Realising an Applied Gaming Eco-system

Research and Innovation Action

Grant agreement no.: 644187

D4.2 – Full Design Document

RAGE – WP4 – D4.2
<table>
<thead>
<tr>
<th>Project Number</th>
<th>H2020-ICT-2014-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>31 March 2016</td>
</tr>
<tr>
<td>Actual Date</td>
<td>24 March 2016</td>
</tr>
<tr>
<td>Document Author/s</td>
<td>Kam Star (PlayGen), James Allsopp (PlayGen), Benjamin Hirsh (PlayGen), Jeremy Cooke (Gameware), Ian Saunders (Gameware), Thierry Platon (Bip Media), Boris Irmscher (Nurogames), Holger Sprengel (Nurogames), Jens Piesl (Nurogames), Marco Polak (Nurogames), Marcus Lindner (Nurogames), Andrew Pomazansky (Nurogames), Sarah Humphreys (Hull College), Gareth Sleightholms (Hull College), Christina Steiner (TuGraz), Rob Nadolski (OUNL), Samuel Mascarenhas (Inesc ID), Wim Westra (OUNL)</td>
</tr>
<tr>
<td>Version</td>
<td>1.1</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
<tr>
<td>Document approved by</td>
<td>WW</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 Research and innovation programme under grant agreement No 644187
### Document Version Control

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change Made (and if appropriate reason for change)</th>
<th>Initials of Commentator(s) or Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>08/01/2016</td>
<td>Initial document structure and draft concepts added</td>
<td>KS, JA, BH, JC, TP, BI, HS, JP, MP, ML, AP</td>
</tr>
<tr>
<td>0.3</td>
<td>15/02/2016</td>
<td>Game design documents compiled into deliverable report</td>
<td>KS, JA</td>
</tr>
<tr>
<td>0.4</td>
<td>01/03/2016</td>
<td>Added table of figures, introduction, executive summary and conclusion.</td>
<td>KS, IS</td>
</tr>
<tr>
<td>0.6</td>
<td>15/03/2016</td>
<td>Feedback and corrections from internal reviewers</td>
<td>KS, SM, CS</td>
</tr>
<tr>
<td>0.7</td>
<td>17/03/2016</td>
<td>Added further pedagogical descriptions and additional notes.</td>
<td>KS, CS, RN</td>
</tr>
<tr>
<td>0.8</td>
<td>21/03/2016</td>
<td>Added further clarifications based on feedback.</td>
<td>KS, JA</td>
</tr>
<tr>
<td>1.1</td>
<td>25/03/2016</td>
<td>Final proofread and comments</td>
<td>KS</td>
</tr>
</tbody>
</table>

### Document Quality Control

<table>
<thead>
<tr>
<th>Version QA</th>
<th>Date</th>
<th>Comments (and if appropriate reason for change)</th>
<th>Initials of QA Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>11/03/2016</td>
<td>Comments provided and actioned</td>
<td>SM</td>
</tr>
<tr>
<td>0.6</td>
<td>15/03/2016</td>
<td>Comments provided and actioned</td>
<td>CS, RN</td>
</tr>
<tr>
<td>1.0</td>
<td>23/03/2016</td>
<td>Comments provided and actioned</td>
<td>WW</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1 EXECUTIVE SUMMARY ........................................................................................................... 8

2 INTRODUCTION ......................................................................................................................... 9

2.1 Pedagogical Approach ........................................................................................................... 9

3 SPACE MODULES INC – PLAYGEN AND SPL ................................................................. 11

3.1 Introduction .......................................................................................................................... 11

3.2 Skills and Learning Objectives ............................................................................................ 12

3.3 RAGE Assets ....................................................................................................................... 14

3.4 Game Design ......................................................................................................................... 16

3.5 Technical Specification ......................................................................................................... 24

4 SPORTS TEAM MANAGER – PLAYGEN AND OKKAM ..................................................... 26

4.1 Introduction .......................................................................................................................... 26

4.2 Skills & Learning Objectives ............................................................................................... 26

4.3 RAGE Assets ....................................................................................................................... 27

4.4 Game Design ......................................................................................................................... 29

4.5 Technical Specification ......................................................................................................... 34

5 DITTO – PLAYGEN AND SPL .............................................................................................. 35

5.1 Introduction .......................................................................................................................... 35

5.2 Skills and Learning Objectives ............................................................................................ 35

5.3 RAGE Assets ....................................................................................................................... 36

5.4 Game Design ......................................................................................................................... 38

5.5 Technical Specification ......................................................................................................... 51

6 IT ALERT – PLAYGEN AND SPL ......................................................................................... 52

6.1 Introduction .......................................................................................................................... 52

6.2 Skills & Learning Objectives ............................................................................................... 52

6.3 RAGE Assets ....................................................................................................................... 53

6.4 Game Design ......................................................................................................................... 54

6.5 Technical Specification ......................................................................................................... 61

7 INTERVIEW SIMULATION FOR JOB SEEKERS – BIP MEDIA AND RANDSTAD........... 62

7.1 Introduction .......................................................................................................................... 62

7.2 Learning Objectives ............................................................................................................. 64

7.3 Rage Assets used .................................................................................................................. 65

7.4 Game Design ......................................................................................................................... 67

7.5 Technical Specification ......................................................................................................... 74

8 HATCH – GAMEWARE AND HULL COLLEGE .................................................................. 75

8.1 Introduction .......................................................................................................................... 75

8.2 Game Flow and Game Modules ......................................................................................... 77


9 INTERVIEW SKILLS FOR POLICE OFFICERS – GAMEWARE AND MINISTÉRIO DA JUSTIÇA

9.1 Introduction ........................................................................................................... 92
9.2 Basic Principles ...................................................................................................... 93
9.3 Scenario Descriptions and Difficulty Levels ......................................................... 98
9.4 Example Scenarios ................................................................................................ 98
9.5 The Interview Engine ............................................................................................ 99
9.6 Simulation Flow and Asset Integration .................................................................. 101

10 WATER COOLER GAME – NUROGAMES AND HULL COLLEGE

10.1 Introduction ......................................................................................................... 102
10.2 Game Design ....................................................................................................... 103
10.3 Skills and Learning Objectives ........................................................................... 110
10.4 Rage Assets ........................................................................................................ 112
10.5 Technical Specification ....................................................................................... 115

11 CONCLUSION ........................................................................................................ 116

12 REFERENCES ......................................................................................................... 117

LIST OF FIGURES

Figure 1: Example time allocation for lesson. ............................................................... 13
Figure 2: Space Modules Inc, RAGE asset flow ......................................................... 15
Figure 3: Space Modules Inc, ITIL Incident flow in game .......................................... 17
Figure 4: Space Modules Inc, customer emotional profile ........................................... 18
Figure 6: Space Modules Inc, single choice GUI ......................................................... 19
Figure 7: Space Modules Inc, multiple choice GUI ..................................................... 19
Figure 8: Space Modules Inc, or choice GUI ............................................................... 19
Figure 9: Space Modules Inc, timed display GUI ......................................................... 20
Figure 10: Space Modules Inc, rank answers GUI ....................................................... 20
Figure 11: Space Modules Inc, scroll answers GUI ..................................................... 20
Figure 12: Space Modules Inc, progression curve graph ............................................. 21
Figure 13: Space Modules Inc, game user mobile interface ......................................... 23
Figure 14: Space Modules Inc, game screen flow ....................................................... 24
Figure 15: Kolb’s model of experiential learning ....................................................... 27
Figure 16: Sports Team Manager, RAGE Asset Flow and Architecture ..................... 28
Figure 18: Sports Team Manager, game flow ........................................................... 29
Figure 19: Sports Team Manager interface designs .................................................... 30
Figure 20: Sports Team Manager, game flow of visual elements .............................. 31
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Ditto, asset flow</td>
<td>38</td>
</tr>
<tr>
<td>22</td>
<td>Ditto, player interaction flow</td>
<td>39</td>
</tr>
<tr>
<td>23</td>
<td>Ditto, gameplay flow</td>
<td>43</td>
</tr>
<tr>
<td>24</td>
<td>Ditto, shape examples</td>
<td>47</td>
</tr>
<tr>
<td>25</td>
<td>Ditto, shape solution example 1</td>
<td>47</td>
</tr>
<tr>
<td>26</td>
<td>Ditto, shape solution example 2</td>
<td>48</td>
</tr>
<tr>
<td>27</td>
<td>Ditto, Star rating score system</td>
<td>49</td>
</tr>
<tr>
<td>28</td>
<td>Ditto, UI wireframe and concept art</td>
<td>50</td>
</tr>
<tr>
<td>29</td>
<td>Ditto, game screen flow</td>
<td>51</td>
</tr>
<tr>
<td>30</td>
<td>IT Alert, flow of data and relations between used RAGE assets</td>
<td>54</td>
</tr>
<tr>
<td>31</td>
<td>IT Alert, game flow with elements per segment</td>
<td>55</td>
</tr>
<tr>
<td>32</td>
<td>IT Alert, 4-player game world layout</td>
<td>56</td>
</tr>
<tr>
<td>33</td>
<td>IT Alert, game HUDs</td>
<td>59</td>
</tr>
<tr>
<td>35</td>
<td>ISJS, asset flow diagram</td>
<td>67</td>
</tr>
<tr>
<td>36</td>
<td>ISJS, game structure 1</td>
<td>67</td>
</tr>
<tr>
<td>37</td>
<td>ISJS, game structure 2</td>
<td>68</td>
</tr>
<tr>
<td>38</td>
<td>ISJS, interviewing phase</td>
<td>70</td>
</tr>
<tr>
<td>39</td>
<td>ISJS, example of simple 2D map</td>
<td>71</td>
</tr>
<tr>
<td>40</td>
<td>ISJS, Interviewing Phase Concept Art</td>
<td>73</td>
</tr>
<tr>
<td>41</td>
<td>ISJS, Interviewing Phase Concept Art</td>
<td>74</td>
</tr>
<tr>
<td>42</td>
<td>HATCH, game flow chart</td>
<td>78</td>
</tr>
<tr>
<td>43</td>
<td>HATCH, analysis example</td>
<td>81</td>
</tr>
<tr>
<td>44</td>
<td>HATCH, project page HUDs</td>
<td>86</td>
</tr>
<tr>
<td>45</td>
<td>HATCH, example layout</td>
<td>90</td>
</tr>
<tr>
<td>46</td>
<td>ISPO, behaviour change during interrogation</td>
<td>94</td>
</tr>
<tr>
<td>47</td>
<td>ISPO, inverted triangle model</td>
<td>94</td>
</tr>
<tr>
<td>48</td>
<td>ISPO, behavioural state</td>
<td>95</td>
</tr>
<tr>
<td>49</td>
<td>ISPO, behavioural state change</td>
<td>95</td>
</tr>
<tr>
<td>50</td>
<td>ISPO, HUDs when interviewing</td>
<td>97</td>
</tr>
<tr>
<td>51</td>
<td>ISPO, HUDs when interviewing 2</td>
<td>97</td>
</tr>
<tr>
<td>52</td>
<td>ISPO, simulation flow and use of assets chart</td>
<td>101</td>
</tr>
<tr>
<td>53</td>
<td>Water Cooler Game, draft the office and its water cooler place</td>
<td>103</td>
</tr>
<tr>
<td>54</td>
<td>Water Cooler Game, draft the water cooler conversation</td>
<td>104</td>
</tr>
<tr>
<td>55</td>
<td>Water Cooler Game, draft of the office and the HUD</td>
<td>107</td>
</tr>
<tr>
<td>56</td>
<td>Water Cooler Game, overview assets and interfaces</td>
<td>116</td>
</tr>
</tbody>
</table>
LIST OF TABLES
Table 1: Kolb learning process ................................................................. 12
Table 2: Space Modules Inc, ITIL impact vs urgency table ................................................. 16
Table 3: Space Modules Inc, components examples ......................................................... 17
Table 4: Space Modules Inc, incident examples ............................................................... 19
Table 5: Sports Team Manager, role stat example ............................................................. 33
Table 6: Ditto, puzzle types ......................................................................................... 40
Table 7: Ditto, types of communication .......................................................................... 42
Table 8: Ditto, league example requirements .................................................................... 45
Table 9: Ditto, experience point reward rate ..................................................................... 46
Table 10: Ditto, achievement example conditions ........................................................... 49
Table 11: IT Alert, subsystems ....................................................................................... 59
Table 12: ISJS, RAGE assets used ................................................................................. 66
Table 13: HATCH, partner work split .............................................................................. 76

LIST OF ABBREVIATIONS

NPC       A non-player character in a game is any character that is not controlled
directly by a player.
GUI       A graphical user interface.
HUD       In video gaming, the HUD (head-up display) or Status Bar is the
method by which information is visually relayed to the player as part of
a game's user interface.
ITIL      Information Technology Infrastructure Library
TCQ       Time, Cost, Quality
1 EXECUTIVE SUMMARY

This deliverable outlines the design blueprints for the RAGE application scenario games and forms the rest of the scope for WP4’s tasks. The game designs have been developed in collaboration with application scenario partners in WP5, and informed by WP1, 2 & 3. Additionally peer-feedback has been provided by game developers across WP4. The designs outline the integration of the RAGE assets developed in WP2 and WP3. Each section provides in detail the game play descriptions, game dynamics and mechanics, pedagogies and technical implementation of the RAGE assets into the game applications as described in detailed in WP5's application documents. The full description of the application objectives and associated learning outcomes has been provided in the project's MS2 Application Scenario Outlines document.

The summaries below provide a brief description of each game and the relevant game developer (WP4) and application partner (WP5) involved.

**Space Modules Inc – PlayGen and SPL**
A socially gamified single player game where the player takes on the role of a customer service representative for a spaceship part manufacturer “Space Modules Inc”.

**Sports Team Manager – PlayGen and OKKAM**
A single player game, where the player must put together the optimal sports team to compete in tournaments, providing training for critical soft skills required in sports employability.

**Ditto – PlayGen and SPL**
A multiplayer cooperative game where two players come together to solve a variety of shape based puzzles, one taking on the role of the architect, the other the role of the builder.

**IT Alert – PlayGen and SPL and OKKAM**
A multiplayer cooperative game for 4 - 6 players. They assume the role of specialist IT service managers, called in to maintain their biggest client’s network infrastructure.

**Interview Simulation for Job Seekers – Bip Media and Groupe Randstad**
A classic point&click adventure game to offer users customized job search courses, with job interviews.

**HATCH – Gameware Europe and Hull College**
The Creative Entrepreneur Game (HATCH) aims to evaluate and enhance Creative Industry-focused entrepreneurial skills across a range of students who are about to embark on creative careers following graduation.

**Interview Skills for Police Officers – Gameware Europe and Ministério da Justiça**
A simulation game to train police officers in communication competencies related to the interview of victims of violent sexual crime and the interrogation of violent offender suspects (primarily sex-crime offenders) through the use of a series of simulated scenarios.

**Watercooler Game – Nurogames and Hull College**
A single player game where the player is hired by a small game studio to develop the team working of the studio through their interpersonal relationships. The player’s goal as an office assistant is to increase the success of the company primary by improving the social skills of the virtual team.
2 INTRODUCTION

The main objective of WP4 is to apply and evaluate RAGE assets by designing, developing and supporting applied games for the application scenario pilots. Accordingly this report provides the designs based on discussions and internal documentations created by application partners in WP5 for each of the 6 application scenarios.

This report is split into individual sections, each covering one game. Each section provides detailed description of the game, the specific skills the game teaches or provides practice for, the RAGE assets that are intended to be used and the functionality that they fulfil, and where applicable the pedagogical approaches. Sub sections have supporting images, technical flows or tables to help explain the concepts. In view of keeping the deliverables as small as possible, each game has approximately 14 pages devoted to it.

2.1 Pedagogical Approach

To ensure pedagogical effectiveness of applied games, the educational dimension and foundation need to be carefully taken into account and incorporated in game design. Given the learning objectives, an applied game should provide an experience that is grounded on pedagogical principles (Greitzer, Kuchar & Hutson, 2007).

All games presented in this document are inspired by a pedagogical approach in the tradition of constructivism (e.g. Fosnot, 1996), considering learning as an active process and promoting learning by doing. Learners are seen as in the centre of the learning process, actively constructing their knowledge instead of being passively exposed to teaching (e.g. Ally, 2004). The adoption of a constructivist perspective appears reasonable for applied games design, since games by nature possess almost all the key features of constructivist learning environments (Tsai, Yu, & Hsiao, 2007). Constructivism is not a specific or single pedagogical strategy, but describes how learning happens, suggesting that learners construct their own knowledge out of their experiences and that learning builds upon prior knowledge of an individual. Constructivist learning theory is based on Piaget (1950) and builds the basis for a range of pedagogical approaches that concordantly argue for active generation of knowledge and meaning from experience.

The games incorporate the following implications for the creation of constructivist learning environments (Ally, 2004; Greeno, Collins, & Resnick, 1996; Hadjerruoit, 2006):

- Learning environment that allows participation in social practices of inquiry, sense-making and learning

  Game experience is characterised by actively participating in simulated situations. The games immerse players in gameplay and learning activities and involve them in interactions with NPCs and/or other players.

- Learning in an active and constructive process with interactive and illustrative instruction using examples and use case

  The learning is embedded in the game story and gameplay. The game and simulation character of the game provide opportunity for discovering knowledge, developing and practicing skills.

- Authentic learning activities relating to real-world situations should be realized
Games establish a direct link and relevance to real-world training and job contexts or realise learning experiences using meaningful metaphors supporting transfer to other gameplay and real-world situations.

- The learning process may be encouraged by collaboration and cooperation
  Learners interact and cooperate with NPCs and/or other players in the game
- Learners ought to have the opportunity to control and direct their learning process
  The game experiences typically allow a high level of learner control and self-directedness
- Self-reflection and generalisation should be realized
  Reflection and meta-cognitions are supported within and outside the game experience, through feedback, reporting and social gamification components, and through reflection and discussion on gameplay experiences in the classroom.

Overall, the games allow players to acquire knowledge and to practice skills in an engaging and meaningful game environment. The application scenario definition and game design have been done by carefully taking into account the characteristics and requirements of the actual learning and performance context and considering how the game will be embedded in this context.
3 SPACE MODULES INC – PLAYGEN AND SPL

3.1 Introduction

Space Module Inc is developed by PlayGen for SPL application partner and incorporates a range of RAGE assets. It is a socially gamified single player game where the player takes on the role of a customer service representative for a spaceship part manufacturer “Space Modules Inc”. Players have to respond to situations by engaging in conversation with customers. The customer satisfaction score at the end of every game level are depending on how well players maintain the conversation. Each customer has a unique emotional profile and provides a new challenge to the player in each level of the game. Players must learn how to manage intense emotions and how to respond to customers in a professional manner in the best way.

After each customer interaction, the player is scored on customer satisfaction, and a breakdown of their responses is shown to allow for self-reflection. The game provides feedback on responses which weren't the best choice in the situation and how to approach these situations next time.

The game and classroom discussions aim to help train students in; utilising conflict resolution strategies in order to keep the situation neutral or beneficial for both parties, improve their ability to defuse stressful situations within a professional work environment, build familiarity with the Information Technology Infrastructure Library (ITIL) Incident Management process. The game also provides practice for players in being able to identify a person's emotional state through verbal and nonverbal feedback and gain further experience in providing effective emotional responses.

Asset usage outline

To provide a unique learning experience for the player and aid in game development, the game aims to utilise a variety of RAGE asset technology which provide some of the key functionality for the game features.

- **Client-side interaction tracking (T2.1A), Server-side interaction storage and analytics (T2.1B), Server-side Authentication and Authorization (T2.4A), Server-side Dashboard and Analytics (T2.4B)** - Provides server setup and storage for game analytics to enable the ability to track player's behaviour and learning outcomes.

- **Emotion Appraisal Module (T3.1A)** - enables each customer to have a unique emotional profile and thresholds and responses to the player.

- **Social Gamification Components (T3.6A)** – provides group goals and achievements together with individual and group leaderboards to allow players to compare scores against each other.
3.2 Skills and Learning Objectives

The learning goal is to train students in vocational education of ICT in communication skills. Space Modules Inc will provide practice in supporting the students’ ability to react to emotionally tense situations, such as dealing with difficult customers, learning to resolve conflict and how to effectively manage and operate within the ITIL business process. The main skills objectives are to improve students abilities on:

- **Emotional Intelligence** - ability to recognise own and other people’s (customers’) emotions and to be able to develop strategies on managing them toward finding the best outcomes and win-win situations.

- **Conflict resolution** - to find the best solution over a disagreement which could be personal, emotional or financial.

- **ITIL Incident Management** - to operate according to the ITIL business process and to ensure the correct procedures are carried out during emergencies.

3.2.1 Pedagogy

Central to the game's ability to teach is its pedagogical framework, which provides a structured learning experience for classroom. The learning pedagogy in this game is inspired by Kolb’s experiential learning model. Experiential learning pursues a pedagogical approach of learning by doing. Applying the Kolb model means encouraging learners to reflect on their actions to foster understanding and transfer. This will enable a smooth transition from a virtual to physical learning environment within a classroom. Table 1 provides an overview on how the learning stages from the Kolb model are integrated in the game and classroom experience.

<table>
<thead>
<tr>
<th>Kolb Process</th>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete Experience</td>
<td>Virtual [Game]</td>
<td>Plays through each scenario</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uses current knowledge and skills</td>
</tr>
<tr>
<td>Reflective Observation</td>
<td>Virtual [Game]</td>
<td>After action review to show the player’s decisions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifying where the player went wrong and where to improve.</td>
</tr>
<tr>
<td>Transition from game to classroom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstract Conceptualisation</td>
<td>Physical</td>
<td>Teacher leads classroom discussion.</td>
</tr>
<tr>
<td></td>
<td>[Classroom]</td>
<td>Students share and compare gameplay experiences.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Students identify common learning obstacles</td>
</tr>
<tr>
<td>Active Experimentation</td>
<td>Physical</td>
<td>Classroom discussion on what could be done differently (in next round of gameplay)</td>
</tr>
<tr>
<td></td>
<td>[Classroom]</td>
<td></td>
</tr>
<tr>
<td>Transition from classroom to game</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Kolb learning process
3.2.2 Lesson Plan

The cycle is repeated several times, possibly across multiple classroom sessions. The table below shows how the time can be split between activities, based on the assumption a lesson is 50 minutes long.

<table>
<thead>
<tr>
<th>Briefing</th>
<th>Gameplay</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td>44</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Example time allocation for lesson.

[10 Minutes] Briefing

- First time - An introduction to what will happen in the next 50 minutes of the lesson and the backstory/setting of the game.

- Multiple sessions - If the game has been run in a previous session, the briefing would consist of a recap on the key points identified in the previous lesson and what could be done differently. This could be reduced to 5 minutes to allow for an extra time during other phases.

[20 Minutes] Gameplay - Students play the game. On average a level should take no more than 5 minutes to complete, some students may be slower than others.

- Free for all - Students can play the game at their own pace.

- Staggered - Students reach a certain level and have to wait until everyone else has completed it. This could be useful, as everyone would have gone through the same scenario.

[20 Minutes] Discussion - After the gameplay, the teacher facilitates a discussion:

1st Half methods:

- Whole class - Teacher asks students to volunteer to talk about their experience. Other students may also participate to discuss differences.

- Groups - Teacher splits the class into groups and gets students to discuss their gameplay experience among each other before asking them to share their experience with the class.

- Pairs - Similar to groups but in pairs.

2nd Half methods:

- Whole class - Teacher asks students to volunteer to talk about what they could do differently, stimulating others to share and participate.

3.2.3 Measurement

The game tracks multiple measurements to determine player score and progression states, these include:
• **Player Choice Score** - each choice a player makes. Making good choices awards a higher amount of points compared to making bad choices.

• **[Average] Customer Emotional State** - customers are emotionally aware and players' decisions could have a positive or negative effect on them. By providing feedback on customer's average emotional state, the game supports players practising strategies in how to best handle a variety of situations.

• **Completion Time** - tracks how long a scenario was played for and allows players to compare with others.

### 3.2.4 Feedback
Each decision a player makes is recorded, when they finish a scenario, feedback and results are provided which include:

• **Player Choices** – a log of all the choices made by the player and solutions/tips to try next time on decisions which may not have been the best choice.

• **Final Score & Rank** - the final score is accumulated through a variety of measurements see section 3.4 for more details. Rank is a total of how well the player has done overall (through all the scenarios played so far) compared to other players.

### 3.3 RAGE Assets
To provide a unique learning experience for the player and aid in-game development, the game will utilise a variety of RAGE asset technologies which will have the base functionality for the game features.

**Client-side interaction tracking (T2.1A), Server-side interaction storage and analytics (T2.1B), Server-side Authentication and Authorization (T2.4A), Server-side Dashboard and Analysis (T2.4B)**

The combination of these assets provide server setup and storage for game analytics system which will drive development and provide support for maintenance once the game is delivered. These assets provide the capabilities to capture player interaction within the game, tracking success and most importantly areas for improving the game, enabling the development team to respond effectively.

**Emotion Appraisal Assets (T3.1A)**

The Emotion Appraisal asset enables customers to portray unique emotional profiles where the player has a different challenge every time they encounter a customer. The asset will provide not only the functionality for emotionally aware non-player characters, but also a controller for animation, such as customer body language and facial expressions:

• **Emotionally Sensitive Customers** - Player's choice and response time will have effects on the customer's emotional state. The asset will help manage emotions states and threshold values.

• **Animation and Facial Expressions** - To visually represent a customer emotion, there is a need for a responsive animation structure. The customer should react and display their emotion not just through voice, but through their body language and facial expressions. The asset's emotion tracking will help provide animation cues to the animation rig of the customer when an emotion is triggered.

• **Customer Satisfaction Score** - One of the tracked values during gameplay is the emotional change of the customer over time (mapped with player's choice). The asset will provide cues to the game activity logger, so that a customer's emotion can be
tracked over time, and then presented to the player during a feedback stage to show how the choices made affected the customer emotionally.

**Social Gamification Assets (T3.6A)**

Provides the ability to turn a single player game experience into a multi-player experience, providing the ability for group goals, progression and achievements to be implanted. Additionally it provide functionality for comparing scores, ranking players and hall of fame display:

- **Individual and Group Leaderboard** – Players’ progression as measured by their scores can be combined as both individual and group scores.
- **Hall of Fame** - A set of breakable records which any player can attempt. A current record holder can be outpaced if another player achieves a better result.

### 3.3.1 Asset Flow

Figure 2 provides an overview of flow of information and architecture of RAGE assets in the game.

![Figure 2: Space Modules Inc, RAGE asset flow](image)

1. Player interacts with the game state
2. Interactions are sent to various RAGE assets
3. Emotional Appraisal takes the player’s dialogue choices and calculates the NPC’s emotional response, and then updates the NPC in the game state
4. Interaction tracking asset tracks all the input the player makes into the game, alongside information about events that occur in the game state
5. Data is stored in an analytics server for external viewing
6. The teacher is able to login to this data via the authentication and authorization asset.
7. The social gamification saves the game states and player interaction data, providing comparison, ranking and goals.
3.4 Game Design

3.4.1 Narrative

Backstory: Players take on a role of a new incident analyst who recently joined Space Modules Inc, the galaxies number one supplier for spaceship and starship parts. The company has a reputation for excellent customer service. It's up to the player to ensure this continues.

Dialogue Structure: Choice can affect the next possible set of responses and emotion state of the customer. Through interaction and experimentation with the customer, the player will begin to understand what responses will have a direct effect on the emotional thresholds. From this they will learn how to deal with specific customer's emotional needs and develop strategies to approach the next set of customers.

A collapsible dialogue tree is used since collapsible dialogue tree allows the player to make a mistake without the customer becoming angry too quickly (depending on their emotional threshold). For example (in Figure 5), if the player was to get three choices right and one wrong giving them a score of 2, Customer A would end in a happy state, Customer B would end in a contented state, and Customer C unhappy. Different customer's emotional thresholds create interesting gameplay.

3.4.2 Game Dynamics

Incidents: Each incident requires the player to follow the ITIL Incident Management framework in order for it to be resolved. Incidents grow in difficulty as the player progresses. Once an incident begins, a timer starts. It stops when the incident is resolved. The result contribute to the final scores. Some incidents may have a time limit. If they are missed, they will have consequential effects on the narrative development. If a player solves the incident on their own without any help from a technician or emergency response, they will receive an incident completion bonus.

- Incident Urgency: how urgent the incident is to be completed. This will depend on the incident and narrative, as some incidents may cause major disruption and others are just small faults, such as the lights flickering.

- Incident Impact: how much effect the incident has on the customer, for example, if life support systems are down, the incident would be categorised as Critical.

The tables below are used to categorise and incident, based on Urgency and Impact within the ITIL incident process. We have provided some examples to show how in-game incidents map to them:

<table>
<thead>
<tr>
<th>Urgency</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>Low</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2: Space Modules Inc, ITIL impact vs urgency table
<table>
<thead>
<tr>
<th>Priority</th>
<th>Description</th>
<th>Incident Type</th>
<th>Module</th>
<th>Effect [on Customer]</th>
<th>Timed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Critical</td>
<td>System offline</td>
<td>Crella Oxygenator</td>
<td>Oxygen levels decrease</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>High</td>
<td>Technical Incident</td>
<td>TV-X00 Radar</td>
<td>Malfunctions after X time, asteroid collision.</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Medium</td>
<td>Software Bug</td>
<td>Impulse DPU2</td>
<td>Engines onboard computer receives wrong data</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>Drive Update</td>
<td>S70 Scanner</td>
<td>scanners system crashes</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Planning</td>
<td>Restart</td>
<td>Ion Drive Type 9</td>
<td>Lights flickering on and off in customers spacecraft.</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 3: Space Modules Inc, components examples

Figure 3: Space Modules Inc, ITIL Incident flow in game

1. Incident is retrieved from the database
2. Once an incident is initiated, the player follows the ITIL Incident Management process [facilitated by the game], also driving the narrative forward.
3. Each incident is timed, the value is then converted into a score, which is used to count towards the final score.
4. If an incident takes too long to complete, it may affect some customer’s emotional state, i.e. they may get impatient or angry (depending on their emotional profile).
5. If an incident has a time limit, it may affect the environment state. For example if a radar module is down for a long period of time, the customer may experience an asteroid collision, causing the screen to shake, lights to flicker and the alarm sounded.
6. Once an incident is flagged as completed [from the customer], the final score is then calculated. If an incident is not completed during the time limit, the final score is calculated and the game progresses down to a negative narrative branch.
3.4.3 Emotionally Sensitive Customers

Each time a player encounters a different customer, they are presented with a unique challenge like identifying the hardware/software fault and managing the customer’s emotional needs. Each customer has a different emotional profile, which is a set of thresholds. Figure 4 shows an example of two customers’ emotional profiles. Customer A has a high tolerance and is difficult to anger, whereas Customer C is easily angered and difficult to keep happy.

![Customer Emotion Profiles]

Figure 4: Space Modules Inc, customer emotional profile

3.4.4 Feedback

Feedback from customers depends on the previous action and the customer's emotional state. If the player made some wrong choices at the beginning, it could be detrimental to the rest of the interaction. Customer could become too unhappy/angry, even if the player provides correct responses afterwards. The game also provides visual feedbacks which are displayed on-screen such as:

- **Explicit response** - when a customer exerts an intense emotion, such as anger, comical animation overlays will display, providing the player with a clear visual indication of the customer's emotional state.

- **Subtle response** - body language and tone may provide the player with an idea of how the customer is feeling.

3.4.5 Player Choice

Players progress through decisions. Each choice has consequences either on the customer they are speaking to or the environment around them:

- **Customer consequences** - Customers are affected by the players’ response time and type. Some customers may be impatient and could emotionally deteriorate if the player doesn’t respond in a timely fashion.

- **Environmental consequences** - If a customer is experiencing a malfunction in their power module, the customer's background environment could be blacked out/low light to represent the severity of the problem. Table 4 provides more examples:

<table>
<thead>
<tr>
<th>Module Type</th>
<th>Module Name</th>
<th>Incident</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engines</td>
<td>Prototype Warp</td>
<td>Software bug</td>
<td>Customer’s ship is stuck in space unable to move until the problem is fixed.</td>
</tr>
<tr>
<td></td>
<td>Drive MK II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>MX500 Radar</td>
<td>Out of date</td>
<td>With radar down the customer's ship is unable to detect incoming asteroids. The problem could result in a collision if not fixed within the time limit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>driver</td>
<td></td>
</tr>
<tr>
<td>Life support</td>
<td>Crella Oxygenator</td>
<td>System offline</td>
<td>A high priority incident which needs to be fixed fast, otherwise the customer’s crew deteriorates.</td>
</tr>
</tbody>
</table>

---

WP4-D4.2

RAGE

Page 18 of 118
**3.4.6 Choice Types**

One of the main aims of the game is to get players to sit back, think and reflect on their decisions, as quick firing through the questions won’t provide any value for the player. The game provides the traditional multiple choices with a few variations to challenge the player's cognitive processes - see below for variations:

**Single Choice** (Figure 5)

The player is required to pick only one option. Answers can appear in a random order each time to prevent any choice patterns emerging, also questions can have a time limit.

![Single Choice GUI](image)

*Figure 5: Space Modules Inc, single choice GUI*

**Multiple Choice** (Figure 6)

The player is required to pick more than one answer.

![Multiple Choice GUI](image)

*Figure 6: Space Modules Inc, multiple choice GUI*

**Binary Choice** (Figure 7)

The player is required to pick one or the other option, but only two answers are displayed.

![Binary Choice GUI](image)

*Figure 7: Space Modules Inc, or choice GUI*

**Timed Display** (Figure 8)

Decisions are not active to be selected until a certain time has passed.
**Timed Display GUI**

![Image of Timed Display GUI]

Figure 8: Space Modules Inc, timed display GUI

**Rank Answers** (Figure 9)

Players must rank the answers by dragging and dropping, and once they are happy with the choice, they can press accept. Another variation - the answers must be ranked in the correct order before the time runs out (no accept button).

![Image of Rank Answers GUI]

Figure 9: Space Modules Inc, rank answers GUI

**Scroll Answers** (Figure 10)

Player must scroll through each answer to find the correct one. Practical for a mobile device, also to ensure player reads all answers.

![Image of Scroll Answers GUI]

Figure 10: Space Modules Inc, scroll answers GUI

**Progression**

3.4.7 **Gameplay**

Gameplay progression is based on the ITIL Incident Management framework, requiring the player to follow the exact same process every time they engage with a customer. Repeating each step enables the player to familiarise themselves with the ITIL process. The process has the following steps:

- **Detection** - Customer makes contact with player and scenario begins.
- **Incident Logging** - Player listens to the customer's problem and logs it as an incident in the Incidents Logs System.
  - **Classification** - Player categorises the incident based on predefined criteria.
Prioritisation - Player prioritises the incident based on the impact to customer and urgency.

Investigation and Initial Diagnosis - Player accesses the Module Database to gather additional information to help to resolve the incident.

Escalation - If a resolution isn’t found or the incident is deemed an emergency, player can choose to escalate it to the another handling group such as:

- Technician - If a resolution cannot be found and it’s not an emergency, a technician can be requested.
- Emergency Response - If player deems the incident to be an emergency, they can forward the incident to the emergency response team.

Resolution and Recovery - If the player is confident, they know how to resolve the incident, they can attempt a resolution. If the resolution is not successful, they can try again or request a technician (incident complete bonus is forfeited).

Closure - Once the customer flags the incident as complete, the scenario comes to an end, where the total score is calculated and feedback is provided to the player.

3.4.8 Experience Points (XP)

Reward schedules determine when and how much XP is given to the player:

- Fixed Ratio Schedules - XP is awarded a fixed amount of actions, i.e. player choice.
- Fixed Interval Schedules - XP is awarded based on a fixed time scale, although depending on the incident time limit.

3.4.9 Difficulty

As players progress, problems become more complex and margin for error decreases. Therefore, during early stages, player can make more mistakes, compared to later stages where it is less forgivable. In the graph below you will see the progression curve.

\[
2 - \text{Current Level} + \text{Previous Difficulty} \times \text{Difficulty Rate}
\]

![Figure 11: Space Modules Inc, progression curve graph](image)

Progression Type

We aim to use a polynomial progression system. The progression factors decline as the level advances, but the total progression ratios fall more slowly than basic progression factors. Polynomial systems generally have a gradation in early levels akin to exponential progressing, however, the rate of reward changes at higher levels.
3.4.10 Score

Scores help keep track of progression for different gameplay elements. There are a variety of scores:

**Customer Satisfaction** - an accumulation of the XP and any incident bonuses awarded to the player.

**Star Rating** - A quick indication of how well the player pleased the customer

- 1 star - 20% of the maximum score earned for the particular level,
- 2 star - 50% of the maximum score earned for the particular level,
- 3 star - 70% of the maximum score earned for the particular level.

The percentages thresholds may increase with the difficulty.

**Average Emotional State** - The emotional state of the customer is tracked over time, and average emotional states are presented to the player.

**Rank**

The total score accumulated by the player is compared and ranked (by the highest first) with other player’s scores. Ranking allows players to compare scores and display how well they are progressing in the game.

3.4.11 Achievement

Hall of Fame, a set of predefined breakable records players can work towards. Current record holders can be outpaced by other players, providing that everyone has an opportunity to have their name in the hall of fame.

Examples of achievements include:

- Highest scoring player
- Most amount of hours spent playing
- Top ranking player for a week/month
- Fastest game completion (100%) time
- Longest active player

3.4.12 Interface Example

See Figure 13 below for wireframe design of the game on a mobile phone.
Figure 12: Space Modules Inc, game user mobile interface
3.4.13 Game Screen Flow
See Figure 14 below for the overview of the game screen flow.

Figure 13: Space Modules Inc, game screen flow

3.5 Technical Specification
3.5.1 Development
- Game engine: Unity3D
- Programming language: C# and Javascript
3.5.2 Target Requirements

- Hardware: Mobile Device. Optionally desktop PC
- Input: Touch screen. Optional: Mouse
- OS: Android 5 or higher, iOS 8 or higher, Windows 7 or higher
- Network: Internet connection
- Additional software: Browser
4 SPORTS TEAM MANAGER – PLAYGEN AND OKKAM

4.1 Introduction

Sports Team Manager is a single player game, where players must put together the optimal sailing team to compete in races. The player becomes the manager of a team, which comprises of multiple roles each with varying skill requirements and roles. The player's objective is to race with the highest performing team by correctly assigning individuals based on their skills and interpersonal relationships, in the process handling and managing conflicts. This is done by interviewing and hiring characters. The game focuses the player attention on the interaction and relationships between the NPCs and the player. The core dynamic of the gameplay is to put characters into roles in the team, place a team in a race, review the performance of the team and then repeat. The learning objectives is thus to support players develop a better understanding of team work, communication, conflict management, emotional intelligence and time management.

The game makes use of RAGE assets to create realistic NPC dialogue and behaviours, and to create the scenarios around them. Assets are also used to monitor the player's interaction and provide relevant feedback and interventions.

4.2 Skills & Learning Objectives

Overarching learning goal of the game is to acquire soft skills for employability. Based on an analysis of the requirements the soft skills required have been identified as: communications, team working, leadership, problem solving and conflict management. Sports metaphor has been identified and chosen as a suitable means for training; a game experience in a sport context should serve as a basis for developing and practising transferrable soft skills relevant for employability. To provide a contextual example to help explain the components and mechanics for the Sports Team Manager game we use a sail racing theme. This however does not represent the final product and may change according to the use case partner’s target audience theme preference.

Communication

The gameplay is based on conversing with virtual characters, who at times will conflict between themselves and/or the player. The player will learn the importance of empathy and emotional intelligence by being rewarded for making correct decisions, during such scenarios. Identifying soft skills are also practiced. Characters with good soft skills will be essential to the team’s success, the implications of which are shown to the player through gameplay.

Team Work

The player witnesses the interactions of the characters and their performance in the races, thus learning the use of good work relationships and specialised role assignment. The conflict management parts of the game will give the player opportunities to make or maintain the relationships between the characters.

Management

The races, which measure how well the player has constructed their team, occur at scheduled intervals. Each action in the game costs a certain amount of time to perform, so it’s up to the player to effectively manage their time in order to be ready for the next race.

4.2.1 Pedagogy

The pedagogical framework underlying the game and how it is embedded in its learning and training context is inspired by experiential learning model of Kolb. The learning stages of this
Concrete Experience: Game is played, student is taught the different learning scenarios via the game mechanics.

Reflective Observation: After the game, player is asked to reflect on the experiences.

Abstract Conceptualisation: Teacher and students come together to reflect collectively. Teacher has access to performance and feedback.

Active Experimentation: Part of the above discussion, each player is asked to think about how they will play the game during the next lesson.

A typical lesson plan may focus on just one scenario, or multiple scenarios and may include period for instruction, play and reflection and generalisations. Scenarios could be shorter or longer, and if the set-up precludes the ability to carry out physical discussion, a virtual board for feedback and reflection could be utilised.

The results of each race, shows the player how well they assembled their team, and where improvements need to be made, in accordance with the learning goals.

4.3 RAGE Assets

Interaction Tracking (T2.1A) / Storage & Analytics (T2.1B). Authentication & Authorization (T2.4A) / Dashboard & Analysis (T2.4B)

Captures the player's interaction with the game, tracking success, errors, help requests and elapsed time. The data is aggregated server-side for further analysis and access to the server is handled with authentication asset.

Emotional Appraisal Module (T3.1A), Emotion Decision Making (T3.1B), IDeA - Identity Driven Agents (T3.1C), Social Importance Dynamics (T3.1D)

These assets are collectively used to produce realistic character interactions, involving or witnessed by the player. Emotional values based on events which occur in the game (such as losing a race or having an argument), and uses them to produce an appropriate reaction. These are either reactive emotional actions or a set of deliberative actions which lead to a particular goal. The remaining assets are used to create realistic NPC-NPC and NPC-player social interactions by providing personality traits to them and influencing the characters' behaviour and actions by their social contexts rather than purely based on their personal goals.

Social Gamification Asset (T3.6A)

This asset is used to track the goals, achievements, progress and scores Players, which is then used to formulate matches as well as leaderboards.

Authorial Agents (T3.5C), Integrated Authoring Tool (T3.7A)

Generates dialogue options for the player and the virtual characters' responses based on emotional state and personal goals. The generated content is aimed at improving the player's understanding of learning goals where improvement is required. The authoring tool assists in creating the learning scenarios, whilst implementing mandatory plot points.
4.3.1 Asset Flow

This section describes the interaction of the RAGE assets and the game components. This is visually represented in Figure 16:

1. Player makes changes to Game State as they interact with the game.
2. Game state holds all possible events that can happen in that state.
3. The interaction tracking asset takes event data from the game state, along with the interaction data.
4. This data is stored on a remote server for analysis.
5. This asset provides a user-friendly interface to aggregate and view data from the game.
6. Access to this information is controlled by the authorization asset.
7. The multiplayer aspect of the game is provided by the social gamification assets.
8. The authorial agents take events, alongside information on the characters and the scenarios, to push the player towards the learning goals.
9. The scenario component is the medium through which the player interacts with the virtual characters.
10. Virtual characters component houses all the NPC data and acts as a high level interface that the other components interact with.
11. The emotions of NPCs are changed by this asset, depending on what event occurred, involving that character.
12. Emotional change is sent to the emotion decision making asset to produce a reaction for the NPC to make.
13. IDeA gives characters personality traits in addition to their personal goals, which influences behaviour and decisions.
14. Social importance causes the characters actions to be influenced by the existence of others, rather than solely themselves.
15. The authoring tool provides a user-friendly method of authoring the agents and scenarios.
4.4 Game Design

The game places the player in the role of a sailing team manager. The overall goal is to put together the most optimally performing sailing team by resolving conflicts and managing the teams interactions. The player interviews virtual characters to identify their skills and personalities. The team has a set of roles, each with overlapping skill requirements. A successful sailing team is not solely based on skill, but also on social relationships between team members. The player manages these aspects through scenarios and conversations involving the NPCs.

4.4.1 Structure and Flow

The game begins with an introduction scenario, which puts the game’s story into context. Then a step through interface tutorial is presented to the player, which follows one iteration of the loop in the flow diagram in Figure 2. The loop then continues for each scenario.

![Flow Diagram](image)

**Figure 16: Sports Team Manager, game flow**

*Start:* The landing/splash page of the game’s website.

*Introduction:* Short scenario to introduce the game’s context and some of the virtual characters to the player.

*Management:* The main interaction phase of the game (See Figure 3). This is where the player manages the sailing team (reconfiguration and recruitment), and may also engage in conversations with team members, usually to deal with conflict.
**Race:** Simple minigame where the player watches the current team configuration compete in a race.

**Review:** Presents the player the results of their team's performance. Details any conflicts or any positive interactions that may occur between the characters in the team.

**Scenario:** Depending on the player's progress towards specific learning goals, the scenario relevant to a particular learning goal is triggered.

![Figure 17: Sports Team Manager interface designs](image)

### 4.4.2 Visual Components

Diagram 19 below covers the different screens of the game, each with a list of the components visible to the player. Elements denoted with a '?' indicate that they appear depending on the
game situation. For example, the tutorial UI will only be visible during the first iteration of the game sequence. Another example, the team member filling a position in the team will not be visible if the position is empty.

**4.4.3 Game Dynamics**

**Challenges**

**Puzzles:** Each character's skill is described by a fixed set of stats and each role in the team has a number of stats associated with it. The stats requirements for each role is hidden to the player who must deduce them by reading the role's description and through interviews. The player will be presented with a group of candidates and will then choose which candidate(s) to hire.
Each set of races is referred to as a season. The game's timeline will feature a sequence of multiple seasons. Each season has a particular bonus stats, where by building a team that has a large cumulative value of that stat player will achieve a higher score. This causes the player to reconfigure their team between seasons, providing varied gameplay.

**Obstacles:** NPCs in the team may conflict with each other and/or with the player. In order to prevent a decline in score, the player must manage these conflicts before the next upcoming race. For example: *The team captain may inform the player that he will retire from the team after the next race.* This will force the player to start searching for a replacement.

**Constraints:** The races which must be prepared for are displayed on a timeline. Each action the player may perform consumes a certain amount of time, so the player must manage their time resource in order to prepare effectively.

The team has a limited number of spaces/roles to be filled, introducing the need to shortlist candidates.

**Progression**

**Difficulty:** Each race has a minimum score requirement to complete, which is calculated as the total of each character's stats they are using (modified by interpersonal relationship scores and conflict). The amount of which a player exceeds this, is shown as a 3-star grading system but converted to race positions: 1st: 3-star, 2nd: 2-star, 3rd: 1-star and 4th: failure.

One of the rewards players receive on successfully completing races is a larger pool of candidates to hire from. This increases the challenge of the interviewing process. It makes it possible to have higher risk team configurations. For a player who is not performing well in the races, the choice of prospective team members is kept small and easy to manage. Seasons also increases in complexity as the game progresses.

**Victory Conditions:** The scenarios end after the player has completed the race seasons. A final race is proposed which does not have a score requirement, but rather the player must score as high a score as possible which will feature in leaderboard.

**Achievements:** Achieving high scores on the grading system will unlock cosmetic items for the player's boat and team members.

### 4.4.4 **Characters (NPCs)**

Some of the characteristics of the NPCs that the player has to consider are:

**Stats:**

- **Body:** physical fitness and endurance.
- **Charisma:** good leadership and modulates personality traits.
- **Perception:** ability to think ahead and outside the box. Situational awareness.
- **Quickness:** reactions and agility.
- **Willpower:** keeping calm under pressure and dealing with fear.
- **Wisdom:** having good judgement, making the right decision.

**Roles**

List of roles in the sailing team. They are split into 3 categories which mirror the physical parts of the boat's layout of where the roles are positioned.

**Command:** These roles are located at the ship's stern. They focus on the final output of the ships movement.

- **Skipper:** team captain.
- **Helmsman:** steers the ship.
- **Navigator:** plans manoeuvres, based on wind conditions.
Energy: These roles focus on strength and endurance to drive the ship, and are located in the mid section of the ship.

- **Trimmers**: control the shape of the sails.
- **Grinders**: hoist the sails.

Maintenance: The roles at the bow of the ship revolve around repairing/replacing sails.

- **Pitman**: manages sail changes and doubles as a grinder.
- **Mid-Bowman**: stows sails below deck and helps with changes.
- **Bowman**: climbs the mast or bow to change sails.

### Table 5: Sports Team Manager, role stat example

<table>
<thead>
<tr>
<th>Role/Stat</th>
<th>Body</th>
<th>Charisma</th>
<th>Perception</th>
<th>Quickness</th>
<th>Wisdom</th>
<th>Willpower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipper</td>
<td></td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Helmsman</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Navigator</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Grinder</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Trimmer</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pitman</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mid-Bowman</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Bowman</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>X</td>
</tr>
</tbody>
</table>

### 4.4.5 Sample feedback and discussion questions

The following are a list of suggested performance feedback and reflective questions. Where possible the system will measure and feedback on these factors, additionally players are encouraged to reflect and discuss them in order to support generalisation of the skills learnt. These include:

- What do you think is essential to building a good team?
- How did you find the stress of time management?
- Would you have done anything different to improve?
- How do soft skills affect team dynamics?
- Which conflicts did you find the most difficult to deal with?
  - Why were they so difficult?
  - Would you have done anything differently?
- What did you find most difficult in building your team?
  - Would you have gone about it differently?
- What effect did the in-game characters have on you?
o Why do you think they caused this?

- Were there any notable characters?
  o Why were they notable?

### 4.5 Technical Specification

#### 4.5.1 Development

- Game engine: Unity3D
- Programming language: C# and JavaScript

#### 4.5.2 Target Requirements

- Hardware: Mobile Device or desktop PC
- Input: Touch screen or Mouse
- WebGL based player. Firefox, Edge or Chrome. Optionally available as an app on OS: Android 5 or higher, iOS 8 or higher, Windows 7 or higher
- Network: Internet connection
5 DITTO – PLAYGEN AND SPL

5.1 Introduction

Ditto is developed by PlayGen for SPL application partner. Ditto is a multiplayer cooperative game designed to improve interpersonal communication and teamwork, particularly verbal and non-verbal communication between two people who may be using audio and/or video chat to cooperate.

Two players come together to solve a variety of shape based puzzles, one taking on the role of the architect, the other the builder. In order to successfully complete the challenge both players need to forge a relationship based on good teamwork, clear communication and effective verbal and non-verbal skills.

Players communicate over audio and/or video chat, the architect is able to see the winning combination of shapes to arrange, while the builder must listen and look to the architect and arrange the shapes to align with what the architect is describing they can see. Each level is won when the winning combination is achieved by the builder, score is determined by how quickly the puzzle is solved. Upon completing the level, both players are asked to provide feedback on how well they believe the other player communicated. Over time, the puzzles increase in difficulty, testing the players’ ability to work together to overcome obstacles, including communication blackouts (no audio), no video feed and reduced ability to see what the other one is doing.

The game and classroom discussions are aimed at providing students with opportunities to practice; understanding each other’s roles in order to effectively work together, communicating instructions clearly and precisely, actively listening and asking questions and support effectively and without becoming agitated.

5.2 Skills and Learning Objectives

The general learning goal of the game is to provide training on communication skills to learners applicable to a large variety of vocational and higher education contexts.

Ditto will test students’ ability to communicate with each other and actively listen to instructions. Each challenge requires a degree of collaboration, increasing in difficulty as students’ progress in the game.

- **Leadership** - asymmetric roles provide a two way perspective to leadership. A player can lead, but also follow, from this they can learn from each other what the good and bad qualities of a leader are (from first-hand experience).

- **Conflict Resolution** - players will sometimes get into heated debates about how to progress forward. Players need to learn to step back and find a peaceful and effective solution over a disagreement.

- **Active Listening** - Listening is important, remembering what’s said also is just as important. Not listening or remembering can cause game progression to slowdown.

- **Verbal/Non-Verbal Skills** - Good articulation and body language is also needed to overcome some challenges. Constraints, such as no audio or no video require the players to utilise the communication channels they have open.

The pedagogy and lesson plans for Ditto follows a similar pattern as Space Modules Inc. described in section 4.
5.2.1 Pedagogy

The main pedagogical concept pursued is collaborative learning. The game applies an approach of collaboration in dyads. From a pedagogical perspective such dyadic approach is considered as highly functional for collaborative learning, since participation and responsibility in the collaborative process is increased (Gokhale, 2012). Concretely, the gameplay integrates collaborative problem solving as instructional strategy (Nelson, 1999). The use of collaborative tasks supports learners in developing and improving their collaboration skills as inherent part of the gameplay (Reigeluth, 2005). Collaborative problem solving consists in a process in which two or more individuals are engaged with a problem by sharing their understanding and effort required to come to a solution. The different roles in the game (only the architect seeing the shape that the builder needs to create) require players to explain themselves and thus reinforce communication and learning in the player dyads.

5.2.2 Measurement

Individual and team progress is recorded in the form of a score which is totalled after each level has been played. Players provide feedback on each other’s communication skill ability through a rating system. The game tracks multiple measurements.

- **Player Experience Points (XP) total** - tracks the total amount of XP a player has earned.
- **Team League Points (LP) total** - tracks the total amount of LP a team (2 players) have earned.
- **Completion time** - how long it took a team to complete a level.
- **Average completion time (all players)** - how long it took all teams (whole player base) to complete a level
- **Games played** - how many levels played by the player, this tie in with the feedback.

See below.

5.2.3 Feedback

The game and players provide each other with feedback on their communication skills and suggestions on areas for improvement. Below are the skills which players rate each other on:

- **Leadership and collaboration** - Has the player been a good leader or collaborator? What could they improve? Rate yourself and other player on a range of abilities.

- **Verbal Communication** - How articulate was the player? What did you not understand? What was clear and done well?

- **Nonverbal Communication** - Was body language used? Was it effective? What could be improved?

- **Active listening** - Did they listen to you? Did they forget any instructions?

- **Conflict Management** - Was there any conflict? Did you resolve the situation? Rate how unbiased you believe the other player is.

5.3 RAGE Assets

To provide a unique learning experience for the player and aid in-game development, the game will utilise a variety of RAGE asset technologies which will have the base functionality for the game features.
Client-side interaction tracking (T2.1A), Server-side interaction storage and analytics (T2.1B), Server-side Authentication and Authorization (T2.4A), Server-side Dashboard and Analysis (T2.4B)

The combination of these assets will provide server setup and storage for game analytics system which will drive development and provide support for maintenance once the game is delivered. These assets will provide the capabilities to capture player interaction within the game, tracking success and most importantly errors, enabling the production team to respond effectively.

Social Gamification Components (T3.6A)

This asset provides multiplayer abilities and matchmaking, leader boards and tournaments . The core of the asset will help provide foundations for the following mechanics:

- **Leagues** - the asset's tournament functionality will support the development of the league mechanic. Leagues bring players together through score comparison over duration of time. Achievements and items will also be used to manage league resources, such as LP and reward schedule.

- **Leaderboard** - A variety of leaderboards are required (see 5.4.9) for more details:
  - Global rank of player and team overall scores.
  - League ranks of team overall scores within the duration of the league.

- **Hall of Fame** - A set of breakable records which any player can attempt. A current record holder can be outpaced, if another player achieves a better result. [See 5.4.10]

Performance Statistics (T2.2)

Time taken (TaskTime) to complete a level is compared with all other times the level has been completed and an average is calculated. By beating the average time the players are able to earn a bonus amount of experience points or league points. The assets functionality will also be used within the league mechanic, where average scores are used to determine team positions.
5.3.1 Asset Flow

1. Player interacts with game state.
2. Game state sends varying data to different assets.
3. Performance statistics compares performance data of current player against other players and finds deviations from global average score. This changes the XP ratio of the game, making it easier for players who are not performing well.
4. Social gamification asset takes this score comparison and data of current player to generate a leaderboard model.
5. This model is used to formulate teams and to display the global leaderboard rankings in the multiplayer component of the game.
6. Interaction tracking asset tracks input player makes, alongside information about events that occur in the game state.
7. This data is stored in an analytics server for external viewing.
8. The teacher is able to login to this data via the authentication and authorization asset.

5.4 Game Design

5.4.1 Player Roles

The game forces the players to work together to overcome the challenges presented. Gameplay relies on a symbiotic relationship to be formed between players: success and failure of one player affects the other. Working together not only means constant two-way communication, but also the ability to actively listen and provide help when necessary. Roles are asymmetric by design and are interdependent.
**Architect**: Has to ensure the winning shape combination is achieved within the allotted time, by providing verbal or nonverbal instructions to the builder. The architect’s traits are as follows:

- The only player who can see the winning shape combination.
- Can see the builders’ movements.
- The default player with a video feed.

**Builder**: Constructs the winning shape combination. In order to achieve this, they must interpret instructions by the architect. The builder’s traits are as follows:

- The only player who can move shapes around the game world.

Figure 20: Ditto, player interaction flow

The player interaction flow is described in section 5.4.12; The architect is presented with the winning shape combination (1), then she proceeds to communicate verbally and nonverbally using the camera and microphone (2). After that, the builder receives the architect's instructions and positions the shapes accordingly (3). The current shape positions are then displayed to the architect in a feedback loop (4).

### 5.4.2 Challenges

**Puzzle Solving**: Actions within the game (construct winning shape combination) can be solved either through deductive or inductive reasoning. Both players work together to find the solution to solve the puzzle. Table 6 provides examples of the types of challenges players will face, combining challenge types increases the difficulty:

<table>
<thead>
<tr>
<th>Visual</th>
<th>Puzzle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Visual" /></td>
<td><img src="image" alt="Position" /></td>
<td>A shape is required to be positioned in a specific place.</td>
</tr>
</tbody>
</table>
### Table 6: Ditto, puzzle types

<table>
<thead>
<tr>
<th>Colour</th>
<th>A shape of a particular colour is to be positioned in a specific place. Colour order will also apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacking</td>
<td>A shape is positioned on top of another shape. Shape order and colour applies.</td>
</tr>
<tr>
<td>Rotation</td>
<td>A shape can be rotated in order to fit the correct combination.</td>
</tr>
<tr>
<td>Multi Piece</td>
<td>Multiple shapes are combined to make another shape. Multi piece shapes snap together once the player moves them close to each other in the correct rotation and position.</td>
</tr>
</tbody>
</table>

**Obstacles & Constraints**: Obstacles disrupt communication and interaction between players, testing their communication skills and capacity to solve problems as a pair team.

- **Hidden information** - the builders’ movements are hidden from the architect, resulting in the architect unable to see what shapes the builder is moving around. A solution to overcome this would be for the builder to verbally communicate to the architect each step they are taking.

- **No video feed** - this obstacle disables the architect’s camera, so the builder cannot see the architect. However, the microphone is still active allowing the architect to provide instructions vocally to the builder.

- **No audio** - this obstacle disables the device microphone, so communication between both players is reduced to body language and hand gestures.

- **One-way communication** - only the architect can speak, they are unable to hear the builder, so they must communicate clearly and trust the builder.

- **Colour variation** - one player may experience a different coloured shape on their screen compared to the other, the players will need to work out the variation between both views in order to solve the puzzle.

- **Timed Combinations** - the target shape combination reconfigures after a certain amount of time has passed, testing the player’s ability to react to changes quickly.
• **Limited** - a combination changes a set amount of times.

• **Limitless** - a combination changes every X time, the game cannot be won until the combination is achieved by the players.

• **Time Limit** - a time limit creates a pressed situation, the players must achieve the winning combination before the time limit is reached - otherwise they fail.

• **Symmetrical display** - both players are able to see the time limit.

• **Asymmetrical display** - only one of the players can see the time limit, they must communicate it to the other player.

• **Timed combinations** - the winning combination may reconfigure after a certain amount of time, this tests the player's ability to respond to a change in the situation.

• **Timed Communication Line** - gives the players a specific amount of time to communicate, after which the communication line is broken and it's up to how well the players listened and remembered what they discussed.

• **Role Reversal** - reverses the roles of the players at a specific time or after X amount of actions. The new roles are then assumed until the end of the level unless role reversal is enacted again.

• **Shape Diversity** - as players progress, more diverse shapes are introduced. For example, they change in size and colour to provide a challenge.

• **World Size** - the size of the world determines not only the position of the shapes, but also the players’ manoeuvrability.

• **Movement Limits** – Counted each time the player touches the shape, Limiting movements provides a challenge, which not only requires spatial thinking, but “three moves ahead thinking” found in games, such as chess.

### 5.4.3 Multiplayer Management

**Player:** Player can set up a quick match between two players or become friends, allowing both players to team up and enter a league. Status indicator lets a player know if another player is online or offline.

**Teams:** A team consists of two players. Players are able to be part of multiple teams, there is no limit. Playing in a team during a challenge awards XP to the players individually 50/50 split and LP, which can then be used to enter and make progress in other leagues. Teams can earn achievements. The team name is chosen by the player who initiates the team creation/player invite.

**Picking a Teammate:** When picking a team mate, there are three options:

• **Search for Player** - players can search for a teammate either by their username or email address. Players can also view their friends list and players they have previously played with who may be available.

• **Random** - matches a player with another randomly.

• **Social Graph** - if the player has connected their social account, they are able to invite their friends to play a game.
Social Graph: Players can use their social account (Facebook or other) to login into the game, enabling social graph features. Players with connected accounts are able to access their friends list and invite their connections to play, also their social media profile picture may be displayed in-game. The game does not post any content to the player’s wall without their permission.

Lobby: Before a game session begins, both players get acquainted with each other in the lobby. If it is the first time both players have played together, a team name maybe provided. The game countdown begins, once the players signal they are ready.

Friends: Friending allows a player to bookmark another player, so they can quickly access them and invite them into a quick match or to enter a league.

Communication: Table 7 provides the types of communication between player to player or system to player. Mobile players who enable push notifications may receive them as they appear, desktop players will receive an email (if given permission), otherwise all messages will be received when the player next logs into the game.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocal</td>
<td>Player - Player</td>
<td>Using a microphone players can talk to each other.</td>
</tr>
<tr>
<td>Visual</td>
<td>Player - Player</td>
<td>Using a web camera players can see each other.</td>
</tr>
<tr>
<td>Invite to Play</td>
<td>Player - Player</td>
<td>Send invite to player. Invitation to play is shown to the receiver.</td>
</tr>
<tr>
<td>Become Friends</td>
<td>Player - Player</td>
<td>Send invite to player. Message to accept or decline friend request is shown to the receiver.</td>
</tr>
<tr>
<td>Become Friends (Accepted)</td>
<td>System - Player</td>
<td>On acceptance, send message back to sender to notify them that their friend request has been accepted.</td>
</tr>
<tr>
<td>Achievement Awarded</td>
<td>System - Player</td>
<td>If achievement condition is equal to true, display achievement award to player.</td>
</tr>
<tr>
<td>Achievement Removed</td>
<td>System - Player</td>
<td>If a player’s record is broken by another player, their achievement is removed.</td>
</tr>
<tr>
<td>Rank Status</td>
<td>System - Player</td>
<td>Displays rank status to the player, showing if they have gone up or down. It also shows the nearest friend’s rank.</td>
</tr>
<tr>
<td>League Status</td>
<td>System - Player</td>
<td>Notifies the player of their current team league status. This will only be used to notify the 1st, 2nd and 3rd teams, and if the team enters a new league.</td>
</tr>
<tr>
<td>Player Offline</td>
<td>System - Player</td>
<td>A graphic component which shows when a player is online (green) and offline (red)</td>
</tr>
<tr>
<td>Update, and Game Errors</td>
<td>System - Player</td>
<td>Game updates, status information and error handling (report to developer).</td>
</tr>
</tbody>
</table>

Table 7: Ditto, types of communication
5.4.4 Gameplay

Figure 21: Ditto, gameplay flow
Annotation

1. **Picking a teammate** - the player selects an option to pick a teammate.

2. **Social account** - if the player has their social account connected, they can select from a friends list, otherwise they can choose to connect their account if they choose to.

3. **Lobby** - once a match has been found and the player is available, both players enter the lobby.

4. **Team** - if it's the first time playing together, a team is created, otherwise previous team data is retrieved.

5. **Level check** - ensures the team start on the correct level, either for the first time or continuing from previous progress.

6. **Game session** - during the game session, the challenge and obstacles are presented to the player. Specific game timers and controllers are loaded, such as achievements, scoring and physics data.

7. **Achievements** - tracks player's progress, once an achievement condition is met, the achievement is awarded to the player.

8. **XP/LP** - game session data, such as time taken, challenge completion, are used to determine how much XP and LP is awarded to the players.

9. **Skill feedback** - once a game session finishes, players are asked to provide feedback on how well they believe the other player did in terms of communication skills. Feedback results appear on the player’s profile.

10. **Results** - scores and bonuses (XP/LP) are accumulated and displayed to the player, including the level star rating.

11. **Progress** - the team advances to the next level, if they score two stars or above, otherwise they have to replay the level.
5.4.5 Leagues

The game explores a variety of tournament styles including group tournaments. In group tournaments there are no decisive final matches. Instead, all teams are ranked by examining the results of all the rounds, based on the amount of LP earned within the duration of the league.

- **Entry** – to enter a league the player needs to create a team, this enables them to earn league points (see 5.4.6)
- **Leagues** – teams progress within a league by earning LP. LP is earned by completing levels and unlocking achievements. Leagues are tiered, as teams advance, they transfer to the next league. Tiering prevents top players from always being at the top, giving no chance to lower level players. Below is an example of the first 5 leagues in the game. Players are scored by the amount of LP they earn within the league duration. LP accumulated outside the league duration does not count towards the final league score.

Players in teams who come 1st, 2nd or 3rd place in a league will receive a league achievement, stating the league and position.

- **Asynchronous Play** - leagues are played across a specific timeframe which could be hours or days, providing teams with enough opportunities to play when it's convenient for them.
- **Tiebreakers** - if two teams accumulate the same score, this results in a tie. Ties are broken by comparing the average time taken to complete a level over the course of the league duration of the tied teams. The lowest value is the winner.
- **Transfers** - A league lasts for a specified amount of time. Players are locked into their current league and cannot transfer to the next league until the current league's duration comes to an end.

5.4.6 Levels

A framing method for presenting players with challenges and obstacles provides the feeling of progression.

<table>
<thead>
<tr>
<th>League</th>
<th>LP Range</th>
<th>Duration</th>
<th>1st Prize</th>
<th>2nd Prize</th>
<th>3rd Prize</th>
</tr>
</thead>
<tbody>
<tr>
<td>No League</td>
<td>LP0</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Bronze III</td>
<td>LP20-29</td>
<td>5days</td>
<td>XP1000 +5LP</td>
<td>XP700 +2LP</td>
<td>XP400+1LP</td>
</tr>
<tr>
<td>Bronze II</td>
<td>LP30-39</td>
<td>5days</td>
<td>XP1700 +5LP</td>
<td>XP1200 +2LP</td>
<td>XP800+1LP</td>
</tr>
<tr>
<td>Bronze I</td>
<td>LP40-49</td>
<td>5days</td>
<td>XP2400 +10LP</td>
<td>XP1500+4LP</td>
<td>XP1010+1LP</td>
</tr>
<tr>
<td>Silver III</td>
<td>LP50-59</td>
<td>5days</td>
<td>XP2900 +10LP</td>
<td>XP1800+4LP</td>
<td>XP1300+2LP</td>
</tr>
</tbody>
</table>

Table 8: Ditto, league example requirements
Level Reward Pool - A set amount of XP and LP can be earned per level. If a player earns the maximum and replays the level, they do not receive any extra XP and LP. This prevents "gaming" the system, where players could replay the level over and over and receive more XP and LP each time.

Experience Points (XP)
Players earn XP by completing levels and achievements. Players can only progress to the next level when they receive a two star or higher rating (see below). If a team fails to complete a level, they will have to re-play it.

Reward pool used for each level:
- XP = time / 100 * (star % x level) x difficulty
- XP is a shared reward, both players receive equal amounts of XP

<table>
<thead>
<tr>
<th>Level</th>
<th>Time (s)</th>
<th>1 Star XP</th>
<th>2 Star XP</th>
<th>3 Star XP</th>
<th>Difficulty</th>
<th>Star %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>30</td>
<td>75</td>
<td>105</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>120</td>
<td>300</td>
<td>420</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>450</td>
<td>1350</td>
<td>1890</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>576</td>
<td>1440</td>
<td>2016</td>
<td>6</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 9: Ditto, experience point reward rate

League Points (LP)
Awarded for winning achievements, levels and league sessions, LP is awarded to a team and should not be confused with XP (awarded to individuals). LP is used as a measurement of a team's progression within the game.

5.4.7 Difficulty
As the player progresses, the levels become more challenging. Difficulty ensures the players experience a constant challenge. The game does this by adjusting:

- **Shape Diversity** - Increasing shape variation increases the opportunity for players to make errors. Players need not only to work together, but to use a more advanced level of communication. Below is an example of shape diversity: form, sides, colour and pattern:
- **Shape Combinations** - There are multiple factors to shape combinations: position, colour, elevation, rotation, multi-piece and pattern. These factors, as they are combined, make the game more challenging. For example, in the diagram below combination 6 not only includes all the previous factors, but also requires the players to improve on their spatial awareness skills.

- **Constraints** - applying constraints can make the simplest of shape combinations more challenging. Below are a few examples of constraints.
1. **Scoring frequency** - as the player progresses through the game, the rate at which XP and LP is awarded decreases. It becomes harder to earn XP and LP, which slows progression down.

2. **Social Challenge** - introducing the player to the games social system and providing them with enough incentive and freedom to move at their own pace.

3. **Introduction** - player is introduced to the main mechanics of the game from the builder's perspective. Soloplay.

4. **First taste of social** - player begins a quick match with another player. Player learns about the architect role.

5. **Friends** - player begins to invite people they played with before as friends or connects to their social account to invite their network of friends.

6. **Team creation and league participation** - player is confident with the game mechanics, has a friends list and begins to create a team for competitive reasons.

### 5.4.8 Score

**Star Rating** - Provides a quick indication of how well a challenge was completed. Requirement thresholds may change as the levels increase in difficulty. *For example, if the players were on level 2, and the time to complete was 60 seconds, and instead they complete the level in 43 seconds, then the player would be rewarded with 3 star rating. As 43 seconds is more than 70% of 60 seconds.*

- **1 star** - 20% of the maximum score earned for the particular level,
- **2 star** - 50% of the maximum score earned for the particular level,
- **3 star** - 70% of the maximum score earned for the particular level.
  - The percentages increase with the difficulty.
5.4.9 Rank
Total score accumulated by the player and their respective team is compared and compared (highest first) with other players’ scores. There are two types of ranks:

Global Ranking
- Player - player ranks are determined by the amount of XP accumulated overall.
- Team - team ranks are determined by the amount of LP accumulated overall.

League Ranking - league rank teams based on how much LP they have accumulated within the duration of the league.

5.4.10 Achievement
The hall of fame displays breakable and permanent records of achievement. Players who enter into permanent records are displayed forever compared to breakable records which can change. The table below shows examples of both permanent and breakable records.

<table>
<thead>
<tr>
<th>#</th>
<th>Achievement</th>
<th>Score/Date</th>
<th>Recipient(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Permanent Records</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>XP Legend</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>First player to achieve 1,000,000 XP</td>
<td>02/11/2016</td>
<td>BrandonX</td>
</tr>
<tr>
<td></td>
<td><strong>Star Champion</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>First team to win 20 tournaments</td>
<td>25/11/2016</td>
<td>Sarah1 &amp; AliceZed</td>
</tr>
<tr>
<td></td>
<td><strong>Gameplay master</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>First team to complete 100 levels 3 star rating</td>
<td>18/12/2016</td>
<td>RossWi4</td>
</tr>
<tr>
<td></td>
<td><strong>Breakable Records</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Gameplay Speedster</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fastest team to complete 100 levels</td>
<td>06/11/2016</td>
<td>Emma &amp; Kate</td>
</tr>
<tr>
<td></td>
<td><strong>Team XP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Highest scoring team</td>
<td>14/12/2016</td>
<td>Will &amp; Dale</td>
</tr>
<tr>
<td></td>
<td><strong>XP Guru</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Highest earned XP level</td>
<td>03/11/2016</td>
<td>SophieKe4</td>
</tr>
</tbody>
</table>

Table 10: Ditto, achievement example conditions
5.4.11 Game User Interface & Concept Art

Figure 26: Ditto, UI wireframe and concept art
### 5.4.12 Game Screen Flow

**GAME SCREEN FLOW**

![Diagram showing game screen flow](image)

**Figure 27:** Ditto, game screen flow

### 5.5 Technical Specification

#### 5.5.1 Development

- Game engine: Unity3D
- Programming language: C# and Javascript

#### 5.5.2 Target Requirements

- Hardware: Mobile Device or desktop PC
- Input: Touch screen or Mouse
- WebGL based player. Firefox, Edge or Chrome. Optionally available as an app on OS: Android 5 or higher, iOS 8 or higher, Windows 7 or higher
- Network: Internet connection
6  IT ALERT – PLAYGEN AND SPL

6.1 Introduction

IT Alert is a multiplayer cooperative game for 4 - 6 players. The players assume the role of specialist IT service staff, called in to maintain their biggest client’s network infrastructure. During each game round, the players must work together to detect, analyse and resolve various problems occurring throughout the network, thus ensuring the client company’s reputation. The players will be in constant communication with each other. The players win if they survive the round, being scored based on the company's remaining reputation and will lose if this value ever falls to zero.

Players will develop their skills in communication and involvement within a team working situation. Optional secret objectives support raising awareness of other player's actions and their effects. The game will utilize RAGE assets for team formation, adaptive level of difficulty, player monitoring and feedback. Analysis of these are in section 6.3.

6.2 Skills & Learning Objectives

The general learning goal is to train students in vocational education of ICT in communication and soft skills relevant for employability. In particular the game focuses on three key skills which include effective communication, teamwork and management of interpersonal relationships and resources.

Communication: Interplayer communication is through voice chat. Player must relay information and verbally coordinate with each other in order to compete the level. The optional secret objectives assigned to each player may require them to behave in a particular way. The secret objectives are used in conjunction with the post-game reflection and feedback questions to get the players thinking about the behaviour of others, thus practising social and emotional awareness skills.

Team Work: Limited resources and movement restrictions support the players to share and coordinate with one another efficiently. The puzzle structures gear each player toward an assumed role within the team (e.g. a player put in a position to constantly relay information may assume the role of the team leader). These roles are purely assumed and not set by the game. These will create a variety of working relationships over the course of the game session, demonstrating the benefits of camaraderie.

Management & Efficiency: Threats are introduced as the game transpires. Threats may occur concurrently and there is a possibility of situations where not enough resources are available to solve all obstacles at once. The threats have been designed to replicate the stress of a busy work environment with staggered deadlines, requiring the players to manage their time and stress to minimize losses within the game’s context.

6.2.1 Strategy

The game uses a collaborative learning approach as the main underlying pedagogical concept (e.g. Dillenbourg, 1999). A group of individual players acts together to solve a joint problem. The problems are presented within a meaningful and authentic context. The collaborative learning scenarios implemented in IT Alert thus incorporate aspects of problem-based and situated learning. The game covers the concrete experience and active experimentation phases of Kolb’s learning cycle. The other phases exist between instances of the game, through discussions in the physical learning environment and facilitated by the games feedback interfaces.

6.2.2 Feedback

Feedback and dynamic difficulty adjustment are made by monitoring interactions within the game. This provides relevant learning and maintains player motivation by using RAGE assets.
and incorporating in-game and after-game feedback on player and team-mate performance. See section 6.4.5 for examples of feedback.

**Example:** Player 1 responds to questions by stating that they felt highly distressed during game play. During the post-game discussion, Player 2 states that Player 1 was the most distressed, so her feedback would reflect this and provide guidance for future games. Remedial actions are suggested which could be practised in the next round.

Feedback is split between game progression and social awareness. The overall success of the team is shared by all the players. This ensures cooperation, making it clear that they win and lose together. Social & emotional awareness is graded upon the individual's ability to correctly analyse the behaviour of other players, the results presented to the player in the post-game feedback and reflection.

### 6.3 RAGE Assets

**Interaction Tracking (T2.1A) / Storage & Analytics (T2.1B)**

Captures keyboard, mouse and voice input from the player, alongside event data from the game. This can be sent to the server side component for later analysis.

**Authentication & Authorization (T2.4A) / Dashboard & Analysis (T2.4B) and Performance Statistics (T2.2E)**

Assigns player a performance rating based on how well they solve the puzzles required to nullify threats. The rating is based on comparing the success of other players.

**Social Gamification Components (T3.6A)**

Formulates teams based on individual performance, motivation and behaviour. For example, a team of players who all show signs of great leadership vs a team consisting of no leader-like players. Configurations of the teams may also be set by the teacher. The asset will handle all the storage of player variables such as score, lives, resources, achievements and goals.

#### 6.3.1 Asset Flow

The diagram in Figure 29 shows the flow of information between the RAGE Assets and the game components that use them. The red numbers are used to sequence the explanation of this diagram.

1. Touch, mouse & keyboard movements, and any communication actions are made by the player.
2. This information is captured by the Interaction Tracking asset. Event data from the current game state is also captured.
3. The data is stored on a remote server.
4. The user friendly interface of the server allows for easy aggregation of the data.
5. The server is secured by the authentication asset so only authorized users have access.
6. Performance Statistics compares the success of the player (taken from the Interaction Tracking asset) against other players within the same game state. This assigns the player a performance rating for each measured skill based on the overall average.
7. The game state represents all the components that formulate the current state of the game at a point in time. The diagram shows interventions from Performance Statistics.
8. The Social Gamification takes the performance comparison and the motivational state to produce useful rankings of the players. Based on predefined rules, team configurations are generated for subsequent rounds of the game based on these rankings.
6.4 Game Design

6.4.1 Structure

The game begins with a short tutorial on how to play, followed by a series of game rounds. Between each round, the teams are reformed with different configurations.

Figure 29 contains a list of game areas and how they relate to each other. After the teacher and students arrive at the game’s website, the game begins at the Main Menu screen and so the flow begins. The teacher has direct access to the teacher panel but may also take part in gameplay if desired.
**Menu & Settings:** Landing page of the game. Players may start a game or tutorial game and change the settings of the microphone, volume or their player's avatar appearance such as colour.

**Lobby:** Shown upon starting the game. Uses the matchmaking system to put players into teams. The team positions to be filled by joining players is also displayed.

**Game:** Main game and play sessions.

**Tutorial:** Single-player demo version of the main game with fewer game elements and tutorial instructions on how to play.

**Feedback:** After each round of the game, players will view a personalised feedback screen. Providing feedback on how well they did and information about how well they and others believe they did or could improve on. This information is also fed back into the Lobby for matchmaking subsequent rounds and into the Teach Panel to for the reflection and discussion phase.

**Teacher Panel:** Exclusive to the teachers, no access to students. Includes visualisation tools for data gathered on student behaviour and interaction (collected by RAGE assets and questionnaires). May also incorporate functions to start, pause and stop games, matchmaking options for team formation and direct communication with players. The questionnaire data can be printed out and brought to the post-game reflection.
6.4.2 Game Dynamics

Game World: The game world resembles a network infrastructure graph, where the subsystems are nodes and the connections are routes between them. The network is presented from a top-down perspective. Avatars, subsystems and threats (apart from viruses) are visible to all players within the game’s main screen.

Subsystems: The subsystems act as both traversable locations and bonuses for the players, with ability for upgrading and providing bonuses, and have two main variables; integrity and security.

Limited Movement: Connections act as pathways for both players and spreadable threats. Each player may move their avatar across the system from one subsystem to another. Once a command is given, they cannot make any changes to their movement until they arrive at the next connected subsystem. Here the leader of the team may need to support organising their team and deploying them strategically across the map. The time taken for this movement is long.

Figure 30: IT Alert, 4-player game world layout
enough so that if another avatar was to enter a connection in the opposite direction, other avatars along the connection will be sent back to the original subsystem they travelled from. This prompts players to coordinate their movements, and efficiently place themselves around the map to respond to threats effectively.

6.4.3 Challenges

The players must collectively overcome a variety of challenges to succeed within the context of the game. They vary in terms of difficulty, complexity and number of participants required (team or solo challenges).

**Maintenance:** Defined by two variables, the players must ensure that this value is maintained at all times. Integrity represents a system’s health and performance. The value will slowly decrease over time, if ever brought to zero, bonus is disabled. A system's integrity can be repaired by any player who is located in that system, at the cost of resources. Shield, a security against virus threats. Value will decrease in unpredictable spikes. A lower security rating increases the likelihood of a (hidden) virus. Tools can be used to increases the security level and reveals the presence of any viruses.

**Threats:** External problems which affect the system and are typically the cause of symptoms found in alerts:

- **Cyber Attack:** aggressive attack on a subsystem which reduces the maintenance value to zero. This can be countered by a player using the trace tool. Any resources left on the target subsystem are removed by this threat.
- **Virus:** hinders subsystem bonuses and can spread along connections, leaving malware in their wake.
- **Malware:** exist in the connections between subsystems that viruses have spread to. Slow down player movement and lowers the reputation of the client company (overall score). Removed by passing along the routes with the cleaner tool.
- **Power Cut:** a system-wide failure which reduces all maintenance values to zero. This is part of the dynamic difficulty adjustment feature, which occurs if the players are finding the game too easy.

6.4.4 Progression

**Upgrades:** Every subsystem uniquely benefits the players by providing improvements, abilities or resources. These can be upgraded for added bonuses using the cash resource. When a threat is affecting a subsystem, its associated bonus will be hindered in some way.

**Rewards:** Players will be rewarded for neutralising threats in the form of bonus resources.

**Difficulty:** Dynamic difficulty adjustment is influenced by the output of the RAGE assets that monitor the players. A change in difficulty effectively manipulates the combination and frequency of the challenges from Section 6.4.3 which the players are faced with.

**Resources**

- **Consumables:** used to repair and upgrade subsystems, generated over time and must be moved by the player's avatar to the desired subsystem.
- **Tools:** used to deal with particular threats. Include the scanner, tracer & cleaner, which are housed at the SECURITY SUITE subsystem.
- **Score:** represented in-game by the company's reputation. Serves as both a measurement of the team's overall success and a life resource (If the reputation ever falls to zero, the players lose).
6.4.5 Sample questions and feedback

The following are a list of suggested performance feedback and reflective questions. Where possible the system will measure and feedback on these factors, additionally players are encouraged to reflect, rate others and discuss them in order to support generalisation of the skills learnt.

- These questions reflect the secret objectives assigned to the players.
  - Which player worked the hardest?
    - How did their actions benefit the rest of the team?
  - Which player helped the least?
    - How did their actions hinder the rest of the team?
  - Which player took charge?
    - Did their actions hinder or benefit the rest of the team?
    - If you had taken charge, what would you have done differently?
  - How did you, (player x), feel playing the game? (happy, frustrated, etc.)
    - How did you think player x (if you are not player x) felt during the game?
    - Who enjoyed the game the most?
    - Who was the most distressed?
- What about their actions lead you to this choice?
- How did their emotional state affect the rest of the team?
  - Which player communicated the least?
    - How did this affect the team?
  - Which player caused the most confusion?
    - How did this affect the team?
6.4.6 Game User Interface Example

See Figure 31 below for an example of the main game screen.

![Figure 31: IT Alert, game HUDs](image)

6.4.7 Mechanics

Subsystems

The table below is a summary of all the subsystems and the effects upgrades and threats have on each.

<table>
<thead>
<tr>
<th>SUBSYSTEM</th>
<th>EFFECT</th>
<th>FAIL</th>
<th>VIRUS</th>
<th>UPGRADE 1</th>
<th>UPGRADE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Database</td>
<td>Reputation Modifier</td>
<td>Lower Rep</td>
<td>Lower Rep</td>
<td>One time Rep bonus</td>
<td>One time Rep bonus</td>
</tr>
<tr>
<td>Security Center</td>
<td>Tools</td>
<td>-</td>
<td>-</td>
<td>Tool Speed</td>
<td>Tool Quantity</td>
</tr>
<tr>
<td>Email Server</td>
<td>Comms</td>
<td>Comms down</td>
<td>Comms distortion</td>
<td>Audio comms</td>
<td>Remote Comms</td>
</tr>
<tr>
<td>Firewall</td>
<td>Defence</td>
<td>Cyber Attack</td>
<td>-</td>
<td>Integrity bonus</td>
<td>Integrity bonus</td>
</tr>
<tr>
<td>Power Management</td>
<td>Generates Megaflops</td>
<td>Income Halt</td>
<td>Income Penalty</td>
<td>Income Bonus</td>
<td>Income Bonus</td>
</tr>
<tr>
<td>Web Server</td>
<td>Generates Cash</td>
<td>Income Halt</td>
<td>Income Penalty</td>
<td>Income Bonus</td>
<td>Income Bonus</td>
</tr>
<tr>
<td>Admin</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Autonomous Storage</td>
<td>Global Storage</td>
</tr>
</tbody>
</table>

Table 11: IT Alert, subsystems
Admin: Starting point for all the players, immune to threats. The admin server also serves as a storage for any unused resources the players may have, but must be moved here manually. Upgrading admin will allow automatic surplus resource storage, as well as leader role assignment. The leader will be able to send commands to each player, but must be voted in once the upgrade is complete. An alternative to the leadership upgrade is a global storage upgrade. This allows players to store resources in subsystems other than the admin server.

Email Server: Though it has no functional use, any problems affecting the email server will cause disturbance in the players’ communication ability with the potential of a total loss of contact. Upgrading the mail server will allow for more efficient methods of communication for the players.

Power Management: Generates energy resource (Megaflops) for the players, used to repair subsystem integrity. Threats will slow or even stop the rate of generation and upgrading will increase the rate.

Web Server: Generates cash resource for players to upgrade subsystems. Like the other generator, threats slow down the production rate and upgrades increase it.

Security Center: Houses the tools which are used to resolve security threats in the network. Upgrades increase efficiency and quantity of the tools.

User Database: Affects the players’ overall score (reputation). Also functions as a life resource as a total loss in reputation results in game over. Threats affecting this subsystem will cause drops in score.

Firewall: Acts as a defence from harder threats (cyber attacks), which become more likely if the subsystem has any negative effects on it. Upgrading the firewall causes system-wide integrity bonuses.

6.4.8 Resources

Scanner: Used to increase security in subsystems and reveal any hidden viruses. There is a limited number of scanners (initially only one). Player must pick it up, drop it into the desired system and interact with it to start the scan. The player is free to perform other actions while the scan is running. Scan will end when the security of the system reaches 100% or when a player interacts with it again.

Tracer: Used to combat a cyber attack on a subsystem. Like the scanner, a player must manually carry it to the subsystem under attack. However, the player must remain in the subsystem to maintain the trace. The speed of the tracer is increased by the number of players sharing its location. The trace will end when a player either leaves the subsystem or picks up the tracer.

Cleaner: Used to delete any malware in the connections between subsystems. The cleaner’s functionality mirrors a vacuum cleaner. It automatically initialises when a player picks it up and turns off when a player drops it. The player carrying the cleaner will automatically remove any malware the player moves over.

Megaflops: Resource used to make repairs on subsystems.

Cash: Resource used to upgrade subsystems.

Heads up display
The HUDs will feature what the player is currently holding, the team resources, and other players on/off mic indicator.

6.4.9 Tutorial
Prior to being placed in the matchmaking lobby, the players will be placed in a short tutorial, which is essentially a single-player demo version of the game played on a smaller scale (3 subsystems: ADMIN, SECURITY SUITE, WORKSTATION, and 2 connections). Tooltips will
inform the player about the game's controls. A single alert will appear in the WORKSTATION and the tooltips will explain how to respond, diagnose the problem with the ADMIN server and resolve it using the SECURITY SUITE. The basics of maintenance and upgrading are also explained.

6.4.10 Controls

Movement: Right-click or touch (if on portable platform) to move from one subsystem to another. The avatar will take the shortest possible route there, if not adjacent.

Interactions: This is the set of interactions each player's avatar can perform.

- **Pick up:** takes a resource and places it in the player's hand. The player can then freely move with this resource. A player can only carry one resource and may not perform any other action while carrying, apart from drop.

- **Drop:** moves the resource from the player's hand to the current subsystem.

- **Repair:** repairs the integrity of a subsystem. The required amount of repair resource (megaflops) must already be placed on the subsystem, else the interaction will not work.

- **Upgrade:** upgrades the bonus of a subsystem. The required amount of cash resource must already be placed on the subsystem, else the interaction will not work. Higher bonus levels require larger quantities of cash.

The HUDs will feature what the player is currently holding, the team resources, and other players on/off mic indicator.

6.5 Technical Specification

6.5.1 Development

- Game engine: Unity3D
- Programming language: C# and Javascript

6.5.2 Target Requirements

- **Hardware:** Mobile Device or desktop PC with microphone headset
- **Input:** Touch screen or Mouse and keyboard
- **WebGL based player. Firefox, Edge or Chrome. Optionally available as an app on OS:** Android 5 or higher, iOS 8 or higher, Windows 7 or higher
- **Network:** Internet connection
7 INTERVIEW SIMULATION FOR JOB SEEKERS – BIP MEDIA AND RANDSTAD

7.1 Introduction

The game "Interview Simulation for Job Seekers" (ISJS - provisional name) uses the form of a classic point&click adventure game to offer users customized job search courses and practise for job interviews.

7.1.1 Game Overview

At the beginning of the game, the user must enter his name and his qualification from which he will be directed to a specific course. A quick questionnaire will then determine his psychological profile.

Once the player has made his CV and cover letter, the real CV and Cover Letter of the player will be used in their digital format (DOC, PDF, etc.) or scanned to simplify the process.

The first meeting of the player will be with a Randstad Personal Advisor (3D virtual character) who will ask various questions to make a personal assessment, establishing the player’s professional career and define the professional project. The advisor will then provide a first series of personalized advice: job search strategy, online activities, organization, etc.

After this first phase, a first course will be defined and the user will be free to visit the various places of a city accessible through a 2D map: the Randstad Center, the player’s apartment and, of course, many companies looking for staff... In each of these places, the player will be able to do a particular action: consult companies’ profiles, receive interview request, get advice, etc.

When players receive a request for an interview, they will of course have to go, on time, at the place of the appointment, using the general map.

During the interview, the potential employer (3D virtual character) will ask questions aloud (subtitles) and the player will respond in natural language (microphone), while emotions are

Figure : ISJS, interviewing phase
tracked by a camera. Note that this game is developed for Randstad France and the language used will be **French only**.

Each player answer will calculate a **score** and this score will decide at the end of the interview, if the player is hired or not. All data will be transmitted to the server for **analysis** and adaptation of the course to the player’s level.

The detail of the score and questions / answers will be accessible after the interview and, if it is a failure, the user will be able to return to the Randstad Center for analysing the failure with the advisor (not only the interview, but all player's actions since the last visit at the center), and **repeat the entire course**…

When a stage of the player’s course is validated, the player receives a **certificate** and is not forced to complete this step when restarting the entire course. When an interview leads to a favourable response, the player receives a final certificate.

### 7.1.2 Learning outcomes and learning context outline

For players, the aim is to improve the way in which they handle finding work, in particular **job interviews**. The game focus on improving the player’s “presentation” skills during an interview, the way players manage to present themselves. With this in mind, ISJS will set up three systems:

- **natural-language dialogues** that simulate job interviews;
- a simplified system to create documents (CV and Cover Letter) and perform job searches;
- and **detailed advice** given by Randstad experts

ISJS is a **downloadable single-player** game. Players could use it **at home**, but in this case there would be no emotion detection with the specific sensor (RAGE asset 2.3D) that Randstad will use. Furthermore, **on-site** test sessions could be performed on Randstad's premises.

In any case, an **Internet connection** will be needed because some RAGE assets need a server and the program will gather specific data to improve the game, its algorithms (in particular related to the creation of dialogues) and its **databases** (key words, job-specific skills and competencies, etc.).

### 7.1.3 Component usage outline

RAGE Components will primarily play a role in the simulation of natural-language dialogues, in order to make them as “human” and realistic as possible.

Following the establishment of an initial **psychological profile** (component 3.4E), the player will meet some **animated 3D characters** (components 3.2A and 3.2B).

Dialogues with these characters will be partly scripted by Randstad experts with a **specific editor** (component 3.3F), but also influenced by feedback from various components: **emotion detection** (components 2.3A and 2.3D) and a **recommendation system** based on skills and competencies (component 3.4C).

Characters (PNJ) will then be able to express themselves via a **voice-synthesis** and text-reading system (components 3.3J). User responses will be interpreted via **voice-recognition and analysis modules** (components 3.3A, 3.3H and 3.3L).

Lastly, a set of components will enable **identification** of the user, as well as **storage** and **analysis** of specific user data for statistical and program-improvement purposes (components 2.1A, 2.1B, 2.4A, 2.4B and 8.2A).
7.2 Learning Objectives

The learning goals of the game are to provide training for job seekers on job application skills, creating application documents and on job interviews. In fact, the main skill practiced is about “selling yourself” in a job interview and, at least, this will mainly be done during the dialogues in natural language with virtual recruiters.

The user will receive a lot of general advice about CV, cover letter, the way to dress for an interview, etc. But giving this general advice is not the main objective of the game. The important advice will be received by the player after each interview.

Affording different interview style of recruiters who will use different interview approaches with different objectives, the player will be quickly prepared for various situations during a job interviews.

ISJS is a solo game and the users will be able to use it as long as they want, even if the “game is finished”, i.e. if the player is virtually hired. The idea is to increase user confidence and improve the understanding of the recruiters’ perspectives for the player. So there is no limit to the number of interview simulation to be played.

In an interview, for each user answer, after transformation of the speech into a text (RAGE component 3.3H), the game will search in a database of keywords to identify the answer options and give a “score” to the answer. This score indicates the accuracy of the answer.

At the end of an interview, all these scores will be added to get a general score which will determine if the player is recruited or not.

Regarding each score for each question, and the general score, the game will determine new advice for the Randstad Personal Advisor (RPA). This advice will be available for the player at the Randstad Center or in his “Randstad Booklet”.

After an interview, the game will send to the server the different scores, the list of the questions and keywords used. This last part will be used to measure the accuracy of the scenario followed, the accuracy of the inputs of RAGE Component which can influence the dialogues (2.3A, 2.3D and 3.4C). Then the server will eventually send some modifications to the databases (keywords, scenario, etc.) to optimize the accuracy of the game regarding the player’s skills.

7.2.1 Pedagogy

The pedagogical approach of situated learning is taken up, which claims the creation of a meaningful context in which the targeted skills can be learned (Brown, Collins, & Duguid, 1989; Lave & Wenger, 1991). Situated learning can be characterised by the following features (Anderson, Reder & Simon, 1996; Wilson, 1993):

- Learning is related to actions of real life and everyday situations
- Knowledge is acquired in relevant situations and is transferable to similar situations
- Learning is a result of a result of social processes of participating, thinking, perceiving, problem solving, and interacting
- Learning is not separated from the world of action

In the ISJS game learning activities enable the learner to appropriately contextualize the information. This is achieved by relating the learning contents to an authentic game world and to the needs and concerns of the learner. This shall support transfer of the knowledge and skills acquired in the game to real-world situations and job seeking context.

In addition, the game incorporates ideas of guidance and scaffold instruction, which argues for the need of giving learners assistance and gradually decreasing this support according to their
learning progress (e.g. Kirschner, Sweller, & Clark, 2006; Larking, 2002; Lipscomb, Swanson, & West, 2004). Guidance is realised in the game by providing an NPC as personal advisor to support and advise the player.

### 7.3 Rage Assets used

<table>
<thead>
<tr>
<th>TASK</th>
<th>COMPONENTS</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identification, tracking and analytics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1A</td>
<td>Client-side interaction tracking</td>
<td>To track the player’s interactions in the game, so that they can be submitted to a server for further analysis.</td>
</tr>
<tr>
<td>2.1B</td>
<td>Server-side interaction storage and analytics</td>
<td>This asset offers a ready-to-deploy server-side implementation of a data collection and storage service. It takes care of authentication and supports current standards for exchange of interaction data.</td>
</tr>
<tr>
<td>2.4A</td>
<td>Server-side Authorization and Authentication</td>
<td>This asset provides a central location where clients can authenticate and locate server-side assets, including analytics.</td>
</tr>
<tr>
<td>2.4B</td>
<td>Server-side Dashboard and Analyst</td>
<td>Analysis of the data received.</td>
</tr>
<tr>
<td>8.2A</td>
<td>Evaluation Asset for Applied Games</td>
<td>This asset will allow in-game evaluation of applied games based on users’ interactions with a game.</td>
</tr>
<tr>
<td><strong>Emotion detection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3A</td>
<td>Real-time Emotion Detection</td>
<td>The asset produces real-time emotion based on facial expression. It uses simple, low cost device: a standard webcam.</td>
</tr>
<tr>
<td>2.3D</td>
<td>Real-Time Arousal Detection Using Galvanic Skin Response</td>
<td>To measure players’ arousal via a special tool and to complete what can be detected by the component T2.3A</td>
</tr>
<tr>
<td><strong>NPC animations and behaviour</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2A</td>
<td>Virtual Human Controller</td>
<td>Virtual Human Controller generates the expressive behaviour of the character including lip-synchronized speech animation, gestures, facial expressions and gaze.</td>
</tr>
<tr>
<td>3.2B</td>
<td>Motion Builder</td>
<td>Motion Builder comes with sample motion capture data of emotional, conversational and idle behaviours in BVH.</td>
</tr>
<tr>
<td><strong>Voice recognition and text analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3A</td>
<td>ReaderBench - Semantic Models and Topic Mining</td>
<td>Searches for keywords in documents. Search for keywords in the players’ answers.</td>
</tr>
<tr>
<td>3.3H</td>
<td>Speech to Text</td>
<td>Voice recognition, which enables text to be produced on the basis of a recording. Converts player responses given during a dialogue into text (mainly keywords).</td>
</tr>
<tr>
<td>3.3L</td>
<td>Speech Act Identification</td>
<td>Identifies intentions during a discussion in natural language and influences dialogues generated.</td>
</tr>
<tr>
<td><strong>Dialogue creation and voice synthesis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3F</td>
<td>Communication Scenario Editor</td>
<td>To develop a scenario between the player and a virtual character. It is the dialogue management tool.</td>
</tr>
<tr>
<td>3.4C</td>
<td>Competence-based Personalisation and Recommendation Asset</td>
<td>Personalization of interviews’ dialogues based on the evolution of the player’s skills.</td>
</tr>
<tr>
<td>3.3J</td>
<td>Text to Speech Asset</td>
<td>Voice synthesis for virtual characters.</td>
</tr>
<tr>
<td><strong>Psychological tests</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4E</td>
<td>Game Configuration Asset</td>
<td>Beginning-of-game questionnaire</td>
</tr>
</tbody>
</table>
Table 12: ISJS, RAGE assets used
7.4 Game Design

Interview Simulation for Job Seekers (ISJS) enables the simulation of a real job search experience. Players will therefore be able to try job interviews a number of times, return to the Randstad Center several times to analyse their situation.

The game could therefore be played over a period of several days, with a system to save player progress. The game would be saved locally, but interaction data would be gathered during game sessions (player profiles and progress, keyword databases, skills databases, logs, etc.). This data be transmitted to servers for analyses and software improvements.

The game is made up of several “places” that the player will be able to go to. Each place is represented in 3D, with 3D NPCs, in a first-person view (no player avatar will be shown on the screen). Movements between various places will be shown on the 2D map.

7.4.1 Game Structure

The entire game will be built around a 2D map of an imaginary town (see Figure 33), which will enable the player to move between the various places shown in 3D: the Randstad Center, the player’s apartment, recruiting companies, etc.

The game begins with a special visit to the Randstad Center during which the player’s professional profile and motivations (CV, cover letter, psychological profile, etc.) are defined.
7.4.2 Player Profile

The player’s profile is created using the information entered at the beginning of the game and the result of the psychological test, in addition to the CV and cover letter.

It will also be linked to a log of movements on the map (places and times), as well as various documents that may be collected (job interview invitation letter, for example).

Elsewhere, as users progress through the game, they will have access to a panel of results, highlighting in particular changes in job interview performance results. All of this information will be accessible to users throughout the game.

Certain information linked to the profile, however, will not be accessible to players, for example results of the psychological test at the beginning of the game.

To give another example, a database of keywords characterising the player will also be created to guide NPC dialogues, which the player will not be able to access.

According to various parameters (creation of documents, reactions during interviews, etc.), this profile will be developed during the game in order to guide game difficulty and progress.

7.4.3 Walkthrough and main features

The first game screen simply asks the player to enter information such as; Surname, First name, Age, Gender, Profession (drop-down menu), Specialisation (if necessary, drop-down menu).

Next, RAGE Component 3.4E will propose a short test to establish an initial psychological profile. Randstad experts use the RIASEC model (http://www.monemploi.com/riasec) of personality structure. This test is simply a Multiple-choice questionnaire (about 15/20 questions) with a fast check-box system. It could also use voice synthesis for a faster use (RAGE component 3.3G).

Next, the player enters the 3D world of the Randstad Center and is welcomed by his or her Randstad Personal Advisor (who is assigned according to the player’s age and gender, from a choice of few graphical models).

Most interactions between the player and NPCs will take place via a natural-language voice interface (voice synthesis and recognition). NPCs will use a library of animations and lip synching (RAGE component 3.2A&B).

The Randstad advisor will, therefore, use speech when interacting with the player for the first time. Following a brief introduction/presentation, the advisor will ask the player to first create a CV and cover letter.
7.4.4 Creating a CV

If the player already has a CV, in digital form (PDF, DOC or scan), this can be directly integrated into the system in order to simplify the tedious process of manually entering data.

RAGE Component T3.3A will be used to extract various keywords and recreate the “in-game” CV, which may be different from the player’s real CV (a fictitious address will be used, for example). The players will be able to edit or create their CV.

Next, the player will be able to choose between several CV layouts. Choosing the correct CV layout according to employers can be decisive.

Next, the first lines of the CV will be automatically filled in using information provided at the beginning of the game. Fictitious information will be added purely in the interests of game consistency: address, telephone number, email, etc.

Four essential sections should then be completed, in any order the player wishes:

- Education
- Experience
- Knowledge
- Various (hobbies, writings, etc.)

For each section, the user will proceed line-by-line. The user will therefore be able to add, delete and edit lines in each section.

In the Education and Experience sections, these lines will of course correspond to various qualifications or jobs, sorted by date.

To complete a line, according to the section of the CV, the player will have a certain number of drop-down menus and fields to fill in by entering data on the keyboard.

For a line in the Experience section, for example, the player can first select start and end dates using drop-down menus, then the position held with another drop-down menu (the list is defined according to the profession). The player can then enter the employer’s name in a free text field, and list various job-related responsibilities in a description field.

The creation of a CV can be somewhat tedious. This is why it is really necessary to simplify and automate the task as much as possible. In addition, the process will be made more enjoyable with contributions from the Randstad advisor.

Once the CV is complete, it is saved along with the player profile. Subsequent modifications can then be easily made by returning to the Randstad Center to see the document creation advisor.

The program will then analyse the player’s CV to extract keywords that will be used to guide later discussions (with a Randstad advisor or recruiter). This keyword database will then be compared with skills/knowledge databases in order to refine the player profile.

7.4.5 Creating a cover letter

As the CV, the Cover Letter will be able to be integrated to the game in a digital format (PDF, DOC or scan). And the RAGE Component T3.3A will be used to extract various keywords and recreate an “in-game” Cover Letter.

If it’s not the case, creating a cover letter will be done by dragging and dropping ready-made sentences! However, their relevance in relation to the job sought and employer targeted, their organisation and simply their presentation could all have a real impact on interview results (see below).

The basic idea in this module is therefore to properly understand instructions and advice given by the Randstad advisor and then to apply it in a kind of puzzle.
Choices made by the player to create the cover letter are then analysed by the program, which will extract a certain number of **keywords** to add to the player profile.

### 7.4.6 Dialogues with the Randstad Personal Advisor

During the first game session, following the creation of the CV and cover letter, the player will have to attend an **initial interview** with his or her Randstad Personal Advisor (RPA).

The aim of interviews with the RPA is essentially to check on the player’s job search **progress**. Previous results are analysed and the RPA gives various **related advice**.

This initial interview will therefore concentrate on the player’s past employment history, in order to establish a **career plan** along with him or her.

For example, while a user’s CV shows many sales-related competencies, and only a few related to marketing, he or she insists on finding a new job in marketing. In such a case, the Randstad advisor will help the user assess the difficulty of doing so, organise him or herself, establish job search strategies, guide online activities, etc.

---

In his or her sentences, the RPA will include certain advice extracted from a **database** created by Randstad Search & Selection experts. Each piece of advice given by the RPA via the voice synthesis module will also be recorded in a “**Randstad Booklet**”, which will be accessible in the Results Panel.

Sentences spoken by the RPA are based on a **scenario** scripted with the Communication Scenario Editor (RAGE Component 3.3F). But they can be influenced by the keywords extracted from the CV and cover letter, as well as those detected in the player’s answers/responses. Furthermore, RAGE emotion detection assets may also periodically influence dialogues in order to give them a more “human” dimension and to refine certain points (the player’s position and posture, for example).

Following the initial interview, the player will be free to move around the map of the town. Before this, the player will only be able to go to his or her flat. Subsequently, the player will be able to visit companies to attend job interviews. In any case, the player will be able to **return to the**
**Randstad Center at any time.** Here, he or she will be able to modify the CV or cover letter if necessary, or have further conversation with the RPA.

**Subsequent dialogues with the RPA** will be based on knowledge gathered about the players, their job interview dialogues and results. Advice given can therefore vary and be refined with each attempt made by the players. Some advice can also be invalidated and other advice can simply vary slightly. All of this is recorded in the Randstad Booklet.

### 7.4.7 Movement on the map and perception of time

To move from one place to another on the map, two clicks suffice: the first will display additional information on the destination and the second will approve the journey. Additional information will include:

- a brief description of the place,
- possibly a reputation indicator when it is a company,
- the journey time.

In ISJS, all actions performed by the player correspond to a certain number of hours or days. When first visiting the Randstad Center, the time taken for each of the player’s actions (creation of a CV and cover letter, interviews with the RPA) is measured in real time. This will also be the case for all interviews, as well as time spent in each of the other places.

**Figure 36: ISJS, example of simple 2D map**

When travelling on the map, the times are fictitious. For example, it will take three game hours to return to the apartment from the Randstad Center, but in reality it will only take two clicks for the environment to appear.

The game date and time are continuously displayed on the screen. This information is important as it will enable assessment of the player’s timekeeping and organisational skills. For example, if an interview is scheduled for 11 am and it takes 2 hours to get there from the apartment, the player will need to leave on time!

In ISJS, days last 8 hours and nights are practically instantaneous. Companies and the Randstad Center close at the end of the day. If the player is not already in his or her apartment, he or she will be “teleported” there. Nights are therefore spent in the apartment where the player “wakes up” every morning.
7.4.8 Actions in the Player's Home

When the player has nothing specific to do, he or she can decide to speed up game time by sleeping in the apartment. The player will then “wake up” the next day...

The apartment environment is relatively simple and interactions are mainly performed here via a virtual PC, which can be used to:

- Consult a list of job offers
- Respond to postings by sending emails
- Receive job interview invitation emails
- Consult recruiting company profiles.
- Consult his/her online profile.

The virtual PC interface will be extremely simple, giving access to only two functions at first: email (to respond to job postings and receive job interview invitations) and web browsing (to consult job postings and company profiles).

The writing of emails is interactive. Like the cover letter, a drag&drop system with ready-made sentences will be used.

From time to time, a new job offer appears in the list. Then, if the player thinks that he has the corresponding profile; the player is able to apply to this job, via a simple email.

A job offer is composed of some criteria/keywords, similar of those used in the players’ profiles. If the profile matches with the offer, the player will receive a job interview invitation email. If there is no match, the player will receive a negative response.

7.4.9 A Job Interview

When the player arrives at a company, he or she meets a recruiter (female or male 3D virtual character) in his or her office (different types of 3D environment can be used to reflect the psychology of the recruiter and the company). The meeting begins with introductions.

Like the RPA, recruiters use the voice synthesis module to express themselves and the player responds in natural language via the microphone. The interview is based on a scenario scripted with the Communication Scenario Editor (RAGE Component 3.3F). But it can be influenced by:

- Emotion detection feedback (RAGE Components 2.3A and 2.3D);
- Keywords founds in the player’s answer (RAGE Components 3.3A and 3.3L);

In an interview, for each answer of a player, after transformation of the speech into a text (RAGE component 3.3H), the game will search in a database of keywords (skills, motivations, jobs, etc.) to identify the possible following answer. If there is no possible answer of this kind, the game will continue with the scripted scenario.

Job interviews are generally relatively short, with the recruiter asking about a dozen questions at most.

The recruiter is looking for an extremely specific profile, in terms of qualifications, as well as psychological profiles. In ISJS, the idea is to enable the player to improve and make progress during job interviews, and therefore to let him or her try several interviews of varying “difficulty”.
The first interview will be very short and the player profile will not be suitable. However, with each attempt made, the profile sought by the recruiter will be increasingly similar to that of the player. Interviews will, therefore, enter into increasing levels of detail. On average, we expect the player to attend 5 or 6 interviews before being virtually hired.

7.4.10 Interface

Player journeys between the various places are made by simply clicking on the destination with the mouse, both on the 2D map and in the 3D environments (seen from a first-person point of view).

In the 3D environments, no camera movements will be possible and the pre-calculated backgrounds will be fixed. 3D virtual characters (PNJ) move little and the player has a first-person view (no player avatar is represented in the game).

Dialogues with the RPA and recruiters’ interviews are entirely conducted in natural language, through the use of the various RAGE Components. So, the in-game interface will, therefore, be fairly basic:

- place, date and time are displayed in the top-left of the screen;
- a Menu button is displayed in the top-right.

This Menu button opens up a number of options:

- Save and load games in progress; quit the game.
- Settings panel (sound volumes, full-screen mode, etc.)
- Results panel.

The last item on the list will enable the player to access his or her profile, linked with virtual job search results (number of emails and interviews, results, etc.) and the Randstad Booklet, which records detailed advice received from Randstad (in the form of text or videos), at any time during the game.
7.5 Technical Specification

BiP media will use Unity 3D as a game engine for ISJS (C# language) for a PC version.

Used PCs have to be equipped with:

- a classic sound card
- a microphone,
- a good graphic card
- Optionally with a standard webcam (emotional recognition).

For a use on Randstad sites, all PC will also be equipped with a real-time arousal detection system using galvanic skin response (RAGE component T2.3D).

ISJS will need an Internet connection.

The final product will be able to be distributing online, by downloading an Install program (.EXE). Compatible OS are depending of the version of Unity 3D used for the final release of ISJS.
8 HATCH – GAMEWARE AND HULL COLLEGE

8.1 Introduction

The Creative Entrepreneur Game (HATCH) aims to evaluate and enhance Creative Industry-focused entrepreneurial skills across a range of students who are about to embark on creative careers following graduation.

The overall game is divided into components - a series of mini-games or ‘chapters’ within the larger game that operate within a live, causal feedback interface. The game will assist the user to identify key skills and knowledge areas required to develop a business plan appropriate to a chosen creative market sector model and/or start-up enterprise.

This business initiative might reflect that of a sole trader or a creative service provider, or a “bricks and mortar”, fixed site enterprise such as a creative product retail store or an Art gallery.

HATCH is intended to support longer professional practice modules such as those found on the various creative BA Hons degrees at the Hull School of Art & Design (and other creative courses across the wider Arts Faculty). The intention with a modular development approach is to facilitate further commercial development and inclusion of additional modules that may be required by other creative industries faculties, departments and institutions across the UK and Europe.

8.1.1 Key activities and outcomes

The general learning goal of this game is to provide training on entrepreneurial skills to final year students and recent graduates in creative subjects.

It is intended that playing HATCH will enable students to appreciate and gain a good understanding of the fundamental steps and knowledge skills required to set up a new business in the creative industries. These may include:

- Creating a Business Mission
- Know your Industry Networks
- Know your Competition
- What type of Company Are You?
- Pricing your Product
- Financing a Start-up
- Cashflow and Financial Planning
- Risk
- Basic Accounting
- An introduction to Tax
- Branding and Market Position
- Marketing Strategies
- Self-Presentation
- Audience identification - Who are your customers?
- Customer Service

In addition the game should enable users to:

- Explore and gain understanding of the entrepreneurial context
- Experiment with effective balance and prioritisation within a “safe” sandbox environment
- Reflect on findings in order to develop effective tool kit for taking products/services to market

Upon delivery, the game should serve as a stand-alone resource - able to be utilised by the teacher/lecturer in a variety of classes and study areas. The game will allow both student and
tutor to access “areas for development” flagged up by the game, this information could be available on a server, allowing for access outside of gameplay.

<table>
<thead>
<tr>
<th>Who</th>
<th>Main contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCG</td>
<td>Requirements and educational/user needs analysis, identification of learning objectives, competencies and observable behaviours, theoretical background and models, content development, asset evaluation, organization of the pilot and analysis, pedagogical and psychological input (theories, educational approaches and methods for teaching soft skills). HCG will provide early game design, flow charts and Game dev</td>
</tr>
<tr>
<td>Gameware</td>
<td>Design development and implementation, assets integration and testing.</td>
</tr>
</tbody>
</table>

Table 13: HATCH, partner work split

8.1.2 Pedagogy

As part of the main game, players will go through a series of game modules (mini games), which addresses on individual learning objectives. This enables an initial focus on the basic steps and skills of creating a business in isolation, before integrating them to a meaningful whole. This approach constitutes and adaptation of the so-called SOLO (Structure of the Observed Learning Outcomes) Taxonomy (Biggs & Collis, 1982). This taxonomy provides a systematic way of describing how a learner’s performance grows in complexity when mastering a range of tasks from unistructural to more relational level. It can be used to define objectives that describe performance goals or targets, as well as to evaluate the level of learning outcomes. The idea adopted in HATCH is to use mini-games to support learning at the unistructural level, by directing focus at one aspect at a time (cf. Abdul Jalil, Plimmer, Warren, Luxton-Reilly).

Kolb’s Experiential Learning Cycle (Kolb, 1984) is used to inspire the structure for the gameplay experience and how it is embedded in the learning context which includes:

- Playing or trying out multiple sessions in the Applied Game – Active Experimentation/Concrete Experience (albeit safe in the virtual in this instance)

- Examine own “scores” or progress by accessing the gameplay data – Reflective Observation (here the option to do this with a tutor to help interpret the data would blend this with the next phase)

- The development of an action plan based on the conclusions drawn from the data the game highlighted to the player. – Abstract Conceptualisation

- Finally going back to try again to apply the knowledge gained through the combination of successes/failures/ and subsequent reflection in the ongoing game (or perhaps the live project in the case of our longer module). – Active Experimentation

8.1.3 Game Run-Time

It should be possible for a novice player to complete one game session in two to three hours, however, it is unlikely that the first session will result in a particularly successful outcome – unless the player is already familiar with the core business concepts the game aims to test.

The game should maintain sufficient interest and course-work value to bear multiple plays. Users should feel encouraged to repeat the process and experience differing outcomes, for differing markets and sector specialities.

Lessons should be learned and further attempts to improve their final assessment will be encouraged and expected.
8.1.4 Platform
The simulation will run as a standard PC application on Windows OS 7 and above.
The simulation will be written in C# ensuring compatibility with any software tools and engines provided as outputs from WP2 and WP3 and to sit within the RAGE eco-system at the end of the project as specified by WP6.

8.1.5 Graphics, Viewpoint and Appearance
We envisage a look and feel based on an infographic style. This approach enables a range of interactive graphic design-led screens comprised of impactful and exciting visual designs that are recognizable, malleable, predictive and reactive to user input.

The intuitive controls and real-time interactivity of the graphic outcomes should readily illustrate causes and effects of user choices and their impact on the business they are seeking to set up.

UI components will consist of sliders, radio buttons or free placement of items representing the player's choices. Together these will allow the user to see the impact of their choices on outcomes and to reflect on these whilst, simultaneously, coming to appreciate the interrelations between key business issues, for example, how pricing is directly affected by supply and demand, or the spending power of the product's target audience.

So, presentation will be entirely 2D in appearance, but should feature plenty of graphically pleasing screen interactions and translations.

8.2 Game Flow and Game Modules
At the outset of the game, the user is presented with choices (or, possibly, a short set of questions) which will determine the type of business which they will be setting up and seeking funding for.

The game works as a succession of decisions, each one reflecting and representing a business planning decision that bears on the eventual outcome. Each decision deals with a specific core activity (e.g. distribution choices, pricing, marketing mix and so on) required in setting up the player's chosen business. Flow through the game is illustrated in Figure 39 below.
Figure 39: HATCH, game flow chart
8.2.1 Business Ventures

The types of business the player may be setting up may include:

- Artist producing products to sell in outlets or to commission
- Retail of cultural products
- Cultural service provider
- Arts company, events-based
- Arts development company – consultancy-based
- Online retailer of art or media-based products

The decisions that players make during the game will be moderated by the nature of the business they have chosen to set up. For example, choices involving marketing strategy or product distribution would require different approaches as between, say, an online retailer and a bricks and mortar art gallery. Indeed, some game activities may be limited to certain businesses and not appear at all in others.

The opening to the game should allow the player to explore the range of business types they are given. This exploration might include descriptions of each sector, and statistics related to that potential choice, such as audience share, financial sector share, average success of start-ups in that sector etc.

The business sector data required by the game model (or engine) will be provided during the development phase by HCUK, but there is no requirement that this is all available in advance since most of the technical development should only require the desired outcomes in one sector to enable work to progress one module at a time.

8.2.2 In-Game Currency and Values

It is unlikely that any given game model might lend itself to an accurate representation of the real-world monetary value of the transactions and/or strategic value of a player’s decisions. The same may apply to issues including market sizing and demographics; timelines for product development, manufacturing and distribution.

For this reason, we favour an approach which uses logical but in-game only metrics. Thus values should be relative to one another – sufficient to convey the financial impact (or otherwise) of decisions but not necessarily tied to figures one may find are truly associated with any given market sector or business venture. This kind of paradigm is commonplace in simulations and resource management games such as Sim-City (Simoleons), Farmville or ‘Tycoon’ games which tend to use dollars ($) but item values are disassociated from their real-world equivalents.

This solution also has the advantage that it is not territory or country-specific, but should work internationally even though actual market characteristics/metrics would vary widely between (say) the UK and much smaller (or larger) countries.
8.2.3 Game Modules

The game's modules (or activities) will each provide the players with a set of decisions and/or tests, presented in an attractive visual style, which seek to give them some exposure to the issues involved in:

1. Self-Assessment & Self-Development
2. Legal Issues in business
3. Financial Issues
4. Project Management
5. Manufacturing
6. Audience and Market
7. Marketing and Branding

8.2.4 Self Assessment and Self Development

Module Objectives:

*Presenting Myself* Presentation and business pitching skills. Confidence building, props, hints and tips. Able to cater pitch content to client (funder, supplier, retailer, corporate etc.)

*Developing Myself* Looking to the future of the business – skills, information, contacts, expertise you will need. Identifying sources/places/courses to assist personal development and what is needed to progress and re-energise your business.

*Know Your Industry* Research / collection of relevant networks and support organisations. Conferences, seminars, skills building available

*Identify Your Skills* SWOT analysis of personal business, entrepreneurial and 'making' skills through strategic questioning, leading onto areas where they may need assistance and suggesting collaborators

**Difficulty (1-5)** 1

Game Mechanics:

In this module the player is presented with a reasonably extensive list of words and phrases, which are each displayed in 'post-in' style labels. They must select and place up to 5 stickers on a 2x2 form or table, under the headings Strengths, Weaknesses, Opportunities and Threats.

Whilst ostensibly a simple task, their first job will be to determine which category (S,W,O or T) the labels belong to. Their second lies in prioritising which labels to choose and which to ‘discard’.

Some of these words relate to personal characteristics, some to aspirations, some to their desired outcomes for their business venture; whilst some may apply to both personal and business objectives (for example, “security is important” could apply to personal or business security).

Each phrase will be ascribed a weighting, in advance, which determines its importance in the player's (pre-selected) business venture. For example, if a player considers themselves highly social, this would not help them in running an online store, but it would be very valuable to a market trader.

The Figure below illustrates the SWOT decision-making process envisaged:
When the players are happy with their selections, they may be presented with a short analysis of their choices in the form of a software-generated assessment. For example (using the incomplete example above), the player has selected:

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Careful, Social, Creative</td>
<td>Price Sensitive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weaknesses</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeracy</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Worker, Prior Experience</td>
<td>Cheaper Product</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threats</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Volatile Market, Few Suppliers</td>
</tr>
</tbody>
</table>

“You describe yourself as careful, social and creative. These are valuable skills in your chosen <gallery owner> business. However, you have not identified any specific strength your gallery has over the competition.

A weakness in numeracy alongside a price sensitive market suggests you will need an experienced accountant or finance officer on board.

Prior experience and fast working present an opportunity to respond quickly to changing market needs, which, alongside your cheaper product offering, make a compelling business case.

You recognise you are in a volatile market with few suppliers available. Your attitude to personal security may result in much personal stress which could be mitigated with a strong team around you.”
Further information (parameters, weightings, sector-specific good and poor profiles) will be sought from HCUK for inclusion in this module.

### 8.2.5 Legal Issues

**Module Objectives:**

- **Know Your Rights**: Copyright, IP, patenting, trademarking
- **Tax Responsibilities**: Introduction to HMRC requirements, self-assessment / other business submission models, tax years, tax-deductibles
- **Am I Insured?**: Deciding which insurances are required for your business and where to go and investigate: public liability, product liability, building insurance, specialist companies/home insurances
- **Sign on the Line**: Understanding contracts – select from options: trading, service agreements, commissions, quotes, retailers, galleries etc. Signposts to examples and sources of information

**Difficulty (1-5)**

5

**Game Mechanics:**

Note: Company law and Intellectual Property (IP) issues vary widely from country to country, but for our present purposes, we shall focus here on a UK-oriented perspective.

This module focuses on determining what type of business vehicle (legal entity) the player might choose as most appropriate to their venture – and the legal ramifications of that choice.

**Phase 1**

- Essentially, the player must pick one from 3 legal entities which they believe best represent the scope of their business venture. For our purposes, these should comprise:
  - Sole Trader
  - Limited Company (Ltd.)
  - Partnership

**Phase 2**

Depending on which entity is chosen, the player will be presented with a ‘Word Search’ matrix. This contains an assortment of apparently random letters and numbers which directly relate to words or government forms (by code number), pertinent issues (e.g. employment law, accounting requirements); or other documentation which they may need to recognise as significant in their chosen business structure.

At this point, the player must correctly identify those words or ‘code’ strings that will apply to their business venture, themselves or their staff. Through successful (or partial) completion of the Word Search, players should have demonstrated that they have researched the terms and meanings associated with the identified word strings and have, therefore, acquired some knowledge as to their implications and any consequent business issues arising.

Key words and/or form references should be tailored in line with the (given) business venture the player plans to be engaged in. A fuller specification will be provided by HCUK but, for the purpose of illustration only, the table below provides some idea of how this game mechanic might be applied in practice:
1) Before (What the player sees)

<table>
<thead>
<tr>
<th>G</th>
<th>2</th>
<th>T</th>
<th>L</th>
<th>5</th>
<th>B</th>
<th>O</th>
<th>M</th>
<th>5</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>I</td>
<td>P</td>
<td>9</td>
<td>I</td>
<td>F</td>
<td>H</td>
<td>M</td>
<td>R</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>L</td>
<td>O</td>
<td>C</td>
<td>5</td>
<td>M</td>
<td>I</td>
<td>5</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>V</td>
<td>5</td>
<td>C</td>
<td>P</td>
<td>F</td>
<td>2</td>
<td>I</td>
<td>6</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
<td>E</td>
<td>A</td>
<td>4</td>
<td>3</td>
<td>X</td>
<td>T</td>
<td>9</td>
<td>I</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>V</td>
<td>R</td>
<td>X</td>
<td>5</td>
<td>2</td>
<td>T</td>
<td>E</td>
<td>N</td>
</tr>
<tr>
<td>T</td>
<td>E</td>
<td>I</td>
<td>T</td>
<td>1</td>
<td>N</td>
<td>I</td>
<td>9</td>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>X</td>
<td>P</td>
<td>N</td>
<td>0</td>
<td>4</td>
<td>A</td>
<td>V</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td>O</td>
<td>E</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>M</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>E</td>
<td>A</td>
<td>R</td>
<td>O</td>
<td>T</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

2) Solution

<table>
<thead>
<tr>
<th>G</th>
<th>2</th>
<th>T</th>
<th>L</th>
<th>5</th>
<th>B</th>
<th>O</th>
<th>M</th>
<th>5</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>I</td>
<td>P</td>
<td>9</td>
<td>I</td>
<td>F</td>
<td>H</td>
<td>M</td>
<td>R</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>L</td>
<td>O</td>
<td>C</td>
<td>5</td>
<td>M</td>
<td>I</td>
<td>5</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>V</td>
<td>5</td>
<td>C</td>
<td>P</td>
<td>F</td>
<td>2</td>
<td>I</td>
<td>6</td>
<td>N</td>
<td>S</td>
</tr>
<tr>
<td>P</td>
<td>I</td>
<td>E</td>
<td>A</td>
<td>4</td>
<td>3</td>
<td>X</td>
<td>T</td>
<td>9</td>
<td>I</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>V</td>
<td>A</td>
<td>X</td>
<td>5</td>
<td>2</td>
<td>T</td>
<td>E</td>
<td>N</td>
</tr>
<tr>
<td>T</td>
<td>E</td>
<td>I</td>
<td>R</td>
<td>1</td>
<td>N</td>
<td>I</td>
<td>9</td>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>X</td>
<td>P</td>
<td>T</td>
<td>0</td>
<td>4</td>
<td>A</td>
<td>V</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td>O</td>
<td>E</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>M</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>E</td>
<td>A</td>
<td>R</td>
<td>O</td>
<td>T</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

Relevant Words or Form References (shades of blue):

- **2T** (unknown placeholder)
- **IP** (as in Intellectual Property)
- **LIMITED** (as in Limited Company)
- **HMRC** (as in Her Majesty’s Revenue & Customs)
NI (as in National Insurance)
LOC (as in Letter of Credit)
PARTNER (as in Partnership, v Ltd company)
PATENT (as in Patent, Patented or Patent Protected)
1060 (as in UK basic rate tax code)
P45 (as in document provided to employees on departure/termination)
NI (as in National Insurance)
P60 (as in document statement of earnings and taxes paid at year end)

Red Herrings (shades of pink)
BOM (as in Bill of Materials – useful acronym to know but not relevant here)
(M)I5 (as in the UK-centric secret service)
V5C (as in Vehicle Tax renewal form)
A1 (as in ‘good condition’ or a major north to south trunk road)
POA (as in Price on Application, irrelevant here)
VM1 (as in Virtual Machine 1 – placeholder only)
M60 (a motorway up North)

Extraneous letters or numbers are displayed above in brown (brown).

Phase 3) When the player is satisfied that they have identified all the words contained in their Word Search grid, they will press a ‘Submit’ button in order to receive their ‘score’ or rating, which identifies their correct answers and indicates <how many correct terms> have been found out of <all valid terms>.

The player may be advised to continue with the exercise and/or to consult selected literature/web-sites to assist them. Alternatively, they may prefer to stick with their results and move on to another module. Scoring in this module will influence inputs and/or outputs in others. For example, a lower score here could negatively affect investor confidence at the final stage of the game – the ‘Dragons Den’ (see 8.2.6 below) only mitigated by very positive outcomes elsewhere. On the other hand, a clear understanding of the technical terms that have, clearly, been researched in this module might encourage a more sympathetic response at the end-game stage.

8.2.6 Financial Issues

Module Objectives:

Cashflow & Planning Developing a 3 year cash flow model using accounting software – step by step, with pointers for finding/calculating figures
Accounting Invoices, receipts, basic spreadsheet bookkeeping to ready accounts to present to the accountant or prepare for self-assessment. Net/gross profit and loss
Start-Up Financing income Calculating how much you need to start up and identifying sources of income
Find Those Funds! Alternative funding sources for a range of business developments

Difficulty (1-5) 4

Game Mechanics:
A clear understanding of financial issues is essential to any business leader. This module should provide a simple template that the user can play with in order to model their venture’s finances and, in particular, gain an understanding that available cash and cash flow is central to operating a sustainable business.

Note: Because all game data that can be represented quantitatively is stored in the main game engine (alongside certain qualitative scores), the financial outcomes modelled here will be affected by decisions made in other modules, which the player may or may not have already experimented with. Likewise and vice-versa, values entered here will affect outcomes in other modules. For example, if ‘Cost of Goods/Trade Price’ is set low here, this would impact on the position of the player’s ‘Pin’ on their TVQ triangle (see 8.2.8 Manufacturing, below). Similarly, Marketing spend indicated here will impact on the player’s available marketing budget (see 8.2.10 Brand and Marketing, below).

For our purposes here, we shall assume that:

- Expenditure on any given stock item or similar (e.g. marketing) will fall due the month after delivery (i.e. standard 30 day terms of payment)
- Overhead costs (rent, salaries, etc.) must be paid in the same month cost is incurred
- Any taxes (i.e. PAYE, VAT) must be paid on time during the month following the quarter in which they were calculated.

Whilst precise details of the preferred line items to be included in the Cash Flow model and associated formulae will be confirmed by HCUK during development, a simple structure is illustrated on the following page.

Two sets of data are presented here. The topmost model (Example 1) demonstrates breakeven in year one, on the back of a small bank loan. Marketing budgets are small and staffing costs kept low (just the one employee to start with - the business owner). Growth in years 2 and 3 are forecast at 10% and 15% respectively. Sales targets are very modest and markup on goods traded is 150% - fairly typical in retail – such that cost of goods is £20, sales price is £50. As seen, after trading one year the business falls quickly into the red. This suggests higher sales will be required; there are few significant cost reduction options available other than, perhaps, cost of goods bought or made for resale. Higher spend on marketing would certainly drive sales but will require either a larger bank loan or investment capital to kick-start the business.

In Example 2, the player has secured a £25,000 investment and a bank loan of £5,000. Accordingly, they are able to spend four times as much on marketing from the outset, and more than double this cost over the first year. Sales grow rapidly, serviced by two employees to begin with and a third joining after 6 months. As a function of their marketing and promotion spend, the player is able to set a higher sales price, increasing their margin to 200%, as well as aspiring to a much higher growth target (25%) by Year 3 – word of mouth and reputation will likely allow this without a further hike over Year 2’s marketing spend.

Note that the player cannot change certain parameters (e.g. PAYE, VAT liability and Loan Interest) figures since these are determined, formulaically from parameters that they cannot modify. Ideally, there would be no need for the player to enter detailed information as they would with a spreadsheet. A template will be provided which is more visually appealing and easier to operate – with most inputs and their consequent outputs being semi-automatic. Ideally, to set up a model, as shown below, simply by entering a few parameters such as Sales Price, Yearly Marketing Budget, Target Sales Margin, Staff Employed (and when), Investment sought etc. Other required parameters would be serviced by the game engine and/or set elsewhere (during other module activities).

8.2.7 Project Management

Module Objectives: Manage the Project - The Planning Stage Project management skills – timescales, practical planning, resources, people, negotiating, budgeting etc. A practical example project relevant to the business idea to run through with scenarios/issues/problems arising to solve. Manage the Project - Delivery & Beyond Building on previous modules -
practical delivery considerations, contingency planning, audience/visitor management, documenting and evaluating

**Difficulty (1-5)** 3 to 4

**Game Mechanics:** Typically, project planning requires the identification of the key tasks required to execute delivery; estimation of the time each task will take; arranging tasks in order of completion and noting any dependencies between them (this task must complete before another can commence); and assigning resources (personnel) against the tasks.

In this game we shall assume a small number of (pre-identified) tasks will generally apply to any business venture, but we may add one or two sector-specific tasks if this is appropriate.

The player must:

1. Choose the order in which each task must be carried out and assign any dependencies
2. Assign resources to a task from a list of up to 4 available 'staff'. Resources will be randomly assigned a characteristic on two simple ranges: Slow to Fast and Low-Cost to Expensive. Thus, say, Andy might be slow and cheap, Sarah fast and expensive, whilst Bruce is average on both counts.
3. Once the player has selected from the available options, they can run the game model to determine if the plan runs on time and whether the costs are appropriate. When the player is happy with the outcomes (they can re-run the model as often as they choose), they press Confirm and the resultant outputs are stored in the main game engine.

A good example of the gameplay proposed here is available at: [http://thatpmgame.com/](http://thatpmgame.com/)

Additional features we propose are the tasks are customised to the business cases being explored in Hatch; the ability to choose ordering and linkage (hence simple critical path analysis); and any cost or timeline implications of the player’s choices having a bearing on the main game outcomes.

---

**8.2.8 Manufacturing**

Module Objectives:

*Investigating my Product* Perceptions of quality, industry standards, fit for purpose, USPs, H&S requirements (if relevant).
Making & Delivering My Product

What skills do you have/need to create your product or deliver your service? What equipment do you need to buy or access? Which suppliers will you need to source? What space do you need? Logistics/distribution issues. A practical module equipping students to deliver.

Price your Product

Introduction to resources, overheads, time and budgeting

Difficulty (1-5) 3

Game Mechanics:

This module makes use of the classic Time v Cost v Quality Triangle to illustrate the relationships between these key factors and the necessary trade-offs between them which will determine the shape of a business offering (product or service) and impact on consumer attitudes (also explored at 3.2.6, Audience & Market).

In this context, the time constraint refers to the amount of time available to complete a project. The cost constraint refers to the budgeted amount available for the project. The quality constraint (sometimes referred to as the ‘scope’ constraint) refers to what must be done to produce the project's end result.

These three constraints are often competing constraints, such that increased quality typically means increased time and increased cost, a tight time constraint could mean increased costs and reduced quality, whereas a tight budget could mean increased time to build and a reduction in quality.

The player is presented with a screen displaying a time, cost, quality pyramid alongside which 3 fields (and/or sliders) are shown, these represent the three parameters and their associated value metrics (cost, time, quality).

In much the same way as a colour-picker works in art packages, the player can move a pin (their pointer) around the triangle and in so doing read the changing outcomes in the value fields. Alternatively, they can enter values directly in those fields – causing the pin to move on the triangle.

The closer the player’s pin to a given vertex point (symbolising Time, Cost or Quality) the higher priority is attributed to the corresponding parameter. Thus, for example, maximum proximity to Time implies a very short production or development period making such very low cost and quality, whilst proximity to Quality (e.g. luxury goods) will result in longer time to market as well as a higher cost (or price).

Since the TQC data is connected to the main game engine, any changes the player makes here are reflected there. This means their actions may lead to their budgets being blown (if previously set), their expected time to market altered (impacting their project plan) or the target audience’s perception of their offering adversely affected (and potential impact on marketing spend requirements).

As with other modules, it is not until the player ‘Confirms’ their selection that the system is fully updated. The player will also have the opportunity to review and change their TQC decisions prior to entering the game’s final assessment phase.

Example outcomes are illustrated below (all data and parameter descriptions are subject to the final model).
8.2.9 Audience & Market

Module Objectives:

Who Are My Customers? Identifying your market, customer demographic and profiling; getting to know their needs. Local, regional, global issues. Test customer sets out with matching your product.

Love My Customers Basics of customer service and experience. Core values, what you need to set in place as a framework. How are you special? Can I go the extra mile? How do I want to be remembered?

Marketing – To Whom? Identifying markets, appropriating your brand and marketing content to the target.

Selling – Me or Thee? Modes of selling - personal, online, through a third party, using an agent etc. Identifying the best way to reach your market and allow you time to make work.

Marketing – Effective Avenues Choosing which method of marketing will suit which customer demographic. Innovative marketing strategies.

Difficulty (1-5) 2

Game Mechanics:

Note: This module’s inputs and outputs are closely connected to those of the Branding and Marketing module (see 8.2.10 below).

A clear understanding of the potential market need for a business’s products or services is essential to a successful venture. Is the product an essential or a luxury? Does it appeal more to one gender than another? Is it more likely to appeal within a specific age-range or not? These (and other) audience characteristics effectively determine the demographic profile of the business’s customers or consumers. For this reason, this module is centred on the choices a player must make in determining their market.

This may be achieved by way of selections the player makes using a graphical interface as shown on the following page (content parameters to be confirmed by HCUK).

<table>
<thead>
<tr>
<th>Social Grade</th>
<th>Social Status</th>
<th>Occupation</th>
<th>Target Market by Socio-Economic Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Upper Middle Class</td>
<td>Higher managerial, administrative or professional</td>
<td>20%</td>
</tr>
<tr>
<td>B</td>
<td>Middle Class</td>
<td>Intermediate managerial, administrative or professional</td>
<td>55%</td>
</tr>
<tr>
<td>C1</td>
<td>Lower Middle Class</td>
<td>Supervisory or clerical, junior managerial, administrative or professional</td>
<td>10%</td>
</tr>
<tr>
<td>C2</td>
<td>Skilled Working Class</td>
<td>Skilled manual workers</td>
<td>10%</td>
</tr>
<tr>
<td>D</td>
<td>Working Class</td>
<td>Semi and Unskilled workers</td>
<td>5%</td>
</tr>
<tr>
<td>E</td>
<td>Subsistence Level</td>
<td>Pensioners or widows, casual workers</td>
<td>0%</td>
</tr>
</tbody>
</table>

In the example above, the player is targeting males (a 64:35 ratio); the vast majority being in their 30’s and belonging to Socio-Economic Group B.
This might imply a reasonably switched-on, tech-savvy market with little time for physical shopping so, perhaps, this profile suggests an online operation for a media-oriented product (music, video or similar). This market is highly likely to be relatively affluent, so higher-priced items should not prove a barrier to purchase, however good quality, good service and customer care will be essential to success with this audience.

8.2.10 Branding and Marketing

Module Objectives:

What's Your Brand?  
Product description, product market placement, logo development

Marketing – My Website  
Using your brand, mission, values and customer knowledge to build an appropriate online presence.

Marketing – Using Social Media  
Introduction to and actual set up of social media marketing tools (appropriate to their business). Learn how to design and launch marketing campaigns.

Marketing – The Good Old Way  
Print media, radio, TV communications, press releases and interviews. Flyering, posters and other formats that is appropriate to the target customers. Distribution networks. Marketing assistance.

Difficulty (1-5)  
3

Game Mechanics:

The object of this exercise is to illustrate how money spent on marketing may be channelled into a variety of different media (e.g. press advertising, website development or broadcast), with differing outcomes in terms of total spend relative to audience reach in a given area of interest (potential customers of the player's business venture).

At the outset, the player chooses a monetary value representing total marketing spend in a year (their budget). Note that their marketing budget may have already been selected as part of another module (e.g. see 8.2.6 Financial above).

Using slider or arrow buttons, the player then attributes how what proportion of the budget they deem appropriate to allocate to their available media channels. As they do so, a pie chart showing the relative percentages spent in each channel is displayed (see below).

Certain channels (Print, Website and Social Media) will require further level of detailed data to be decided upon and to be entered in a similar way. For example, Social Media may itself include specific references to FaceBook, Twitter, Google Ads and similar. Again, sliders or buttons are employed to select what percentage of the available budget ascribed to Social Media will be spent on each activity.

A similar mechanism could be used to allocate budget to various forms of print media (e.g. ads in local or national papers; flyers or fly-posters; billboards or bus-stops)

Whilst it would be desirable for all the required interactions to take place on one screen, this may not be possible without cramming too much into the limited real estate. So it is more likely we will employ ‘tabs’ or similar in order to move from one set of choices to another.

In particular, there needs to be further discussion regarding how the player may come to understand the typical requirements of managing a successful web site: have they included contact details, covered legal requirements, have the appropriate content and so forth.
When the player is happy with the marketing mix they end the module via a Confirm button and the resulting data is stored in the main game data engine.

8.2.11 End Game Scenario

The end goal of the game is to demonstrate a good understanding of the issues explored in each module. Together, these should indicate that the player has rounded entrepreneurial skill set – sufficient to pass muster during this final phase.

The player’s choices and corresponding outcomes will be ‘assessed’ by a (virtual) business angel or investor group (i.e. a ‘Dragon’s Den). The quantitative or qualitative data stored in the game model will determine their views that, in effect, would be provided by way of a risk assessment of the shape of business venture the player has envisioned.

Key risks would be specified by HCUK as assessed against individual, combinations or all of the game’s modules.

For example, what is the level of investment required to finance the business relative to the return on that investment after 3 years? If returns are low (below a prescribed percentage) then

**Figure 42: HATCH, example layout**

**BRANDING AND MARKETING EXAMPLE LAYOUT – CHOOSING MARKETING MIX, INCLUDING SOCIAL MEDIA BREAKDOWN**

The resulting data is stored in the main game data engine.
the Dragon’s response would be that the player should re-assess their model with particular attention paid to <whichever module is damaging the ROI most significantly>.

If returns are satisfactory or even very high, the player may still fail to secure investment if, for example, they have not demonstrated sufficient understanding of issues covered in the Legal or Personal Development modules.

A marginal decision by the investor might be influenced by quite subtle factors. For example, if the player is expecting to take out a bank loan (Financial module), they would take the view that their funds are likely to be more secure since, to some extent, they are underwritten by the bank. On the other hand if all the numbers appear to ‘add up’ but the target consumer profile is not a good match in a given creative sector, the investor will choose not to invest and give this as their reason and advise tuning inputs in both of the Audience and Marketing modules.

If the investors decide to offer the desired start-up funds, the player may have to make one final decision. How much equity in their venture are they prepared to offer in return for the investment? Pushing back too hard against the investor demands may yet scupper the deal…

8.2.12 The Game Model (Hatch Engine)

The game model, referred to throughout the preceding module descriptions, will tie together the input(s) and output(s) – the I/O - of each module into one, binding set of results. Many of these module I/Os will be mutually dependent, such that a pricing decision made in one module (e.g. Finance), will determine available budget in another (e.g. Marketing). However, as far as possible, it is the intention that the modules can be played in any order, which means that no decisions are absolute until the player has committed (confirmed) their decisions in all 7 modules – at which point their business venture can be finally assessed in the End-Game (see 8.2.11 above).

Subject to final development limitations and further design decisions throughout this iterative process, the high-level model is shown in the schematic below:

- Game Modules (as described in the preceding section) are shown as pre-defined processes.
- Data Type Linkages between modules are labelled in Yellow. These are high level only, further detail parameters will be added during development under HCUK guidance. Note that outputs from the Finance module are not shown, other than those which feed the Assessment stage, however we envisage that the Finance module’s parameters will, in fact, flow bi-directionally.
- Final Assessment is derived as a function of three qualitative value ratings (Legal Competence, SWOT assessment and Appropriate Market Mix); as well as two quantitative values (Capital Requirement and projected ROI), shown in Pink. These ratings will be subject to weighting values during development and test to arrive at a fair scoring mechanism.

8.2.13 Additional Support Materials

Additional resources should be made available to players, inside the game and outside the game (tests, advice, external rewarding mechanisms, access to detailed course documents and texts).

Beyond the basic gameplay, the player can "step out" of the game to analyse the data collected from game-play at any time. Learning from the data and feedback autonomously or with the tutor in a tutorial situation, formatively or as a summative assessment exercise, this can then feedback into the players continuing gameplay' or more practically, into the player/student’s module portfolio, and of course their actual behaviour in any given real world creative business enterprise.
9 INTERVIEW SKILLS FOR POLICE OFFICERS – GAMEWARE AND MINISTÉRIO DA JUSTIÇA

9.1 Introduction

“Information gathering constitutes the main point of the investigating officer’s (IO) work. The IO’s ability to gather useful and precise information from witnesses and victims is essential for the justice system’s efficiency. However, the recovering of precise and complete information is something difficult to obtain.” - James Stewart (1985) Director, US National Institute of Justice

The primary goal of this simulation is to train police officers in communication competencies related to the interview of victims of violent sexual crime and the interrogation of violent offenders (primarily sex-crime offenders) through the use of a series of simulated scenarios.

The focus of the simulation will be the communication processes involved during police interviews, especially bearing in mind the required verbal and non-verbal communication competencies of the IO in gathering information from witnesses, victims, suspects and offenders.

The simulation will provide a number of scenarios that will progress from notionally easy and gradually increase in difficulty. The training of communication competencies will be set within three different categories (witnesses, victims and offenders). In this document, we present the core structure of the simulation which will be updated during the ongoing development process with Policia Judiciaria.

9.1.1 Learning Objectives

The overarching learning goal is to provide training on interview skills to police officers. In summary, ISPO is a simulation of police interview procedures intended to provide a training aid for police officers who need to upskill their interviewing techniques and competency relating to:

a) Victims of a serious sexual crime including rape and violent assault.

b) Offenders or suspects who may have committed such a crime.

c) Potential Witnesses who may have information relevant to a criminal investigation.

Additionally, the simulation should include an element of matching descriptions of events and other such evidence in order to arrive at a conclusion. For example, is the suspect the actual offender or is there a possibility of mistaken identity?

Also, in the real world, there is the possibility that no crime has been committed and that allegations are therefore false. The simulation will also be capable of running such scenarios.

Scoring or rating of the user’s performance in using the simulation is intended to provide guidance as to where an officer’s interrogation techniques are inadequate or can be improved as well as to establish their level of understanding of the core competencies which they are required to demonstrate.

9.1.2 Simulation Objectives

Simulation objectives are achieved when:

a) The interviewer collects the required amount of necessary information about the crime, using appropriate interview communication techniques

b) The information can be obtained using difficulty levels defined by the simulation. Difficulty levels are related to the use of more complex communication strategies. Instead of ‘difficulty’ setting per se it would be desirable to offer a set of interviews each ranging from straightforward to much more complex cases.
c) The simulation scenario ends if the interviewer uses inadequate communication skills and the interview cannot proceed, i.e. that the situation is irrecoverable.

Policia Judicaria (PJ) will provide details covering the communication techniques which should be measured and scored. These characteristics will include some or all of the following:

- Emotional Control
- Resilience and resistance to stress
- Communication skills
- Empathy
- Conflict management capacity
- Ability to analyse verbal and non-verbal communication

### 9.1.3 Pedagogy

The ISPO game builds upon the pedagogical foundation of experiential learning. An experiential learning environment is implemented through a training simulation that conveys police interview scenarios that are relevant to the real-world practice of police officers. Players are immersed in the simulated situations by actively participating and directing activities that resemble their actual job practice (e.g. Rooney, 2012).

The interview scenarios create an authentic practical, social context for training; interview skills can be acquired, practiced, and mastered in a situated virtual context. The game thus serves as a safe environment to learn the targeted skills, where it is possible to experiment with behaviors, to make and learn from mistakes without having to embrace severe or negative outcomes, and to receive feedback (Beard and Wilson, 2002; Lewis & Williams, 1994). In this way, knowledge transfer and future retrieval of skills in real-world settings shall be supported (Thalheimer, 2009).

### 9.1.4 Simulation Run-Time

It is intended that each simulated Interview should take place in real-time, i.e. over a 30 to 45 minute period, similar to a real interview.

### 9.1.5 Technical Background

The simulation will run as a standard PC application on Windows OS 7 and above.

The simulation will be written in C# ensuring compatibility with any software tools and engines provided as outputs from WP2 and WP3 and to sit within the RAGE eco-system at the end of the project as specified by WP6.

### 9.2 Basic Principles

A police officer will almost always have to succeed in guiding the interviewee (regardless of their being witness, victim or suspect) from one undesirable state (for example, un-cooperative, upset or agitated) during which time accurate information will be difficult to isolate, to another state such that answers are clearer, fuller or more accurate.

Good interview practice involves a number of key stages and employing relevant skills alongside these. See Figure 43 below:
So, in order to progress during each phase of the interview the officer will have to employ, with increasing effect, a range of skills including Active Listening, Empathy, Rapport and Influence in order to trigger the desired Behavioural Change in the Interviewee.

The officer’s main tool is the choice, the phrasing and the timing of the questions that they pose or their choice of statements (perhaps clarifying something that has already been stated). In ISPO they must choose these questions from a carefully prepared and weighted set of options (multiple choice) and the answers that they receive will be affected by those questions AND the interviewee’s emotional state at the time of asking. Emotional state changes may be evidenced non-verbally as well as verbally. So the IO should also use visual references in deciding how to proceed. Depending on the answer to a question, the next set of question choices is presented to the IO and this process is repeated through to the end of the scenario.

A useful way to look at the phases of an interview is shown below:
9.2.1 Behaviour States and Attitude Change

In any given training scenario, the interviewee's behavioural state will be described, by the simulation model, as a function of two or more (numerical) values. These reflect the key characteristics of that particular interviewee. The chosen parameters may vary widely, but by way of example, we shall consider the case of a suspect with psychopathic tendencies.

Amongst other characteristics, psychopaths are manipulative and aggressive; the conceptual opposites of which are compliancy and calm. We can visualise these characteristics on a 2-axis graph as shown in Figure 48.

Any progress from the lower-left area to the upper-right area would represent the desired changes in the psychopathic interviewee's behavioural state see Figure 46:

Each question or statement, which the officer chooses, will have an effect on the psychopathic interviewee's state. A poor choice may diminish compliance or calm, a good one may positively affect both, and another may improve one state and diminish the other. Weightings ascribed to questions (or absolute values) will determine the extent of change and to which parameter(s) it will apply.

It should be clear that ‘good’ choices will have a summative, positive effect and poor ones result in negative effects. As the interview procedure approaches the final phase it becomes...
increasingly important that the officer maintains a positive trajectory – or they run the risk of a major setback and a difficult path in regaining the interviewee’s confidence.

It is very important to note that an apparently ‘good’ question – one that results in a positive or very positive effect on the interviewee’s behaviour – could nevertheless result in a negative assessment of the officer’s results. In the case of the psychopath, for example, if the officer has achieved satisfactory Calm and Compliance states, but he/she has done so by appearing to agree that the suspect’s actions were, in some way, acceptable. This would contravene acceptable practice and may result in immediate termination of the simulation for that reason.

9.2.2 Graphics, Viewpoint and Appearance

With the exception of any start-up or menu options, the primary viewpoint will comprise a simple depiction of the interviewee seated across a table from the user who is therefore facing the interviewee from the front.

The background scene will be plain walls, possibly with some windows or other simple graphic to add some life to the view (e.g. a wall clock), but the main focus will be on the avatar of the interviewee with particular attention paid to the head, torso and arms.

Facial expressions, movement of the head, eyes and arms of the avatar will provide a range of non-verbal clues to assist the interviewer in assessing the interviewee’s emotional state.

The overall graphical look should therefore be as photorealistic as possible, with the head and torso of the interviewee and perhaps one or more interviewers in a plain environment such as an interview room with cream or white walls. To this end, we intend to use video footage of a wide variety of poses and moods, interpolated (one scene to another) in a seamless manner resulting in a film-like experience.

Synchronised voice and facial expressions should be as close to reality as possible.

9.2.3 Head Up Display (HUD)

The HUD should be available at all times, regardless of viewing mode and include a list of items outlined below. Also see Figure 47 and Figure 48

- Visual Time Reference (i.e. a Clock)
- Pop-up text panels with item selection on a mouse-click to choose from a list of questions (or could use keyboard to type item number selected)
- Pop-up text reply panel (to supplement any speech which is an optional feature)
- Pause button, to support pausing and reflecting.
- Help button (to provide guidance on using the application or ask for advice from a ‘virtual colleague’)
- A Review mode will enable playback of the interview and the questions and their answers as recorded over the course of the interview.
9.2.4 Virtual Colleague

It is usual for an interview to be conducted by two officers.

This might imply simultaneous play by two users who could assist each other in making decisions about what questions to ask should run the simulation. However, as an alternative, the second officer could also be a virtual character who provides advice either spontaneously (if the situation is becoming critical, for example) or at the user’s request. If this solution is confirmed the second officer could appear in a ‘pop-up’ window on the HUD.
9.3 Scenario Descriptions and Difficulty Levels

We envisage three types of scenario with each one representing a base level of difficulty

Level 1) Witnesses

Witness interviews will generally be the easiest to conduct.

An eyewitness is a person who witnessed a crime and that by the request of the police will describe what he or she saw. In this context, the work of the police officer must be based within a learned set of communication skills in order to capture the accurate description that the witness should provide.

These communication skills are: compassion, empathy, engagement.

Level 2) Victims

Victims of a violent crime such as rape and violent assault.

Level 3) Suspects

Offenders or suspects who may have committed a violent crime.

At Level 2 and 3 the officer will be expected to employ more complex communication skills and strategies related to the techniques used for interviewing victims and interrogating offenders.

At each Level, a variety of scenarios will be scripted which provide increasingly complex interactions such as un-cooperative witnesses (who may be confused or fearful); victims who are unclear in their testimony (e.g. traumatised, angry or ashamed); or offenders who may be aggressive, psychopathic or cunning and evasive in their responses.

A simulation scenario will end if the interviewer’s communication skills are inadequate in gaining the desired responses after a period of time, in which case the user will need to repeat the scenario until they have demonstrated the required communication skills.

9.4 Example Scenarios

The following example scenarios are under discussion between GWE and PJ. They are based on real-world cases and are subject to confidentiality.

- SUSPECT: Interview with a female suspected of performing sex acts with under-age boys. She is the victim of mental disorders that make it difficult for her to understand the unacceptability and moral implications of her behaviours. A patient and non-judgemental approach is required in order to encourage a full account of when and who was involved in such events.

- SUSPECT: Interview with an antagonistic male suspected of grievous harm and sexual assault against his wife. He has psychopathic tendencies, is misogynistic and uncooperative. The interviewer will need to treat him in such a way that he effectively admits to what happened without becoming angry and obstructive. To achieve compliance, the interviewer may (almost) have to agree with him that his attitude to sex within marriage is justifiable.

- VICTIM: Interview with a male victim of gang rape carried out because he was unable to pay debts to a loan shark. He is mortified and traumatised by the experience and must be gently encouraged to recount the details of what took place and who was involved.
• VICTIM. Interview with a female streetwalker who, after consensual sex, was the victim of physical and aggravated sexual violence at gunpoint. The incident happened some months previously and she has not previously come forward for fear that she might not be believed. The interviewer must provide reassurance as to her safety and that her evidence will be taken seriously by the authorities.

• WITNESS: After closing time, a female cleaner at a bank is uncertain that a sex act between two strangers she witnessed in a private lobby was consensual or not. She needs to be put at her ease in the police station and encouraged to give a full account of everything she saw and heard.

9.5 The Interview Engine
The Interview Engine forms the most part of how the simulation works.

As described, the user is presented with the task of interviewing a witness, suspect or victim of a sexual crime and has to conduct the interview from start to finish. They will be assessed according to how well they succeed in achieving their objectives in terms of information gathered, time taken, errors made, interventions required and, hence, overall performance.

The interview process consists of a series of multiple choice questions, beginning with the basic information required to establish the victim’s name, their comfort level, to win confidence and to gain the key information required within the given scenario.

The selection of questions available to the user will take the form of a series of ‘scripts’ which will be provided by the team at PJ. Some of these questions will be crucial to an investigation, some of them will not. Some may be completely inappropriate in the context presented, whilst others will convey subtle variances in approach which may encourage the interviewee to become more open, less upset, and more accurate in the way that they respond.

In the background, the interviewee (the victim’s or the suspect’s avatar), will be ‘emotionally’ affected by the questions that they are asked or even by the order in which they are asked (see Section 9.4, above).

The current emotional state of the avatar will determine what answer to any given question the subject will present to the interviewer. In turn, depending on the answer given, another set of questions will be chosen by the engine, from a range of possible questions in the scenario script.

As described, we envisage that asking the most appropriate questions is likely to result in answers that would be more likely to be true, accurate or clear. Equally, poor questioning will result in obtuse or unreliable answers or even no answer at all. The next, subsequent, question choices will be directly affected such that the interview could end prematurely (be terminated) if the user has arrived at an impossible position and would, in real life, have to be replaced by another officer to take the process over.

The means by which the application will assess questions and answers will be reliant on the concept of the Metadata associated with these.

9.5.1 Metadata
Each question (and choice of question) will have metadata associated with it. Likewise, any answer should carry relevant metadata associated with that answer. Each metadata type will have more than one associated parameter, and a range of values.

The project will deliver a simulation framework and an editing tool which PJ can use to populate a series of questions and responses, along with fields and number ranges which signify the type of metadata (emotional, physical, relevance, etc.) and the range or probability of significance associated with a given metadata type. It is important that we use ranges of values rather than
fixed ones, to provide as real a representation as possible – meaning that the simulation can provide slightly different results with each use.

These may be very varied in nature. For example, a question such as "what is your name" could carry associated metadata such as "time" which is a number between 0-1 and a likelihood or several likelihoods. This would indicate that the user chooses to ask and use the victim's name more often than not.

For each interview scenario, PJ will specify questions, metadata and answers. Answers will be given a metadata value or range of values so we can deduce the outcome by weighting all relevant metadata and criteria.

Another example of metadata is a series of emotions connected to a question, again with likelihoods according to the timeline of the interview and likelihoods when the interviewee is in a certain behavioural state or condition.

So a question such as "did he really do it" with a metadata of "truthfulness" with a low value if the interview has a high scared state might score lower value than had the same question been asked later on in the process.

The deduction algorithm will find the nearest suitable output by cross analysing metadata, current parameters values and then choosing an Answer (and associated Question choices) from the relevant outputs. The more metadata provided or, indeed number of parameters (we can deal with an n-dimensional behaviour profile), the more realistic the outcome will be. The tool to enter inputs should be designed to assist entry as much as possible such that many metadata types can be entered quickly for each question and cross referenced.

The system can be tested with any type of question and answer. Metadata types will be fixed. Each metadata type needs programming into the system, so it can weight that metadata at each stage and calculate its current relevant value.

**Some Metadata examples**

- Time position during interview 0-100%
- Openness 0-100% how <compliant> is the interviewee?
- Anger to Resigned 0-100%
- Reticence to Verbose 0-100%
- Obtuse to Truthful 0-100%
- Repetitions: max repetitions to be used by interviewer
- Similarities: what questions is it similar to?
- Invalid combinations: don't ask if questions in this list have already been asked
- Calmness required: only ask if interviewee is within a required/acceptable calmness range
- Priority: how important is it to ask this?
9.6 Simulation Flow and Asset Integration

At a high level, progression through the game will take a very simple path as shown in Figure 49 below: After the user has logged in and selected their scenario, questions and answers determine win or fail states which are then assessed using a series of RAGE assets, leading to the conclusions and evaluations being presented to the player.

![Figure 49: ISPO, simulation flow and use of assets chart](image-url)
10 WATER COOLER GAME – NUROGAMES AND HULL COLLEGE

10.1 Introduction

The “Water Cooler Game” that will be developed by Nurogames as part of the RAGE project together with Hull College and several asset developers.

Overview Game Outline

In the “Water Cooler Game” the player is hired by a small game studio to develop the team working of the studio through improving interpersonal relationships. As an office assistant the player’s goal is to increase the success of the company primarily by improving the social skills of the virtual team. Therefore the player can visit the water cooler to start conversations that prompt group working/conflict management behaviour/value based questions and multiple-choice answers for the player to choose from. Besides that the player has to handle the workflow by interacting with the staff members and organizing the work package as best as possible to achieve good ratings and awards for the shipped games.

The game is designed primarily for students engaged in subjects to which there is a digital skills bias, in which “soft skills” are often seen as not as important by the student. Therefore a gameplay based feedback will be implemented for the players and the tutor. In this report the teamwork skills are analyzed.

The primary target audience are diverse gender, mixed nationality students, studying in English in England, of mixed age and related ability, from Level 3 (16/17yrs +) through to Degree and MA Level students 18+ with no ceiling on the upper age limit. The pilot will involve around 300 participants across two sets of evaluation.

The basic structure of the game should attempt to allow for a certain amount of potential “re-skinning” at the developers end in order that it might be more readily repackaged to work within a range of work-place scenarios (workshop, post office, bank, police station etc.).

10.1.1 Learning outcomes and learning context outline

The specific skills the game teaches and provides practice for are soft skills, i.e. conflict management skills, attitudes towards equality and diversity in the workplace and team working skills. The gameplay based feedback report can be complemented by a tutor feedback and be used to form the basis of an actual “live” group working scenarios.

10.1.2 Asset usage outline

First of all one of the used assets will handle the user account creation and authentication to manage the access to the different functionalities of the project (the dialog editor, the game and the dashboard for analytics and reports using Server-side Authentication and Authorization (T2.4A). Here also the different user roles will be respected.

To optimise the game design balancing, to analyse the behaviour of the user and to reflect the learning progress a browser dashboard asset for analytics and reports (T2.4B Server-side Dashboard and Analysis) and a tracking tool (Client-side interaction tracking (T2.A)) will be used. This will also require a server side interaction tracking asset (Server-side interaction storage and analytics asset (T2.1B)) and interaction storage asset (Player Model (T2.4C)) that is linked to the account and authentication asset.
To allow the users to play the game on different devices and locations a cloud based user data storage must be handled as well. This is also covered by the Player Model Asset (T2.4C). This will allow users to switch the devices without losing their game progress (i.e. score, level).

For the content of the game a dialog editor browser dashboard asset (Communication Scenario Editor and Player (T3.3F)) will be used to create and prepare the conversation between the user and the NPCs during the game development. A server side dialog player (Communication Scenario Editor and Player (T3.3F)) must send these texts to the game client dynamically depending on several parameters defined in the editor and the current situation in the game.

10.2 Game Design

The game will be developed for Desktop PC (Windows) with an optional support of Android Smartphone and Tablets. During development English language is used. The technical workflow of the game client will allow also other languages.

10.2.1 Story, Setting, Role of the player

The player is hired as an office assistant by a small games studio with the goal to increase the productivity of the company. The productivity is reflected in the amount of finished and shipped games and their quality.

The productivity of the company is depending primarily on the NPC team members’ individual mood levels that are not ideal from time to time. When a NPC is working its individual mood level decreases little by little. These levels can be influenced by the player by the answers selected during dialogs with the particular team members. This social interaction is taking place at the so-called “Water Cooler” where the NPCs take a break from time to time. If the choices of the player are made with a “social cleverness” the staff members can be cheered and motivated again to increase the mood levels and the company’s productivity. To improve the water cooler interactions in future the player can print out and examine a report that shows, assesses and reflects the player’s behaviours during the past conversations. See Figure 50 and Figure 51: Water Cooler Game, draft the water cooler conversation.

Figure 50: Water Cooler Game, draft the office and its water cooler place
The productivity of the company is depending secondly on the status of the “production pipeline” between the different departments that gets slow or stuck without the player’s help. To manage the production workflow the player must move the different work packages from desk to desk and from department to department to avoid that someone has to wait or gets overworked.

The productivity of the company is depending thirdly on the number of employees. The amount of staff members is limited by the player’s level and the size of the office and that is reflected by the amount of available desks. To expand the office the player has to collect experience points and level up. The experience points are awarded for several interactions during the game, especially as rewards for the shipped games.

In addition to the office expansion a level up is also rewarded with decoration objects. These things can be used optionally to embellish and customize the game studio.

10.2.2 Main goals

- “Increase the productivity of the game studio!”
- “Talk to your team members!”
- “Select the right answers to increase the mood level of your dialog partner!” using Communication Scenario Editor and Player (T3.3F)
- “Examine a report that shows, assesses and reflects your past conversations!” using Server-side Dashboard and Analysis (T2.4B)
- “Reflect and improve your own and your team’s social skills!”
- “Manage the workflow in the office by move work packages from desk to desk!”

10.2.3 Subgoals

- “Finish and ship games!”
- “Collect experience and level up!”
- “Expand the office for new employees and to produce more games!”
- “Finish games as fast as possible!”
- “Get good ratings and awards for shipped games!”
- “Decorate the office!”

Figure 51: Water Cooler Game, draft the water cooler conversation
10.2.4 Length of game (play times and sessions)

- The game will be able to be played indefinitely with no fixed final win scenario.
- A series of short or extended gameplay sessions should be possible.
- If the player is unable to maintain the productivity at all the company will fail to finish any games until the player does better again. In this situation a “game over” could be implemented, but the better way would be to teach the player how to do better next time.
- An option to reset and restart the game at any time should be implemented.

10.2.5 Register, Login and User data

- Before playing, the user has to register for the game (i.e. with a Username and Password).
- There must be different roles for the tutor and “normal players”. Tutors can apply for a tutor account that must be authorized by the admin.
- The account creation and authentication process must happen on a website also to allow a tutor to set up and invite new students and for the report functionality.¹
- To allow the user to play on different clients and not only on one pc the user data must be stored on a server and be synchronized.²
- After the registration the user can login and go on with playing on any compatible device where the game is installed.

10.2.6 User input

The game will be developed for desktop PCs but should also prepare a possible release on Android smartphones or tablets as well. Therefore also the usability concept will respect both platforms.

Platform specific inputs must be prohibited if possible, but there might be some cases where a text input is needed, i.e. for the login process. In this case the keyboards of the particular platform must be used.

There will be mainly three interactions that must be handled:

a) Interactions with the game world objects.
b) Interactions with the NPCs.
c) Interactions with the GUI.

All of these interactions will be controlled mainly by:

1. Mouse point and click interactions in the case of playing the game on a desktop PC.
2. Finger touch interactions in the case of playing the game on an android smartphone or tablet.

¹ The used asset for these requirements will be: Server-side Authentication and Authorization (T2.4A/UCM)
² The used asset for these requirements will be: Player Model (T2.4C/OUNL)
10.2.7 Navigation and representation of the player (Avatar)

The player is represented as an avatar that moves through the game world controlled by the user. The avatar moves automatically to the target spot the player has clicked before.

10.2.8 Navigation through and representation of the game world

The whole game studio is represented as a single room that will be expanded stepwise after a level up. In addition a “zoom in and out” functionality will be implemented that allows the user to change the view. The detail view of the camera can be moved to change the frame position.

10.2.9 Water Cooler NPC Interaction

The water cooler will be the central interaction spot in the game world. Here the player can talk with one of NPCs from time to time. These “conversations” prompt group working/conflict management behaviour /value based questions and multiple-choice answers for the player to choose that cover a range of attitudes, subjective to objective and in between.

This water cooler feature will be implemented as a RAGE asset that allows creating the dialog content (editor) and also controls and delivers the content of the conversations to the game client (player). It also must rate the user’s answers so the game client can handle the consequences of the player’s choices afterwards.

The questions and answers for the water cooler conversations will be used as predefined dialogs that are created, written and arranged in the RAGE asset (editor).

For the Rage asset (editor) standards will be defined to ensure that they fit into the user interface, i.e. character amount maximum, word amount maximum, and answer amount, dialog total length.

The RAGE asset (player) will deliver the content for the conversations to the game client. It should be able to use additional parameters to select the dialogs, i.e. the player’s current level, the NPCs static attributes, a predefined specific order.

It will be triggered by the game client when and with which NPC a conversation will be started. Either the user can select the NPC (limited amount) or the programs game mechanics will select it (if the player has not started a conversation for a long time).

During a conversation the workflow of the company must be paused, to avoid that the player hurries through the dialogs with pressure of time.

Each selected answer of the user will be delivered to the RAGE asset (player) that rates and stores it. After the end of a conversation the result will be sent back to the game client as a percentage score. The results of these water-cooler discussions will affect the mood level of the particular NPC and thus the overall productivity of the office as well as the rating of the shipped games. Happy NPCs due to a good rating may alleviate stress in the workplace for a while and mediate interpersonal relationships for example, while a poor rating will decrease the mood levels of the co-workers in the office.

10.2.10 Feedback Report (Print Out)

An analytics and reporting browser dashboard asset (Server-side Dashboard and Analysis (T2.4B)) allows not only developers and use case owners but also the users and tutors to “step out” of the game to analyse the data collected from gameplay. Learning from feedback, autonomously or with the tutor in a tutorial situation, formatively or as a summative assessment exercise, this can then feedback into the players continuing gameplay’ or more practically, into the player/student’s “live” group working behaviours. The players can print out reports of their “interpersonal skills development”, and perhaps use this as a method to re-model their actions/behaviours in real group projects.

The used asset for these requirements will be:
Communication Scenario Editor and Player (T3.3F/UU)

---

3 The used asset for these requirements will be:
Communication Scenario Editor and Player (T3.3F/UU)
For the feedback report two dashboards (analytics and reporting, login and authentication) will be implemented as RAGE assets. The dashboard can be called up directly in a browser using a specific URL or can be opened by using a button represented in the game world or HUD i.e. as chart symbol.

Before users can see their individual reports they have to use the login (developed as a RAGE asset) on the website first. With the help of the login authentication process the report could be customized depending on the user’s role.

Normal users (players) would only see their own report. The content of the report would be the history of their multiple choice interactions (e.g.: time, question, answer and scoring). The design of the report (its presentation) could be defined by the use case owner for an optimized layout that prepares it for a good use in a pedagogical way. It could also contain some feedback that explains the reason for the kind of rating and add some instructions to offer solutions to get better.

Tutors would be able to see the reports of all their students. The content of the report would be the history of the multiple choice interactions (e.g.: time, question, answer and scoring) and could be also contain more information if needed and specified by the use case owner.

Admins (developers and use case owners) would have access to the complete feature set of the asset.

For all user roles some configuration options could be offered to the user as well, i.e. the possibility to filter the data, an "export to “pdf” button, a charts view.

10.2.11 Workflow Interaction

The player’s role in the game is to manage this production workflow. The game development process will be implemented as a simplistic expression of collaborative working. It should impart how important each person is during a team workflow. Therefore the production will be divided in several pieces that must be processed by the following different departments step by step in the right order:

Project Manager > Writer > Level Designer > 2D Artist > 3D Artist > Animator > Coder > Sound/Music > Tester > Project Manager

![Figure 52: Water Cooler Game, draft of the office and the HUD](image)

The player’s goal is to avoid that someone has to wait or gets overworked. Therefore the player must move different work packages from desk to desk after they have been finished within the
right order. This will happen through point and click and/or maybe drag’n’drop interactions combined with an inventory. The status of the “production pipeline” will affect the duration of the development process of the game.

- The player can start the development of a game by visiting the Project Manager’s desk.
- Each game has a specific amount of work packages that have to be executed by the different departments after and after until they can be released at the end of the production line. A development duration limit has to be reached by the player to get rewards.
- Each NPC and the player’s avatar can receive a limited amount of work packages. These are represented as inventory slots with different parameters (not finished, finished, kind of work package, game type).
- The player can start more than one game simultaneously. Therefore the work packages have a specific ID so the user can see to which game it belongs.
- Depending on the level the player can start producing games with different work package amount requirements.
- Each shipped game will be rewarded, depending on the kind of game, the duration and the quality (the rewards are balanced in a way that many little games give more rewards than one big game).
- The NPCs individual mood level influences the quality of each work package.
- Each work package has a predefined duration that is needed by the NPC to finish it. This duration is also influenced by the mood level of the NPC.
- The total duration of the development of the game depends on the NPCs mood levels, but also on the players’ interactions: they have to move the work packages in an optimal and fast way.
- If the developing process is finished the player has to ship the game to receive the rating and, hopefully, an award.

10.2.12 NPCs

The NPCs will be predefined and created during the development of the game with a specific set of attributes. The behaviour of the NPCs will be controlled by the game client.

The NPCs are not meant to represent the “real” industry balance of job roles within a company of this type, but simply to provide characters with a potential range of personality types to interact with at the water-cooler. They will demonstrate a variety of physical and personality oriented combinations like, gender, name, profession and appearance (i.e. hair colour, haircut, skin colour, kind of cloth, cloth colour). Each NPC has its own desk.

The game client will handle an automated behaviour of the NPCs depending on their attribute status, the company’s attributes and the player’s interactions, i.e. waiting at the desk, work at the desk, different moods at the desk, moving to the water cooler, leaving the water cooler.

10.2.13 Games that must be developed

The games that the player has to develop have different attributes, i.e. ID, name, amount of required work packages, required department, and maximum development duration for rewards.
10.2.14 Work packages
The work packages that the player has to move have different attributes, i.e. status, department history, required duration, quality for each executed production step related to the workers mood level. These will be predefined, but some of them will be influenced by the player's interactions.

10.2.15 Ship finished games
After the last production step in the production workflow the player has to put the work packages back to the Project manager. If the latter has received all required work packages for a game, a result screen will be displayed, the game is rated and the player is rewarded.

10.2.16 Result and rewards
For each shipped game the player is rewarded with experience points. The player has to collect experience points and level up. The amount of the rewards will be predefined for each game, but is also influenced by the player’s interactions (i.e. depending on the duration needed to finish the game, the quality of the game affected by the water cooler conversations). For very good results the player gets game rewards in addition.

10.2.17 Embellishment of the office
The office can be pimped up by levelling up. Parts of the environment or objects will be replaced with more valuable variations.

10.2.18 Level system
The player has to collect experience points and level up. The experience points are awarded for several interactions during the game: answers during the water dialog conversations, interactions during the workflow management, shipped games. The required experience points for each level will be predefined and will increase constantly.

10.2.19 Unlocked rewards
Level ups will sometimes be rewarded with office expansion (amount of available desks), new staff members, new kinds of games that can be developed, embellishment of the office.

It will be predefined which level unlocks the expansions, the games, the staff members and/or the embellishments and what kind of attributes they will have. There can be level ups that unlock no, some or all of the possible kinds of rewards. The final unlock order will be balanced during the development progress.

10.2.20 Difficulties
To make the start of the game easy at the beginning there will be unlocked only games that require a low amount of work packages and departments. Higher levels will unlock higher amounts of employees to increase the difficulty of the management of workflow and that will require more conversations at the water cooler. The water cooler conversations themselves might also become more difficult and challenging. The new kind of games will require more work packages in higher levels.
10.2.21 **GUI and HUD**

Some parameters will be shown permanently to inform the user, i.e. player / company name, company life time, company productivity, number of shipped games, number of awards, experience points, company level, required experience points for next level up, number of employees / staff overview, water cooler status. In addition some buttons will be displayed, i.e. to pause the game and to display the report.

10.2.22 **Avatar and NPCs inventories**

The inventory of the player’s avatar will be displayed at the bottom of the screen. It has a limited amount of slots. Here the player can temporarily store the work packages. The inventory of a NPC will be displayed next to its desk, if the player’s avatar has arrived there. It also has a limited amount of slots. Here the player can store the work packages and pick them up if they are finished.

10.2.23 **NPC mood status**

The personal mood in the office staff/NPC’s being interacted with by the player will be signposted over the character’s heads.

10.2.24 **Graphic style**

The game world will be created in an isometric top down view. It will be illustrated in a kindly and colourful art style that fits to the target group. This art style will also be used also for the avatar and the NPC: a comic style that is not too realistic, and detailed and funny but not too cute.

The avatar and the NPCs must be designed also with the regards to their visible attributes (i.e. hair colour, haircut, skin colour, kind of cloth, cloth colour) and optional animations.

The design of the desks make it obvious which department they represent: Drawing board for artist, PCs with manuals for coders, PCs with 3D toys for 3D artists. The design of the desks will include a specific colour that will be related to the required work package. The design of the GUI and the HUD must fit to the game world style, but first of all it will allow a good usability and readability.

10.2.25 **Balancing**

A RAGE asset for Performance Statistics (T2.2E) will help the game designer to analyse and optimize the balancing of the gameplay (e.g. experience points for level up, playing durations).

10.3 **Skills and Learning Objectives**

The specific skills the game teaches and provides practice for are:

- conflict management skills
- attitudes towards equality and diversity in the workplace
- team working skills

The game will also promote the development of further skills, such as:

- Objectivity over subjectivity
- Inter-personal skills
- Leadership skills
- Acceptance of ambiguity as a reality of professional working practice
- Critical self-reflection skills
- A personal (and effective) working/design process.
The high level objectives are:

- to develop students that identify and interpret conflicts in group working as an expression of diversity, a potential growth opportunity, both personal and for the group, and an opportunity to expand upon one’s own experience instead of considering it a negative issue. Particularly one to be avoided altogether

- for the student to attempt conflict management practice, understanding their subjugation of their own personal values and attitudes for the combined success of the group, and thus the project.

- to allow students already engaged in often high level learning to augment their employability through a solid understanding that soft skills, conflict management and a recognition of the advantages of embracing diversity in the workplace and collaborative project working.

10.3.1 Pedagogy

The game design incorporates the pedagogical approach of experiential learning applying Kolb’s Experiential Learning Cycle (Kolb, 1984). Regardless of whether the game was played during a short (single day session) or used as part of a much longer Module, the player would go through the learning stages of the Kolb model:

- Playing or trying out multiple sessions in the Applied Game – Active Experimentation/Concrete Experience (albeit safe in the virtual in this instance)

- Examine own “scores” or progress by accessing the gameplay data – Reflective Observation (here the option to do this with a tutor to help interpret the data would blend this with the next phase)

- The development of an action plan based on the conclusions drawn from the data the game highlighted to the player. – Abstract Conceptualisation

- Finally going back to try again to apply the knowledge gained through the combination of successes/failures/ and subsequent reflection in the ongoing game (or perhaps the live project in the case of our longer module). – Active Experimentation

10.3.2 Context of Use

The game will be used as part of a broader education module that uses traditional practical group working as a major part. The game, as an ongoing/“endless” scenario, could be used concurrently alongside the practical sessions, while in shorter courses the game could be used in and of itself.

Both scenarios would result in a series of short or extended gameplay sessions (as appropriate) followed by discussion with the tutor/lecturer looking at data collected by the game showing the players given responses to stimuli designed to test.

Therefore the gameplay based feedback will be implemented in a report for the players and the tutor. In this report the teamwork skills are analyzed. This combination of gameplay based feedback, and tutor feedback would form the basis of an ongoing plan for the player/student when moving into actual “live” group working scenarios.

The game is a training/educational instrument and a self-assessment tool, to focus students’ on particular areas for study/learning paths and choices related to strengths and weakness flagged up by the game.
The game is aimed at an individual learning strategy (within a larger group of peers) supported in studio with tutorial discussion, prior to submersion in a “live”, peer-to-peer and client oriented group learning project with definite client expectations and where the impact of poor group working will affect the students ability to evidence learning/professional skills acquisition and so hinder their summative assessment potential.

10.3.3 Measurement

The “Water Cooler Conversations” of the game in which the student/player is confronted with emotive/confrontational/positive/negative contextual conversational Q&A by NPCs will be used to measure the success or effectiveness in the game or outside the game.

The responses offered to the player in each instance will have a weighting, perhaps group/set oriented, or simplistically numeric; these will then be grouped over time (though ideally influencing the systems choice of further interactions) to provide an overview of the players attitudes/given responses into typical group working/work based social interactions as measured against a framework that looks at positive/negative and perhaps ambivalent attitudes/required skills in the field of soft skills group working.

The “continual play” style of the game should allow for a monitoring of a change in values/approach on the part of the player. For example, should the player start poorly, with responses within the game that cause “productivity” (in this instance the measure of success in the wider game) to go down, this should engender the player to perhaps change the approach and try alternative strategies. Over time, these changes in strategy could be tracked as the data based on player choices is recorded and logged.

External to the game, the final summative measurement would be the student’s qualification/level of qualification within the wider module/course into which the applied game is placed as a component element of a wider teaching strategy.

10.3.4 Feedback

The analytics and reporting the browser dashboard asset (Server-side Dashboard and Analysis (T2.4B)) allows the users and tutors to “step out” of the game to analyze the data collected from gameplay. The players can print out reports of their “interpersonal skills development”, and perhaps use this as a method to re-model their actions/behaviours in real group projects. Ideally a situation in which both tutor and student can review the data/“scoring” would be best.

The design of the report (its presentation) and its layout will be prepared for a good use in a pedagogical way. It could also contain some feedback that explains the reason for the kind of rating and add some instructions to offer solutions to get better.

Besides that the players get feedback to their performances directly in the game through the NPCs mood levels, the productivity value of the game studio and the rating of the shipped games.

10.4 Rage Assets

10.4.1 Server-side Authentication and Authorization (T2.4A/UCM)

The project will be used by different users that will need access to different kind of application scenarios that are connected to one or more servers. Because of that an asset is needed that handles the different user accounts.

- This asset should handle the registration and authentication process for
  - the dialog editor,
  - the game and
  - the dashboard for analytics and reports.
• Although these things are different in a technical point of view there should be only one registration process for the users in total (therefore the accounts must be linked).

• The registration must be offered on a server side browser dashboard, where the user can create an account with different options (i.e. user name, role, password, email verification, ID).

• Different user roles must be handled:
  o Admins must get access to a user management section, i.e. to set the roles that are requested by new users.
  o Editors must get access to the dialog editor.
  o Tutors must get access to the detailed analytics and report dashboard.
  o Standard users must get access to game and the user report dashboard.

10.4.2 Server-side Dashboard and Analysis (T2.4B/UCM)

This asset must offer a browser dashboard that allows visualizing the data of the server side interaction storage. The kind of data that is shown and the kind of configuration options (i.e. filter, calculate, analyse) are depending on the users’ role that is defined in their account (i.e. tutors must see the data of all players, normal players just their individual report). NUROG and HCG will specify the requirements for this Analytics Dashboard for the Water Cooler Game and OUNL and UCM will configure and implement the dashboard based on the framework of UCM.

• The dashboard can be called up directly in a browser using a specific URL or can be opened by using a button represented in the game world or HUD i.e. as chart symbol.

• Before players can see their individual report they have to login (handled by the “Account creation and authentication asset”) on the website first. With the help of the login authentication process the report could be customized depending on the user’s role.

• The players want to get the report that assesses their behaviour during the “Water cooler Dialogs”. The content of the report will be the history of their multiple choice interactions (e.g.: time, question, answer and scoring). The design of the report (its presentation) will be defined by the use case owner for an optimized layout that prepares it for a good use in a pedagogical way. It should also contain some feedback that explains the reason for the kind of rating and add some instructions to offer solutions to get better.

• Tutors will be able to see the reports of all users (students). The content of the report will be the history of the multiple choice interactions (e.g.: time, question, answer and scoring) and could be also contain more information if needed and specified by the use case owner.

• The developers, game designers and use case owners want to know how the players are interacting with the game to optimize it (i.e. the balancing) and would have access to the complete feature set of the asset.

• For all user roles some configuration options must be offered to the user as well, i.e.
  o The possibility to filter the data,
  o An “export to *pdf” button.
  o Charts view.
10.4.3 Client-side interaction tracking (T2.1 A / UCM)
- This asset must track the behaviour of the players and their interactions while playing (i.e. amount, kind, order, frequency and lengths) and send it to a server side storage.

10.4.4 Server-side interaction storage and analytics asset (T2.1 B / UCM)
- This asset must receive and save the data that is collected by the client side interaction tracking asset
- Also other data that is generated and sent by other assets (i.e. the interaction of the dialog player) must be received and stored.

10.4.5 Communication Scenario Editor (T3.3F/UU)
Dialog editor for the “Water Cooler”-Feature must handle the following:
- offer a browser dashboard
- allow to write, arrange, test prepare and export the dialogs
- export the dialogs to the server side interaction storage asset
- different languages and the particular spelling and grammar rules

10.4.6 Communication Scenario Player (T3.3F/UU)
Dialog player for the “Water Cooler”-Feature must handle the following:
- receive an event of the game client with additional data (i.e. NPC attribute)
- select predefined question and answers, depending on the different parameters (i.e. rating of the previous answer, amount of already answered questions, NPC attributes)
- sends them to the game client
- after the user has made his choice on the client the dialog player has to receive the answer
- save the answers of the user
- rate the answers (i.e. “30 %”)
- send the rating of the answers back to the client
- export the data to the server side interaction storage asset

All interactions with the “Communication Scenario Player” will be technically realised via a RESTful Interface.

10.4.7 Performance Statistics (T2.2E/OUNL)
The Performance Statistics Asset will be implemented using the “Server-side Dashboard and Analysis asset (T2.4B/UCM)” of UCM and be part of the Analytics Dashboard of the Water Cooler Game. NUROG and HCG will specify the requirements for this Analytics Dashboard for the Water Cooler Game and OUNL and UCM will configure and implement the dashboard based on the framework of UCM according to the requirements and specifications of HCG and NUROG.
This asset will help the game designer to analyse and optimize the balancing of the gameplay (e.g. experience points for level up, playing durations) as a balancing analytics and optimization dash board.

This asset will also give both the tutor/teacher and the student feedback of the student's performance and be the basis for the periodic conversations between the tutor/teacher and the student.

10.4.8 Evaluation of Applied Games (T8.2A/TUGRAZ)
This asset will allow in-game evaluation of the Water Cooler Game based on users' interactions with a game. The evaluation asset will collect data in a non-invasive and continuous manner. It will monitor the player's usage behaviour and interactions in a game and translate it into meaningful information about different aspects of game quality and user experience.

10.5 Technical Specification

10.5.1 Development

- Hardware: Desktop PC (Optional: Tablet / Smartphone)
- OS: Windows, optional deployed for Android
- Game engine: Cocos 2d-x
- Programming language: C++

10.5.2 Target Requirements

- Hardware: Desktop PC, optional: tablet / smartphone
- Display resolution / aspect ratio: Multiresolution - Optimized for 1920 x1080 (Full HD) / Landscape mode
- Input: Mouse, Keyboard, optional: touch screen
- OS: Windows 7 or higher, optional: Android 5 or higher
- File format: *.exe (windows executable), optional:*.apk (Android App)
- Network: Internet connection
- Additional software: Browser

10.5.3 Assets

The game client will not provide any interfaces to the RAGE assets. The game client addresses the interfaces of the RAGE assets.
CONCLUSION

This document provides the blueprint for development of the software aimed at application scenario pilots based on the preliminary design documents’ integrated feedback, incorporating game design, instructional design as well as iterations on concepts and functionalities carried out in conjunction with WP1, 2, 3 and 5.

As is customary, designs do change as development progresses, and such changes will be needed to respond to further feedback and clarification as the development effort continues in WP5, toward the delivery of D4.3, the initial version of game applications for scenario pilots and D4.4, the final version of game applications for scenario pilots.
12 REFERENCES


