Realising and Applied Gaming Eco-system

Research and Innovation Action

Grant agreement no.: 644187

D2.3 - First bundle of client-side components

RAGE – WP2 – D2.3

<table>
<thead>
<tr>
<th>Project Number</th>
<th>H2020-ICT-2014-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date</td>
<td>30 June 2016</td>
</tr>
<tr>
<td>Actual Date</td>
<td>15 July 2016</td>
</tr>
<tr>
<td>Document Author/s</td>
<td>e-UCM</td>
</tr>
<tr>
<td>Version</td>
<td>0.4</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>RE</td>
</tr>
<tr>
<td>Status</td>
<td>Draft</td>
</tr>
<tr>
<td>Document approved by</td>
<td>Wim</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>28 June 2016</td>
<td>First version</td>
<td>MFM</td>
</tr>
<tr>
<td>0.2</td>
<td>30 June 2016</td>
<td>Initial round of comments</td>
<td>KB, DV, BB, MFM</td>
</tr>
<tr>
<td>0.3</td>
<td>1 July 2016</td>
<td>First “final” version</td>
<td>MFM</td>
</tr>
<tr>
<td>0.4</td>
<td>12 July 2016</td>
<td>Factored in D2.3e asset</td>
<td>JD, MFM</td>
</tr>
</tbody>
</table>

### Document Change Commentator or Author

<table>
<thead>
<tr>
<th>Author Initials</th>
<th>Name of Author</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFM</td>
<td>Manuel Freire</td>
<td>UCM</td>
</tr>
<tr>
<td>KB</td>
<td>Kiavash Bahreini</td>
<td>OUNL</td>
</tr>
<tr>
<td>DV</td>
<td>Dessislava Vassileva</td>
<td>USofia</td>
</tr>
<tr>
<td>BB</td>
<td>Boyan Bontchev</td>
<td>USofia</td>
</tr>
<tr>
<td>JD</td>
<td>João Dias</td>
<td>INESC-ID</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></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

Page

EXECUTIVE SUMMARY
INCLUDED ASSETS
  2.1a Client Tracker (UCM)
    Status
    Links
  2.2b Domain Model Asset (TUGraz)
    Current status
    Links
  2.2c Competence Assessment Asset (TUGraz)
    Current status
    Links
  2.3c Motivation Assessment Asset - (TUGraz)
    Current status
    Links
  2.3d Real-Time Arousal Detection Using Galvanic Skin Response (USofia)
    Current status
    Links
  2.4c Game Storage - Client-Side (OUNL)
    Current status
    Links

CONCLUSIONS
EXECUTIVE SUMMARY

This document describes the first bundle of core WP2 (user data analytics) client side components, including their specifications, use-cases, and working prototypes.

Included assets contain a description of their current status, and links to their full designs and downloadable versions.

This deliverable only describes operational SW assets (even though beta) that are tested and documented. It should be noted, however, that various additional software assets (2.2d Cognitive Capacity Measurement and 2.3a Real-time Emotion Detection) are near completion for inclusion in games during the first pilot round. Those assets are still scheduled for inclusion in the final bundle deliverable D2.2.
1 INCLUDED ASSETS

2.1a Client Tracker (UCM)

After a game is developed, a common need is to know how the players play, what interactions they follow within the game and how much time they spend in a game session; collectively, these are known as game analytics. Analytics are used to locate gameplay bottlenecks and assess game effectiveness and learning outcomes, among other tasks. This component sends analytics information to a server; or, if the server is currently unavailable, stores them locally until it becomes available again. It is a central part of the RAGE analytics ecosystem.

The two versions of this asset are in the process of being merged.

Status

Fully functional, although testing outside of Unity has been minimal.

Links

- Sources
  - https://github.com/e-ucm/unity-tracker (stand-alone unity version)
  - https://github.com/e-ucm/ClientSideTrackerAsset (RAGE client-side asset version)
- Design
  - https://docs.google.com/document/d/1_XtAMvI9mNhR5Qxc9RHF6OWhRBjyL2lg6biAwtZegAQ/edit?usp=sharing
- User documentation
  - https://github.com/e-ucm/unity-tracker/blob/master/README.md
- Example integration with game
  - https://github.com/e-ucm/QuizDemo
2.2b Domain Model Asset (TUGraz)

The goal of the Competence Assessment Asset is to uncover the competence state of a player while playing a game. In the game development process the competences covered by a game are defined and structured. Furthermore, the events or tasks in a game are identified that give evidence whether an individual competence is available or not. During game play this information is used by the asset to identify and update the competences that a player has available. The assessment result can be used by other assets to adapt the game play or to select an appropriate game. In addition, the result can be visualized in the dashboard of the UCM infrastructure. The figure below shows an illustration how the competences that are demonstrated by the player during the game play are assessed by the asset.

The main value of this asset is to define a domain knowledge in terms of the competences that should be learned with a game.

Current status

- The main functionality of providing the domain model is complete
- Missing feature is the external authoring tool that is currently being developed

Links

- Sources [https://github.com/RAGE-TUGraz/CompetenceBasedAssets](https://github.com/RAGE-TUGraz/CompetenceBasedAssets)
- Demonstration [to be released] [http://css-kti.tugraz.at/projects/rage/assets/demonstration/Demonstration-DomainModelAsset.pdf](http://css-kti.tugraz.at/projects/rage/assets/demonstration/Demonstration-DomainModelAsset.pdf)
2.2c Competence Assessment Asset (TUGraz)

The goal of the Competence Assessment Asset is to uncover the competence state of a player while playing a game. In the game development process the competences covered by a game are defined and structured. Furthermore, the events or tasks in a game are identified that give evidence whether an individual competence is available or not. During game play this information is used by the asset to identify and update the competences that a player has available. The assessment result can be used by other assets to adapt the game play or to select an appropriate game. In addition, the result can be visualized in the dashboard of the UCM infrastructure. The figure below shows an illustration how the competences that are demonstrated by the player during the game play are assessed by the asset.

The main value of this asset is the detection of the competences a player possesses. In many cases, acquiring competences is the main goal of performing a learning activity. Through the assessment it can be validated if the learning goal is being achieved or if there is still a competence gap.

**Current status**
- main functionality is complete
- missing feature: connection with the Interaction Tracker (available by mid of July 2016)
- missing feature: connection with the Game Storage Asset (unclear)

**Links**
- Sources
  [https://github.com/RAGE-TUGraz/CompetenceBasedAssets](https://github.com/RAGE-TUGraz/CompetenceBasedAssets)
- Release
- Design
- Demonstration [to be released]
- User documentation
2.3c Motivation Assessment Asset - (TUGraz)

The Motivation Assessment Asset aims at assessing the player's motivation to learn while playing a game. One's motivational state is a crucial aspect within a game and strongly influences how often the game is played or if it is positively perceived. By analysing interaction within the game confidence, attention and satisfaction as three components of motivation are measured and provided to other assets for further processing, e.g. to the motivation-based adaptation asset (T3.4) for maintaining or supporting motivation.

The key value is that this asset provides information about a gamer's current motivation to learn and its changing states over gaming episodes. The motivation assessment is done non-intrusively and may be used for adaptation of the game in order to maintain and enhance motivation.

Current status
- main functionality is complete
- missing feature: connection with the Interaction Tracker (available by mid of July 2016)
- missing feature: connection with the Game Storage Asset (unclear)

Links
- Sources: https://github.com/RAGE-TUGraz/MotivationBasedAssets
- Release: http://css-kti.tugraz.at/projects/rage/assets/software/MotivationAssessmentAsset.zip
2.3d Real-Time Arousal Detection Using Galvanic Skin Response (USofia)

The real-time arousal detection using galvanic skin response asset detects in real time human arousal based on measuring electro-dermal Activity (EDA), also known as skin conductance, galvanic skin response (GSR), electrodermal response (EDR), skin conductance response (SCR), and skin conductance level (SCL). EDA is related to the activity of the sweat glands, which are regulated by the sympathetic nervous system and it is measured from particular player in order to produce real-time signal features such as: mean tonic activity level, phasic activity represented by mean and maximum amplitude of skin conductance response, rate of phasic activity, SCR rise time, SCR 1/2 recovery time, and slope of tonic activity. The level of arousal may be useful for emotion detection and for adaptation purposes. The asset receives a filtered raw signal from a simple, low cost biofeedback device allowing sampling rate up to 0.8Khz. Measurements are carried out with two electrodes placed on two adjacent fingers. Recording, filtering and feature extraction might be executed on a computer (server) different than the game machine, in order to speed up all the required processing. The results are communicated from the server-side to the client component in order to be used for game adaptation.

Current status

The asset produces two main metrics featuring user arousal based on the GSR signal, namely:

- Current level of phasic user arousal measured in N levels, i.e. from 0 to N-1, where N is user-defined and has a default value (the level is indicated by the area under the curve of the filtered GSR signal within the time window using a sampling rate chosen by the asset user). The user arousal level is determined after the first time window is expired and is based on statistical analysis of human GSR signals, thus, the accuracy is based on statistically significant data. With collecting more data from given user, the asset starts providing more accurate data.
- Current level of tonic arousal measured in N levels, i.e. from 0 to N-1, where N is user-defined and has a default value (the level is indicated by the mean amplitude of the tonic component of the signal within the time window and using a sampling rate chosen by the asset user).

As well, the asset produces real-time features of GSR signal measured from particular player such as: phasic activity represented by mean and maximum amplitude of skin conductance response (all in micro-siemens), rate of phasic activity (response peaks/sec), SCR rise time, SCR 1/2 recovery time, mean tonic activity level, and slope of tonic activity (in micro-siemens/sec).

Links

- Sources https://github.com/ddessy/RealTimeArousalDetectionUsingGSR
- End-user documentation https://github.com/ddessy/RealTimeArousalDetectionUsingGSR#installation-and-usage
D2.3 - First bundle of client-side components

- Video demo
  https://www.youtube.com/watch?v=fJvbghlZi1c
- Usage presentation
- Design
2.3e - **Client-Side Real-Time Emotion Recognition**

This asset is responsible for integrating multiple emotion recognition modalities and combining them into a single classification. In its final version it will incorporate 5 different modalities: facial expression recognition, text recognition, speech recognition, touch recognition and GSR/EDA recognition for arousal. The client-side version is designed to run directly in the client application and communicate using web-services with server-side Emotion Recognition Assets (e.g Speech and Text Emotion Recognition Assets). This asset is prepared to run with any number of emotion recognition modalities, and their configuration and incorporation into the classification fusion process is straightforward.

**Current status**

For this first release, three emotion recognition modalities were integrated (the ones available during this release): the Sentiment analysis asset from UPB, the speech emotion recognition asset from INESC-ID, and an internally developed asset for EDA recognition which will be later replaced by the Real-Time Arousal Detection asset from TUSofia. Three classifier fusion policies were implemented for this first release: max, weighted average, and kalman fusion. The max policy just uses the maximum value of its classifiers for predicting a particular emotion, while the weighted average policy takes into account a weighted average from all classifiers contributing to a particular emotion. The kalman fusion policy is the most complex, and it is based on the system proposed in [1] using a kalman filter to combine multiple sensors. The main advantage of this policy is that it can more easily deal with situations where one of the sensors is not currently active, by using information from past observations and explicitly assuming a higher uncertainty about these observations.


**Links**

- Sources  
  https://github.com/GAIPS-INESC-ID/FAtiMA-Toolkit
- Download (Source Code)  
- Download (DLLs + Demo + Documentation + Demo Description)  
  https://www.dropbox.com/sh/mbwd1ew0vlqc6sg/AADKPbzV3guO5PhQZY6m_Wxka
2.4c.c Game Storage - Client-Side (OUNL)

This asset offers a simple storage of (hierarchical/historical) data for inter-asset information exchange, or to persist to a server (using the Game Storage - Server-Side asset, 2.4c.s).

The model is situated client-side, and is intended to replace the Player Model Asset (which it replaces). It can store any object that can be serialized (for C# with the build-in XmlSerializer class, for TypeScript JSON using the browser JSON object). Lists and Models are also allowed as value, enabling storage of hierarchical/historical data as well.

Current status

Working prototype; local storage working fine, and initial integration tests with server-side are successful. Work in progress:
- missing cross-platform tests
- connectivity errors currently not handled

Links

- Sources  
  https://github.com/rageappliedgame/ClientSideGameStorageAsset
- Design  
  https://rage.ou.nl/filedepot_download/36/509
2 CONCLUSIONS

WP2 presents significant challenges in its main goal of producing assets that focus on different perspectives of data collection (e.g. interaction data vs. emotion data) but can be seamlessly combined in game development projects.

This first deliverable includes prototypes for WP2’s core client-side assets, and demonstrates significant progress towards the goals of WP2.

All assets in this deliverable will continue to be improved, based partly on the feedback resulting from inclusion into RAGE pilots.