



Realising an Applied Gaming Eco-system

Research and Innovation Action

Grant agreement no.: 644187

D4.3 – Initial version of game applications for scenario pilots

RAGE – WP4 – D4.3

Project Number	H2020-ICT-2014-1
Due Date	November 2016
Actual Date	November 2016
Document Author/s	Kam Star, Ben Hirsh, Jack Coldridge, Ellis Spice, Thierry Platon, Alida Ruggeri, Boris Irmischer, Jens Piesk, Marco Polak, Marcus Linder, Holger Sprengel, Andrew Pomazansky, Sarah Humphreys, Gareth Sleightholms, Ian Saunter
Version	0.9
Dissemination level	PU
Status	Final
Document approved by	KS/WW



Document Version Control			
Version	Date	Change Made (and if appropriate reason for change)	Initials of Commentator(s) or Author(s)
0.1	1/11/2016	SportsTeam Manager added	KS, ES, AR
0.2	9/11/2016	Job Quest added	KS, TP
0.3	10/11/2016	Space Mods Inc and IT Alert! added	KS, BH, JC
0.4	11/11/2016	WaterCooler	BI, AP, SH, GS, HS, ML, MP, JP
0.5	14/11/2016	Formatting and harmonizing the text	KS
0.6	14/11/2016	Added HATCH section	IS
0.7	15/11/2016	Added ISPO section	IS
0.71	15/11/2016	Formatting and proof reading	KS
0.8	25/11/2016	Addressing comments from internal reviewers	KS, IS, AP, TP
0.9	30/11/2016	Minor amendments	KS

Document Change Commentator or Author		
Author Initials	Name of Author	Institution
KS	Kam Star	PlayGen
JC	Jack Coldridge	PlayGen
BH	Ben Hirsh	PlayGen
ES	Ellis Spice	PlayGen
AR	Alida Ruggeri	PlayGen
BI	Boris Irmischer	Nurogames
JP	Jens Piesk	Nurogames
MP	Marco Polak	Nurogames
ML	Marcus Lindner	Nurogames
HS	Holger Sprengel	Nurogames
AP	Andrew Pomazansky	Nurogames
SH	Sarah Humphreys	Hull College Group
GS	Gareth Sleightholms	Hull College Group
TP	Thierry Platon	BIP
IS	Ian Saunter	Gameware

Document Quality Control			
Version QA	Date	Comments (and if appropriate reason for change)	Initials of QA Person
0.71	15/11/2016	Comments provided in-line & as QA template	KRS
0.71	15/11/2016	As above	SM

TABLE OF CONTENTS

	Page
EXECUTIVE SUMMARY	5
1 SPORTS TEAM MANAGER	6
1.1 Overview.....	6
1.2 Use of RAGE assets.....	6
1.2.1 Role-Play Virtual Character (T3.5).....	6
1.2.2 Integrated Authoring Tool (T3.7).....	7
1.2.3 Interaction Tracking (T2.1).....	7
1.2.4 Storage & Analytics(T2.2).....	7
1.2.5 SUGAR Social Gamification Assets (T3.6).....	7
1.3 Game screenshots.....	8
1.3.1 Pre-game Menus.....	8
1.3.2 Team Management.....	9
1.3.3 Position Pop-Up.....	9
1.3.4 Team Member Pop-Up.....	10
1.3.5 Recruitment Pop-Up.....	10
1.3.6 Post-Race Event Screen.....	11
1.3.7 Learning Pill Pop-Up.....	11
1.3.8 Results Screen.....	11
2 JOB QUEST	12
2.1 Overview.....	12
2.1 Use of RAGE assets.....	12
2.1.1 Virtual Human Controller (T3.2).....	12
2.1.2 The Player Profiling (T3.4).....	14
2.2 Future RAGE asset utilisation.....	14
2.3 Game screenshots.....	15
3 SPACE MODULES INC	18
3.1 Overview.....	18
3.2 Use of RAGE assets.....	18
3.2.1 Role-play Virtual Character.....	18
3.2.2 Integrated Authoring Tool.....	19
3.2.3 Interaction Tracking.....	19
3.2.4 Storage & Analytics.....	19
3.2.5 Social Gamification Components.....	19
3.3 Game Screenshots.....	20
3.3.1 Pre-game Menus.....	20
3.3.2 Game Interface.....	20
3.3.3 Feedback Screens.....	21
4 IT ALERT!	22
4.1 Overview.....	22
4.2 Use of RAGE assets.....	22
4.2.1 Social Gamification.....	22
4.2.2 Client & Server-side Interaction Tracking.....	23
4.2.3 Storage & Analytics.....	23
4.3 Future RAGE asset utilisation.....	24
4.3.1 Game Configuration Asset.....	24
4.3.2 Performance Statistics.....	24
4.3.3 Heterogeneous Adaptation.....	24
4.3.4 Text-to-speech Client/Server.....	24
4.4 Game Flow.....	25
4.5 Game screenshots.....	25
4.5.1 Game start splash screen.....	25
4.5.2 Social Asset Authentication.....	26
4.5.3 Main Menu.....	26
4.5.4 Create Game.....	26
4.5.5 Join Existing Game.....	26
4.5.6 Game Lobby.....	26

4.5.7	Player Settings	27
4.5.8	Team Feedback.....	27
4.5.9	Game Settings	27
4.6	Game Scenarios.....	27
4.6.1	Scenario 1: The basics	27
4.6.2	Scenario 2 : Virus classification, spreading behaviour, and failure.....	30
4.6.3	Scenario 3 : Teamwork, scoring and cooperation.....	31
5	WATERCOOLER GAME	32
5.1	Overview.....	32
5.2	Use of RAGE assets.....	33
5.3	Game screenshots	34
5.3.1	Login Screen	34
5.3.2	Game Mode Menu.....	34
5.3.3	Studio and Avatar	34
5.3.4	Water Cooler	35
5.3.5	The results.....	35
6	HATCH – THE CREATIVE ENTREPRENEUR.....	36
6.1	Overview.....	36
6.2	Key activities and outcomes.....	36
6.3	Game Flow and Game Modules.....	37
	Self-Assessment & Self-Development.....	37
	Legal Issues in business.....	37
	Financial Issues	37
	Project Management.....	37
	Manufacturing	38
	Audience and Market.....	38
	Marketing Channel.....	38
6.4	The Game Model (Hatch Engine).....	39
6.5	Game Development and Rage Asset Utilisation	40
6.5.1	Client side interaction tracking. (T2.1).....	40
6.5.2	Server side storage and analytics (T2.1).....	40
6.5.3	Client side non-obtrusive competence assessment (T2.2)	40
6.5.4	Server side authentication and authorisation (T2.4)	40
6.5.5	Server side dashboard and analysis (T2.4)	41
6.5.6	Heterogeneous Adaptation – HAT (T3.4).....	41
6.5.7	Social gamification components – SUGAR (T3.6)	41
7	ISPO – INTERVIEW SKILLS FOR POLICE OFFICERS	42
7.1	Overview.....	42
7.2	Simulation Components and Flow.....	42
7.2.1	Script Authoring	43
7.2.2	Video capture and editing.....	44
7.2.3	Simulation player	44
7.3	The Interview Model	45
7.4	Rage Asset Utilisation	45
7.4.1	Client-side interaction tracking, reporting and analysis (T2.1, T2.3, T8.2).....	45
7.4.2	User-login and authentication systems (T2.4).....	45
7.4.3	Heterogeneous Adaptation – HAT (T3.4).....	45
7.4.4	Cognitive intervention (T3.4)	46
7.4.5	Speech recognition and I/O (T3.3).....	46
8	CONCLUSIONS	47
9	REFERENCES	47

LIST OF FIGURES

Figure 1 Sports Team Manager Flow.....	6
Figure 2 Job Quest Game Flow	12
Figure 3 Use of Behavior Markup Language in Job Quest	13

Figure 4 Assets and Components in Job Quest.....	14
Figure 5 Space Modules Inc Game Flow	18
Figure 6 IT Alert! Game Flow	25
Figure 7 Watercooler Game Logo	32
Figure 8 Overview Assets and interfaces for Watercooler game	33
Figure 9 HATCH Game Market Demographic	38
Figure 10 HATCH game Social Media Channels.....	38
Figure 11 HATCH Engine Game Flow	39
Figure 12 Main components of ISPO	42
Figure 13 ISPO Overview Loading Screen	43
Figure 14 ISPO Questions Script Editing Tool	43
Figure 15 ISPO Video Capture and Editing	44
Figure 16 ISPO run-time screenshot (windowed mode)	44
Figure 17 Use of RAGE assets in ISPO.....	46

EXECUTIVE SUMMARY

This intermediate report provides an overview of the current versions of the RAGE games made using the RAGE assets for the first round of pilot testing and formative evaluation. The document serves as internal communication and discussion in RAGE among game companies and asset developers together with case owners and evaluators. Whilst detailed description of the design of the games, together with their learning outcomes and piloting can be found in D4.2 and D5.1, updates and changes to the designs, game flow and use of assets are included here.

The document is structured as follows: each game is contained in its own section, which is then divided into a section covering a short overview, a section on the RAGE assets being used and a section providing actual game screenshots from the current version of the game, together with narrated description.

1 SPORTS TEAM MANAGER

SportsTeam Manager is developed by PlayGen for the OKKAM use case. It is written in C# and uses the Unity 3D game engine.

1.1 Overview

Sports Team Manager is a single player game where players assume the role of a sailing team manager. Players must hire, fires and communicate with their team members, deciding on which team members are placed into each position per race and resolve conflict situations as they arise. The diagram below presents the game flow during an individual race session.

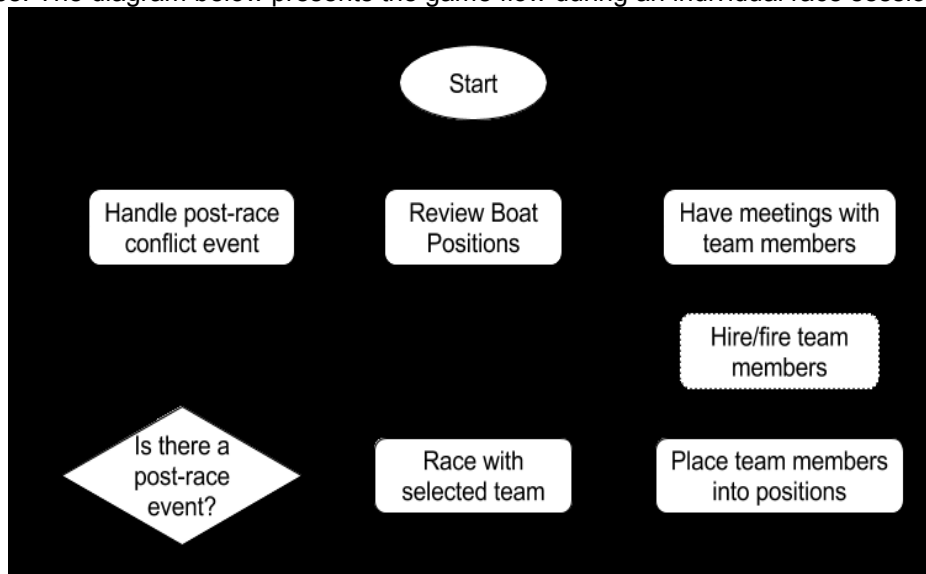


Figure 1 Sports Team Manager Flow

The player must first review the positions they need to fill on the boat, taking note of the required skills for each. Next, they must meet with their NPC team members, taking into account the skills and inter-team relationships already known, and asking questions where further information is needed. Using this information they should, if required, recruit new members into the team and place individuals into positions. After racing with the selected line-up, players will occasionally have to handle events with team members. After the event stage concludes, using the result and pieces of feedback from the race session, players begin the gameplay loop again, but now with additional information to assist in their decision making.

1.2 Use of RAGE assets

1.2.1 Role-Play Virtual Character (T3.5)

This asset, along with the related Emotional Appraisal (EA) and Emotion Decision Making (EDM) assets, is used to model the emotional state and decision making of NPCs based upon their known knowledge, referred to as 'beliefs'. The Role-Play Character (RPC) asset is used to facilitate communication between the above assets and can also include the use of the Social Importance (SI) asset, although this is currently unused within Sports Team Manager.

The Emotional Appraisal asset takes in events, usually based on input from the player, to update the emotional state of NPC team members. This effect on the emotional state is based on the current state of the NPC, the emotional weighting of the event and the valence of emotions this produces. In turn, the product of all the emotional states creates a mood value for the NPC, with this value used to decide which facial expression are displayed in-game. The

Emotional Decision Making asset, meanwhile, can be used to calculate the reaction of NPC to events (such as the next dialogue state to move into), without any direct impact on their emotional state.

In addition to this, the Emotional Appraisal asset is also used to store the 'beliefs' of every NPCs and save these beliefs over multiple play sessions. For NPCs, these beliefs are made up of information such as their last position in the team, skill ratings, opinion ratings and event states. An Emotional Appraisal file is also created for the players' in-game manager character, which stores information such as current boat type, team colours and team allowance amount. Furthermore, events sent to characters are saved, meaning a history of events can be preserved. This allows a history of every team selection to be stored within the file for the manager. As all of this information is stored regularly, it can be also be reloaded in further play sessions, allowing for the possibility of a persistent game.

1.2.2 Integrated Authoring Tool (T3.7)

This asset provides the ability to define a large collecting of Role-Play Virtual Characters into a single scenario. A connected system allows dialogue choices to be made authored for both NPCs and the player.

In the case of Sports Team Manager, an Integrated Authoring Tool file is created in order to contain references to all of the Role-Play Virtual Characters dynamically created at the beginning of and during each game. This file also contains all of the dialogue options for the player and NPCs during various parts of the game, such as team member meetings and post-race events. An additional Integrated Authoring Tool file is also used in order to store all of the available learning pill dialogue.

1.2.3 Interaction Tracking (T2.1)

This asset is used to track user interaction with the UI of Sports Team Manager, including interaction with dialogue options in NPC conversation and the opening and closing of pop-ups for positions, team members and learning pills.

1.2.4 Storage & Analytics(T2.2)

The information gathered using the Interaction Tracking asset is sent to and aggregated on a system based around the storage of the analytics data. This data is used to produce visual information that can be more easily reviewed and analysed by both developers and teachers using the related front-end.

1.2.5 SUGAR Social Gamification Assets (T3.6)

This asset will be used within Sports Team Manager in order to gather information on player progress through the game, with this information used to reward players with achievements for their performance/actions and leaderboards to compare their performance with others.

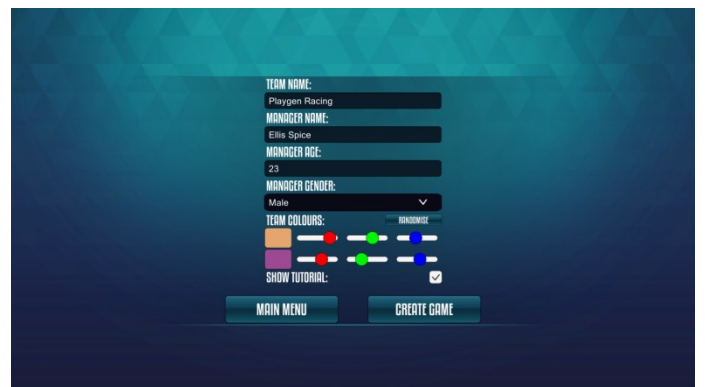
1.3 Game screenshots

1.3.1 Pre-game Menus

The game start menu, allows players to start a new game, continue a previously started game and change the volume and language settings.



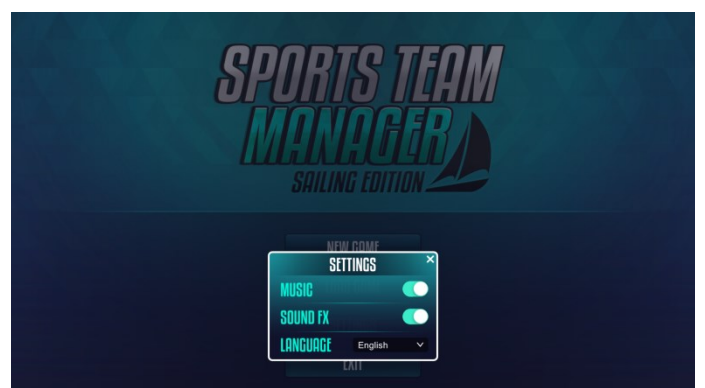
On the 'New Game' screen, players are given spaces where they must provide the name of their team and the name, age and gender of the team manager (who they will play as). Players also pick a team color and whether the game should display the tutorial at the beginning of the game.



The 'Load Game' screen allows players to continue all previously started games on that device.



The 'Settings' pop-up allows players to toggle whether sound effects and music should be played and to select which language is used throughout the game.



1.3.2 Team Management

The Team Management screen is the main hub screen throughout the game. From here, players are able to gather further information on positions and team members by clicking on them. In order to race, players click and drag team member icons onto the position they wish to place them in.

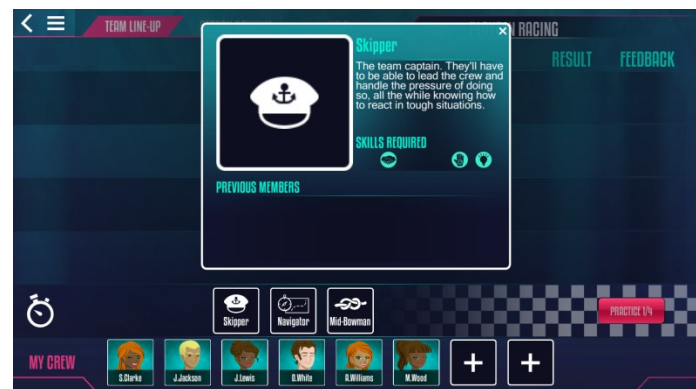


After a few race sessions have been completed, players are also able to access their complete history of selections, including the result and feedback for each.



1.3.3 Position Pop-Up

The position pop-up screen, accessed by either clicking on a position icon or the position button on the team member pop-up, displays what skills are needed in the position and a short description for the position.



When the pop-up is opened whilst someone is placed in that position, their icon is displayed in the top-left instead of the position icon. After some racing, players are also able to see which team members have been placed in that position, sorted by most common to least. Both of these icons can then be clicked to take the player to the meeting screen for that team member.

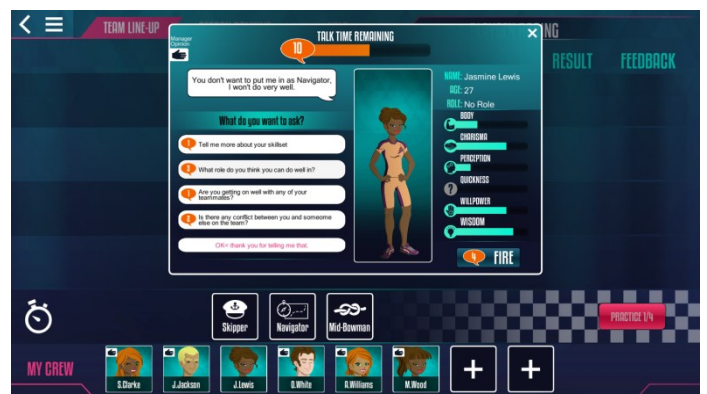


1.3.4 Team Member Pop-Up

The team member pop-up screen, accessed by clicking on a team member icon either on the team management screen or position pop-up, displays all the known information for that team member. Players are able to ask this team member questions to reveal further information, but at a small time cost. Opening this pop-up also displays this team member's opinion of every other team member and the manager.

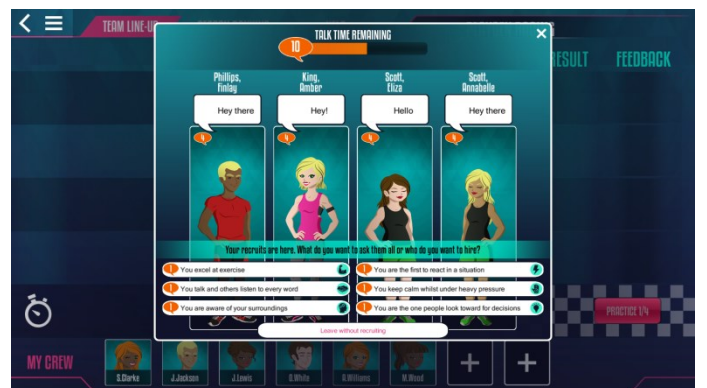


After asking about skillset, skill ratings are revealed to the player, as displayed with the bars on the right-hand side of the pop-up.



1.3.5 Recruitment Pop-Up

The recruitment pop-up, accessed by clicking on the plus icon buttons on the team management screen, allows players to question four possible recruits and hire them into their team.

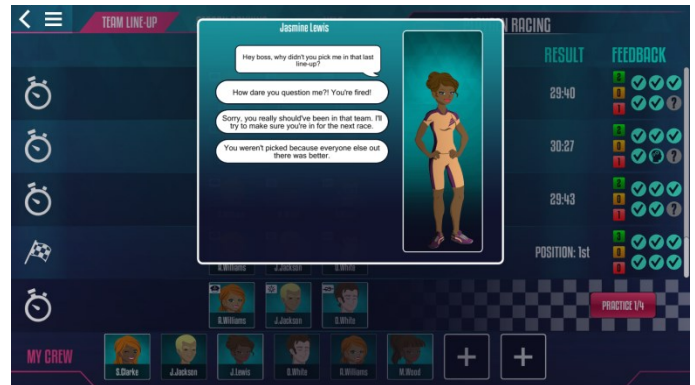


By clicking on any of the skill statements at the bottom of the screen, each recruit replies with how much they feel that statement describes them. This means their skillset can be determined. The player can then hire a recruit.



1.3.6 Post-Race Event Screen

After race sessions, team members can sometimes come to the player in order to talk to them. Much like the team member screen, players can reply back to the team member by selecting from a list of dialogue options, with the dialogue selected progressing the event and possibly having an impact on the team member and, in rare cases, the whole team.



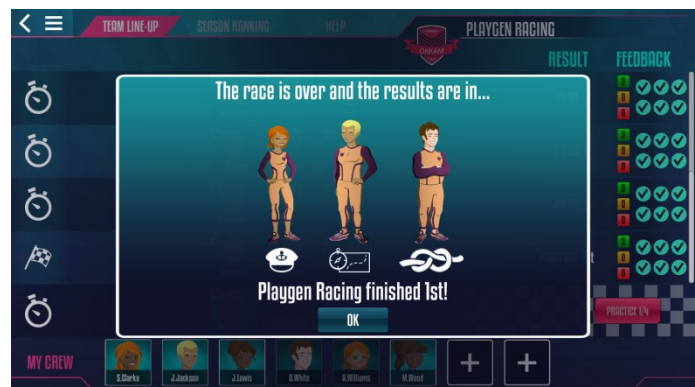
1.3.7 Learning Pill Pop-Up

After finishing a post-race event, this pop-up is displayed, giving players a 'learning pill' piece of feedback depending on what options they selected.



1.3.8 Results Screen

After a race, a results screen shows all team members who participated in the race and the position in which the team finished.



2 JOB QUEST

Job Quest is development by BIP Media for the Randstad use case. It is written in C# and uses the Unity 3D game engine.

2.1 Overview

Job Quest is a single player game simulation of a real job search experience.

For users, the aim is to improve the way in which they handle their job searches, in particular job interviews. So it will be mainly about improving the player's "presentation" skills during interviews, the way he sells himself. During the game, the player will receive different sort of advices, mainly from a Randstad Personal Advisor (RPA) NPC.

At the beginning of the game, a quick questionnaire will then determine the player's psychological profile. Then the first meeting of the player will be with a Randstad Personal Advisor who will ask him various questions to make a personal assessment, establishing his professional career and define their professional project.

Then the player receives a request for an interview. Affording different kind of recruiters who will use different interview approaches with different objectives, the player will be quickly prepared to many kind of situations during a job interviews.

During the interview, the potential employer will ask questions aloud (subtitles) and the player, in this initial version, will respond by selecting an answer among several choices. Note that this game is developed for Randstad France and the language used will be French only.

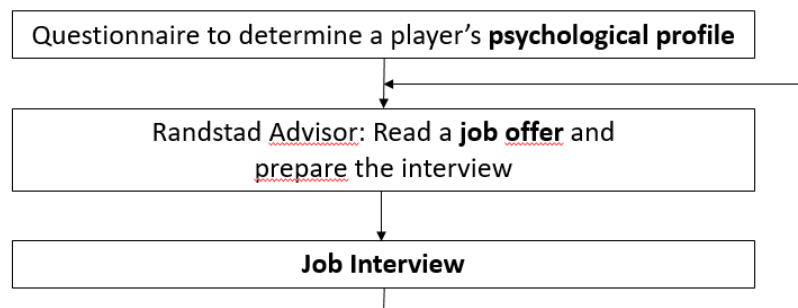


Figure 2 Job Quest Game Flow

2.1 Use of RAGE assets

2.1.1 Virtual Human Controller (T3.2)

The Virtual Character Controller is made of several functionalities but it is mainly based on using the **Behavior Markup Language (BML)**: an XML description language for controlling the verbal and nonverbal behaviour of (humanoid) embodied conversational agents (ECAs).

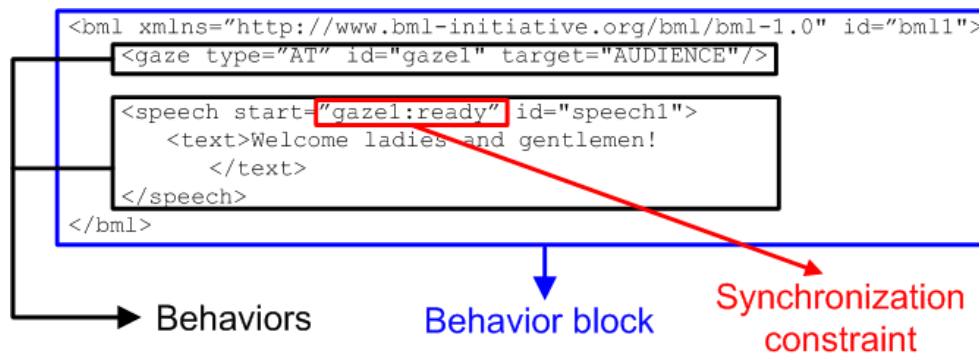


Figure 3 Use of Behavior Markup Language in Job Quest

A **BML block** describes the physical realization of behaviours (such as speech and gesture) and the synchronization constraints between these behaviours.

The synchronization (SyncPoints) allows to specify when a behaviour starts, ends or meets a peaks in regard with a global timer or with others behaviours. (e.g. "Wave your hand 2 seconds after the beginning of your speech, then look to your right and smile just after the end of the speech")

In Job Quest, we've designed our own version of the Dialog Editor (although in future versions BIP will use the Scenario Editor Asset (T3.3) that exports XML files containing BML blocks to describe a Virtual Characters' behaviour, but we've used the **BML Parser** and **Realizer** (newly designed in a modular way).

A Character Manager class instance dispatch the synchronized behaviours (resulting from SyncPoints completion) to the appropriate character. Then, a Character Interpreter instance triggers the appropriate behaviour (Speech, Gaze, Face expression or Gesture) using:

- CereProc Text-to-speech SDK to create audio files from text and to produce arrays of phonemes
- Rogo Digital LipSynch to process these phonemes into lip animations, and to express facial emotion by adjusting the blend shapes of the mesh
- The Head Look Controller and the Blink Controller from Utrecht University (another features from the Virtual Human Controller) in order to manage the direction of the gaze and eyes blinking
- Parameters from the Unity Engine standard Animator in order to trigger gesture animations

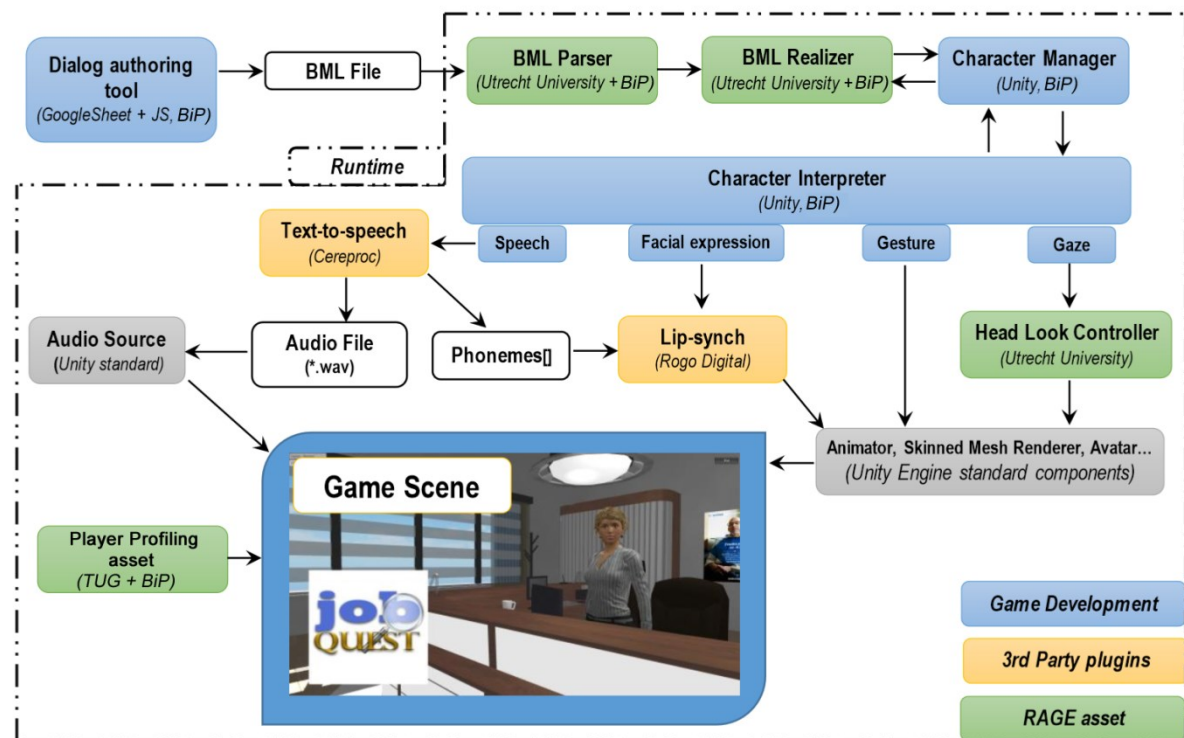


Figure 4 Assets and Components in Job Quest

The Virtual Human Controller is still a work in progress but it is already full of promise. We have used only a small portion of its possibilities: the Head Look Controller, for example, can animate a chain of bones inside a mesh, thus giving a smoother look to the animations. Several functionalities described in the BML specifications have not yet been developed in the new version of the BML Realizer, and we are looking forward what could be done in conjunction with a smart IA implementation.

2.1.2 The Player Profiling (T3.4)

In Job Quest, the **Player Profiler Xml Creator** (Authoring tool) is used in order to create and export a XML file for a 48 questions **RIASEC test** (Holland 1997).

An Evaluation Test Manager to manage a Player Profiler instance and an array of Question Handler instances. These ones are attached to the User Interface game objects, in Unity, that are displaying quiz questions unitarily. They are responsible for receiving the player inputs, displaying and formatting the text and for handling the toggle buttons.

2.2 Future RAGE asset utilisation

In the upcoming version of Job Quest we will aim to incorporate :

- **Scenario Editor (T3.3)** to replace the current *Dialog Manager*
- Evolutions in the **Virtual Character Controller (T3.2)** to get a standardized interface and to work with “standard” 3D models
- **Semantic Models and Topic Mining (T3.3)** : CV and cover letter analysis to extract keywords and to work with **Competence Assessment (T2.2)**
- **Competence-based Adaptation (T3.4)** will be used to provide personal recommendations

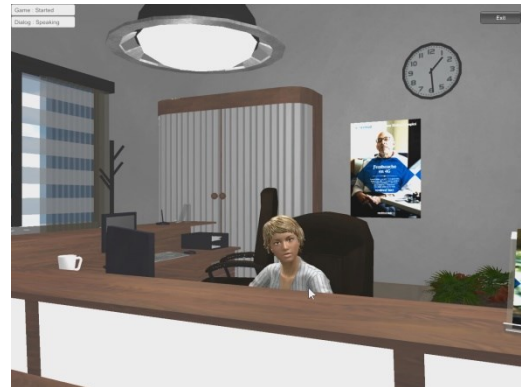
- Emotion detection assets (T2.3).
- Text-to-Speech / Speech-to-Text assets (T3.3) in French.

2.3 Game screenshots

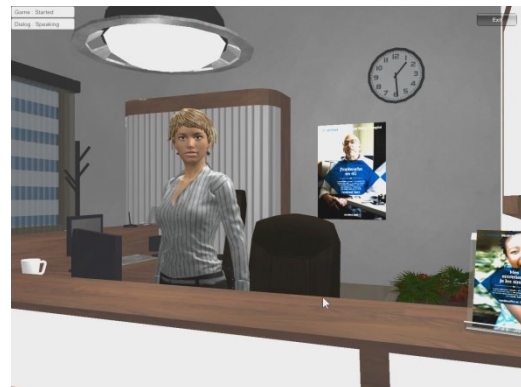
The start screen presents the player with options to either get advice about the presentation first or to go directly to the job interview sequence.



Selecting "Presentation", the player meets his Randstad Personal Advisor for the first time.



The Advisor welcomes the player (voice synthesis) and explain what she can do him. The camera is free so the player can move the mouse to observe the room.



A multiple-choice window opens on the top of the screen and the player have 4 choices:

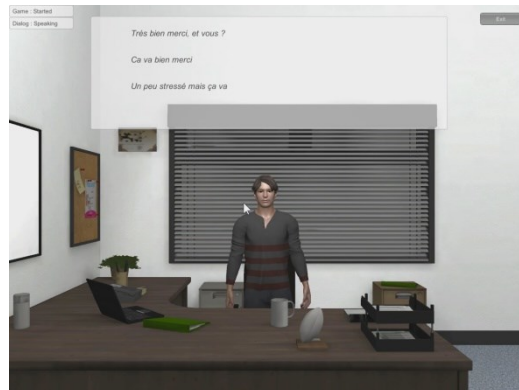
- Look at the job offer
- Pass the evaluation test
- Go to the interview



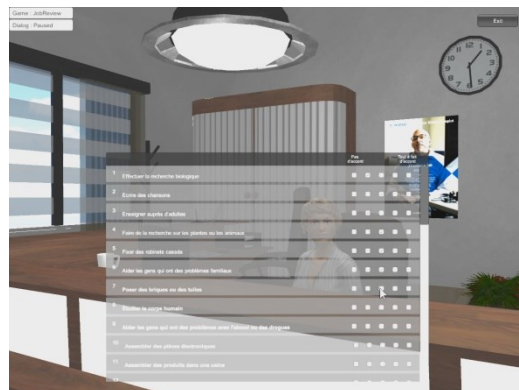
If the player chooses the first option, the job offer is displayed in a new window with all details: profile and information needed, missions, remuneration.



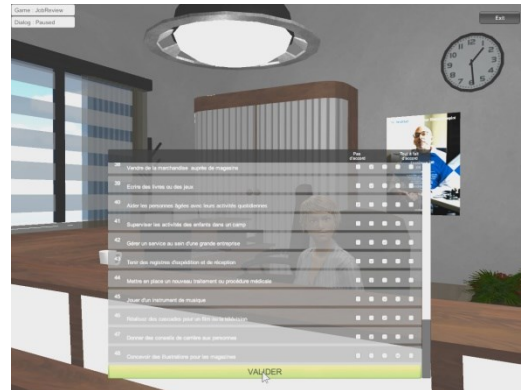
If the player chooses to pass the test, a new window opens and he'll have to give a level of importance (1 to 5) to 48 profiling questionnaire.



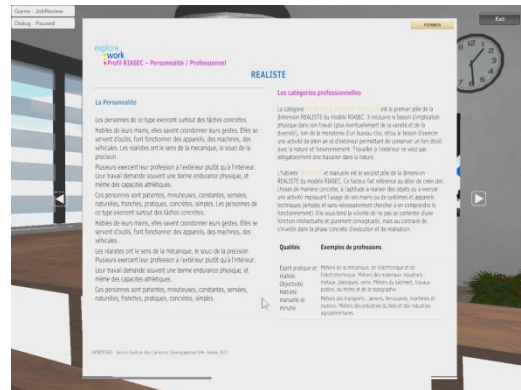
And the result is displayed: three pages describing the three principal personality traits of the player. This result is taken into account during the construction of the interview dialogue.



If the player chooses to pass the test, he will be “teleport” in a new office where the recruiter welcomes him.



The player could be prompt with some action to realize. Here for example, he has to click the left button of the mouse to sit down.



Then the interviewer asks a question aloud (TTS) and the player must respond by clicking on one of the answer proposed in the top window. Each answer is evaluated and influences the next question. The discussion usually includes 5-6 questions.



The recruiter can react emotionally to the response of the player. Here for example, he grimaces when referring to an excessive salary.



3 SPACE MODULES INC

Space Modules Inc is development by PlayGen for the Stichting Praktijkleren use case. It is written in C# and uses the Unity3D game engine.

3.1 Overview

Space Module Inc 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”. Customers get in touch about hardware and software faults they are experiencing, some maybe angry, others uncooperative or stressed, it’s up to the player to manage the situation and decide how best to respond.

Players respond to situations by engaging in conversation with the customer. The customer satisfaction score at the end of every game level will depend on how well they 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 level, and a breakdown of their responses is shown to allow for self-reflection. The game also provides feedback on responses which weren’t the best choice in the situation and how to approach these situations next time.

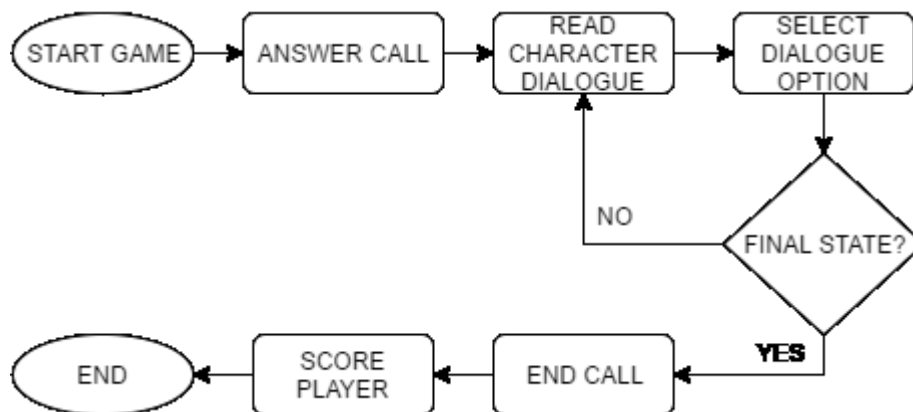


Figure 5 Space Modules Inc Game Flow

This flow diagram shows the execution of a single game level. Once the player has selected which level they would like to play, they must answer the incoming call from the NPC. The main loop of the game now initialises; the player is presented with dialogue text from the NPC and must choose one of the available dialogue options in response. This repeats until the final state of the conversation where the dialogue interaction ends the call. The player’s score is then passed to the review screen to be shown to the player once the game ends.

3.2 Use of RAGE assets

3.2.1 Role-play Virtual Character

This collection of assets is used to model the decisions and emotional reactions of the NPC. Each asset in the collection provides components to replicate a particular aspect of human emotion in virtual characters. The Emotional Appraisal asset handles player action’s as input to update the NPC’s emotional state. The Emotional Decision Making asset dictates the reaction of the NPC based on the changes to their emotional state.

In the context of this game, the visual representation of the character's emotional state (their facial expression) is managed by a controller which takes emotional state output from the Emotional Appraisal asset (i.e. "Joy", "Distress"). When there is a change in emotional state, the controller updates the character's visible facial expression to show the player the emotional aftermath of their actions.

The Emotional Decision Making asset selects the appropriate dialogue response to send back to the player. The game gets the output response from the asset and produces it for the player to review. The response produced is based on the current state of the dialogue and the emotional state of the character. These two assets working in tandem cater for meaningful interaction between player and character.

3.2.2 Integrated Authoring Tool

This tool provides a GUI for editing the assets specified under Role-play Virtual Character. It provides an easy method of creating scenarios of various characters and dialogues, editable by non-technical project participants.

The scenario file generated by the tool holds a collection of characters with their dialogue options and appraisal rules (beliefs). The dialogue options for the player are also specified. The game loads a character, identified by the 'character name' property and gets the initial state of the available dialogue. This displays to the player the initial dialogue from the NPC and initial dialogue options the player has to respond with.

3.2.3 Interaction Tracking

This asset is integrated into the game to track what the user is interacting with and the timings of those interactions. In the game, the asset logs which dialogue options the user selects when interacting with the character.

3.2.4 Storage & Analytics

The data gathered from the Interaction Tracking asset is stored and aggregated on a server running the Storage & Analytics asset. Here the educational partner can collect and analyse results at their leisure and draw educational conclusions about the game.

3.2.5 Social Gamification Components

This asset is used to track the player's achievements earned in the game. The scores that the player achieves in each level are stored on a server and used to evaluate when a player has attained an achievement (i.e. score 3 stars in all levels).

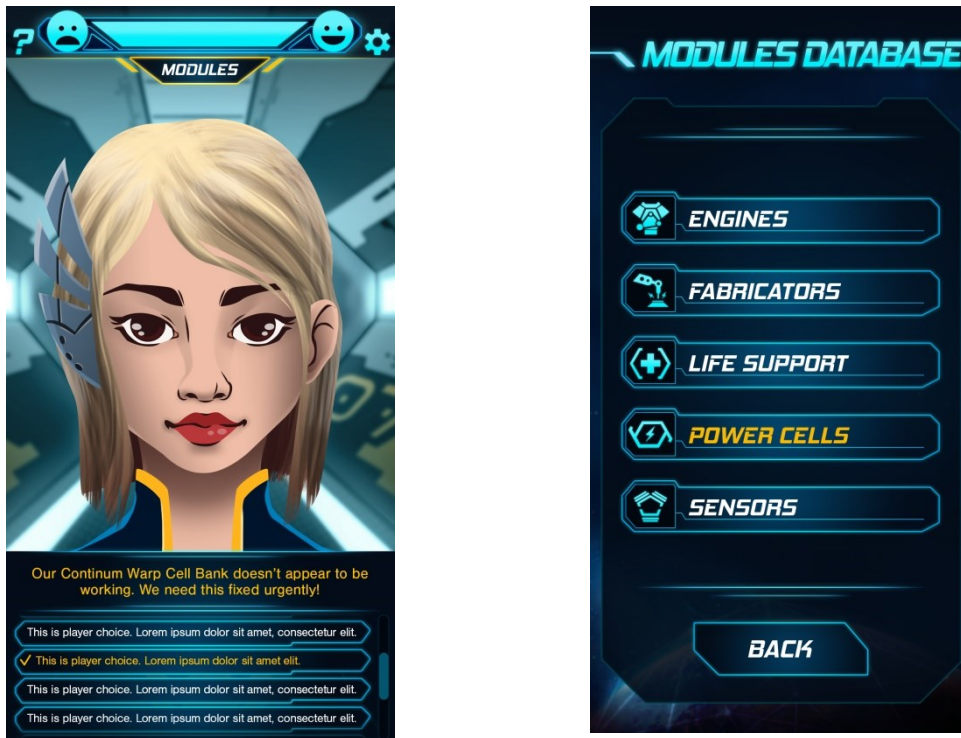
3.3 Game Screenshots

3.3.1 Pre-game Menus



The pre-game menu screenshot above show the sequence a player follow to get into a game. The level select screen (“Customer Line”) shows the highest scores of previous attempts on each level.

3.3.2 Game Interface



The main game playing screen is shown here. The player has a conversation with the character, selecting appropriate dialogue options that show a consideration for etiquette and technical understanding.

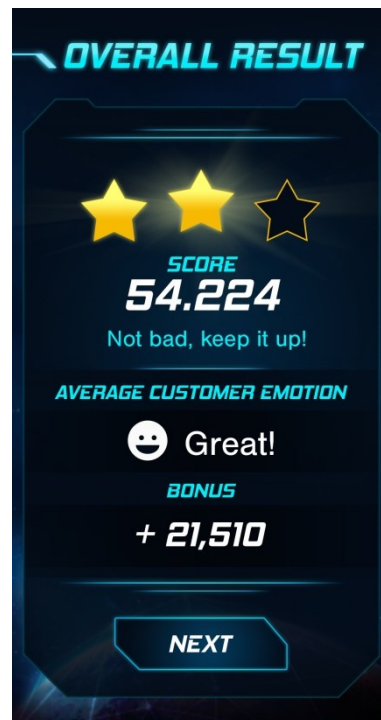
For technical problems that require a particular solution to solve, the player can check the ‘Modules Database’ for information. Each component that the customer can report as faulty falls within several categories listed in the root of the database.



The player will identify the correct malfunctioning component from a list of components in its category.

Details on how to repair the component is shown to the player when selected.

3.3.3 Feedback Screens



Review screen shows the conversation history between the player and the character. It gives feedback on individual responses made by the player at the end of each game.

This is the final view before returning to the pre-game menus. It displays the final score of the player for the level and a star rating.

4 IT ALERT!

IT Alert! is developed by PlayGen for the Stichting Praktijkleren use case. It is written in C# and uses the Unity 3D game engine.

4.1 Overview

IT Alert is a real-time multiplayer cooperative game for 2 to 8 players. The players assume the role of specialist IT service agents maintaining the integrity of a networked system architecture. The game round consists of the players working together to detect, analyse and resolve various issues and threats affecting the network and ensuring the system's continuing operation. The game teaches communication and cooperation in an IT environment.

The game features a number of predefined scenarios and may be expanded to offer procedurally generated play mode. The scripted scenarios will serve, in sequence, to introduce the players to the gamut of actions and actors they will encounter throughout the game and to instruct them on how to handle situations as they emerge.

The scripted scenarios will have success conditions that require the players to resolve predetermined events within a specified time. The game will also feature a survival mode where the players score is determined by the time-to-failure of the system and the difficulty of the problems resolved.

The game will feature voice communications between players and will be balanced in such a way that success is impossible without cooperation and communication between players. The multiplayer nature of the game requires a pre-game lobby for players to organise and converse while waiting for a certain player count or to reflect via voice and text chat on previous game sessions.

The overall objectives are to practice ability to work in a team, sharing limited resources, prioritising and delegating simultaneous actions, increasing emotional awareness of others and practicing communicating information with clarity and precision, as well as improving the ability to remain calm under pressure. Additionally the performance evaluation will aim to measure communication skills through tracking players' ability to provide clear instructions, measuring time taken to complete level compared to average and taking in post-game team feedback.

4.2 Use of RAGE assets

The initial release of the game will incorporate a small number of RAGE assets, but potential applications of a larger set have been identified.

4.2.1 Social Gamification

The SUGAR Social Gamification asset provides tracking of game domain specific player behaviour such as completing the in-built scenarios, completing a survival mode scenario or performing various in game actions. SUGAR provides an achievement tracking and notification component which will notify the player when an achievement criteria has been satisfied and will allow them to view their progress towards visible achievements.

SUGAR will eventually provide matchmaking features for the game based on the player's social graph or current skill level. Currently this functionality is implemented with a combination of game specific logic and API's provided by the networking stack that is currently being utilised.

The table below details a non-exhaustive list of social achievements that have been implemented so far:

Description	Criteria
Teamwork (Level 1)	Complete the first multiplayer scenario with a team of at least three players.
Teamwork (Level 2)	Complete {n} multiplayer scenarios with a teams comprising {x} different players.
Full Spectrum Analysis	Analyse all three primary virus strains in a multiplayer match.
Security Researcher	Analyse {x} virus specimens in multiplayer games.
Close Call	Survive a scenario in which all nodes have been infected simultaneously.

4.2.2 Client & Server-side Interaction Tracking (T2.1)

The currently implemented assets are used to track player actions and progression through the game scenarios. The game implementation contains a developer configured mapping between the events taking place in the game simulation and analytics payloads that can be transmitted to both the Social Gamification asset (SUGAR) and the Interaction tracking assets.

4.2.3 Storage & Analytics (T2.4)

Data provided by the Interaction Tracking asserts is stored and aggregated on a server running the Storage & Analytics asset. This facilitates analysis of the game data by the game developer and education provider using the analytics dashboard components and provides a consistent data storage format for consumption by the other RAGE assets discussed below.

The table below details a non-exhaustive list of analytics events that have been implemented so far:

Event	Category	Contextual Indicator
Failed item activation	Cooperation/Communication	Multiple players attempting to perform actions on same system node
Item drag target miss	UI	Player struggling to use drag-drop item behaviour
Virus spread unseen	Performance Metric (Team) Situational Awareness	Increasing occurrence indicates players are not observing or responding to system failure
Voice chat activate/deactivate	Communication	Relative total timings can be used to infer dominance or emergent leadership
Player movement complete (distance units)	Cooperation	Total movement cost relative to average provides chain efficiency metric
Player item activation	Performance Metric (Individual)	Correlation between item type and player determines role segregation strength
Infected/uninfected system ratio (periodic sample)	Performance Metric (Team)	Winning/losing trend over time

4.3 Future RAGE asset utilisation

4.3.1 Game Configuration Asset (T3.4)

The asset could be used as an alternative to manual configuration of the game scenario by players in the pre-game lobby where less previous game experience is present in the player group. This could be used in an automatic capacity to select appropriate configuration in a quick-match situation.

4.3.2 Performance Statistics (T2.2)

The performance statistics asset might be integrated into a SUGAR module to provide leader board or player ranking functionality from data captured by the analytics system.

A metric of the player's performance relative to the larger population might be used to dynamically modulate the difficulty of the game scenarios.

4.3.3 Heterogeneous Adaptation (T3.4)

This asset could be used to select suitable default parameters or scenarios for the player in a quick match or survival mode and the track and modulate the apparent level of difficulty of those scenarios or parameter sets.

4.3.4 Text-to-speech Client/Server (T3.3)

IT Alert relies heavily on verbal communication between players to facilitate cooperation and coordination of actions therefore a means of generating verbal cues for players to changing game situations might be desirable.

However this asset is only currently available with Portuguese language support and so not suitable for the Dutch/English speaking user base that will be initially involved with IT Alert. The online-only mode of operation is not a hindrance to a network based multiplayer game that already relies on internet connectivity to operate.

4.4 Game Flow

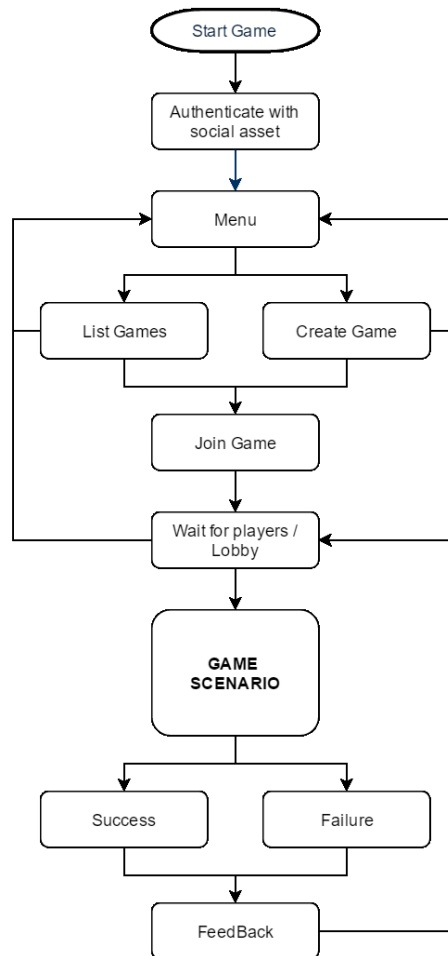


Figure 6 IT Alert! Game Flow

4.5 Game screenshots

This sections contains actual screenshots of the current game.

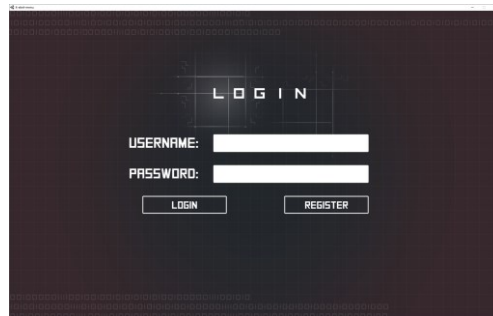
4.5.1 Game start splash screen

The splash screen is a placeholder that is shown while the game systems are initializing and has no interactive elements.



4.5.2 Social Asset Authentication

The login screen allows the player to register and authenticate themselves with the social gamification asset (SUGAR). This screen may be suppressed when using external authentication provided by the education partner.



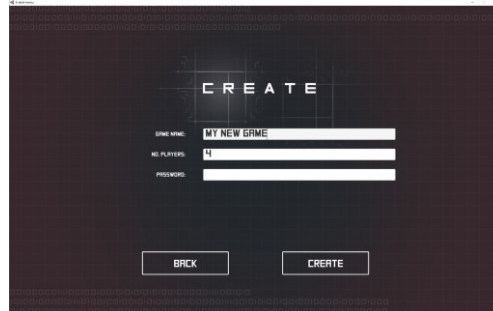
4.5.3 Main Menu

The main menu allows the player to select the game action to perform by clicking the corresponding button.



4.5.4 Create Game

This allows the player to create a new game by specifying the title, number of players and optionally a password to restrict access to specific players.



4.5.5 Join Existing Game

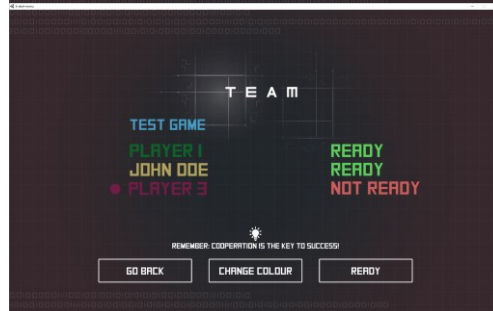
This displays the list of all currently available game lobbies that are currently waiting for player count. The player is able to join a specific game by clicking on the button in the join column.



4.5.6 Game Lobby

The game lobby shows the current players and their readiness state. Here the player is able to select their player colour from those not in use by other players.

Voice chat is enabled in the lobby so the players may communicate before the game begins.



4.5.7 Player Settings

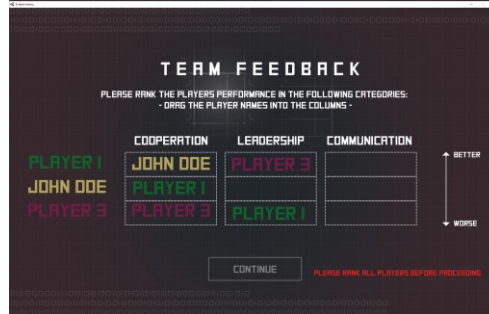
This dialogue allows the player to select the colour that will be associated with their player element them in the game scenario.



4.5.8 Team Feedback

Here the players are required to rank the other players according to three procedurally selected feedback categories.

The rankings are initially unpopulated to prevent the player from accepting a default configuration without interaction.



4.5.9 Game Settings

The game settings menu allows the player to configure audio and graphical settings for the game.

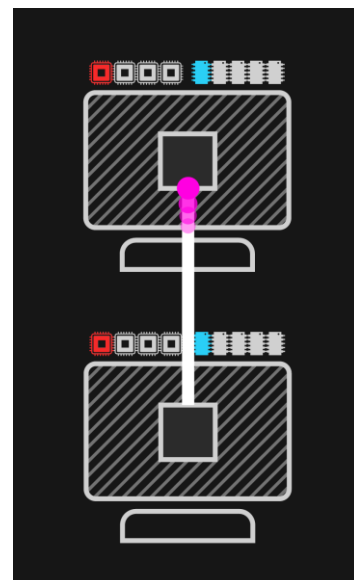


4.6 Game Scenarios

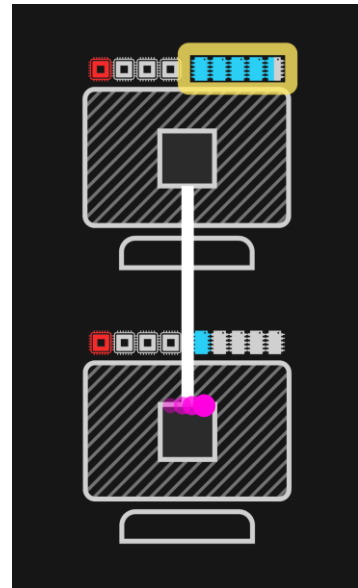
4.6.1 Scenario 1: The basics

The first scenario introduces the player to the virus NPC and demonstrates the tools and actions needed to record, analyse and then resolve an incident on a small system architecture.

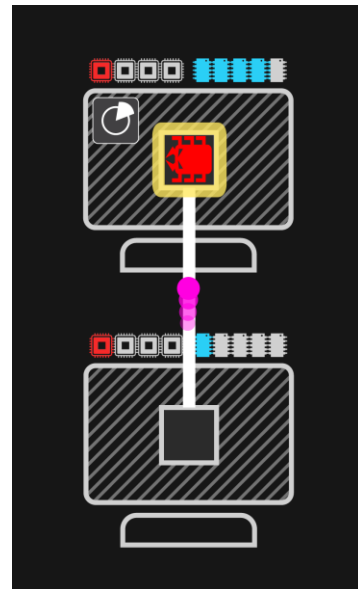
First the player is presented with a simple system comprising only two nodes. The player is spawned on one node and the tutorial introduces the player to basic movement.



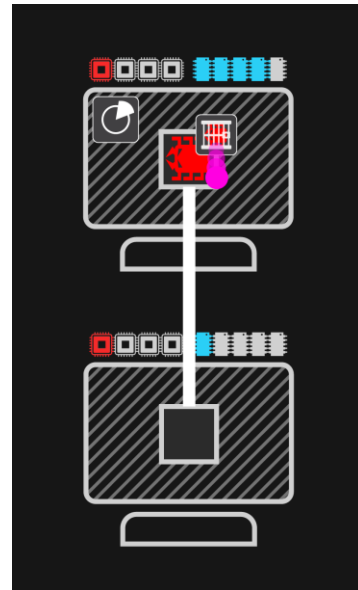
A virus is spawned on one of the nodes still invisible to the player. The tutorial draws the player's attention to the increased memory usage on the system node indicating the observable symptom of the infection.



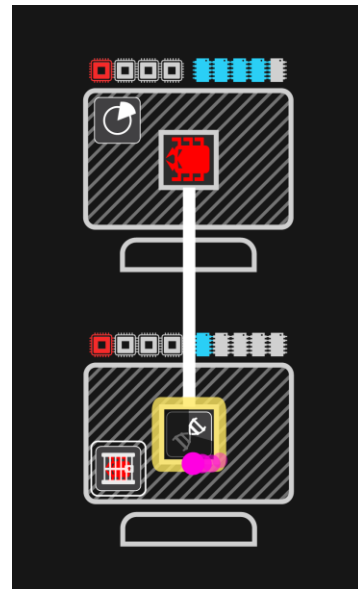
The player is introduced to the first item, the scanner, and instructed how to use this to detect and reveal (log) the system problem by dragging the tool from the storage location on the node into the centre activation location. After the tool activation timer has elapsed the virus is revealed to the player.



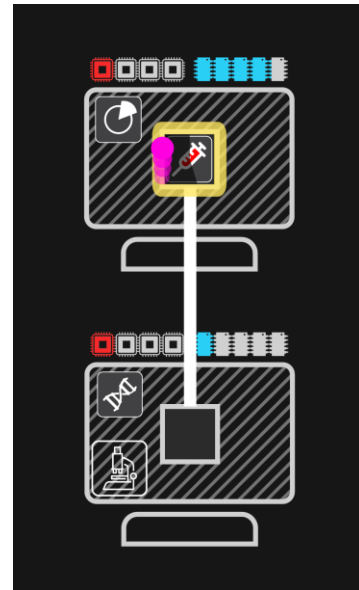
The player is introduced to the second tool, the capture tool, which is spawned on the second system node and the player is instructed to navigate to that node and collect the tool and return it to the infected node to be activated. Once activated the player gathers a sample of the virus which is represented by the virus sample item which is now following the player's motion.



The player is instructed to carry the virus sample to the second node which is now revealed to be an analysis workstation. Upon depositing the sample in the activation area the player is introduced to the third tool, the analyser which when activated produces an antivirus tool in the colour corresponding to the captured virus.



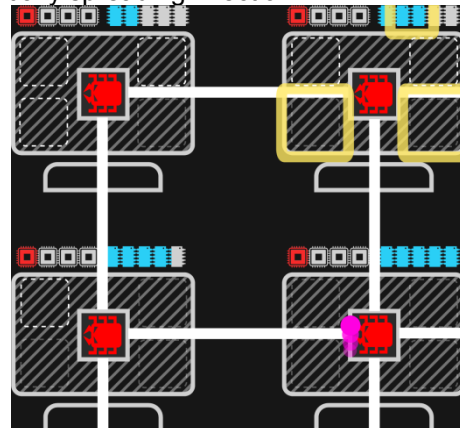
The player is now instructed to carry the antivirus tool to the infected system node and to activate it there to eliminate the infection. This ends the first scenario and the player is returned to the tutorial menu.



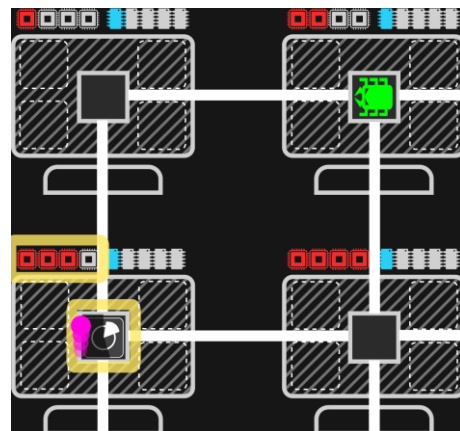
4.6.2 Scenario 2 : Virus classification, spreading behaviour, and failure.

The second scenario introduces the player to the different virus species and demonstrates their respective effects on the system. The player is also introduced to the self-replicating behaviour of the viruses and experiences failure due to the exponentially spreading infection.

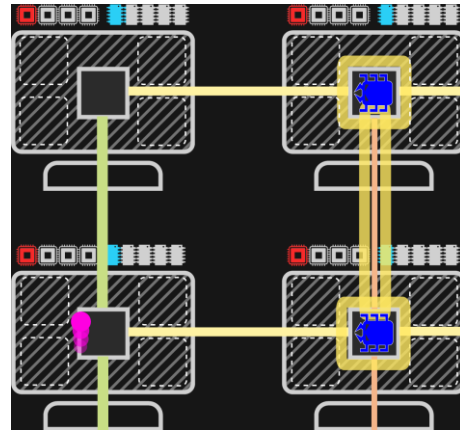
The player is presented with a system comprising nine nodes in a 3x3 arrangement. Viruses are spawned on four of the nodes and the player is instructed to observe the symptoms, in this case memory consumption. It is explained that as more memory is consumed the ability of each nodes to store items is reduced proportionally.



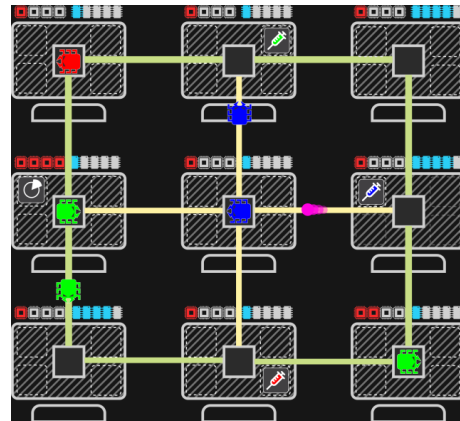
The previous viruses are removed and new viruses spawned invisible to the player. The player is instructed to move between the infected systems and use the scanner tool on each, observing how the level of CPU consumption is inversely proportional to the activation speed of the tool. As the tool is used the player reveals that the viruses are of a different strain to those seen previously affecting memory.



The previous viruses are removed and new viruses spawned invisible to the player. The player is again instructed to move between tools using the scanner, observing how the network connections are being affected by the infected nodes, slowing movement proportional to the number of adjacent infected systems.



Lastly a number of viruses and antivirus tools are spawned and the viruses begin spreading around the network. It is thus demonstrated to the player how rapidly the viruses can take control if left unchecked.



Finally the player is shown a failure condition once the viruses have taken hold of all system nodes.



4.6.3 Scenario 3 : Teamwork, scoring and cooperation.

The third scenario introduces the player to multiplayer interaction on a larger map, the survival timer is introduced and the malware actors are introduced on opposing sides of the map requiring the team members to prioritise movement and transfer items between players to survive. The scenario introduces the post-game feedback screen to the players.

5 WATERCOOLER GAME

Watercooler game is developed by Nurogames for the Hull College use case. It is written in C++ and uses the Cocos2D game engine.

5.1 Overview

In the “Water Cooler Game” the player is hired by a small game studio to develop the team working in the studio through his interpersonal relationships. As an office assistant his goal is to increase the success of the company primary by improving the social skills of the virtual team. Therefore he can visit the water cooler to start conversations that prompt group working/conflict management behaviour/value based questions and multiple-choice answers for him to choose from. Besides that he has to handle the workflow by interacting with the staff members and organizing their 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 analysed.

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.).

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.



Figure 7 Watercooler Game Logo

5.2 Use of RAGE assets

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)¹. 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 reports² and a balancing tool³ will be used. This will also require a client side interaction tracking⁴ and server side interaction storage⁵ 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 storage⁶ must be handled as well. This asset 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 asset⁷ will be used to create and prepare the conversation between the user and the NPCs during the game development. A server side dialog player⁸ must send these texts to the game client dynamically depending on several parameters defined in the editor and the current situation in the game.

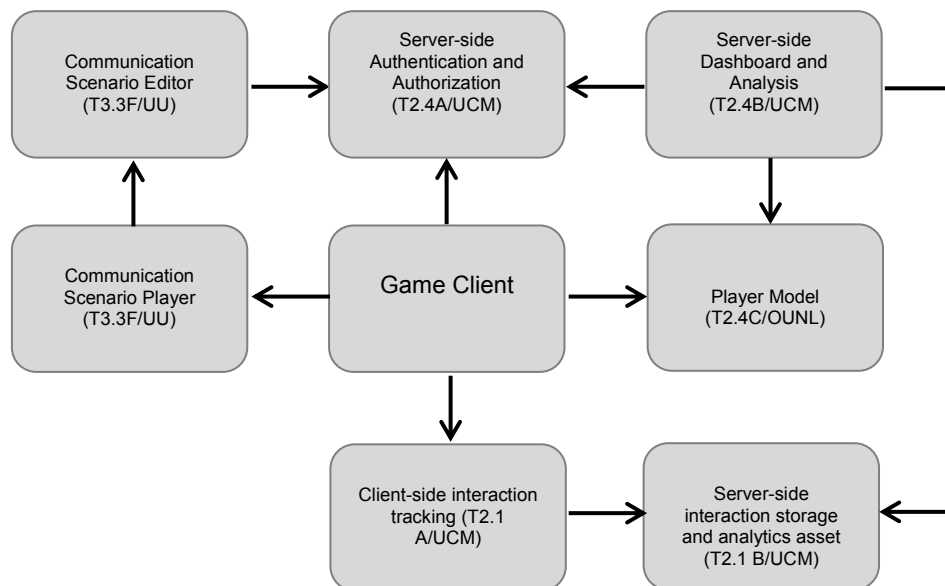


Figure 8 Overview Assets and interfaces for Watercooler game

¹ Server-side Authentication and Authorization (T2.4)

² Server-side Dashboard and Analysis (T2.4)

³ Performance Statistics (T2.2)

⁴ Client-side interaction tracking (T2.1)

⁵ Server-side interaction storage and analytics asset (T2.1)

⁶ Player Model (T2.4)

⁷ Communication Scenario Editor and Player (T3.3)

⁸ Communication Scenario Editor and Player (T3.3)

5.3 Game screenshots

5.3.1 Login Screen

After all client resources have been loaded the Login Screen appears. Here the user has different options to choose:

- 1) Sign up
- 2) Play as Guest
- 3) Start with Username and Password

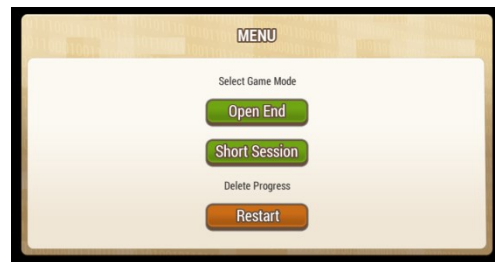
Before playing, the user has to register for the game (i.e. with a Username and Password). The account creation and authentication process will happen on a website also to allow a tutor to set up and invite new students and for the report functionality. To allow the user to play on different clients and not only on one pc the user data must be stored on a server and be synchronized.



5.3.2 Game Mode Menu

After the Login the Game Mode Selection Menu appears. Here the user has different options to choose.

- 1) The game will be able to be played indefinitely with no fixed final win scenario.
- 2) Alternatively a gameplay session is possible.
- 3) An option to reset and restart the game at any time will be implemented.



5.3.3 Studio and Avatar

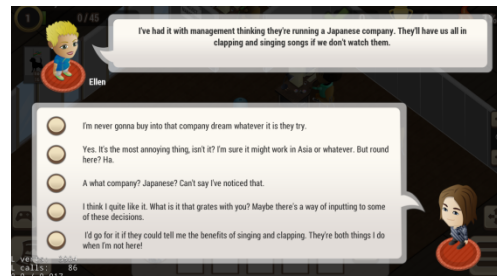
The player is represented as an avatar that moves through the game world controlled by the user. The avatar moves automatically to the target spot the player has clicked before.

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



5.3.4 Water Cooler

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 behavior/value based questions and multiple-choice answers for the player to choose that cover a range of attitudes, subjective to objective and in between.



5.3.5 The results

The results of the 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 and 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.



6 HATCH – THE CREATIVE ENTREPRENEUR

The HATCH game is developed by Gameware Europe for a Hull College use case. The game is written in C# and is developed in Microsoft Visual Studio and is a standard .NET windows forms application.

6.1 Overview

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. Further information on target audience can be found in D5.1.

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.

6.2 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 will include:

- Pricing your Product
- Financing a Start-up
- Cashflow and Financial Planning
- SWOT analysis and self-assessment
- Branding and Market Position
- Production and product management issues
- Basic legal requirements
- Audience identification
- What level of investment is required to grow the business

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 an effective approach 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.

6.3 Game Flow and Game Modules

A full description of the game design can be found within the document D4.2 (Full Game Design) and has not changed in any substantial way since that time. In summary, at the outset of the game, the user is presented a choice to 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.

The decisions that players make during the game are 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.

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

Self-Assessment & Self-Development

Player picks from a list of personal and business attributes in order to provide an assessment of their own strengths and weaknesses *and* the perceived opportunities and threats to their new business.

Assessment is based on relevance of their choices to their chosen business.

Legal Issues in business

Player has to demonstrate a basic understanding of legal terms in connection with business dealings including tax affairs, minimum wage and employment law, incorporating their business and so on. Final design of this module is undecided at this point

Financial Issues

Player will need to understand and use a number of financial issues in order to come up with a viable business plan and attractive growth prospects. Their choices feed a financial model which drives most of the game engine and will have the most impact on their final assessment.

At a minimum, the financial model inputs to and outputs from the financial model include cost setting (e.g. COG, day rate, commission rate, royalty rate etc.), overheads, staff number and cost, gross and net margin, marketing budget, sales projections and growth forecasting, gross and net profit, cash flow outcomes and investment requirement.

Project Management

Player must allocate resources to production tasks over variable timescales and use logic to arrive at the most efficient means of doing so (meeting time and/or budgetary constraints)

Manufacturing

Here, the player must think through and decide upon the trade-off between providing Quality, keeping costs down, and time taken to produce their outputs. Time has a bearing on the project management module; costs effect outcomes in the Finance module; and Quality will influence brand value and sales.

Audience and Market

This module enables the player to choose the target demographics for the product or services. Choices include age group, geographical area, gender and socio-economic group. Decisions made here directly affect the appropriateness of their choices in the Marketing Channel module and indirectly affect sales figures

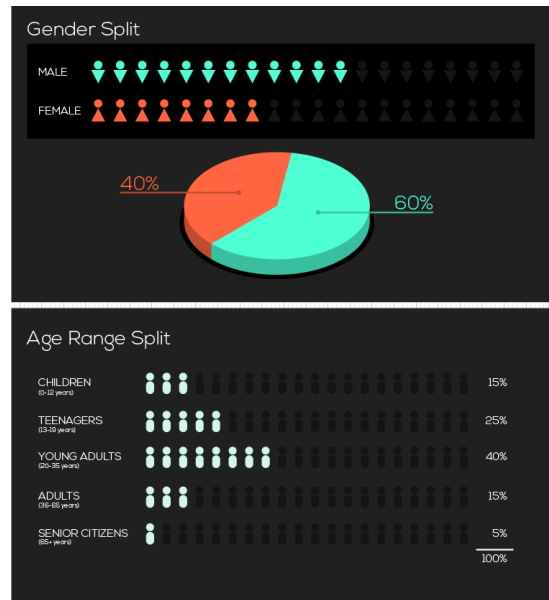


Figure 9 HATCH Game Market Demographic

Marketing Channel

The player chooses through what advertising/marketing channels they will spend their marketing budget. Players are assessed against how closely their choices bear against real-world proportions according to the business they are operating and their chosen target demographics.

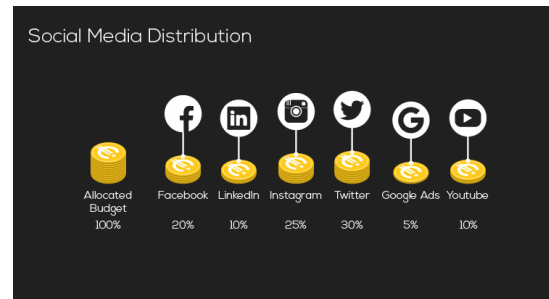


Figure 10 HATCH game Social Media Channels

Common features of all modules are as follows:

- Ability to pull on Meta data. (Derived from the setup phase at game start when the business-type is selected)
- To be sensitive to output from other complete modules (where applicable)
- To allow output to be read by other modules
- To produce an interim report. This should collate the information from the current module and any other completed stages. This step is designed to offer nurturing feedback and not assess.
- To provide relevant data when the player moves to the final assessment phase during which the player's choices are analysed. The final outcome will provide a success rating and/or guidance on where further work and/or modifications to their proposal might be required.

6.4 The Game Model (Hatch Engine)

The game model will tie together the input(s) and output(s) 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 stage

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

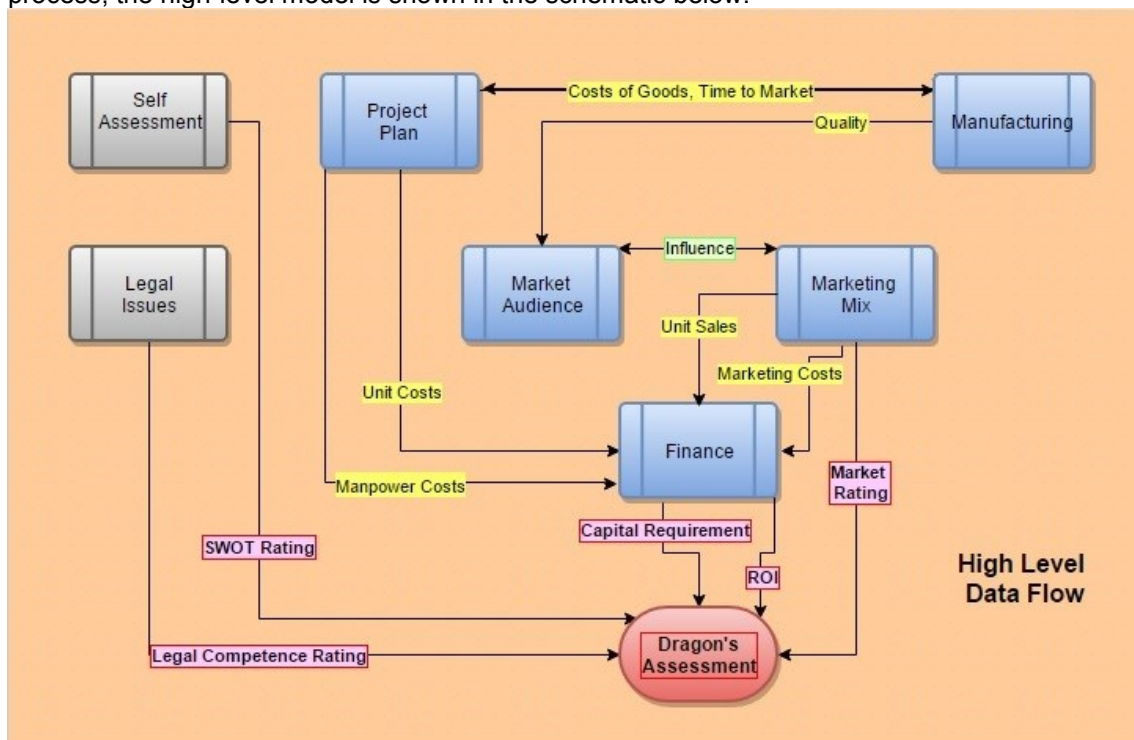


Figure 11 HATCH Engine Game Flow

In the schematic above :

- Game Modules 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), all shown in Pink. These ratings will be subject to weighting values during development and test to arrive at a fair scoring mechanism.

Table 1 HATCH Game Scoring Table

Module	Potential Max Rating (Score)
Self-Assessment / SWOT score	100
Legal Issues	100
Project Planning	150
Manufacturing	100
Market Audience	100
Marketing Mix	150
Finance	300 (including ROI rating)
Assessment Total Max	1000

6.5 Game Development and Rage Asset Utilisation

The game is written in C# and is developed in Microsoft Visual Studio and is a standard .NET windows forms application.

The Gameware game engine allows certain components to be run in parallel or selected via admin functions. This allows us to compare the behaviour and/or usefulness of the application using either RAGE components, or our proprietary solutions, or both, as needed. This approach is also useful for integrating any RAGE components which are not yet complete, as these can be bypassed for the time being.

HATCH utilises RAGE assets that enable peer/tutor input, personalise learning, social gamification and interactive story telling. The server side RAGE asset building blocks that are being utilised will enable real time client side interaction tracking, server side storage and analytics and non-obtrusive competence assessment, server side authentication and authorisation, server side dashboard and analysis and evaluation. The use of the RAGE assets therefore enable data tracking, analytics and intervention as appropriate in a learning environment.

6.5.1 Client side interaction tracking. (T2.1)

Client side interaction tracking is being used to store user selected options. We have our own tracking logging input events and timings thereof. At present, this is purely for monitoring purposes.

6.5.2 Server side storage and analytics (T2.1)

Server side storage will be used to store game/settings for a player. Server side analytics will be used to show user input patterns, play history etc.

6.5.3 Client side non-obtrusive competence assessment (T2.2)

Competence assessment will be used prior to the start of each mini-game to optionally auto-select a key difficulty level for the user. The difficulty level will adjust parameters in certain game module, such as increasing complexity of the project management task or requiring a higher percentage Return on Investment at the game's final assessment stage.

6.5.4 Server side authentication and authorisation (T2.4)

Server side authentication will be used for security setting as appropriate.

6.5.5 Server side dashboard and analysis (T2.4)

Server side dashboard will be used later to show analytics.

6.5.6 Heterogeneous Adaptation – HAT (T3.4)

The HAT module may be used prior to start of each certain modules to determine appropriate difficulty levels.

6.5.7 Social gamification components – SUGAR (T3.6)

SUGAR will be added towards the end of the project to allow sharing of results and other useful features – player achievements and rewards as well as player grouping.

7 ISPO – INTERVIEW SKILLS FOR POLICE OFFICERS

The ISPO simulation game is developed by Gameware Europe for the Policia Judicaria use case.

7.1 Overview

ISPO is a simulation of a police interview procedure intended to provide a training aid to police officers who need to question:

- 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 or inadequate or can be improved as well as to establish their level of understanding of the core competencies which they are required to demonstrate.

ISPO will provide several means by which a police officer's interviewing technique and skill can be measured and scored, including their ability to demonstrate; Emotional Control, Resilience and resistance to stress, Communication skills, Empathy, Conflict management capacity, Ability to analyse verbal and non-verbal communication and Information Gathering and comparison.

7.2 Simulation Components and Flow

The ISPO system provides 3 core software modules elements for scripting, recording and running interview simulations.

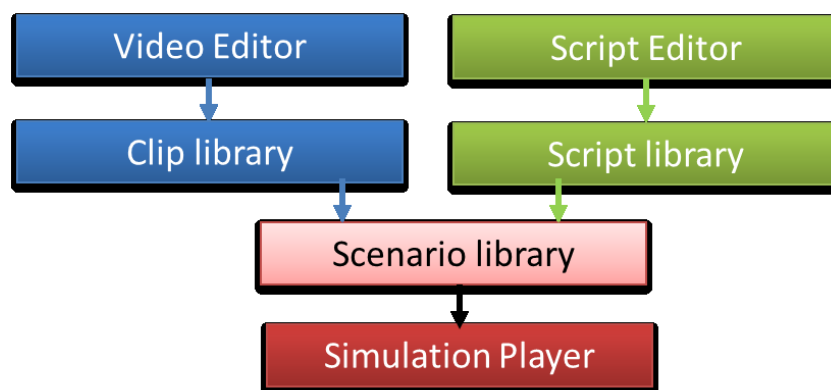


Figure 12 Main components of ISPO

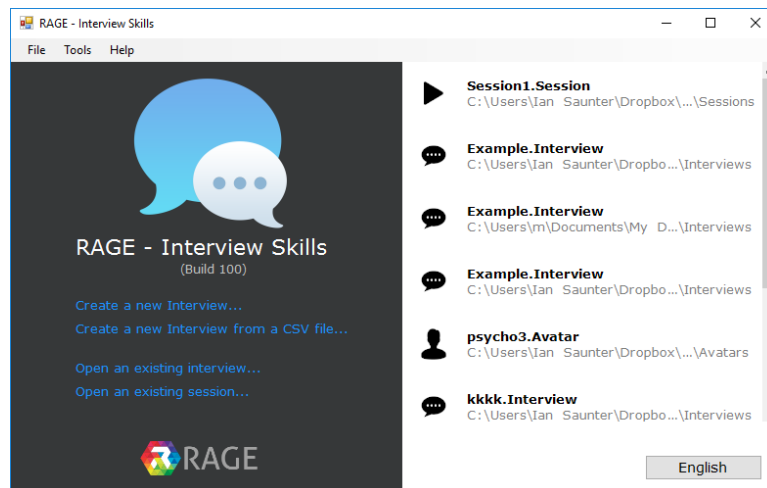


Figure 13 ISPO Overview Loading Screen

7.2.1 Script Authoring

A Script authoring tool is provided which enables trainers, administrators and moderators to create interview scenarios. This can either be built from scratch or to build on existing scenarios to alter difficulty or realism.

Potential questions can be set along with configurable parameters denoting, for example, the priority assigned to that question in a real situation, or the aggressive nature of asking that question (the appropriateness of which may be denoted by its timing within the interview).

A set of potential answers to each question can then be scripted, again associated with parameters which might include the informational value of the answer (i.e. trivial or important); the truthfulness of the answer; or the tone (emotion) which is ascribed to the answer. Scripts can be entered directly through the editing tool or imported, in .csv format, from spreadsheets.

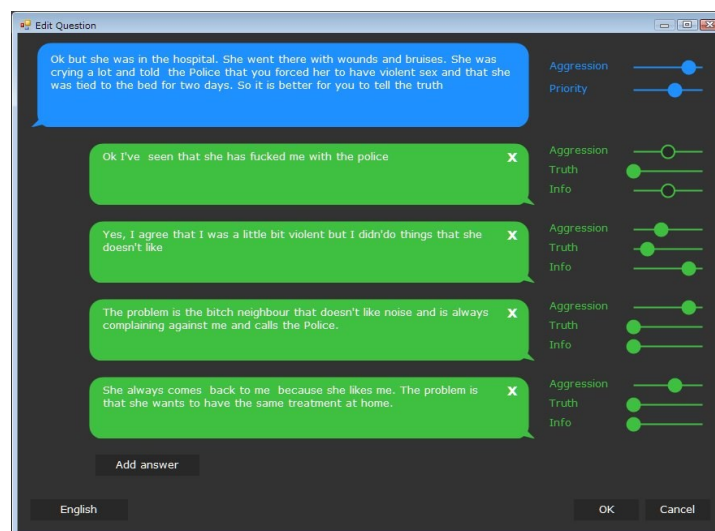


Figure 14 ISPO Questions Script Editing Tool

7.2.2 Video capture and editing

A Video-capture and editing tool will be provided. This enables trainers (et al) to film and edit sequences of actors into short clips which correspond to the interviewees mood, reactions, and responses to questions.

Each clip can be assigned values to a range of configurable parameters which relate to the emotional state (of the interviewee) when the clip is displayed. This different video clips may be displayed according to the state of the interviewee depending on the nature of the last question asked *and in turn* their response to the question.

Video clips produced by the editor can be used in multiple interview scenarios – there should be no need to re-shoot additional content once a sufficiently rich set of clips are available. Therefore, it will be possible to create many interview scenarios featuring the same ‘actor’, sometimes performing as suspect, witness or victim.

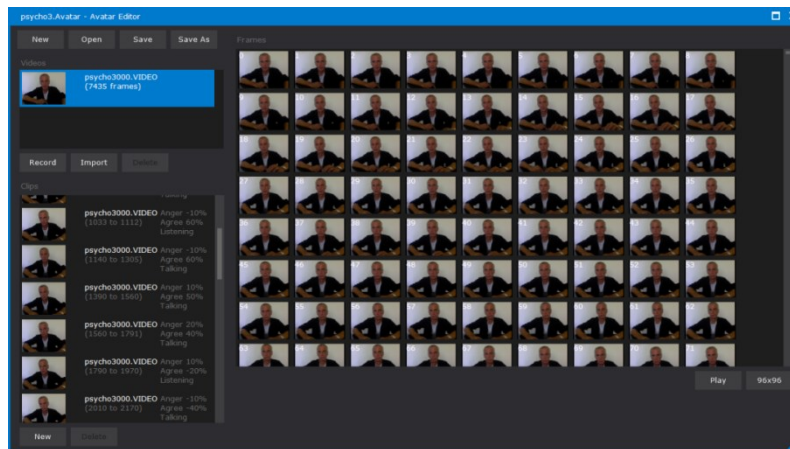


Figure 15 ISPO Video Capture and Editing

7.2.3 Simulation player

A run-time simulation which combines Scripts with Video clips in order to play out any interview scenario produced using the simulation tools.

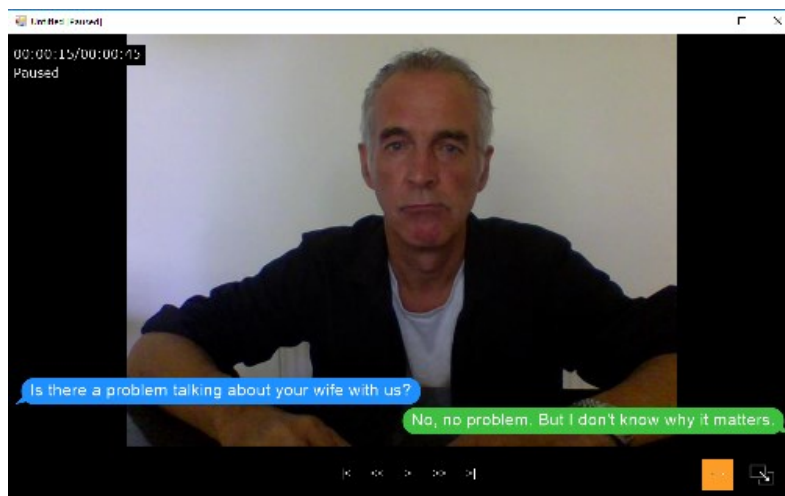


Figure 16 ISPO run-time screenshot (windowed mode)

7.3 The Interview Model

Interviews comprise a series of multiple choice questions and/or selected actions followed by a response. Questions carry a meaning stored in metadata associated with each one. Metadata affects one or more emotional parameter modelled in the interviewee's state.

Responses are chosen from a list of possible answers to each question, dependent on the current state. Metadata is also passed into the model with each response. Visual cues in the form of the interviewee's body language provide further clues to the user (non-verbal communication).

The user's objective is to achieve a favourable state change to achieve more reliable, accurate or truthful responses from the interviewee and to complete the interview process with a complete picture of Events.

7.4 Rage Asset Utilisation

The ISPO simulation is written in C# and is developed in Microsoft Visual Studio. It is a standard .NET windows forms application.

For the application, we have developed our own engine, with some proprietary components such as storage, UI tracking, behavioural analysis, TTS, speech to text etc. The engine allows certain components to be run in parallel or selected via admin functions. This approach allows us to compare the behaviour and/or usefulness of the application using either RAGE components, or proprietary solutions, or both - as needed.

ISPO will utilise a number of RAGE technology assets including, but not necessarily limited to:

7.4.1 Client-side interaction tracking, reporting and analysis (T2.1, T2.3, T8.2)

Client side interaction tracking is being used to store user-selected options. We have implemented a tracking and logging of input events. At present this is purely for monitoring. Server side storage will be used to store game/settings for a player. Server side analytics will be used to show user input patterns, play history etc.

7.4.2 User-login and authentication systems (T2.4)

Server side authentication will be used later for security as appropriate. Server side dashboard will be used later to show analytics.

7.4.3 Heterogeneous Adaptation – HAT (T3.4)

The HAT module may be used prior to start of each certain modules to determine appropriate difficulty levels. HAT module will be used prior to start of each interview to determine difficulty levels.

The difficulty level will adjust what questions are displayed at each stage of the interview. on easier difficulty levels, the application displays fewer, more relevant questions to that point of the interview.

On harder difficulty levels, little or no question filtering is done by the application itself, and the user must determine the relevance, usefulness or side effects of asking a particular question to the interviewee at that point in time.

Similarly on easier difficulty levels, a 2nd interviewer (AI) will intervene and ask a relevant question, assisting the user and steering the interview towards its end goal of getting the interviewee into a suitable state of mind, and retrieving the most relevant information from them.

7.4.4 Cognitive intervention (T3.4)

Cognitive intervention will be used during interview to assist the in-application AI. The behavioural aspect of the Role-Play virtual character will be used to determine behavioural adjustments during the course of an interview.

This is intended to be purely for comparative purposes with the custom built AI for the application, which adjusts the behaviour as per Policia Judicaria's instructions directly based on timings/appropriateness/aggression and relevance of subsequent questions asked, as well as behavioural starting conditions.

7.4.5 Speech recognition and I/O (T3.3)

Text-to-speech (TTS) will be available (optionally - from the UI) to speak either questions or answers in the application.

The RAGE TTS Asset supports Portuguese only. We have also added Microsoft SAPI TTS support, which supports any language the installed voices cater for. The RAGE Speech to text Asset is English only.

Speech to text will be optionally useable to allow the interviewer to speak into the microphone to ask a question. The application then determines the question the interviewer is asking.

We have also added our own Speech to text support, as an alternative. This works based on the expected phrase length and frequency patterns and is language independent. In effect, this is a multiple choice choosing assistant.

Within the simulation flow, we can visualise these assets fitting into the user experience as shown in the figure below:

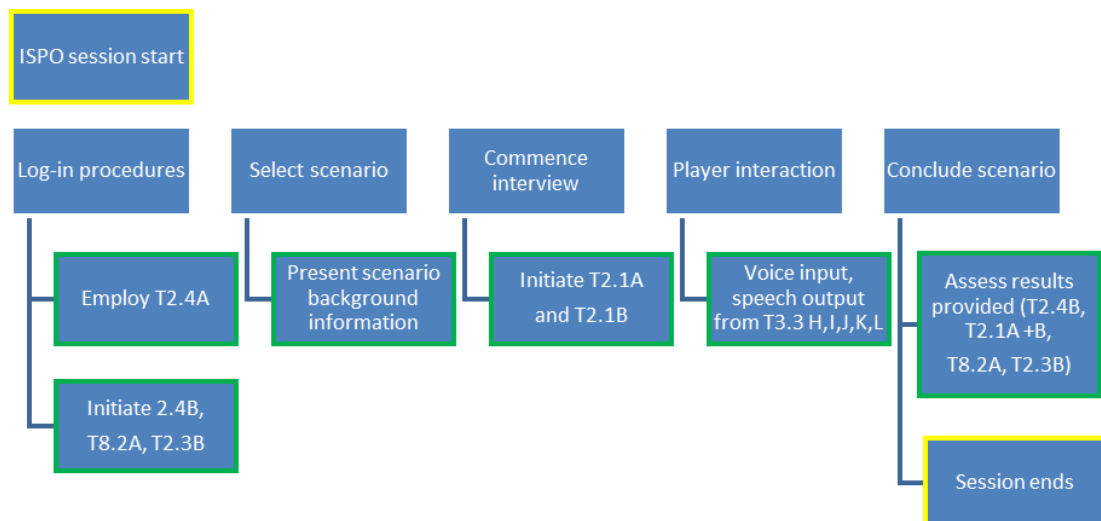


Figure 17 Use of RAGE assets in ISPO

8 CONCLUSIONS

This interim report provides a current overview of the initial versions of the games made using the RAGE assets for the first round of pilot testing and formative evaluation. The games will be piloted according to the detailed description found in D5.1 and other deliverables of the project.

The final version of the games taking into account the formative feedback from this initial version is due in M38 as D4.4, the final version of game applications for scenario pilots.

9 REFERENCES

Holland, J. L. (1997). *Making vocational choices: A theory of vocational personalities and work environments*. Psychological Assessment Resources.