



Grant Agreement No. 687676

Innovation Action

ICT-20-2015

D3.5 Game Design Document

Due date	Month 12
Actual date	Month 12
Deliverable author(s)	Laurent Auneau
Partner(s)	Succubus Interactive
Version	1.0
Status	Draft
Dissemination level	Public

Project Coordinator

Coventry University

Sylvester Arnab

Priory Street, Coventry CV1 5FB, UK

E-mail: s.arnab@coventry.ac.uk

Project website: <http://www.beaconing.eu>



Version control				
Version	Date	Author	Institution	Change and where applicable reason for change
0.1	12/12/16	Laurent Auneau	Succubus	
0.4	13/12/16	Laurent Auneau	Succubus	Merged Geomotion's review
0.5	22/12/16	Laurent Auneau	Succubus	Merged Imaginary's review
1.0	30/12/16	Laurent Auneau	Succubus	Merged Sebit's review
1.1	31/12/16	Jayne Beaufoy	COVUNI	Language check and Abrev. table

Quality control				
QA Version	Date	QA Responsible	Institution	Change and where applicable reason for change
0.5_Sebit	23/12/2016	A.M. Turker	SEBIT	See track changes
1.1	31/12/2016	Ioana Stefan, Antoniu Stefan	ATS	See track changes

Release approval				
Version	Date	Name	Institution	Role
1.1	31/12/2016	Jannicke Baalsrud Hauge	BIBA	QM

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

TABLE OF CONTENTS

EXECUTIVE SUMMARY.....	5
1 INTRODUCTION.....	6
1.1 BACKGROUND	6
1.2 ROLE OF THIS DELIVERABLE IN THE PROJECT	6
1.3 STRUCTURE OF THE DOCUMENT	6
2 RUNTIME COMPONENTS	7
2.1 GAME ENGINE	7
2.1.1 3D isometric view	7
2.1.2 Close-up dialogs.....	8
2.1.3 Inventory.....	8
2.2 MINI-GAMES	9
2.3 CONTEXT AWARE CHALLENGES	11
3 CREATION PIPELINE.....	13
3.1 THE VAULT.....	13
3.2 GAME PLOTS.....	14
3.2.1 Plot example – Liberate Ocean Springs	14
3.2.2 Plot editor	15
3.3 MINI-GAMES.....	16
4 CUSTOMIZATION	18
4.1 APPROACH.....	18
4.2 EXAMPLE.....	19
5 CONCLUSION	21
APPENDIX: DEFINITIONS OF TERMS.....	22

LIST OF FIGURES

<i>Figure 1: In game mock-up</i>	<i>7</i>
<i>Figure 2: Close-up dialog</i>	<i>8</i>
<i>Figure 3: Word-o-copter mock-up.....</i>	<i>9</i>
<i>Figure 4: Type of location-based challenges</i>	<i>11</i>
<i>Figure 5: Generic Scenario Editor</i>	<i>15</i>
<i>Figure 6: High level logical diagram</i>	<i>17</i>
<i>Figure 7: Content creation pipeline</i>	<i>18</i>

LIST OF ABBRIVATIONS

Abrv.	Description
API	Application Programming Interface
FTUE	First Time User Experience
NPC	Non-Player Characters
POI	Point of Interest
STEM	Science, Technology, Engineering and Mathematics

EXECUTIVE SUMMARY

The BEACONING platform is a learning environment supporting interaction among its different users, made of multiple layers, where multiple users can log in:

- teachers to customize scenarios, give access to them, and exploit the analytics reports about competencies developed;
- students to pervasively play these scenarios, with procedural adaptation support for disabilities;
- and authors, such as learning designers and game designers, to create new scenarios.

The present document does not take into consideration the gamification, social and analytics layers of the platform, but rather focuses on the "game" part, more precisely how scenarios are created, and what players can expect. Therefore, this document describes what happens when a student launches a scenario, and how the authoring pipeline allows a sequence of multiple different authors to collaborate sequentially and create such a scenario.

Players

Students can log into the platform, access their list of available scenarios, and launch a selected one to try it for the first time or to continue it.

These scenarios are made of a succession of "point and click" game scenes, dialogs with non-player characters, and activities such as quiz like mini-games and context-aware challenges. Learning can happen at any level and any moment in this experience, as, from the player's point of view, the learning content is intertwined with the gaming content in all different phases of the scenario.

Authors

As a simple example, the creation process of a new lesson-path could be done this way:

1. A learning designer logs into (the authoring part of) the platform;
2. They create a new lesson-path entry;
3. Using the authoring tool, they define what kind of educational content will be delivered in their course by creating their own lesson-path;
4. They browse a gallery and choose a ready-made game plot backbone that they consider fit for purpose;
5. They pick from another gallery multiple mini-games that will help to make their lesson-path more engaging;
6. They customize the content of the plot and of these mini-games;
7. They save their work on the platform, and let all the teachers access and use their work for their students.

To be able to provide such an authoring flow, the BEACONING platform relies on a few key concepts:

- "The vault" is a back-office repository containing all the narrative elements. A specific tool is able to access and combine these elements, so that game designer can use them to create an engaging gaming experience;
- Using this tool (the Plot Editor), along with a defined API for mini-games and a dedicated authoring tool for lesson-paths, all types of authors around the world will be able to contribute to the platform;
- Finally, a last customization step ensures that the experience given to the player answers to the expectations of the teacher, without losing the previous work done along the authoring chain (game scenario author, lesson-path author).

1 INTRODUCTION

1.1 BACKGROUND

The BEACONING platform will present a game experience to its end users, and for this the consortium bases its approach on existing capacities from its partners. On the gaming side, there are the already existing technologies that are going to be adapted:

- the 3D isometric adventure game engine from Succubus;
- the generic Scenario Editor from Succubus;
- the client and server-side geolocation runtimes from Geomotion.

These elements will be described later on in the document, whereas missing but needed parts will be defined and developed on demand. The platform will be created using an open architecture, with properly documented APIs, which allows third-party tools to integrate and exchange data (ex. game engines created by other producers), since these two components are not open source.

1.2 ROLE OF THIS DELIVERABLE IN THE PROJECT

The approach of the BEACONING platform is to put storylines at the heart of the gaming experience and to achieve a proper way to blend learning and gaming. To do so, we defined a set of rules:

- Audience should be transported into their avatar's universe to attach to it emotionally;
- Must have initial and intermediate twists to catch attention;
- Must present a long-term mission/thrill to hold interest on the long term;
- The long-term thrill/objective should not be self-centred (altruism);
- Learning under pressure is not efficient: learning as fun and entertainment, not as an obligation;
- The narrative should set the learner in mature situations;
- Must present one or several characters (identification process);
- It is better to represent the learner inserted in a group rather than acting alone - social aspect of the narrative/learning;
- A mentor (virtual) must accompany the learner (the mentor's feedback should be editable by the teachers) to provide help whenever needed;
- The narrative must allow splitting the story into quests/missions;
- The narratives must allow STEM topics in a natural way.

This deliverable will propose a method to ensure all these rules are enforced.

Another of the main ambitions is to have an editable technical platform open to the community of creators. Additionally, new runtime parts, such as dedicated mini-games should be added after the project is released. This document describes how we will succeed in this task.

1.3 STRUCTURE OF THE DOCUMENT

The document is split into 3 parts:

- A description of the runtime components, such as the game engine, the mini-games and the context-aware challenges;
- A description of the game scenarios content creation pipeline;
- An approach to how learning designers and then teachers can use this content to create a backdrop for their learning scenarios.

There are also 4 annexes: The Vault, Game plots, Mini-games, and the Plot Editor manual

2 RUNTIME COMPONENTS

This chapter explains all “runtime” components, that will be in interaction with the end users. At the time of the writing, we have identified 3 main elements: the point and click adventure game engine, the mini-games, and the context-aware challenges.

2.1 GAME ENGINE

The game engine is the first entry point for the player. This is where the story telling begins and when the first interactions happen. All the graphic content displayed further on is provided by the consortium and accessed through The Vault (see 3.1).

2.1.1 3D isometric view

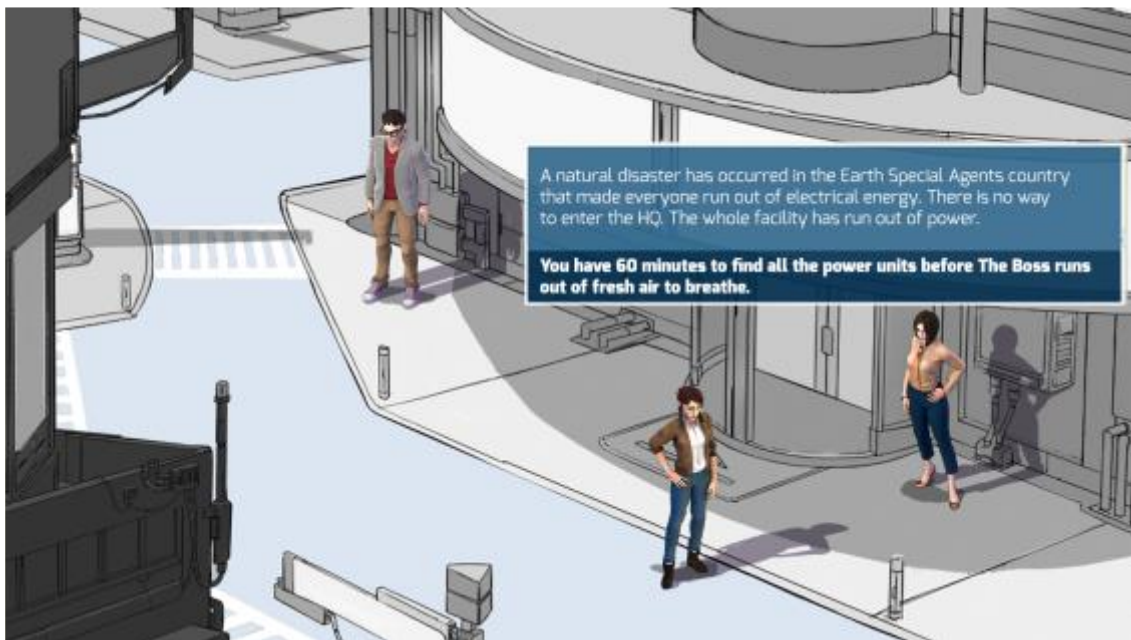


Figure 1: In game mock-up

The engine lets the player move their avatar around, click on interactive elements, talk with non-player characters (NPC) and launch activities (mini-games and context-aware challenges). The game engine also lets the player go from one background to another, and progressively unlock story content, intertwined with learning content.

2.1.2 Close-up dialogs

For important dialogs, a “focused view” or “close-up dialog” show high resolution animated characters talk with facial expressions and body language. If recorded voices are provided, the lip synchronisation will be automatically performed.



Figure 2: Close-up dialog

2.1.3 Inventory

Finally, an inventory will keep track of items found and/or carried by the player in their scenario. It can be anything that will have an icon (possibly uploaded by the content creator) and a text description.

2.2 MINI-GAMES

Mini-games are very specific game mechanics, meant to be different from the meta-game storytelling approach, aiming to bring fun, and challenge the knowledge of the players. Access to these games will be possible on mobile platforms through a web view component, to let game studios not belonging to the BEACONING consortium to add more of these later on. It also means that these new games should run in a webpage, with low requirements on hardware to render them.

The player's native mobile App will provide the 3D game environment, and inside a web view component, according to the lesson path, one or more challenges will be prompted to the player using the mini game as instrument.

Mini games can be assessment, just fun challenges or aimed at improving the STEM knowledge level of the player.

Depending on the gamified lesson path, a certain score in the mini game may be needed to let the player progress in the game experience.

One of the key advantages deriving from the use of mini games, inside a higher level meta game, is that doing this way the game ecosystem will be open in the future to host new mini game challenges, and the addition of new mini games will be almost seamless, not implying changes to the overall game infrastructure.

As mentioned above, the mini game set will be an open ecosystem, so the mini games list described in Annex 1 is just a starting point and not exhaustive set of all the mini games that in the future will be possible to add to the BEACONING ecosystem.

Word-o-copter

This mini game challenges the player to create short words while keeping an eye on moving helicopters getting them to land safely. The words, which have to be built with those helicopters, will be the answers to questions appearing on the screen and can be about several STEM topics. Enemy aircrafts try to hit and crash the player's helicopters. Using an own armed aircraft at the bottom of the screen, the player can defend their helicopters.

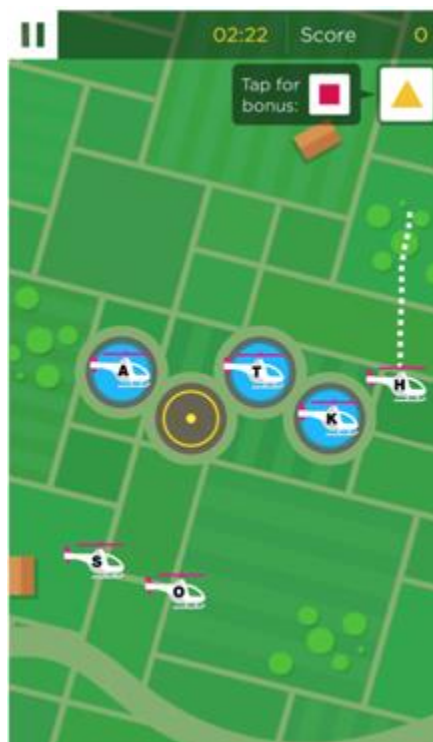


Figure 3: Word-o-copter mock-up

Helicopters will randomly appear on the screen, the frequency of new helicopters appearing will be one of the customizable parameters of the game. Suddenly, a quiz will be displayed on the screen. This can be a question which has to be answered, an equation or a physics problem which has to be solved. If the quiz has been configured to be solved within a certain amount of time, a timer will be displayed in addition to the time counting down.

After the question has been displayed, the player can start guessing the right answer, and as soon as they are sure about the answer, they type the solution into a typing box. This action triggers the helicopters which start flying into the screen and they will automatically try to land on their pad going to compose the right answer.

Several enemy aircrafts will appear on the screen, which try to shoot down the answering helicopters. To defend them, the player moves a shooting aircraft at the bottom of the screen. Tapping on the right or on the left side of the screen, the player will move their aircraft left or right.

The goal of the player is to make the answering helicopters land on their pad keeping them safe from enemies shooting. The player can choose between 3 types of shooting (weak, medium, strong) and according to the shooting level it will be easier to destroy enemy aircrafts.

The player will have some bonuses to use to get help on the answers, according to the number of bonuses they use, their shooting power will be decreased so they will need more shooting time to destroy flying enemies. There are 3 types of help bonuses:

- Some helicopters with the correct letters of the answer will already land, so that a part of the answer will be already pre-filled;
- For numeric answers, there will be a help such as greater than X and lower than Y;
- For other kind of answers there will be a multi option choice.

According to the type of answer, the system automatically provides the right kind of bonuses to be used.

2.3 CONTEXT AWARE CHALLENGES

Context aware outdoor challenges can be used in lesson paths and created by a location-based authoring tool integrated in the BEACONING Platform. This enables teachers to extend the students “game-based learning experience” beyond the boundaries of the classroom and allow them to explore the surroundings of the school, the neighbourhood or the city interacting with a real-life map through the BEACONING App.

Location-based challenges can be of different types:

- **Follow the Path:** linear location-based game challenge where students have to find and check-in at a specific Point of Interest (POI). When they find a POI the system will show a clue with information about the location of the next one;
- **Treasure hunt:** linear exploratory location-based game challenge where the goal is to find a hidden treasure in the real world. Individually or in groups, students will have to find clues that give them information that has to be interpreted to find the final location of the treasure;
- **Scout:** non-linear exploratory location-based game where the students have to walk around to find POI defined by the teacher. Each POI contains information and challenges;
- **Rat Race:** linear competitive location-based game where two or more teams of students have to participate in a race and be the first to reach the finish line. From the starting point to the goal the students have to solve challenges in different POI;
- **Conquest:** non-linear competitive/cooperative location-based game where the students have to conquer different zones of the city solving challenges. The first team to conquer all zones wins;
- **Jigsaw:** linear competitive/cooperative location-based game where the goal is to be the first team to arrive to a specific location solving puzzles finding different parts.

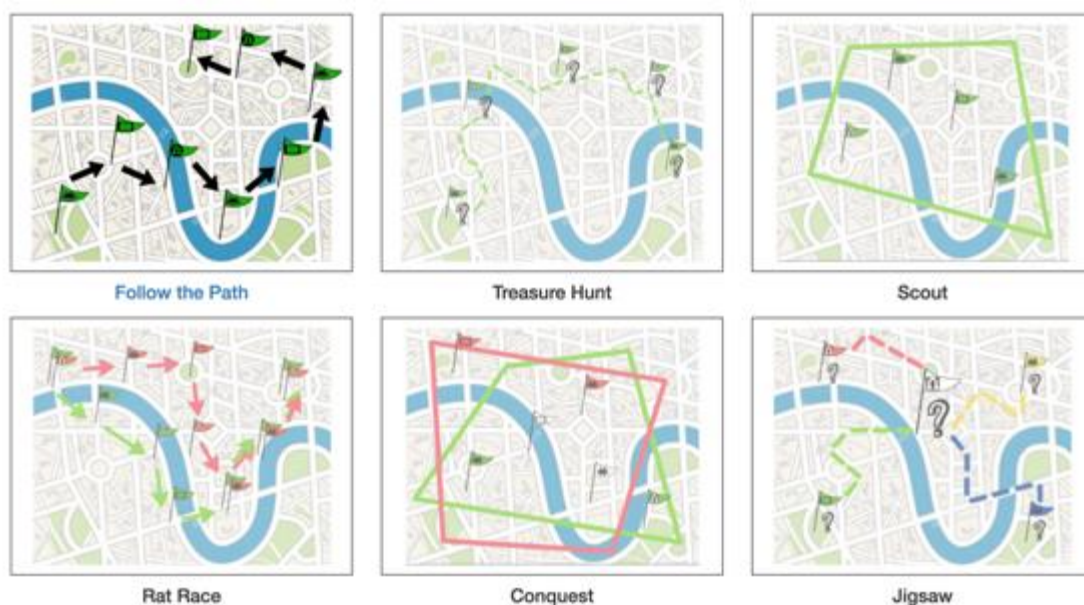


Figure 4: Type of location-based challenges

Teacher can decide if a context aware challenge fits the pedagogical goal of the lesson path setting the condition if game is still playable but depending on the scenario the progress may be locked as long as the challenged is not passed.

Technologies available to create location-based challenges are unified in the BEACONING platform. Technologies available are:

- **GPS:** should be used for tracking user outdoor location. The precision of this technology might vary on weather. That's why we highly recommend using GPS whenever user location does not need to be exactly precise;
- **BEACONS:** can be used both indoors and outdoors. When using outdoors system owner need to pay attention to device description and restrictions (e.g. temperature, high humidity in area). Beacons are both waterproof and can work in low temperature (been tested in - 15 C) however the last one may affect the battery life;
- **QR Codes:** QR codes usage depends on type of surface they are printed on. This type will decide if QR code can be used indoor and outdoor or only indoor. We recommend using QR codes only indoors because most of them will probably be printed on paper.

3 CREATION PIPELINE

As the BEACONING platform aims at living past the consortium, most parts of it are thought to be "extensible". So, although the starting content will be provided by consortium members, we want to open the door to other contributors, and give them a way to add content. There are two types of game additions that the platform will accept: new plots and new mini-games.

This chapter explains the tools and assets that will therefore be open to the community to create more of these additions. Although our choice is to keep the consortium as the warrant for the generic narratives coherence, to achieve such a goal the scenario is split into 2 parts: the **generic** narratives that set the background and defines what is the long-term story being told, and the **specific** narratives, that could be seen as a TV show episode.

3.1 THE VAULT

The generic narrative is the responsibility of the platform editor (the BEACONING consortium), so that everything stays in control and in coherence even though anyone can create new episodes from it. This part is called "The Vault", and is a story background created as to be ready to be used by external game designers. The Vault is where all the narratives of the game are stored. This is where we define all the individual stories of each non-player character of the universe, as well as the background of the world the game is happening in. It is also where we can find mandatory playable stories such as how the player joined the agency (First Time User Experience), and where we'll progressively unravel story telling elements (that we'll add with time, even after the platform is up and running).

The purposes of these narratives are multiple:

- Firstly, it must provide an immersive and convincing environment which is engaging and makes players willing to come back, because they want to know what will happen next;
- Secondly, it must be rich and varied enough to provide many pretexts for missions so that the lesson-path can be blended into an overall credible storyline;
- Thirdly, a good narrative will provide fun and fun makes people more open to learning;
- Finally, the generic narrative contained in the Vault will enable the specific narratives and other elements to blend seamlessly and avoid an otherwise incoherent gaming experience.

The Vault is a part of the BEACONING platform with restricted access to the core dteam, but its content will be inserted into the game experience automatically, with a little parameterization made by the authors. In the Vault we can find:

- A library of graphics resources, such as backgrounds, characters (NPC) and items to be chosen from;
- Bits of interactive dialogs with specific non-player characters, letting the player discover some stories about them, and also choose how they want to react to what they say. These dialogs are triggered depending on the previous choices of each player in the platform (a bit of random also help sometimes to make each user experience even more unique);
- Static content, as in non-interactive, such as texts, picture, audios or videos. These can be found throughout the game and will give more insights to the player on the world they're playing in. The players can collect these items, and it will reflect how much of the big story they have revealed yet;
- Interactive stories, such as the First-Time User Experience (FTUE), that will seamlessly blend into the experience.

Earth Special Agents

In these narratives, the learner plays a member of a global agency whose aim is to help the world. S/he is part of a young skilled agent team and each team member is really skilled in one specific area (renowned journalist with connections, sport athlete, etc.). The stories take place all over the world in various contexts: meet governors; go to local villages; meet famous leaders; and so on. The player will use their own skills and those specific ones of the team, to help the agency. As member of the agency, the player will receive instructions and will be supported by a mentor who guides them and provides help.

A first draft of these narratives can be seen in Annex 2.

3.2 GAME PLOTS

Plots are non-linear stories with "game mechanics" made to be played, and replayed, with varying background elements, chosen by the game designer but also dynamically picked from the Vault. Plots are meant to drive, motivate and engage the player through a game experience that is limited in time. Plots have lots of parameters, so that they can be customized. Most importantly, they can vary in length (number of chapters or challenges), for instance to properly fit a lesson-path. Multiple different lesson-path can rely on the same plot, so the plot authors (game designers) should take into consideration a great number of parameters to ensure enough variety when plots are replayed.

3.2.1 Plot example – Liberate Ocean Springs

The player is asked by The Boss to liberate Ocean Springs, a Neutral's city, as soon as possible. To do that the player will have to overcome three challenges that will test their skills and knowledge to decrease the reputation of The Dictator.

In the first challenge the player has to find the secret flat and discover the hidden code, by interacting with a mini-game that opens the door to meet other Special Agents of the Team. They will be given special items that will be very helpful to overcome forthcoming challenges.

The second challenge is a set of questions that will be presented to the player to be solved at the "Knowledge Competition Derby" against The Dictator. Questions will be shown with an increasing difficulty level and the first ones gives the player less reputation than the latest.

On the third challenge the player will have to move around the city or school in a given time and find location-based evidences that will reveal information of the bad intentions of The Dictator. The first evidence will be shown in a map on the mobile device. The evidence will give a clue of the place that the next evidence is hidden, only when player is close enough to the next evidence this will be shown in the map.

Depending on how many reputation points the player has on their scoreboard, there are 3 different endings where The Dictator loses its reputation but finally escapes, opening the possibility to authors to link this to new or existing plots.

A first draft of plots that will be implemented on the platform can be found in Annex 3.

3.2.2 Plot editor

To let the gaming community easily extend the game content on the platform and create new plots to choose from, a generic scenario editor will be open to **game designers**. This graphical tool will let third parties create new “plot backbones”, that will be used later on by **learning designers** to base their content on.

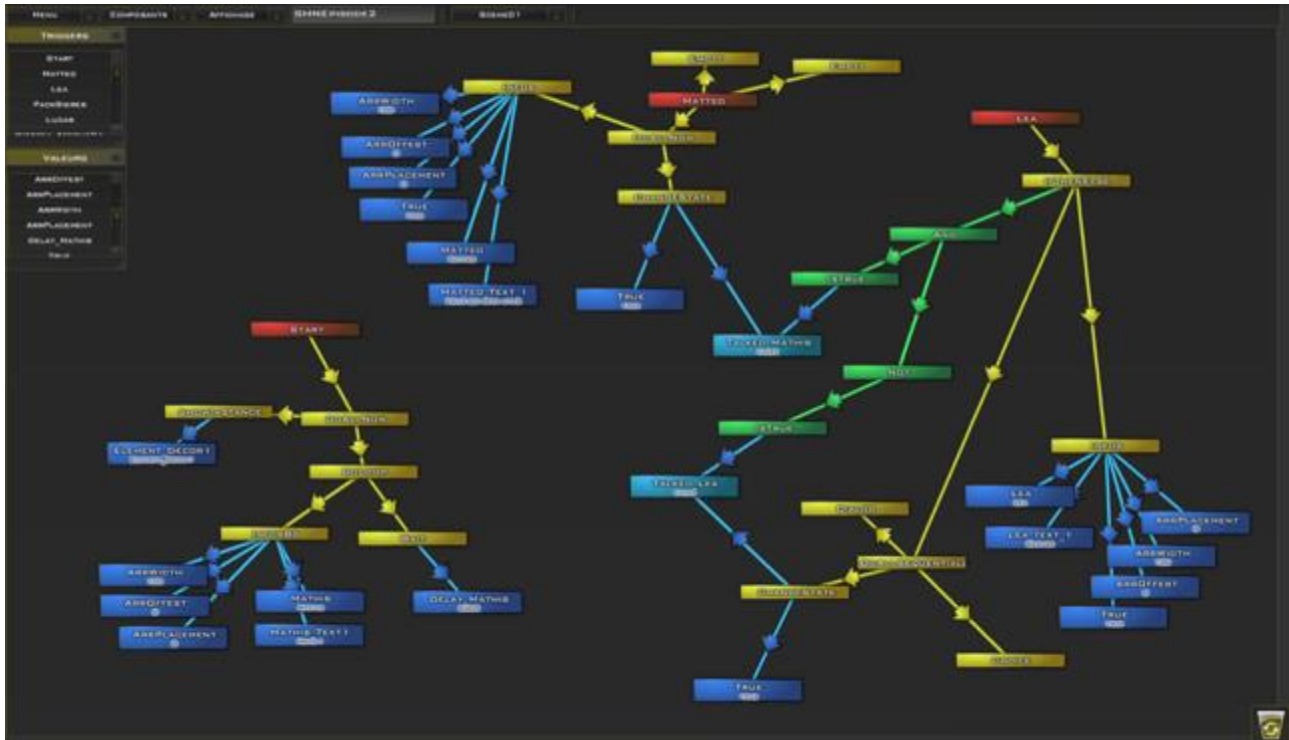


Figure 5: Generic Scenario Editor

The plot editor is based on Succubus' scenario editor, and acts as a visual scripting editor. More than the ease of use, the key point is that it lets people manage multiple types of unknown resource at the moment of the edition. For instance, text files (for dialogs), names of protagonists, GPS coordinates and so on can be open parameters than will be later customized by the learning designer, or by the teacher.

The Scenario Editor lets simple interaction behaviours be defined, based on a set of existing axioms. Things can be done such as “When user clicks on the blue door, if blue key is in inventory open the door, else pop-up an info message stating the door is closed”. The model can get as complex as needed (functions, variables, collections, etc.) so it can model any game scenario that is commonly encountered.

The manual of the scenario editor can be found in Annex 4.

3.3 MINI-GAMES

As the BEACONING pervasive game learning ecosystem is, in a long term view, supposed to be an open ecosystem, where the number of mini-games will increase over time, the idea is to start with a first set of mini-games that will be short, as reusable as possible, easy to configure and easy to be translated in several languages, by simply changing the loaded configuration.

Once this starting set of mini games are defined, it will be possible to add further, more topic specific games, to nurture the BEACONING game ecosystem.

According to this scenario, the overall BEACONING game architecture is designed to address the need to let the platform be open to accept new mini games in a way that will be as seamless as possible.

Therefore, the aim is to develop a native (iOS and Android) mobile App for players (students) that the user can download and install from Google Play Store or Apple Store, providing the narrative 3D gaming experience, and inside this native environment a web view component will host web based mini games that will be triggered by events and interactions inside the 3D game.



The idea to have web based mini games encapsulated into an external native environment is crucial for two reasons: the first one is that doing it this way it will be possible to add new mini games simply deploying them in the server, allowing them to be loaded by the web view component through an HTTP call, the second one is that it will be possible also for the authoring tool to generate "on-the-fly" a live preview of each mini game with the configuration just authored.

This approach also allows the release of new mini-games without republishing of a new version of the mobile App. In fact, this architecture will implement a full decoupling between the external native mobile App and mini games development roadmaps.

The picture below presents a high level logical diagram of the components involved in the mini-games architecture and the key communication flows.

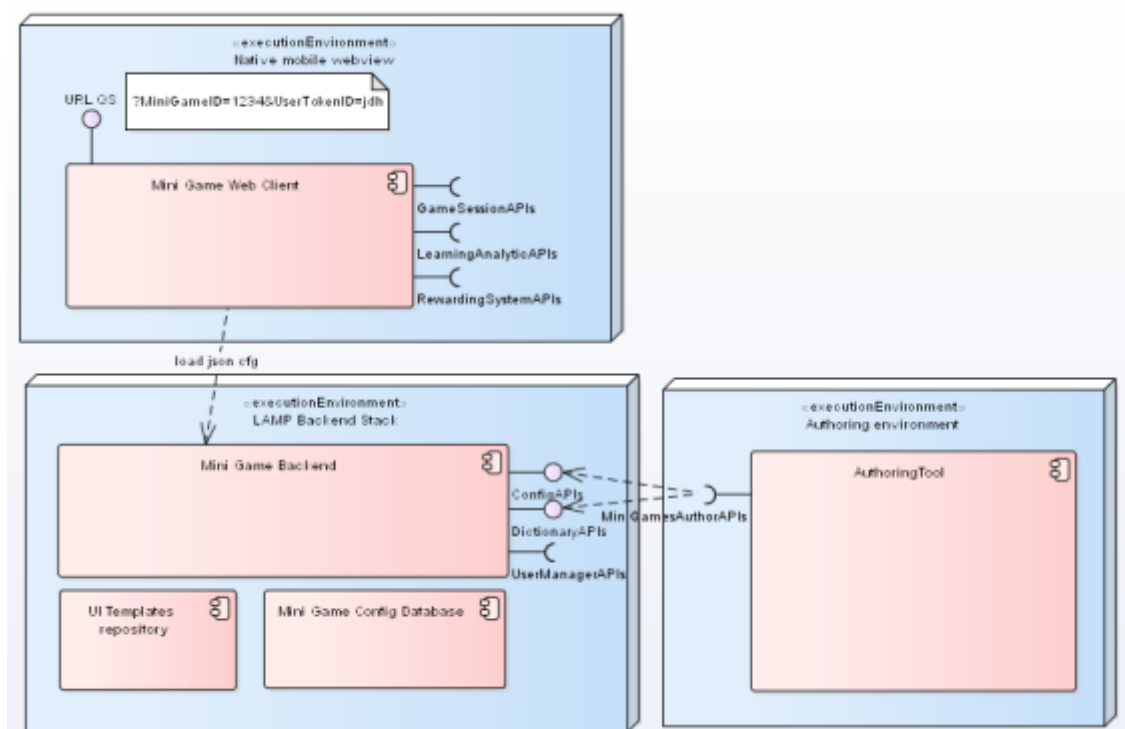


Figure 6: High level logical diagram

The web based mini-game will be invoked by the higher-level game environment through an HTTP call, passing one parameter that specifies the selected configuration and another parameter that define the user.

The idea is that each mini game, according to the loaded configuration specified by the input parameter, will be able to address different STEM topics, different levels of difficulty, and will provide several others possibly different behaviours according to the loaded game session configuration.

The mini-games backend will expose a set of RESTful APIs to the authoring environment to provide full customization and configuration endpoints.

4 CUSTOMIZATION

4.1 APPROACH

Each plot backbone coming out of the scenario editor will not be immediately playable, because it lacks learning content. This content will be added further down the pipeline by learning designer, and then customized by teachers before being played by students.

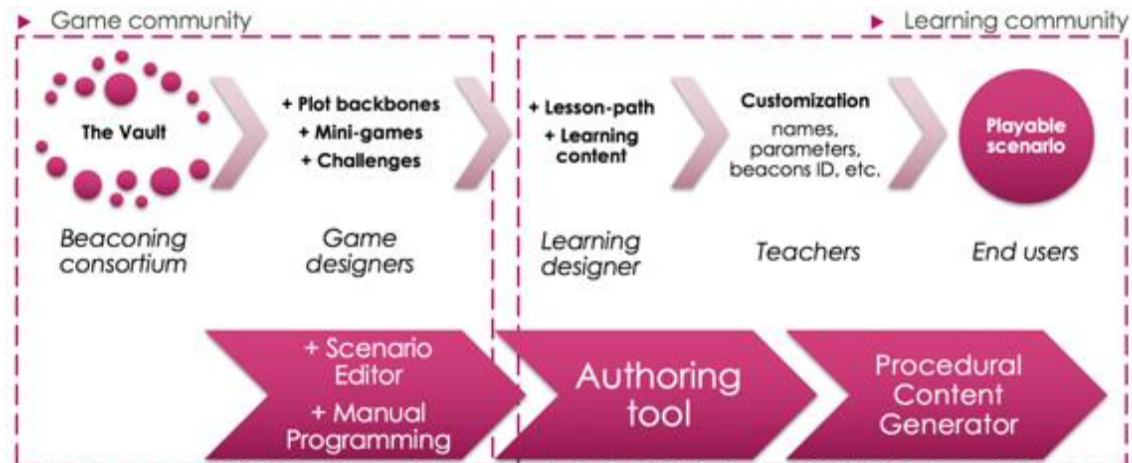


Figure 7: Content creation pipeline

Although this part is not exactly fitting into the game design approach, it is mandatory to explain at least briefly how what is created by the “Game Community” will be transformed before being used in the scenario played by the student by the “Learning Community”. The previous chapter therefore focused on the “game community” part, by describing the Vault and the game editors, and this current chapter describes what will happen further down the content creation pipeline.

Lesson-path, plots and mini-games have a list of parameters that needs to be defined before starting the game. This "customization" task is an iterative process where everybody in the chain will bring a few more of these parameters. Some of these need to be defined and closed right away, others can be left open so that someone down the chain can take care of them. The final step is done by the procedural content generation tool, that will fill all the blanks.

- **Learning designers:** when creating a lesson-path, a learning designer will pick a plot, and customize a few of its parameters, but he can also create more parameters, to fill their lesson-path needs;
- **Teachers:** a teacher can pick a lesson-path for their classroom, and customize many parameters, to make the experience fitting to their own needs, but also to the context. A few examples: GPS parameters, language, names of places to go to, hardness of the mini-games, pedagogical content specialization, length of the game, and so on;
- **Players:** a student may also have some parameters they may wish to customize such as playing at home or from school, and to choose from multiple mini-games style the one they prefer (same content will be delivered);
- Finally, after all of these people have defined or refined parameters, everything that’s still open is left to the procedural content generation tool.

4.2 EXAMPLE

This paragraph introduces an example of a customization process of one plot, according to the selected lesson plan.

Lesson plan	Plot	Mini games
Maths > Algebra > Quest 1	Liberate Ocean Spring	Solve-IT

Learning designer plot personalization

General settings	
Number of missions	3
Number of challenges	3
Total reputation points of the player	22
Total reputation points of The Dictator	45
Challenge 1 mini game settings	
Mini game	Solve-IT
Levels	easy
Questions and answers	12 x 4 34 x 12 24 + 43 etc.
Rewards	10 power units earned
Challenge 2 settings	
Mini game	Solve-IT
Location	Coventry (UK)
Levels	medium
Questions and answers	127 x 44 341 x 123 248 + 431 etc.
Challenge 3 settings	
Type	location based challenge
Location	Coventry (UK)
Evidences	Power Unity
	Lat: 8,769868 Lon: 45,03208

	Image.png
	The power generator needs to be activated with the Power Units collected to repower the headquarter.
	Badge: Medal of Honour of the Earth Special Agency

Teacher mini game personalization

Mini game	STEM topic	Other
Solve-IT	Algebra	Pausable:Y N Mandatory: Y N

Procedural content generator plot personalization

Plot scene	Dialogs
Power Unit	<p>The power generator needs to be activated with the Power Units collected to repower the headquarter.</p> <p>Go inside the Power Generator and place power units collected!</p> <p>--</p> <p>Lights on! Congratulations, you did a great job. You've repowered the headquarter and saved the Boss.</p> <p>The Earth Special Agency will reward you.</p>

Player settings

In the customization chain the players has the chance to override one or more settings according to their needs.

Game engine settings	Mini game settings
	Language switch disabled
	Selection of the accessibility theme
Audio:ON	Audio:OFF

5 CONCLUSION

This deliverable introduced an innovative way to structure a “community friendly” platform. Multiple problems were addressed:

- Creating a compelling and coherent non-linear game like storyline is a hard task that can be properly done only by a game studio;
- To avoid the game content to become boring, external third parties should be able to add more game content to the platform, but without breaking the existing narrative coherence;
- Integrating such experiences in a game engine normally asks for computer science skills combined with game design skills, and that would be a major challenge to the platform development as it needs a lot of content to strive;
- These tasks get even harder for learning games, where narrative content is mixed with learning content, and game designer are not learning designers, and vice-versa;
- Finally, extending the mini-games offer is also needed.

The objective of opening a game engine to a wider community than to video game skilled professionals is solved by:

- providing separate narrative storylines (The Vault) created by the consortium that are automatically integrated into the final scenario;
- providing a visual scenario editing tool to let game designers not skilled in computer sciences create and integrate game scripts (plots) directly into the platform without any further help;
- decoupling game and learning content creation, thus letting learning designers choosing existing game plots from a library, instead of (poorly) creating them themselves;
- defining a protocol between the main application and mini-games to let anyone create new such mini-games, without having to publish a new mobile version of the application;

APPENDIX: DEFINITIONS OF TERMS

Game designer

The game designer uses the plot editor to create new game scenarios for the community. They use bits from the Vault to link their content with long term narratives.

Learning Designer

By specializing existing game plots, a learning designer authors the learning content in the platform.

Teacher

A teacher is responsible for customizing and delivering the gamified lesson-path to their students.

End-user

This is the student, the player, the one that will use the platform to enjoy its content, and not create it.

The Vault

Asset repository for all graphics and also used to bring controlled and long term game narratives.

Plot

Specific short term game scenario created by game designers, and used by learning designers to gamify their lesson-path

Plot Editor

Based on the generic scenario editor, it is the tool used to let game designers create new plots.

Mini-game

Highly repurposable (content, difficulty, game rules) quiz-like game.

Avatar

Also known as the player character, this is the representation of the end-user inside the game.

Non Player Character (NPC)

These are all the others characters inside the game that are played by the computer with a scripted behaviour