European Commission

ReInHerit

Redefining the Future of Cultural Heritage, through a disruptive model of sustainability

ReInHerit
Redefining the Future of Cultural Heritage

www.reinherit.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004545

Project

Project Number	101004545
Project Acronym	ReInHerit
Project Title	Redefining the future of cultural heritage, through a disruptive model of sustainability
Starting Date	01/03/2021
Duration in Months	36
Funding Scheme	Coordination and Support Action
Call (part) Identifier	H2020-SC6-TRANSFORMATIONS-2020
Topic	TRANSFORMATIONS-19-2020 *Culture beyond borders – Facilitating innovation and research cooperation between European museums and heritage sites*
Website	www.reinherit.eu

Deliverable

Work Package	3
Task	T3.5: Story based game development
Deliverable	D3.6 Story Based Game Report
Dissemination Level	
Type of Deliverable	Report
Leader	CYENS – Centre of Excellence
Due Date	30/06/2023
Submission Date	
Keywords	

Version History

Version	Date	Author	Notes
1	26/06/2023	Panayiotis Charalambous, Christos Othonos, Theodoros Kyriakou, Andreas Panayiotou Niki Kyriakou	

Acronyms and abbreviations

Field of View	**FOV**

Disclaimer

This document reflects only the author's view, and the Research Executive Agency is not responsible for any use that may be made of the information it contains. The current deliverable is a technical game design document, providing the guidelines on how to develop a story-based game.

Table of Contents

1 Introduction

The main aim of task T3.5 was the development of a story-based interactive game addressing various aspects of cultural management, utilizing CH assets. Using interactive technologies and game-based learning, the aim of the game is to increase awareness of and appreciation for EU culture. The game targets players over the age of 13 with functionality that enables any interested player to get more information about certain artifacts, thus covering different types of audiences.

During M12-M30 several brainstorming sessions took place between the technical and cultural heritage professionals of the ReInHerit consortium, led by CYENS. The scope, style, and platforms that the game will run were thoroughly discussed.

During the first phase of the game development different experiences and game mechanics were designed and developed to enrich the educational experience of players. This allowed the development of an early game prototype (without the final assets). A final game was then deployed (includes 3D assets or artifacts scanned from the Bank of Cyprus Cultural Foundation - BoCCF) and is available on the ReInHerit digital hub.

The current deliverable provides detailed guidelines for museums and cultural heritage organizations that wish to build a similar game or evolve the current game developed by the ReInHerit project.

2 Game Concept

*"That's where **gaming**'s strength lies, not as storyteller but **as story generator**… Only when games accept that unique strength, take pride in it, and stop borrowing the clothes of others, will they truly achieve their potential as the only truly new creative medium of the last 100 years."*

'Systems Vs. Stories', *Eurogamer*, 22 June 2013

2.1 The Design process

2.1.1 Multidisciplinary team

For the development of task '*T3.5: Story based game development'*, a multidisciplinary team was set up composed of technical and cultural heritage professionals bringing together diverse and complimentary expertise aiming to deliver. The technical team of CYENS was composed by digital artists and developers while it was complimented by cultural heritage professionals of BoCCF, with expertise on cultural heritage management, archaeology, history, and curation. The multidisciplinary team focused on delivering a technically sound game while ensuring historical and archaeological accuracy.

During M12-M30 several brainstorming sessions took place between the technical and cultural heritage professionals of the ReInHerit consortium, led by CYENS. The scope, style, and platforms that the game will run on were thoroughly discussed. An interdisciplinary team of experts (archaeologists, museologists, historians etc.) ensured the scientific/historical accuracy of the game and its added value in heritage management.

An initial ideation meeting was organized by CYENS on December 14th, 2021 (Figure 1), where all consortium partners participated; several other brainstorming sessions followed. During the sessions the objectives, requirements (educational aspects, historical context, technology requirements, type of game, define the site(s) and/or museum(s) etc.) for the development of the story-based game were addressed and discussed among the partners.

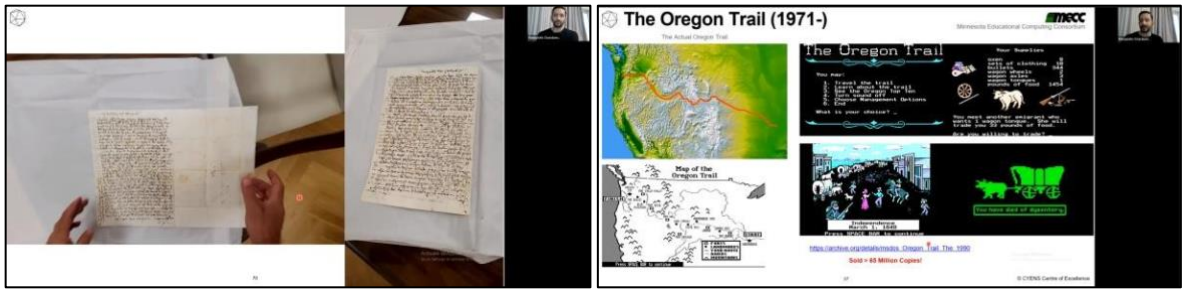


Figure 1: The first brainstorming session organized by CYENS for the development of the Video Game was organized over Zoom in December 2021.

The timeframe, the available technologies and expertise were also addressed during the sessions. Figure Complementary to the brainstorming sessions, the CYENS team, which is by default multidisciplinary as it is composed of programmers, 3D artists, archaeologists, visited the BoCCF Museum where the BoCCF personnel (archaeologists and art historians) provided tours to the museum areas (exhibition area, storage spaces, staff areas). The aim of such in situ visits was mainly to understand the mechanisms of a museum.

Figure 2-3: Brainstorming and Tour at BoCCF

Over the course of the remaining months, CYENS and BoCCF met several times both physical and online to clarify the concept of the game and track its evolution over time (Figure 4).

Figure 4 Follow -up brainstorming sessions with representatives of BOCCF and CYENS were held over the development of the game since the game would be utilizing assets from the BoCCF's museum[1].

In addition to all the brainstorming sessions with the partners, CYENS team had regular weekly meetings to set goals and discuss issues with the cultural heritage professionals of BoCCF. Given the set timeframe to develop the task, this was a necessity since it allowed the team to iterate ideas, face technical challenges and be able to adapt, sometimes leading to some game mechanics being redesigned. These meetings were both physical and online, always aiming for agility and efficient collaboration between technical and CH partners.

Beyond the collaboration developed between the partners for the delivery of T3.5 a significant milestone of the project was the skills and knowledge exchange between technical and CH partners. The technical partners gained significant awareness on the processes and procedures of managing cultural heritage organizations/museums and the CH partners were introduced to 3D documentation practices and techniques.

2.1.2 Game Concept – Museum Management

The main game concept was to build a museum management game, where the player assumes the role of a museum director. Their decisions shape and curate the museum experience for visitors. And depending on the success of that museum, the player is rewarded with more game content. The main inspirations for this concept were games like the following:

[1] https://www.boccf.org/en-gb/homepage/

- Roller Coaster Tycoon, Franchise[2]
- Two Point Hospital, PC Game[3]
- Theme Hospital

The aim is to create a simulation game that allows you to create your "own emergent story". A fundamental goal is to make the players think of the design process and management of a museum in a fun, creative but not overwhelming manner.

The team decided to make the base game 3D with a top-down camera[4] and to keep the graphics simple and stylized. Any 2D graphics required such as the images of non-real artifacts that the player should find as part of the gameplay would be supplemented using Generative AI technologies (see Section 3.4). It must be underlined that none of the actual assets that can be hosted in the virtual museum are generated with AI – only scanned artifacts of real objects can be hosted in the museum. Generative AI is used to create only non-existent artifacts that the players must identify as fake otherwise they get penalties in the game. It was decided to have this as part of the gameplay to add some form of added layer of information about the dangers of malicious use of AI in the real world and how they could potentially be used to create counterfeit art.

2.2 Source Code and Links

The game is available at the ReInHerit Digital Hub using the following link Videogame | ReInHerit Digital Hub (reinherit-hub.eu).

The source code of the game with all assets can be found in the ReInHerit Digital Hub and in ReInHerit Github account https://github.com/CYENS/ReInherit-Video-Game.

[2] https://www.rollercoastertycoon.com/
[3] https://en.wikipedia.org/wiki/Two_Point_Hospital
[4] https://behind-the-scenes.net/popular-camera-angles-used-in-video-games/#Top-down_view

3 The game

3.1 Concept

Story: The player serves as the protégé of the Bank of Cyprus Cultural Foundation Director. The director entrusts the player with the museum, for the player to prove its worthiness. The player is granted the opportunity to assume the responsibilities of running the museum in the director's place for a duration of one month. If the player manages to meet expectations (or exceed them), they will become the new director. The player must rise to the occasion, demonstrating their exceptional management skills and (re)inherit the museum with confidence and skill!

Gameplay: A top-down simulation style game, where a player manages a museum taking a series of roles. The gameplay can be enriched with other options where the player will be able to create the museum rooms, place exhibits, manage/restore artifacts and maintain order amongst the visitors.

In the following sections an overall description of the roles and technical details related to the implementation and modification of these functionalities is provided, to enable other organizations to set up their own game. A programmer with skills in game engines (Unity preferable) can do most of the modifications in the game. In case new assets must be scanned, this would require someone with photogrammetry/laser scanning expertise. For any other given added functionality in the game (e.g., AI, art, etc), the personnel/outsource requirements would increase appropriately.

3.2 Game Flow – The Player's Journey

The game flow in video games refers to the detailed breakdown and description of the ***player's journey*** through the game. It starts with the initial user experience such as the game's startup and menu navigation, progresses through gameplay loops including core mechanics and level designs, incorporates key events, challenges, and concludes with the endgame scenario, in case the game has an end point. Thus, the following subsections focus on the description of the game flow.

3.2.1 Overview

This subsection presents with detail the game flow of the game.

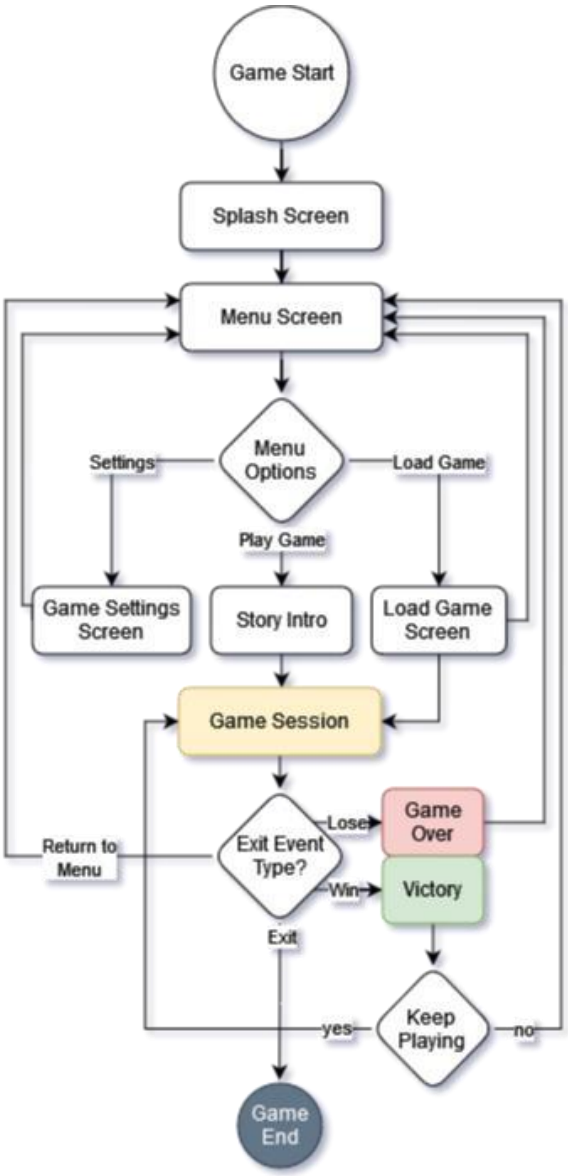


Figure 5 Game Flow

The game starts with a Splash screen, the players are then taken to a Menu Screen with several options:

- New Game: Starts a new game.
- Load Game: Takes players to a new screen that allows them to choose a previously saved session to continue the game from.
- Game Settings: Takes players to a screen with sound settings.

Quit: Exits the game by starting a new game, players are presented with a short Story Intro showing the plot of the game. Followed by a short animation of the museum director (which players take control of) walking up to their office desk and taking a seat. On the director's desk is a model/maquette of the museum.

A fade transition takes the players into the game session. The camera hovers over the desk, and the museum model becomes their stage. The players keep playing the game until one of few things happen:

- Game Over: The player fails the game due to a very low score or bankruptcy.
- Victory: After a set amount of rounds the player is victorious and is presented with an end cutscene. After which point, they decide if they wish to continue playing the game, or whether to return to the menu.
- Exit: The player willingly exits the game to return to the menu.

Once the game starts, players enter the “Closed Museum” mode. During that mode they can make changes and modify their museum. When they are ready, they can press a button to open the museum. This process is shown in Figure 6.

The “Open Museum” mode lasts for a certain amount of time, and during that period visitors come to the museum to view the exhibits. After that amount of time, players are shown some events, of which some are positive (monetary awards, good reviews, new artifacts, etc.) and others are negative (damages, bad reviews, etc.). Then, following that event, the museum closes once more. This loop continues several times until the end of the game.

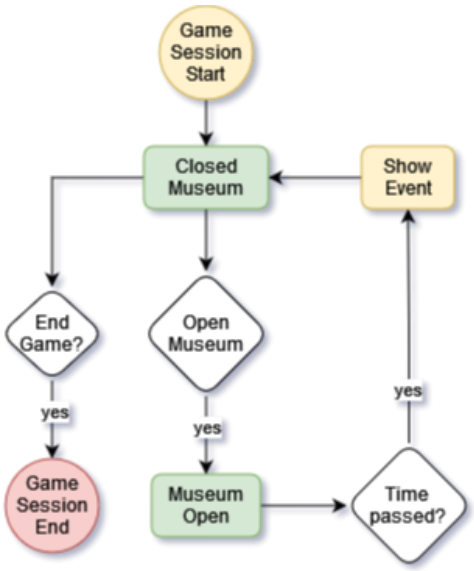


Figure 6 Game Session Cycle

3.2.2 How the game progresses

The game has several progress metrics that players can use as indicators for their progress:

- Day: The current “round” of the game.
- Artifacts: Players acquire more artifacts as they play the game
- Museum Size: As players progress through the game positively, the game allows them to expand their museum in size. The allowed space is shown with a grid.

3.2.3 Task Difficulty

Progressive increase in task difficulty/challenges faced by players is a very important aspect of any game and it is considered an art. The difficulty of the ReInHerit game increases progressively as players advance through the game. This happens with the following ways:

- While the museum building space expands, a larger budget is needed for cleaning and operation tasks.
- When more artifacts are placed in the museum, more maintenance and cleaning is required.
- The number of visitors may increase during gameplay; thus, enough artifacts need to be placed and maintained in the museum to keep them interested.

3.3 Game Rules

Defining appropriate game rules in a video game is also an essential process. They can be described as a set of guidelines that controls the players' actions and their main usages are to define what players can and cannot do, set the objectives they should pursue, and define the consequences of their actions. These rules serve to structure gameplay, creating a balanced and engaging experience while fostering strategic thinking, problem-solving, and progression. Specifically, mostly simple, and straight-forward rules have been used in the ReInHerit game, with the aim of not increasing the complexity of the game unnecessarily, something that will be an overhead for future developers. The subsections below present the most important rules that have been incorporated in the game.

3.3.1 Winning and Losing the Game

These rules basically control when, and if, the game ends. There are two conditions that the players need to be aware of: the Victory (win) condition, and the Failure (lose) condition. If players reach a number of rounds, without meeting any of the failure conditions, they win the game. The failure condition is met when:

- The museum is left with zero budget (files for bankruptcy)
- The museum has an extremely low rating from visitors. This may be due to poor management, dirty museum and poorly maintained artifacts.

3.3.2 Opening and Closing the Museum

The museum is in one of two modes: *Open* or *Closed*. By default, the museum starts out closed. In this mode the player is free to make any changes and additions to the museum. This includes adding/subtracting rooms, adding/moving/upgrading artifacts, cleaning up the floors and more. The player also can open the museum by pressing a button, but **only-if** the following conditions are met:

- No workers (e.g., Keepers) are on the exhibit floor, performing tasks.
- At least one room block is added, that is connected to the entrance room of the museum.
- At least one artifact is placed in any of the available museum's rooms.

Once the museum is open, the player is unable to close it until the preset opening time is up. The museum will only close automatically after a certain amount of time has passed. During that time, visitors will appear and interact with the artifacts and each other.

3.4 Game Roles

The game contains a list of different roles. Some of them are controlled by the player, the behavior of other roles is controlled by an AI system. Additionally, some of these roles can be visualized with characters (e.g. visitors) while some of others are not visible in the game (e.g. the architect). Likewise, these game roles have been selected mainly based on discussions with professionals from the cultural heritage sector, while also deciding whether they will serve a purpose in the final game.

1. **Museum Director**: This role is assumed directly by the player, as their choices shape the museum. Their directions essentially guide the workers of the establishment and will result in either success or failure.
2. **Architect**: Building custom museums is one of the main mechanisms of the game. Thus, the players can decide the structure of their museum by taking the place of an architect. Details on how the player controls this mode are mentioned in Section 4.3.
3. **Curator**: The curator of the museum is responsible for deciding where the artifacts will be placed. The players indirectly control the curator by entering the “Curator Mode” and placing exhibits in desired places anywhere in their museum. The functionalities of this mode are presented in Section 4.6.1.
4. **Keepers**: The keeper is a type of worker that moves artifacts from their storage to the exhibit hall and vice versa. They may also install exhibit cases where the artifacts are stored inside. While the player does not control these workers directly, they will appear when the curator has decided where to place an artifact. They will perform their tasks and then exit through the elevator that they came from. A more detailed description of this game role can be found in Section 4.6.1.
5. **Visitors**: The primary responsibility of visitors are to bring vibrancy to the museum, while also serving as a mechanism for assessing the player’s performance. Accordingly, visitors’ fundamental activities consist of moving in the museum while observing artifacts and engaging with other visitors. Their behavior is entirely controlled by an AI system which is described in Section 4.6.2.
6. **Conservator**: This type of employee works behind the scenes and is not visible in the game. However, their responsibility is to restore an artifact to the state it was.
7. **Janitor**: The responsibility of the janitor is to keep the museum clean, for the visitors that enter the museum. The cleaning process only takes a few moments.

4 Technical Section

4.1 User Interface and User Experience

4.1.1 User Interface

In the gaming context, a User Interface (UI) includes all visual design elements like menus, icons, control prompts, and on-screen indicators that facilitate a player's interaction with the game. It's a fundamental component that communicates critical game information effectively, ensuring players can navigate, understand, and interact with the game smoothly and intuitively.

4.1.1.1 Start UI

In this subsection the interface of the start screen is described. There are four buttons in the start screen (Figure 7 – Left Image). When the “PLAY” button is pressed, the game starts from the beginning and when the “QUIT” button is pressed the game closes. Furthermore, when the “CONTINUE” button is pressed, the player can continue the game from a previously saved game state (Figure 7). When the "OPTIONS" button is pressed, the player can adjust the game volume and set the desired graphics quality (Figure 7 – Right Image).

Figure 7 Start Menu

4.1.1.2 Game UI

In this subsection the interface of the game is described. An easy-to-use interface was selected. The following figures present the main components of the user interface, combined with a number tag, and the usage of each one is described below. Firstly, component 1 shows the current round, available in-game money, score and total number of visitors. Secondly, component 2 is a button that when selected, the current state of the game is saved. Thirdly, component 3 is the button that enters the player into building mode; the dropdown list contains a group of buttons that gives the player the ability of adding new rooms, deleting existing rooms and connecting a pair of rooms by creating a door. After one of these functionalities has been selected, the user can execute it by clicking and dragging in any available position on the blue grid. Then, similarly, component 4 enters the player into curating mode; in this mode the player can select any of the available artifacts that are contained in the inventory window, and then can place it in any available space in the museum. Furthermore, component 5 is a button that when pressed changes the museum’s state to open. Finally, the slider in component 6 enables the player to speed up the game. The interface with components 1-6 is presented in Figures 8-9.

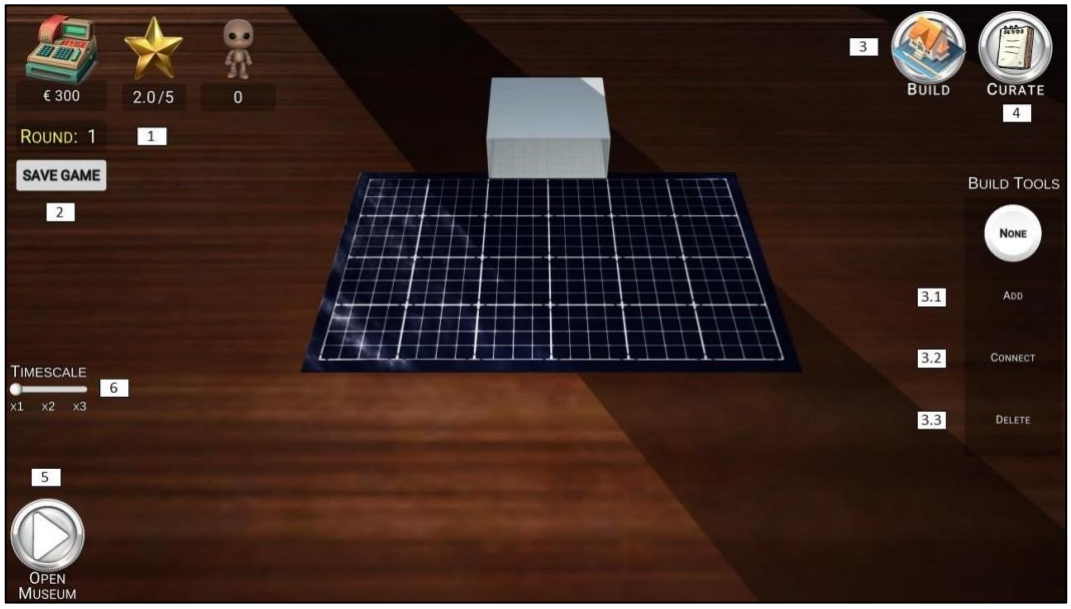


Figure 8 User Interface - Main

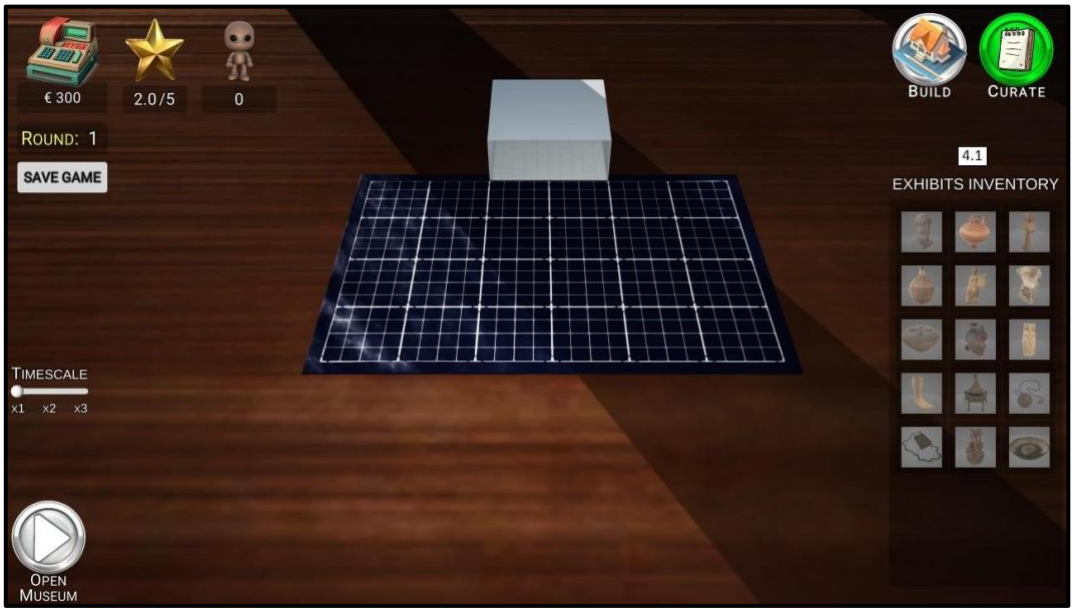


Figure 9 User Interface - Extension

Finally, when an artifact is placed in the museum, the player can click on it and get the options of component 6 (Figure 8). When button 7.1 is pressed, the selected artifact enters in observation mode; the camera zooms-in and presents the artifact and at the same time a popup window appears containing information about the artifact (Figure 9); this information is provided by the BOCCF to reflect the history of the artifact. Buttons 7.2 and 7.3 in Figure 10 give the ability of removing the artifact from the museum and modifying its position respectively. Likewise, when button 7.2 is clicked, the case of the artifact can be upgraded to a glass one, which can give higher scores by visitors. Finally, button 7.5 cleans the artifact.

Figure 10 User Interface - Artifact Control

Figure 11 The player can see a high-quality photo of the artifact in the top-left corner combined with a brief description. A more detailed description can be found by following the provided link.

4.1.2 User Experience

4.1.2.1 Camera System

The camera, especially in top-down games, plays a critical role since it functions as the eyes of the player. It is important to ensure that the camera consistently maintains an optimal angle to allow for easy and efficient navigation in the environment.

The player has the capability of controlling a top-down camera using keyboard and mouse input. It includes operations like panning left and right, zooming in and out, and rotating the camera. The player can move the camera in the front, back, left and right directions using the

WASD keys, as shown in Figure 11. Furthermore, the camera can zoom-in and zoom-out using the scroll wheel of the mouse, while also, the player can rotate the camera by holding the mouse right click in combination with the WASD keys. The zoom and rotate operations are presented in Figure 13 and Figure 14 respectively.

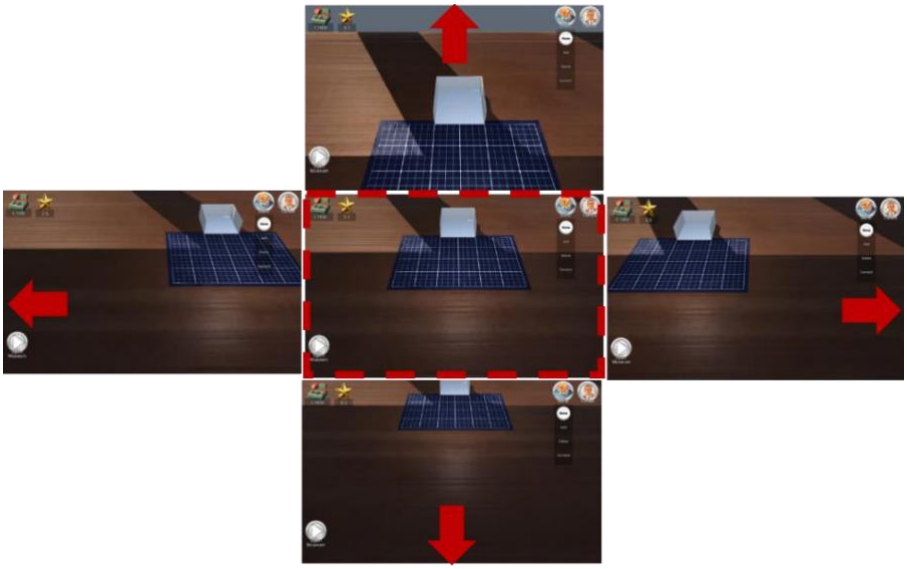


Figure 12 The player can move the camera Left by pressing the "A" button, Right by pressing the "D" button, Forward by pressing the "W" button and Backwards by pressing the "S" button.

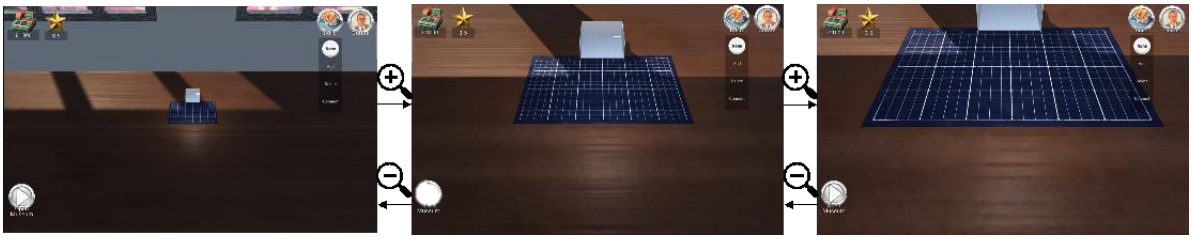


Figure 13 The player can zoom-in by scrolling the mouse wheel up and zoom-out by scrolling the mouse wheel down.

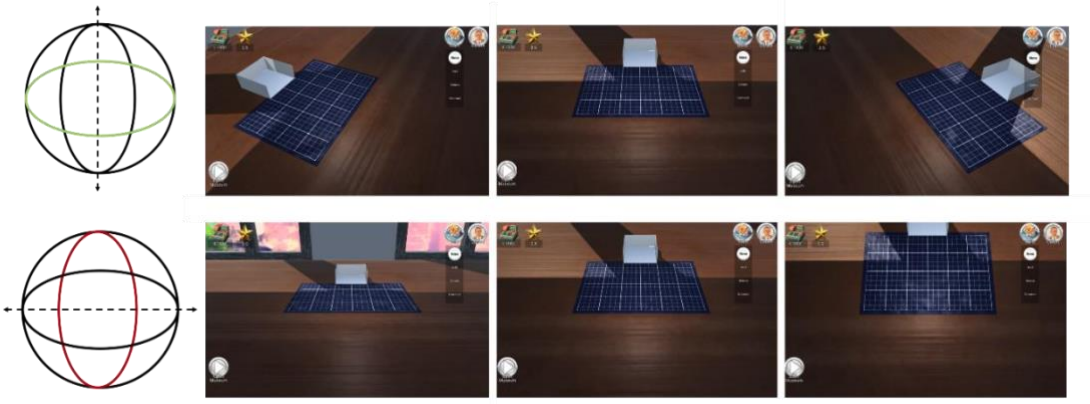


Figure 14 The player can rotate the camera on the Y-axis as shown in the above images, by pressing the right click on the mouse and the buttons "A" or "D" and rotate the camera on the X-axis using "W" and "S" buttons.

4.2 Art Style

"Good graphics won't save a game with bad gameplay, but good gameplay won't do it alone without highquality visuals."[5]

The ReInHerit game belongs to the "Fantasy Realism" category because it combines elements of both fantasy and reality. For the fantasy part, there are imaginative elements like characters and effects; the real world is represented with real-world artifacts that are currently exhibited at BoCCF and have been scanned and imported. This art style was chosen to accurately show museum artifacts, allowing players to observe their realistic appearance and retrieve information on their historical content and value. In addition, the incorporation of fantasy and playful elements has been used to enhance the entertainment value of the game and attract a younger audience to engage with the game.

4.2.1 Characters

The characters have cartoonish appearance with exaggerated proportions, while clothing and apparel can be added.

4.2.2 Museum Model

The art style of the museum model had to be designed in a way to match the fantasy and cartoonish appearance of the characters in combination with the realistic artifacts that are placed in the museum. Likewise, after some suggestions from cultural heritage professionals it was pointed out that *the current trend in museums' architecture follows a more minimalistic non-cluttered design to allow visitors focus on the artifacts rather than be distracted by external factors*. Thus, considering all the requirements mentioned above, a minimalistic design has been followed.

4.2.3 Background environment – Skybox

A skybox is generally used to make video game scenery appear bigger, more impressive, and less artificial. Basically, a skybox can be characterized as a box that surrounds the player's entire field of view with a "sky". Thus, applying a skybox allows game developers to define the farthest visuals of the game, while also enhancing the realism of the environment.

A skybox asset can be created from scratch, downloaded from an asset store, or generated using various tools. Specifically, in the case of the game, a generative AI tool called "Blockade Labs – Skybox AI"[6] has been used. This tool combines AI and user input, which can be in the form of text or drawing and generates a skybox. Finally, after generation, the tool gives the ability to the user of downloading the skybox asset as a picture, which then can be imported to the desired game engine[7]. A demo of the generated skybox that has been used in the game is shown in Figure 14, featuring a blurry town in the background.

[5] https://kevurugames.com/blog/choosing-an-art-style-for-your-video-game/
[6] https://www.blockadelabs.com/
[7] https://docs.unity3d.com/560/Documentation/Manual/HOWTO-UseSkybox.html

Figure 15 AI generated skybox made with "Skybox AI" by Blockade Labs.

4.3 Building Museums and the progression in the game

The functionality of building custom museums is one of the main concepts of the game. The player can take the role of an architect and build a custom museum. The player starts with a limited budget and building area, and while they progress through the game acquires more monetary rewards and the building area expands; this means that a larger and more complex museum can be created, more artifacts can be placed in the museum, something that allows more visitors to enter the museum.

The main functionalities of the architect mode, as mentioned in Section 4.1.1.2, are adding new rooms, deleting existing rooms and merging rooms using a door. In this section the highlevel implementation of the architect mode and how the player executes each of the three tasks will be described.

First, a grid area has been utilized where each cell can contain a maximum of one room. A room is a fixed size block that can be placed in any empty cell on the grid area. Second, a graph-based approach has been followed that is useful for keeping track of every room placed, while also using this method, the process of identifying adjacent is easy and fast. Furthermore, when a new room is placed or an existing room is deleted, there is a check that identifies which type of partitioning is suitable between neighboring rooms. The available partitions are a solid wall, a door or no separation that indicates that multiple rooms are merged to form a larger one. The steps that the player needs to perform to achieve every available task are presented below.

Adding room/s: Adding a new room/s is a simple process. The player can click in a single cell of the building grid to create a new single room. Likewise, as presented by the Figure 16 below, there is the ability of selecting multiple grid cells simultaneously and multiple rooms will be created and merged to form a larger area.

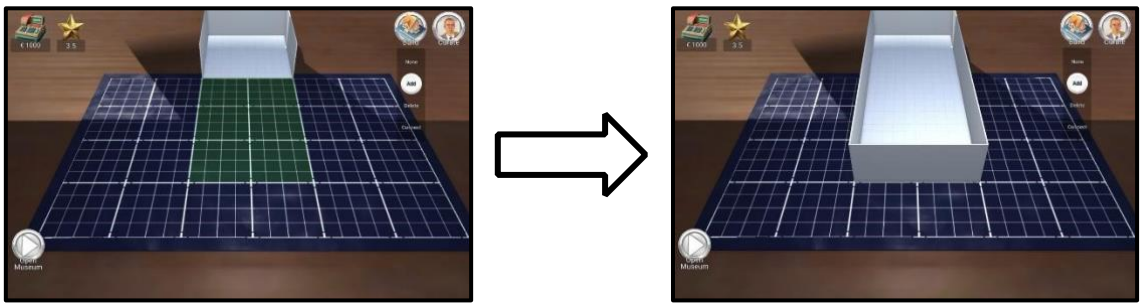


Figure 16 Creating a museum from a blank canvas.

Deleting room/s: The process of deleting rooms is identical to the process of adding new rooms. The player can delete a single room by clicking on it or by selecting multiple existing rooms for batch deletion. A demo of the process is shown in Figure 17.

Figure 17 The process of deleting existing rooms.

Connecting rooms: This functionality assumes that there are already at least two adjacent rooms. Then, the player can create a door that connects the two rooms by selecting them as shown in the following Figure 18.

Figure 18 The process of connecting two existing rooms.

4.4 Assets

In a video game, assets refer to all the elements used to construct the game world, including 2D and 3D models like characters, environments, and objects, as well as textures, sounds, animations, and scripts. This section focuses on the generation of the assets that have been used in the game.

4.4.1 Artifacts

This section provides details related to the artifacts that have been used in the game. The following subsections present how the real artifacts have been digitized, various techniques and tools that have been used to utilize them in the game and steps on how new artifacts can

be added. Finally, besides the 3D models of the artifacts, information that describes them is included in the game. Thus, the player can observe them and learn more details about each one of them while playing the game.

4.4.1.1 Collection - Photogrammetry

As mentioned above, there is the need to generate 3D models of museum artifacts. There are multiple methods and techniques to achieve that; however, each one has its advantages and disadvantages. In the case of the game, a process called photogrammetry has been used. The main reason that this method has been utilized is that no specialized equipment is required, besides a camera. Likewise, another critical consideration is that artifacts are objects of significant historical value and moving them is not always an easy process, thus a method that is able to work in-place inside the museum is preferrable.

Photogrammetry is a technique used to measure and record physical spaces or objects, translating them into digital 3D models. The process involves capturing multiple overlapping photos of an object from different angles. Subsequently, using a software like Meshroom[8], which is a free and open-source 3D Reconstruction Software, points between the photos are analyzed and matched to build a highly accurate 3D representation. This method is widely used in fields such as surveying, architecture, archaeology, and increasingly in video game development, as it allows for the creation of highly detailed and realistic 3D assets based on real-world subjects.

Photogrammetry can produce excellent results but is a process that must be executed carefully as some factors can highly affect the end result. First, lighting is a critical factor, thus ideal lighting conditions must be ensured. Second, coverage is another important factor. The collected photos need to cover the entire object, from every available angle, while having enough overlap between them to enable the reconstruction software to match them successfully. Finally, the material of the object may affect the quality of the reconstructed model, as shiny and reflective or monochromatic surfaces makes the process more difficult because of reflections and fewer unique points for matching respectively. Figure 19 shows the capture process and the final 3D model for one artifact.

[8] https://alicevision.org/

Figure 19 The process of Photogrammetry was used to carefully digitize the artifacts to be used in the video game. On the right, one of the reconstructed models, "Temple Boy - Cypriot Sculpture 4th century B.C." is shown. The model can be interactively viewed at: https://skfb.ly/oDN7X.

4.4.1.2 Level of Detail (LOD)

In the context of computer graphics, Level of Detail (LOD) is a set of algorithms used to manage the complexity of 3D models based on their distance from the player's viewpoint, helping to optimize performance. As an object gets farther from the camera, its lower LOD versions - simplified models with fewer polygons - are presented, ensuring that the high-detail models are reserved for when they are most needed. This enables a scene to maintain high visual fidelity where it matters most, while minimizing unnecessary strain on the system. Specifically, to achieve that the scanned models are imported into Blender[9], a free and opensource 3D creation suite. Then, a Decimate modifier, which can reduce the vertex/face count of a mesh, is added to the model and gives the ability of setting a ratio that affects the overall quality of the model. More details about the process mentioned above can be found in Section 4.4.3.4. Finally, using this technique, the same model can have various versions with different levels of details. Figure 20 below presents an artifact model and its different versions, where higher levels contain a greater number of polygons.

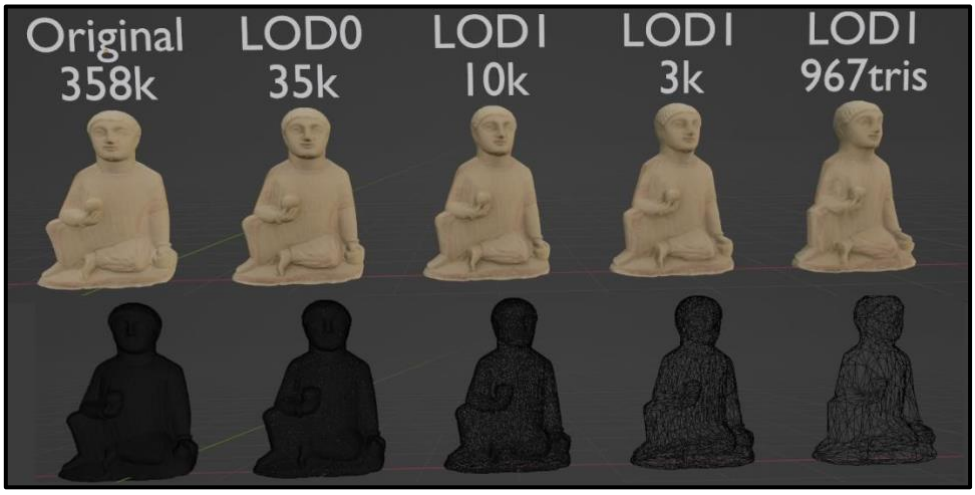


Figure 20 Various versions with different LOD for "Temple Boy - Cypriot Sculpture 4th century BC" 3D model.

[9] https://www.blender.org/

4.4.1.3 Metadata associated with artifacts

The models of the artifacts that have been used in the game have been generated using the process described in Section 4.4.1.1. However, besides the 3D model data, more data that accompany the models are required. These data are often called "Metadata" and their purpose is to provide descriptive and structural information that helps in understanding, managing, and organizing the primary data it refers to. Specifically, in the case of the artifacts that have been used in the game, details like category, period, style, material, dimensions, and description have been fetched. These details have been provided by BOCCF and can be found in the BOCCF online collection[10].

4.4.2 Adding New Artifacts

One of the goals of this project is to enable other museums to use the source code, structure and documentation that we provide for the game to allow them to incorporate their artifacts. Thus, the following subsections describe the process of adding new artifacts in the game.

4.4.2.1 Exhibit Manager

The Exhibit Manager is a top-level Singleton[11] script (i.e., there is one and only copy object of this manager; more are not allowed) that is attached to the "ExhibitManager" gameobject, found under the Managers gameobject. It provides other scripts with an easy and abstract way to handle exhibit objects indirectly. See Figure 21 for more details.

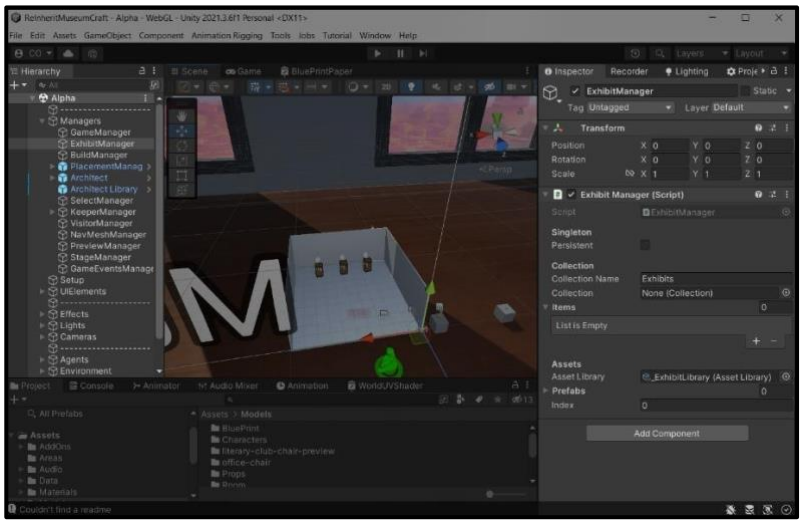


Figure 21 Exhibit Manager

When the ExhibitManager script first runs it seeks the collection object with the name "Exhibits" which contains the exhibit objects. It then grabs all the existing exhibit objects (if any) and places them in the items array. In Figure 22, it can be observed that three exhibit prefabs have already been positioned within the Exhibits collection. Once the game is

[10] https://www.boccf.org/en-gb/homepage/museums-collections2/museum-of-george-and-nefeli-giabra-pierides-collection-donated-by-clio-and-solon-triantafyllides/

[11] https://gamedevbeginner.com/singletons-in-unity-the-right-way/

executed, these prefabs will be automatically registered in the ExhibitManager, ensuring their proper functionality and inclusion within the game environment.

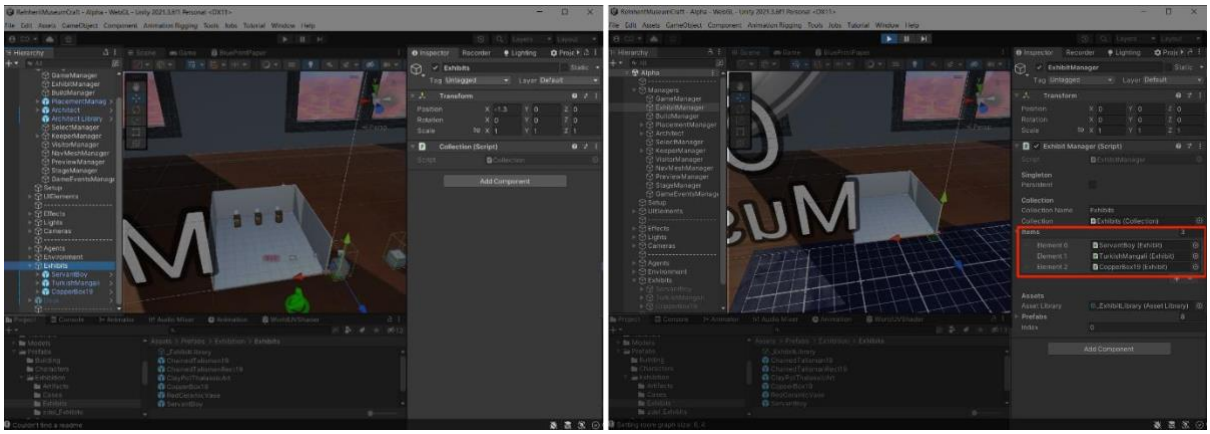


Figure 22 Automatically Registered Prefabs

Another noteworthy feature of the ExhibitManager is that it contains a list of all artifacts in the game. This list is an AssetLibrary with the name "_C". The library object can be found in the Asset directory Prefabs/Exhibition/Exhibits. The underscore in the name of the file is simply to place that file at the very top. After selecting it, the Inspector window will show all the objects that are included in that Library. Ideally, the list will update itself in case more exhibits are added. However, if the update does not happen automatically, users can click the "Refresh" checkbox to trigger the update. Screenshots of the process can be found in Figure 23. The ExhibitManager keeps an internal index, which it uses to decide the next Artifact to be gifted to the player after certain events.

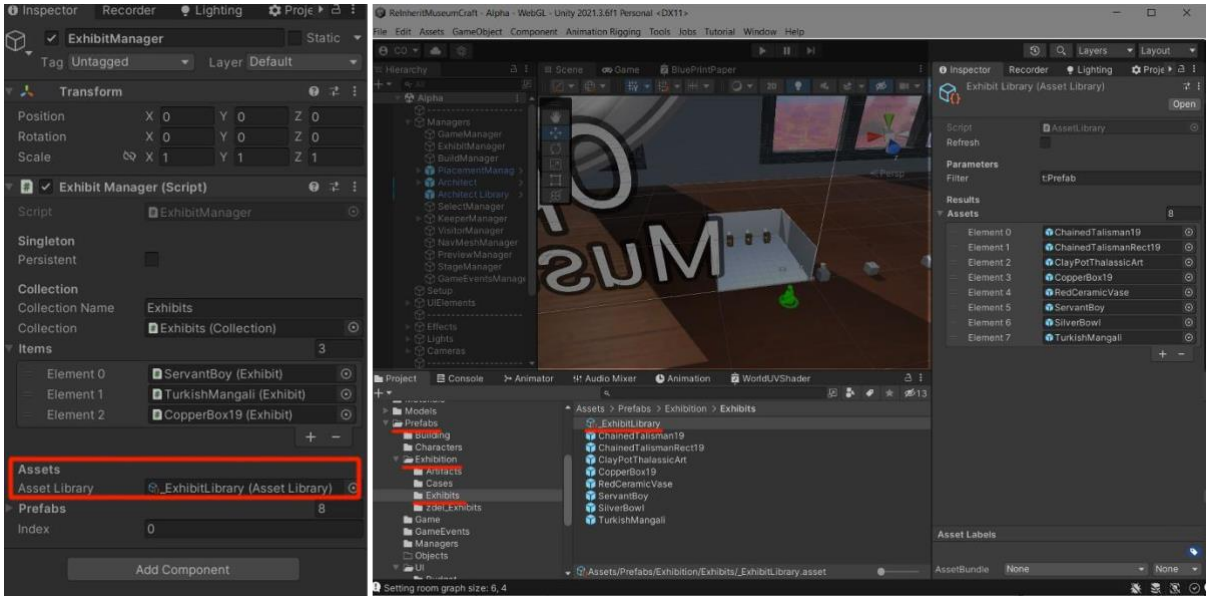


Figure 23 Exhibit Library

4.4.2.2 Exhibit Inventory

While the Exhibits collection object contains all the exhibit objects, there is also a need for a way to visualize that collection to the player. To fulfill this purpose, the ExhibitInventory script is utilized. The game object containing the script can be located in the Scene Hierarchy under

UIElements > GameUI > WhenClosed > ModeSelectionUI > CurateModeUI. The script requires a direct reference to the Exhibits collection to add the thumbnails of each exhibit inside the inventory box during Curation mode. An example is shown in Figure 24 below.

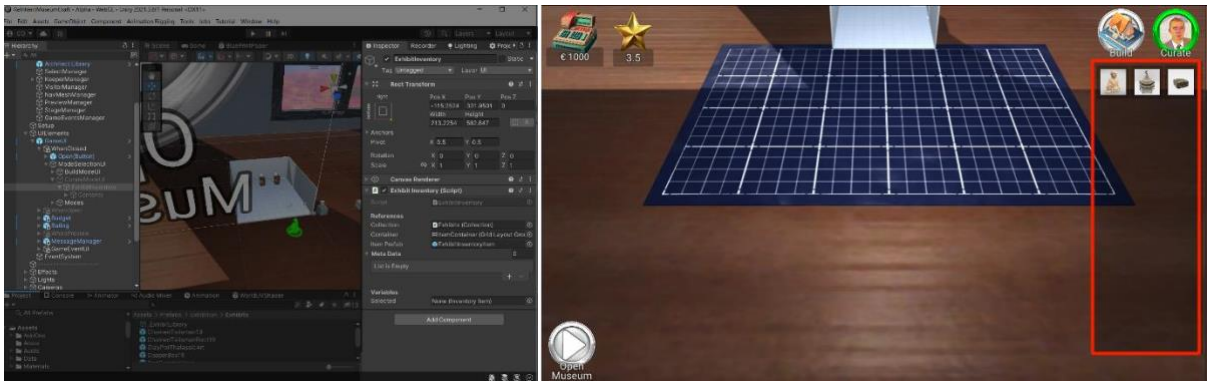


Figure 24 Exhibit Inventory – Visualize Collection to the Player.

4.4.3 Preparing and Importing a Photogrammetry Model

This is a short guide that assists users in importing photogrammetry or laser-scanned models into Unity. Numerous approaches can be employed for this purpose, but the guide focuses on using the free and open-source modeling software Blender[12]. It assumes that the user has already acquired the software and also has a basic understanding of operating the tool.

4.4.3.1 Importing

Once the physical object has been scanned, it needs to be processed using software such as MeshLab[13] (for editing and smoothing meshes) and exported in a suitable format like OBJ. Blender comes with a default OBJ (Wavefront) importer. Alternatively, users can consider using the FBX format. In case the desired format is not supported, there are various Blender plugins available online that can facilitate the import process. Lastly, it is important to remember to delete the default cube in Blender. Screenshots are shown below.

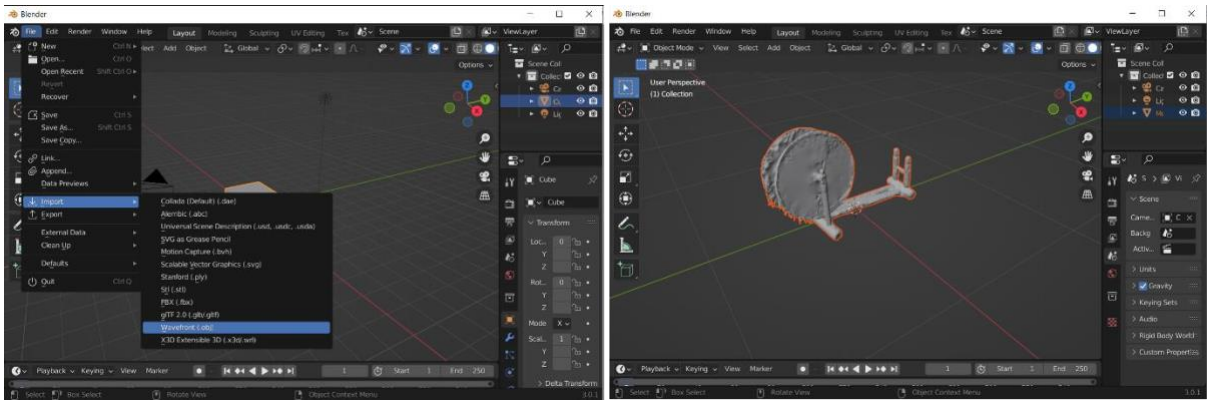


Figure 25 Import OBJ file In Blender

[12] https://www.blender.org/
[13] https://www.meshlab.net/

4.4.3.2 Modify Original Diffuse Texture

Often, the scanned model will have a large color (diffuse) texture attached to it. The size of the texture allows for a lot of scanned details to be preserved in order to make the scanned geometry more detailed. Unfortunately, large textures may cause problems in-game. These are the most notable problems that should be kept in mind:

- The size of the image increases exponentially with the dimensions. This means that by doubling the width and height of the image, the size is effectively quadrupled. This will have an impact on the size of a project and compression at a later stage may mitigate the problem to an extent.
- Larger textures impact loading times, which is especially important for web-based games.
- Many older or low-end graphics cards are unable to handle large textures.

It is worth noting that compression (like using JPEG) will mitigate the file size issue despite the enormous size. However, the image will still take a larger size in memory once the game is loaded and will require significantly more time to decompress the file on startup. Therefore, a recommendation is made to duplicate the image file and subsequently scale it down to a more reasonable size, such as 1024x1024 pixels. Additionally, it is advised to assign a descriptive and meaningful name to the duplicated texture, preferably preceded by an underscore and followed by the letter D (indicating Diffuse). Users may also choose to convert the texture to a PNG format instead of JPG if they prefer. It is essential to refrain from deleting the original texture, particularly if there are no existing backups of the original data, to prevent potential data loss. Then, the modified texture will need to be assigned to the desired object. A demo of the process is presented below.

Figure 26 Duplicate, Convert and Resize a Texture.

4.4.3.3 Assign Modified Texture

The next step involves duplicating the original model as a precautionary measure in case any mistakes are made during the process and the user needs to revert. To facilitate easy recognition from the Scene Hierarchy, the objects are renamed accordingly. The letter combination "OG" is assigned to the original model, while "HD" (High Definition) is used for the duplicate, which will be utilized to create the additional levels of detail. Optionally, the user can choose to hide the original model (Figure 27).

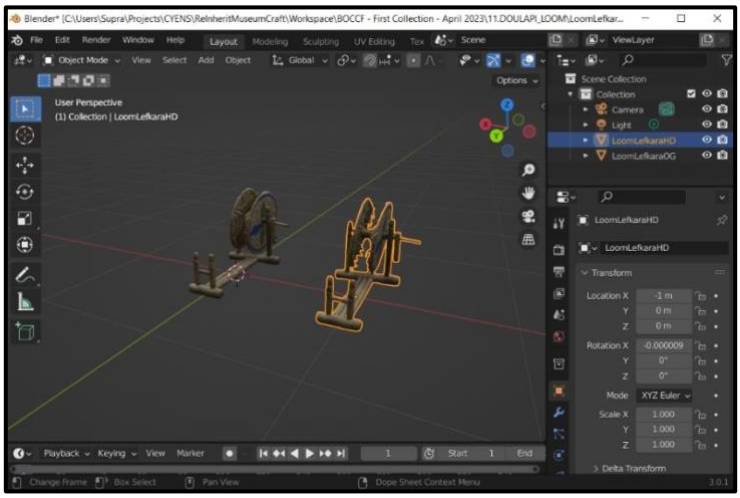


Figure 27 Duplicate and Rename Object

The next step involves creating and assigning a new material to the HD object, utilizing the scaled texture instead of the original one. This approach ensures that the model utilizes the "optimized" texture, resulting in improved efficiency and performance. The overall process can be seen in Figure 28.

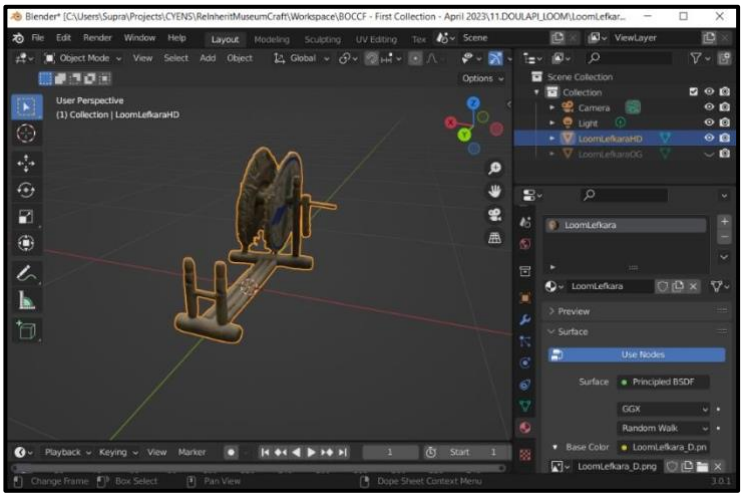


Figure 18 Assigned "Optimized" Texture

4.4.3.4 Geometry Optimization

Although it is possible to directly export the HD model and import it into Unity, doing so may lead to performance issues, especially if the same geometry as the scanned model is used. Scanned models often possess an exceptionally high number of triangles, which can place a heavy burden on low-end graphics cards, especially when multiple artifacts are simultaneously displayed within the game.

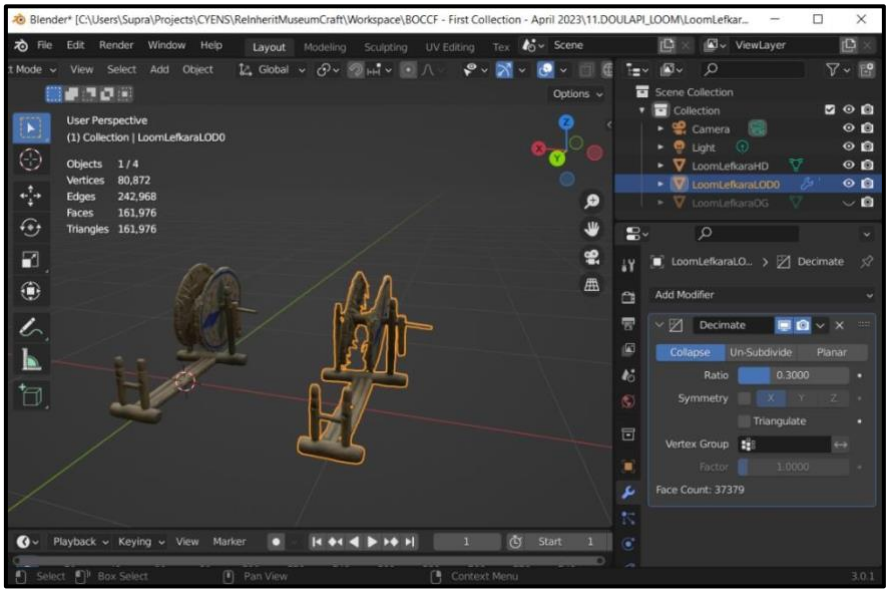


Figure 29 Duplicate And Decimate Modifier

4.4.4 Generating/Applying Textures for building blocks

Textures are digital images or patterns that are mapped onto the surfaces of 3D models to give them color, detail, and visual depth. They can represent a variety of surfaces such as skin, metal, fabric, or terrain, and are often combined with additional layers like normal maps or specular maps to achieve effects like bumps, reflections, or shadows. Usually, these textures are created by artists as they have to follow a specific art style to match the context of the game. This subsection is focused on the generation of textures that have been used in the video game and specifically building blocks including walls, floors, and scene objects.

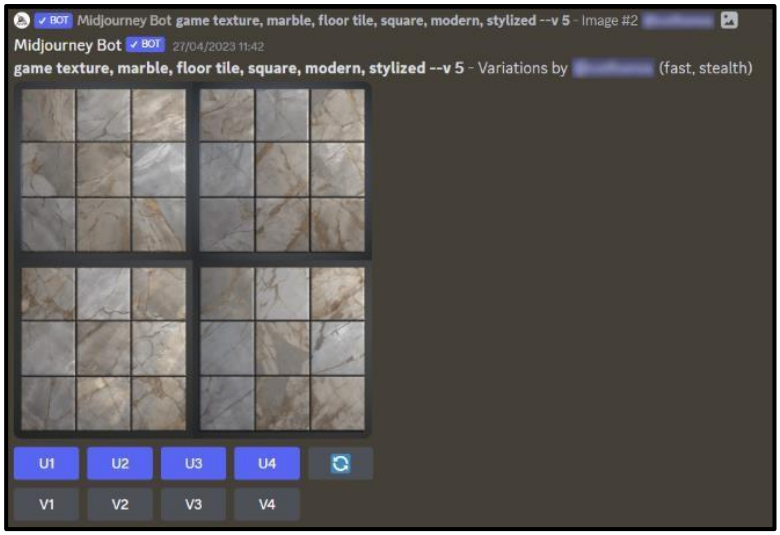


Figure 30 An example of generated images to be used as floor texture.

The game development team is equipped with a strong background in software engineering and programming. Thus, besides using ready-made textures, a good idea was to experiment with evolutionary technologies like generative AI. One tool that has been utilized is Midjourney[14], which can generate images based on text prompts. Photoshop[15] is a wellknown and professional editing tool with many capabilities, however other free alternative tools, like Gimp[16] and Krita[17] are suggested for photo editing. Although Midjourney is not an ideal tool and is not designed for creating textures with patterns, like floor tiles, it is able to produce remarkable results. An example of generated images that were intended to be used as floor texture, is presented in Figure 30.

While the results are satisfactory there are some issues. First, is the inability for creating seamlessly connecting textures. This can be a problem for environments that rely on a large continuous and repeating texture. Second, another issue is the lack of secondary detail maps, such as normal maps. These can be generated automatically in most cases, using tools like Materialize[18], by inferring the height and normal maps from the color of the texture. Unfortunately, many textures do not follow that assumption and will generate incorrect normal maps. Thus, for the final version of the game, the floor texture has been downloaded from PolyHaven[19], which is an online library that provides free textures with public domain licensing.

After the desired texture has been found, it can be applied as floor or wall material using the following process. The approach that has been followed for the building of the museum is that several instances of a prefab room, with a predefined size, are connected together. Thus, changing the building appearance is just a matter of modifying the prefab object of the room. The prefab room is located under “Assets/Prefabs/Building” directory with filename “Room”. The hierarchy and the 3D model of the room are shown in the figure 31 below.

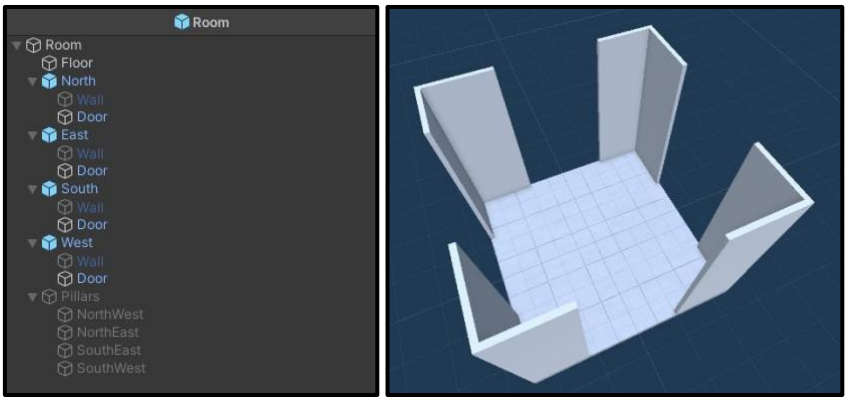


Figure 31 Hierarchy and 3D model of the room prefab.

[14] https://www.midjourney.com/
[15] https://www.adobe.com/ie/products/photoshop.html
[16] https://www.gimp.org/
[17] https://krita.org/
[18] https://boundingboxsoftware.com/materialize/
[19] https://polyhaven.com/

First, the appearance of the floor can be altered by assigning a new material to the “Floor” gameobject. Second, the texture of the walls can be changed by assigning a new material to all “Wall” and “Door” gameobjects in the prefab hierarchy.

4.4.5 Generating Sprites and Icons for the User Interface

The user interface of the game requires a lot of 2D art. This is used for icons, buttons, and other images that visually guide the player through the game tasks. As mentioned in Section 4.4.4, the use of generative AI was the best choice for creating these types of graphics, thus Midjourney has been used for this task too. Below is an example of prompts used to generate these icons and sprites. Midjourney uses these prompts to generate 4 variants that match user’s input. The keywords used are important and will drastically affect the results. For instance, as presented in Figure 32, labels such as “clip art”, “game sprite”, “game art”, “isometric” has been included in the text query to get the intended look and feel.

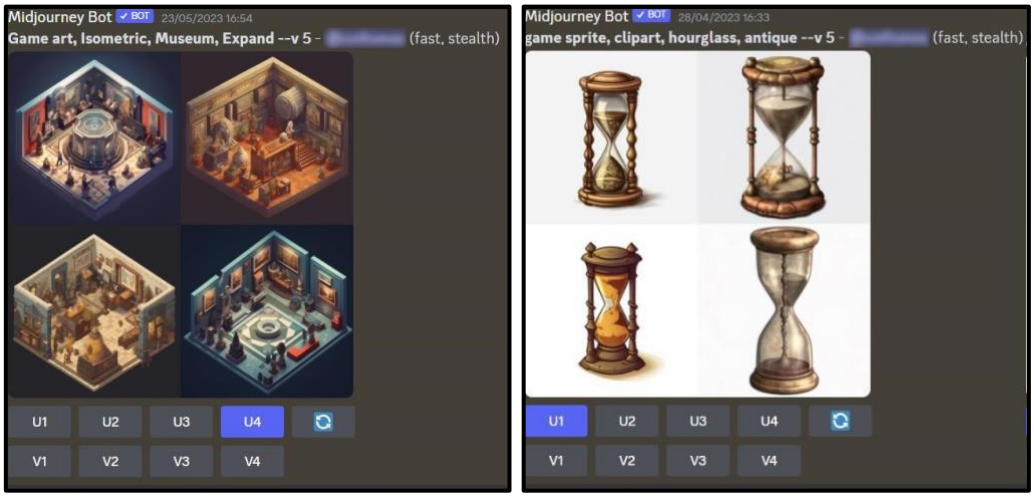


Figure 32 An example of generated images to be used as sprites and icons for UI.

4.5 Character Navigation

In video games character navigation refers to the mechanism that enables game characters, or agents, to move and navigate through the environment. Depending on the game's genre, this may involve walking, running, jumping, climbing, swimming, driving, or flying, among other forms of movement. In the case of the ReInHerit video game, character navigation includes mostly walking and the process and tools that have been used to achieve that are described in the following subsections.

4.5.1 Character Navigation

Unity offers an AI package with features that enable game agents to navigate with intelligence. The package offers a solution to two distinct navigational challenges. First, how to calculate a path to the desired position and second, how to get there. The first problem is global and static because the agent must consider a significant portion of the scene to choose the best path, whereas the second problem is local and dynamic since the agent must be concerned with local collision avoidance. The Unity Navigation System is based on three key components,

"NavMesh Surface", "NavMesh Agent", and "NavMesh Obstacle", while also other auxiliary components are available including "NavMeshModifier" and "NavMeshLink".

All available components are visualized in Figure 33(image taken from the official Unity manual[20]).

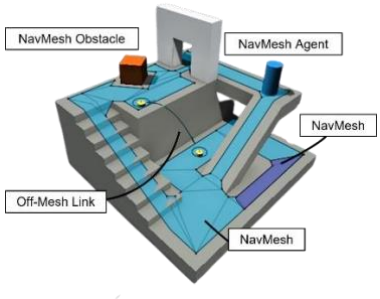


Figure 33 All Components of Unity's AI Navigation package.

4.5.2 Navigation Mesh (NavMesh) Components

As mentioned above, that Unity Navigation System provides a list of individual components that can be combined to achieve the desired results. This section describes each individual component and their intended use.

NavMesh Surface: defines a walkable surface that an agent can use to move around the environment. A NavMesh can be created by baking different surface areas, each of which can have a variable area cost. Lower values indicate that the area is easier to walk over. The area cost of the surface specifies how difficult it is to walk across a certain area. The agents specifically employ the A* Algorithm to determine the shortest route to the target. A graph of interconnected nodes has to be created in order to apply A* in an environment; this is done by placing a point on each polygon of the mesh. Thus, for the A* algorithm, the aforementioned area costs are used as connection costs. The cost to move between two nodes is calculate as $distance * linkCost$.[21]

NavMesh Agent: The agents that need to traverse NavMesh surfaces are outfitted with the NavMesh Agent component. The component can be utilized to describe a list of parameters like, agent's radius, height, speed, angular speed, and acceleration.[22] Likewise, by using this component, the agent is characterized as a moving object that is able to avoid collisions by utilizing Reciprocal Velocity Obstacles (RVO) algorithm[23]; RVO algorithm is describes in the Section 4.5.3.

NavMesh Obstacle: A moving, or a static object can be designated as an obstacle using the NavMesh Obstacle component. During the simulation, agents attempt to avoid these objects, but they can also be utilized to totally block a certain path.[24]

[20] https://docs.unity3d.com/Packages/com.unity.ai.navigation@2.0/manual/NavigationSystem.html
[21] https://docs.unity3d.com/560/Documentation/Manual/class-NavMeshSurface.html
[22] https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html
[23] https://ieeexplore.ieee.org/document/4543489
[24] https://docs.unity3d.com/Packages/com.unity.ai.navigation@2.0/manual/AboutObstacles.html

NavMesh Link: This component is able to link two navigation surfaces and create an alternative path for a specific type of agent.

Figure 34 Higher surface cost around artifacts

NavMesh Modifier: As discussed earlier, the surface of navigation mesh contains areas with traversal costs. Using this component, the user can define an area using a bounding box and manually set its cost.[25] This could be helpful in case there is the need of having areas with various surface types; for instance, higher cost can be set when agents walk over mud. This component has been utilized in the game for assigning a higher traversal cost in a radius around the artifacts, with the aim to discourage visitors from passing in front of other visitors observing an artifact.

4.5.3 Collision avoidance using Reciprocal Velocity Obstacles (RVO)

Characters will be moving in the environment – one of the fundamental actions that they need to be doing is collision avoidance between them. Several algorithms have been introduced in the literature; one of the most popular is the Reciprocal Velocity Obstacles (RVO) approach that has been introduced in Robotics and is widely used in video games with multiple characters. This is the algorithm that is being used for the developed game.

The Velocity Obstacle (VO) concept, which was first introduced for the navigation of robots in the actual world, is the basis of the Reciprocal Velocity Obstacles (RVO) algorithm, which is used for multi-agent real time navigation. The foundation of RVO is the assumption that every agent expects that every other agent uses the same decision-making procedure to avoid collisions. When one agent anticipates a collision, it only travels halfway out of the path of the collision direction and assumes that the other agent would follow the same process by moving halfway to the opposite direction. A visual illustration of the process is presented in the Figure 35 below.

[25] https://docs.unity3d.com/Packages/com.unity.ai.navigation@2.0/manual/AreasAndCosts.html

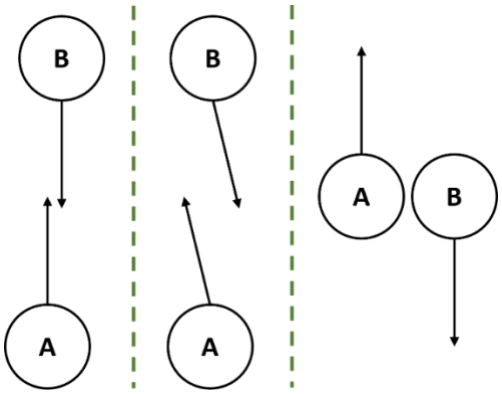


Figure 35 An example of RVO collision avoidance

4.6 Character Behavior Control

Besides the ability to navigate in an area, there is also the need for tools to control the behavior of the various agents in the game. There are different techniques and tools to define the behavior of agents, however the simple and easy-to-use approach of state machines (SM) has been used. A state machine can be defined as a computational model that can be in exactly one of a finite number of states at any given time. The SM can change from one state to another in response to some external inputs and the change from one state to another is called a transition. There are two main characters in the game, Keepers, and Visitors where their behaviors are presented in the following subsections.

4.6.1 Keeper

A keeper is responsible for moving and placing artifacts in the museum. After some tests, it has been decided to use multiple keepers in order to avoid long waiting times when multiple tasks have to be completed.

4.6.1.1 AI

First, a keeper manager keeps track of all the tasks that the keepers need to perform. When a new task is created, the next available keeper is assigned to move the artifact. When a keeper needs to carry an artifact, these steps are followed. First, collect it from the entrance point and start moving towards the placing position. Second, select the nearest standing point around the placing position and set it as destination. Third, when the keeper arrives at its standing point, rotate towards the placing position, and place the artifact. Finally, the keeper returns to the spawn position and disappears. This process combines a simple state machine and the navigation system described above.

4.6.2 Visitor

Visitors' main tasks are to give life to the museum that the player builds while also acting as an evaluation mechanism for the progression of the player. Thus, the basic actions of visitors include moving between artifacts, spending some time on each one and interacting with other visitors.

4.6.2.1 AI

The behavior of a visitor is more complex than the keeper's behavior. However, a similar approach has been followed, which is based on state machines. Specifically, a Unity's utility named "StateMachineBehaviour" has been used[26]. This utility is based on Unity's Animation Controller. Basically, the utility provides the ability for assigning scripts to each state of the animation controller, and besides defining the animation played in current state, is able to set the behavior of the agent through code. Likewise, each state can have a list of end conditions and as many transitions required between other states. Specifically, each script contains three key functions "OnStateEnter", "OnStateUpdate" and "OnStateExit", which contain the code for when entering, during and exiting the current state respectively. For the need of the game, for controlling the behavior of the visitors, they have been generated four states, "Idle", "Observe", "Walk" and "Talk-Blend". The structure of the animation controller that has been used is shown in Figure 36 below.

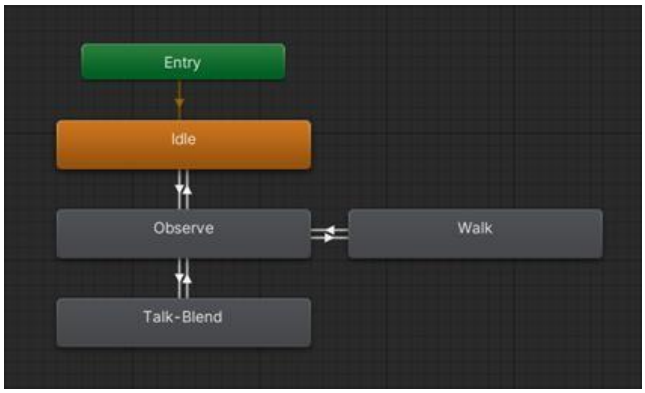


Figure 36 Behavior Controller for visitor agents

First, a visitor is in "Idle" state if not currently observing an artifact nor walking towards a destination. Thus, this state is basically used for initialization and then visitor enters "Observe" state. In this state, the visitor can follow one of the following three branches:

- Stay stationary and observe an artifact for a period of time.
- Communicate with another visitor while observing an artifact.
- Select a different artifact and navigate towards it.

When the museum is in open state, a manager has been used, that is responsible for spawning visitors in predefined positions around all artifacts placed in the museum; this is the starting state of the visitors. Then, based on the controller described above, visitors move inside the museum and execute their behavior. Finally, a visitor is moving towards the exit position if all available artifacts have been observed. In case the museum goes back to closed state, all visitors disappear.

[26] https://docs.unity3d.com/ScriptReference/StateMachineBehaviour.html

4.6.2.2 Evaluation of experience

The behavior of the visitors has been described in Section 4.6.2.1. While visitors are navigating, they observe the various artifacts that are placed inside the museum. During that process, visitors may decide to start a conversation with other visitors and present any kind of emotions. The available emotional states are “Neutral”, “Happy”, “Angry”, “Disgusted”, and “Bored”. For every emotional state, an emoji icon has been selected and any time a visitor needs to express one of these emotions, the appropriate emoji appears inside a bubble above the visitor (Figure 37). Thus, this is an easy, fun, and intuitive way of giving the player the sense that there is life inside the museum, while also acting as an indication of how well the player progresses through the game.

Figure 37 Visitors observing the artifacts while expressing their emotions.

4.7 Character Animation

Character animation for games involves creating lifelike, dynamic movements for game characters through a blend of artistry and technology, using techniques like rigging, motion capture, and keyframe animation. It's a crucial aspect of game development that helps in enhancing the player's immersion, storytelling, and overall gaming experience.

4.7.1 Character Control

To control the movement of the characters, such as the keeper and the visitors, the “Third Person Controller” from Unity Starter Assets is used. Starter Assets provide an excellent base for various projects, promoting flexibility and ease of use. They offer a plug-and-play solution for character movement and behavior, significantly reducing development time and effort. Third Person Controller comes with a complete armature character controller that is fully rigged and animated, using a Humanoid rig.[27]

[27] https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-urp-196526
[30] https://docs.unity3d.com/Manual/ConfiguringtheAvatar.html

4.7.2 Humanoid Avatars

A Humanoid model is a distinct structure that consists of at least 15 bones that set which parts of the model correspond to the legs, arms, head, and body. They are arranged in a manner that roughly mirrors the structure of a real human skeleton. Due to the similarities in skeletal structures among different humanoid characters, animations can be transferred from one humanoid character to another, thereby enabling features like retargeting and inverse kinematics (IK) (Section 4.7.3).

A detailed description of humanoid animations is provided in Unity's manual[30]. On the left side of Figure 38 showcases the skeleton hierarchy of an example Mixamo [28] character, offering a structured view of how each bone relates to the others. Mixamo tool is presented later in Section 4.7.2.2. The middle portion of the figure provides an internal view of the character, which shows how the skeleton appears inside the mesh. This visual insight helps understand how the bones move in relation to the character's outward appearance. Finally, the right side of the figure displays the mapping interface where each bone of the character is assigned or 'mapped'. Here, specific bones of the character model are linked to the corresponding bones in the reference humanoid model, facilitating accurate retargeting of animations.

By using this approach, has been achieved an effective retargeting of animations from Unity's Third Person Controller Character to their custom character model, including all the preset animations - walking, running, jumping, and any other movements. The retargeting process using humanoid models in conjunction with the Third Person Controller saved significant amount of work in developing animations.

Figure 38 Humanoid Avatars

4.7.2.1 SketchFab

Sketchfab[29] is an online platform that allows users to upload, share, and discover 3D models. It is often referred to as the "YouTube of 3D" because it provides an accessible solution for viewing and sharing 3D content on the web. It supports a multitude of file formats, allowing users to upload their work from a variety of 3D modelling tools. Once uploaded, these models

[28] https://www.mixamo.com
[29] https://sketchfab.com/

can be viewed in 3D directly in the browser without needing any special plugins. The platform provides a range of opportunities for its users. Artists can showcase their portfolio, game developers can find assets for their projects, educators can share interactive 3D content, and businesses can present products in a unique and interactive way. The free and open-source character that has been used in the game is shown in Figure 39.

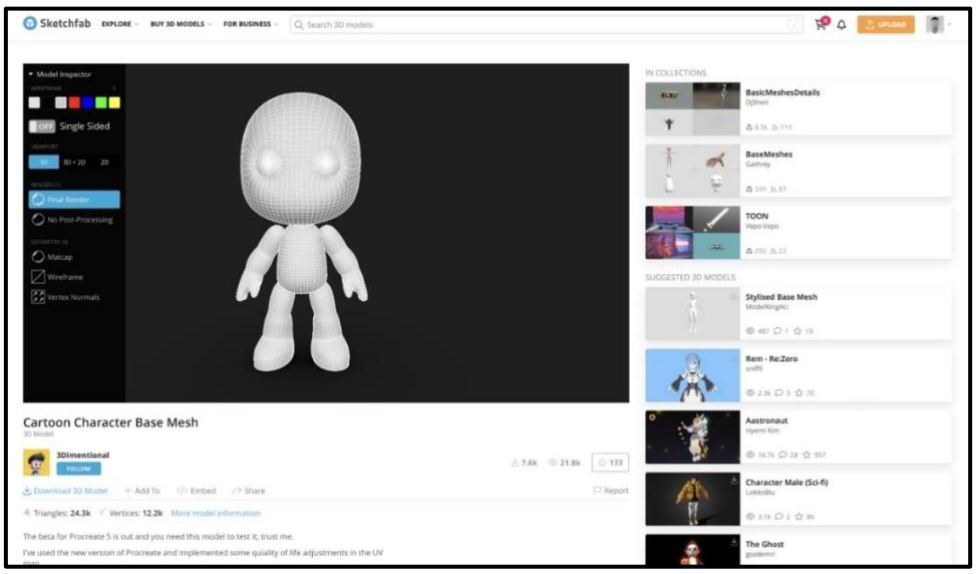


Figure 39 The Sketchfab Character used in the game. Available at: https://skfb.ly/oqqJK

4.7.2.2 Mixamo

Mixamo is an online free platform developed by Adobe that enables users to easily animate 3D characters. It offers a wide selection of high-quality, ready-to-use 3D character animations, which can be customized to suit specific requirements. This platform has been used to rig a custom character that has been used in the game.

As depicted in Figure 40 (Left part), users can select from a library of pre-existing 3D characters available on the platform. This vast catalog offers a variety of characters, encompassing different styles, forms, and complexities. It caters to various genres and settings, providing a robust starting point for many projects. Once a character is chosen, users can apply a variety of animations to the model, such as walking, running, jumping, and many other actions as shown in the same Figure 40 (Right part). These animations can be tweaked to create custom movements.

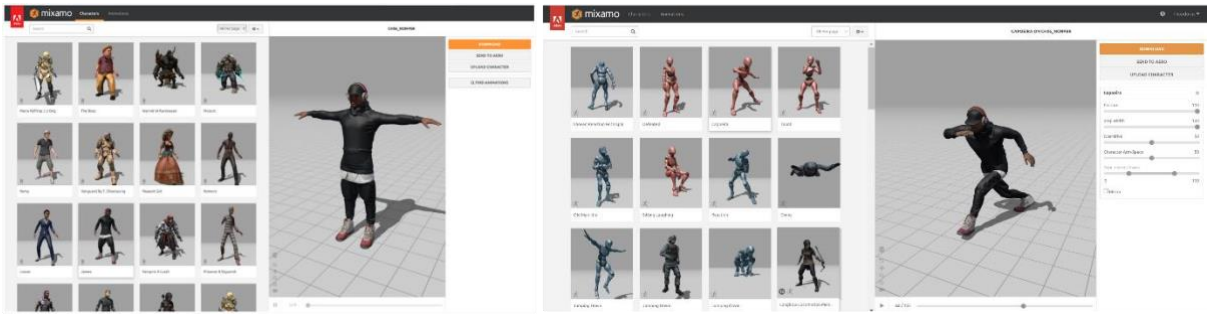


Figure 40 Mixamo Characters and Animations Library

Alternatively, users can upload their own character models. This feature allows for a high degree of personalization and flexibility, as users can animate unique characters designed specifically for their projects. Once uploaded, these custom models can be rigged and animated using the interface of the Mixamo platform by applying any of the available motions. As presented in Figure 41, the character that has been used in the game has been uploaded and rigged (Left part of the figure), in order to test a list of various animations, like a kick (Right part of the figure).

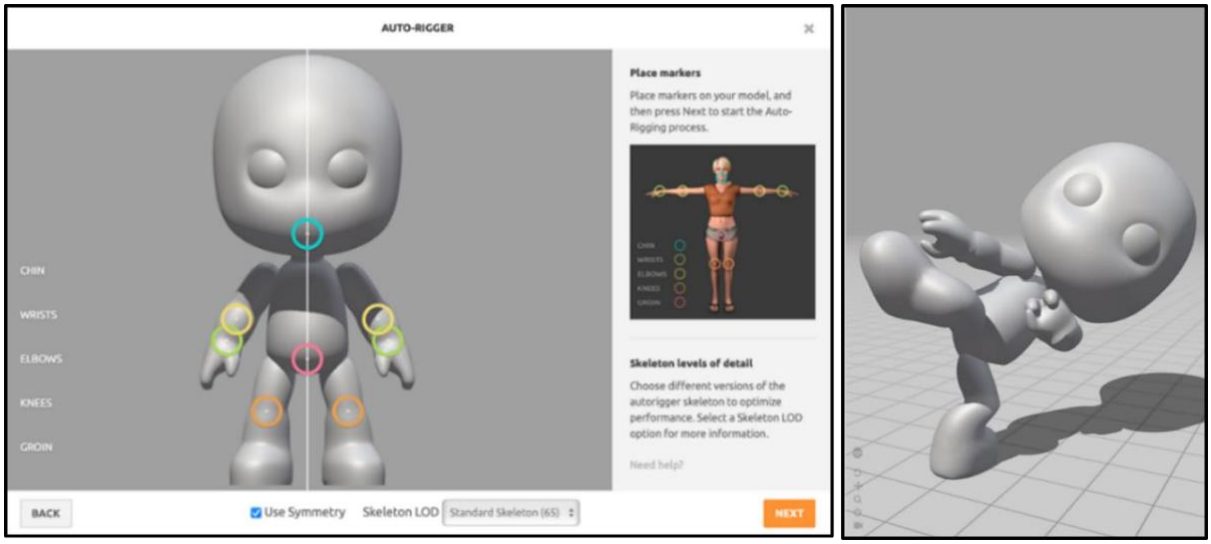


Figure 41 Mixamo Applying animations to custom character.

4.7.3 Inverse Kinematics (IK)

Inverse Kinematics (IK)[30] is a technique in computer graphics and animation to calculate how a character or object should move. It is particularly useful for creating realistic movements in characters' limbs, like hands and feet with not a lot of effort. The term "Inverse Kinematics" denotes a problem-solving methodology that involves reversing the calculation process utilized in Forward Kinematics (FK) (hence the name "Inverse"). In FK, the analysis commences from a designated "root" point, such as the shoulder of a character, and subsequently determines the position of the "end effector" such as the wrist of the hand.

[30] https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13310

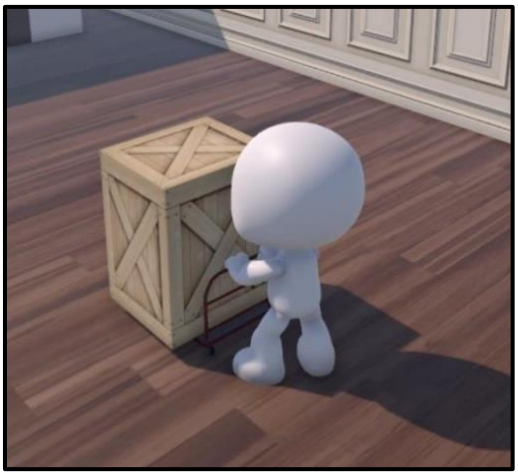


Figure 42 Inverse IK applied on the keeper

On the contrary, in the case of IK a target position is defined for the end effect, such as placing a character's hand on a doorknob or foot on a stair, and it is the responsibility of IK to compute the angles and positions of the joints and limbs necessary to attain the desired configuration. In Unity, Inverse Kinematics[31] tools can be used to create more natural motion paths for animation. For example, if a game developer wants a character to pick up an object, the developer can just specify the target position (the position of the object), and the IK system will calculate how the arm should move to reach it.

Regarding the development of the game, IK method has been utilized to achieve precise positioning of the keeper's hands on the trolley during walking. This method eliminated the requirement for additional animations beyond the existing 'walking' animation, ensuring the hands' accurate placement. The effect of applying IK on our character is illustrated in Figure 42.

4.8 Game Events

Game events are special occurrences in a game's narrative or gameplay that are designed to provide unique experiences, challenges, or rewards to players. The following subsections present the various game events that have been generated and briefly describe how they work.

4.8.1 Game Event Manager

At the end of each game round, the “GameEventsManager” script is responsible to handle the events that appear to the players. This script is attached to the gameobject which is located at Managers > GameEventsManager. The creation of seven event categories is presented in Figure 43.

[31] https://docs.unity3d.com/Manual/InverseKinematics.html

The various categories of events:

1. Artifact: Relevant Authority (e.g. Department of Antiquities, Deputy Ministry of Culture) gives the museum a new artifact to display.
2. Award: Either a grant, a museum award, or a donation from a generous donor.
3. Damage: Some kind of damage has occurred, and the player needs to take action to fix it.
4. Expand: The Office of city planning allows you to expand the space of your museum.
5. Nothing: Nothing happens during these types of events.
6. Quiz: Quizzes players on their knowledge, to distinguish between the fake and real artifacts.
7. Review: A reviewer gives the player a score on social media. This can boost the museum's popularity.

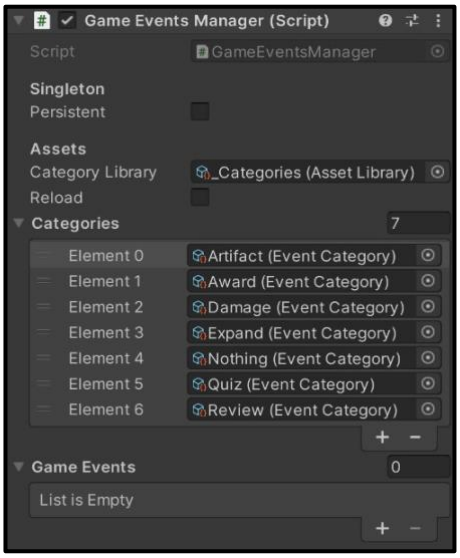


Figure 43 Event Categories

Furthermore, when a round ends, the "OnRoundEnd" method is called, which evaluates, and updates conditions based on the current round's game event. The process initializes a shuffled list of game events that will occur at the end of each round. An example initialization of events is presented below.

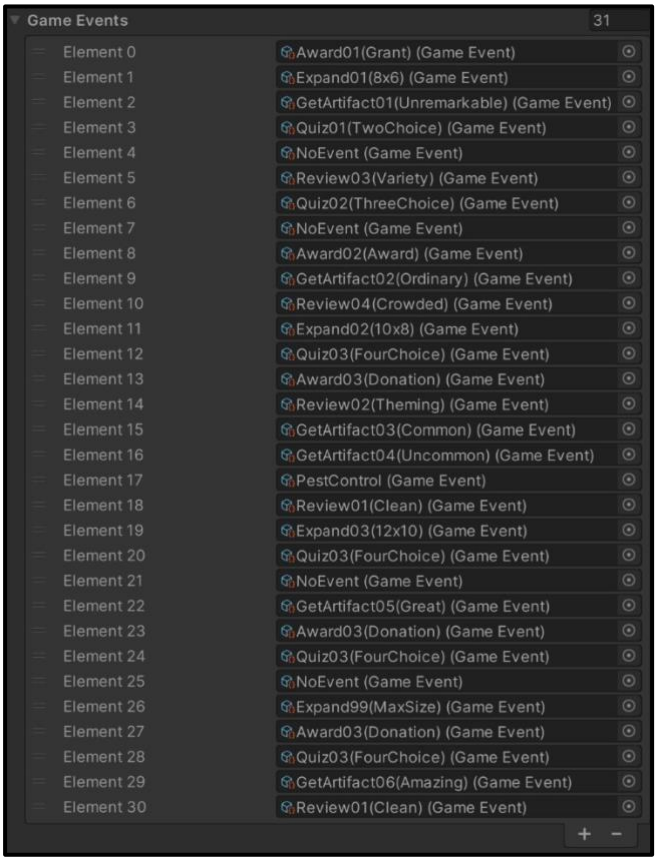
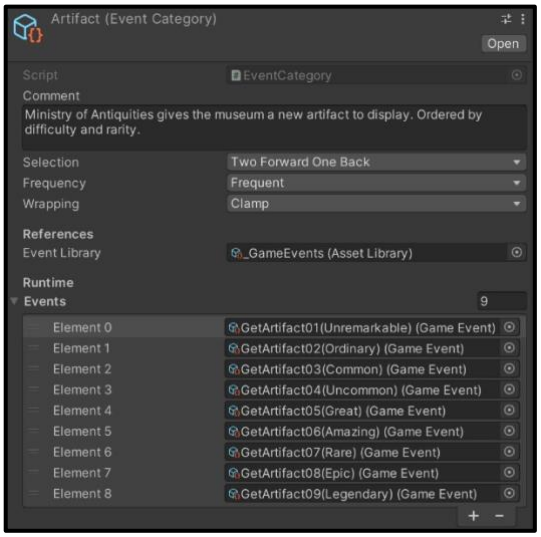


Figure 44 Shuffled List of Game Events.

The rest of this subsection contains a detailed description for each subcategory event that can occur in the game.

Event Categories:

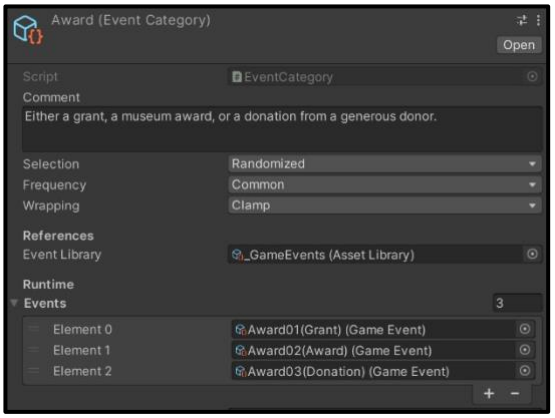
1. Artifact: gives the ability to the player to acquire a new artifact. A list of sub-categories has been created and ordered by difficulty and rarity.



   In order to acquire:

   a. Unremarkable Artifact: a minimum rating of 0
   b. Ordinary Artifact: a minimum rating of 0.5
   c. Common Artifact: a minimum rating of 1
   d. Uncommon Artifact: a minimum rating of 1.5
   e. Great Artifact: a minimum rating of 2.5
   f. Amazing Artifact: a minimum rating of 3
   g. Rare Artifact: a minimum rating of 3.5
   h. Epic Artifact: a minimum rating of 4
   i. Legendary Artifact: a minimum rating of 4.5

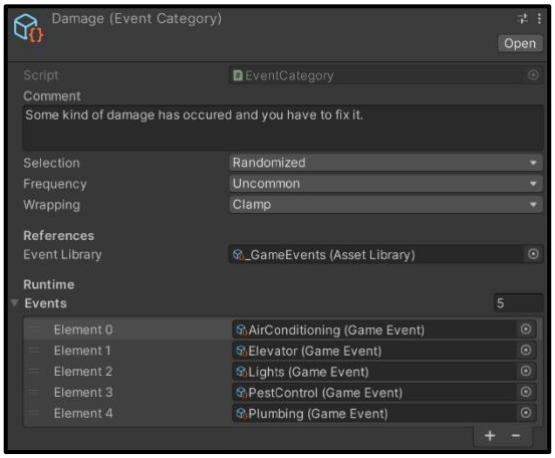
2. Award: the player gets one of the three available rewards based on performance.



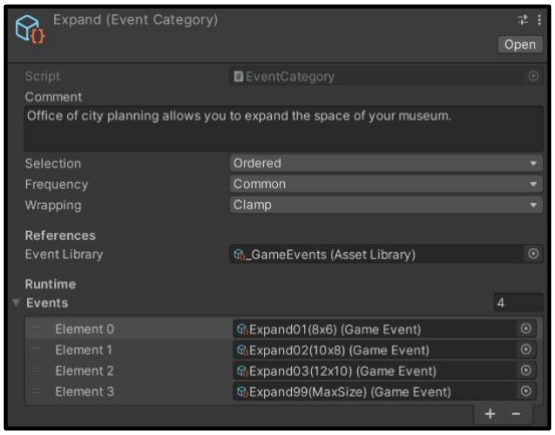
   In order to acquire:

   a. Award01 (Grant): a minimum rating of 4
   b. Award02 (Competition): a minimum rating of 4.5
   c. Award03 (Donation): a minimum rating of 3.5

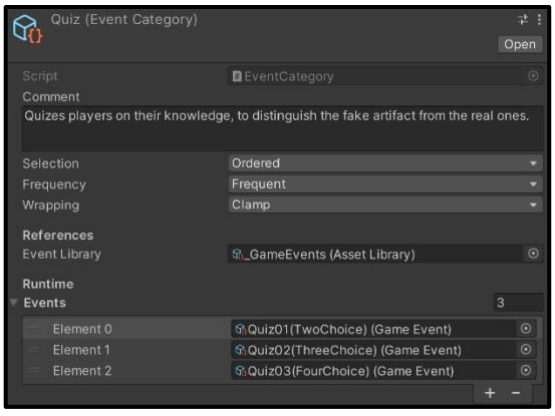
3. Damage: 5 sub-categories have been created. The minimum amount of funding that is required to succeed those events is 300. Otherwise, a penalty will be taken. a. Air-conditioning
   b. Elevator
   c. Lights
   d. Pest Control
   e. Plumbing



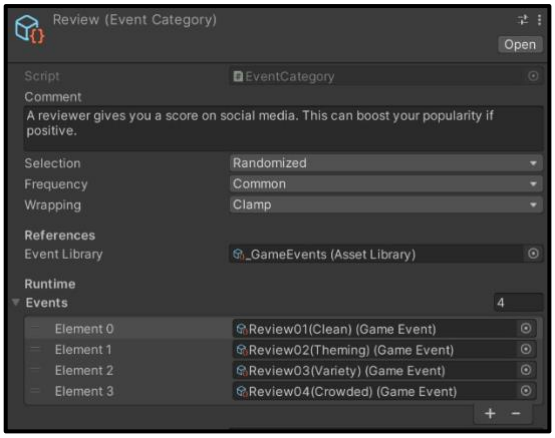
4. Expand: 4 sub-categories have been used that do not have any requirements to complete.
   a. Expand area size to 8x6.
   b. Expand area size to 10x8.
   c. Expand area size to 12x10.
   d. Expand area size to max size.



5. Quiz: 3 sub-categories with different difficulties have been utilized. The player needs to find the original artifact in a quiz game to earn it. This category is discussed extensively at a later stage in Section 4.9. a. Two choices Quiz
   b. Three choices Quiz
   c. Four choices Quiz



6. Review: 4 sub-categories have been generated.
    a. Clean: Need a minimum level of cleanness 0.75/1
    b. Theming: Need a minimum level of theming 0.75/1
    c. Variety: Need a minimum level of variety of artifacts 0.75/1
    d. Crowded: Need a minimum level of spacing 0.75/1



These events have been implemented as part of the project. Future users that wish to create their own event categories, they can follow the procedure of right-clicking on the Project Window and selecting the options: Create > Data > GameEvent. Subsequently, they can proceed to configure the event according to their specific requirements, including the ability to establish subcategories and create prefabs for desired actions. The created events can then be easily incorporated by dragging and dropping them onto the GameEventsManager component.

4.9 Artifacts Quiz

As presented in Section 4.8, the game includes various event categories. Regarding the quiz category, where the player needs to distinguish between fake and real artifacts, it has been decided that it would be interesting to include a dedicated section to further discuss the process and technologies that have been utilized.

At some point in the game, the players receive a notification that a crate containing ancient artifacts has been delivered to them by the relevant authority (Figure 45). The players are warned that the contents of the crate might include not only genuine artifacts but also replicas or forgeries. The players' task is to carefully examine the items and identify the authentic artifact with the aim of avoiding mistakenly displaying a fake piece in their museum. In order to accomplish this, the players may need to utilize their knowledge in history, art, and archaeology or rely on hints and tools provided within the game. Successfully identifying the genuine artifact will result in the relevant authority granting the players the ownership of the ancient item, which they can then proudly showcase in their museum, enhancing its prestige and reputation.

However, if the players are unable to determine the authenticity of the artifact and select a counterfeit, they will not receive the genuine item. This may impact the museum's credibility and the players' progress within the game. The challenge of identifying the real artifact among the fakes adds an element of intrigue and suspense to the gameplay, encouraging players to hone their critical thinking and observational skills as they strive to improve their museum's collection.

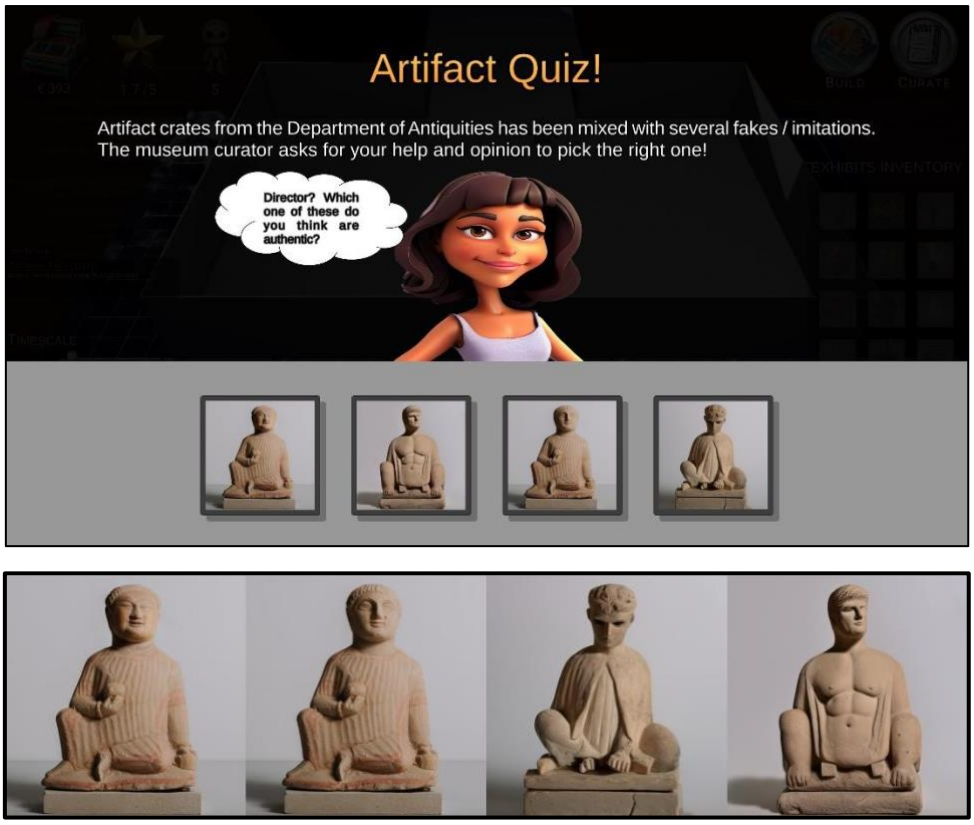


Figure 45 Example of the quiz game. Only one of these images (the 2nd) is a photo of the real artifact – the others were modified using Generative AI (Stable Diffusion) to be of Asian style (1st), like a super-hero (3rd) and like a famous politician (4th). The purpose is for the users to be able to identify the original and to only allow original pieces to be hosted in the museum.

4.9.1 Technologies Used

Generative AI techniques have been leveraged, specifically Stable Diffusion[32] and Midjourney, to edit and generate images. The avatar / narrator of the game has been generated using a Generative AI tool and appears in various events during gameplay. Additionally, three out of the four artifact images displayed at the bottom part of Figure 46, have been edited using the inpainting method derived from Stable Diffusion. The following paragraphs provide explanations for both methods.

1. Midjourney: Midjourney was initially mentioned in Section 4.4.4. This tool has been utilized to create various images and icons used in various parts of the game. Moreover, the program allows users to create new images that build upon a previously used image, ensuring a consistent visual flow throughout a series of images. An example of using this concept of building upon existing images is presented in Figure 46.

[32] https://huggingface.co/spaces/stabilityai/stable-diffusion

Figure 46 Midjourney – Creating New Images That Build Upon a Previously Used Image.

2. Stable Diffusion: is a similar tool like Midjourney, however provides the functionality of "inpainting", which is described below in this section. A user can create images from text prompts as shown in Figure 47. We note again that the **only use of these generated images is to generate forgeries that the users must identify**.

Figure 47 Stable Diffusion – Various text-prompts input and their image results.

As mentioned above, Stable Diffusion tool offers an additional mode called "inpainting" which allows users to edit specific areas of an image. An example of this feature is demonstrated in Figure 47. To utilize this mode, users input an image and mark the area they wish to edit. Subsequently, they provide a written description of what they want to appear in the marked area. The tool then generates a new image that incorporates the desired modifications as specified by the user.

Figure 48 Stable Diffusion – An example of inpainting feature that it can be used to modify an existing image.

Furthermore, in order to experiment with these tools, a summer camp that was organized at CYENS in 2022, was an excellent opportunity for students to engage with generative AI technologies. The students exhibited a keen interest in the process of generating AI art. At the end of the camp, each student was able to take home a framed copy of their own creations as a memento (Figure 48).

Figure 49 Summer Camp 2022 on Video Game development and AI students at CYENS.

Having witnessed the enthusiasm and success of the students in creating AI-generated art during the summer camp, the team became convinced that integrating such AI-generated art into the ReInHerit game might contribute in sparking the interest of younger students.

5 Digital Ethics

The project consortium has already established an Ethics Panel which reviews the ethics requirements annually, established since the beginning of the project. The Ethics Panel reviews the digital ethics of the deliverables as well. The questionnaire was developed with

the advice of digital ethics expert from CYENS and UNIFI to further support the digital ethics monitoring.

6 Conclusion

The game developed by ReInHerit project, under T3.5: Story based game development, led by CYENS aims to provide an interactive game utilizing cultural heritage assets providing unique experiences to different users (general public, cultural heritage sector professionals).

The current document aims to deliver to all interested cultural heritage organizations, detailed guidelines on how to develop and deliver a similar game to further support their activities or engage with the public.