README

Background information

This game resulted in a group project at the University of Cologne. Each member had certain tasks to fulfill, in order to build the game. The group contained five people, two from the archaeological branch, three from the computer science branch. The purpose of the game was to create an ethically correct game, which means it should not represent the typical archaeological game, such as Tomb Raider. It should be a gaming experience with a learning component. The focus of the group was on the Roman Empire, which has a large extent of information one can embed into a game.

Plot

The game's name is (Re)Discover and the reasoning is simply linked into the story. The game takes place in the near future, where the civilization has lost valuable knowledge of certain artifacts from the past. The character's grandfather has this information hidden in a journal that the character finds and navigates through the world with. The user is playing a female character that has no name. Her name was left out, so more people might bond with her better. The task is to find three main artifacts that are scattered throughout the gameworld. One has to gather these artifacts through RPG-elements, such as an inventory and a quest system. In order to keep the world vivid, a dialogue system is integrated, which has several replies as choices. These dialogues serve the purpose to learn more about the Roman Empire and also to be engaged in the task more, since one has to be cautious on specific information. The main character and the NPCs are MetaHumans, a creation by Unreal Engine, that hands one the opportunity to have high-poly and realistic characters, made by the MetaHuman creator.

Game structure

We know that ethical decisions and morality were a big factor in this project. Still, we have decided to leave the choice up to the user, how to play. There is an ethical level system integrated, that determines the outcome of the game. The point of the game is to gather the three main artifacts, while also gathering information about the Roman Empire through dialogues with the NPCs. There is a lot of information scattered, for instance about Roman pottery or Roman watchtowers. If a quest is accepted, the reward will be a potion, which will boost the ethical level of the user. If all three artifacts are gathered, three potions in total will be in the inventory, ready to be used. After using the third potion, the ethical level has risen to 100, triggering the ending of the game. The user can make their own ethical decisions, in order to critically evaluate the character's behavior. One can also accept quests from the black market, which will promise more money and somewhat the same potions. These potions decrease the user's ethical level, when used. Since the black market's person mentions that the potions may be faulty sometimes or boost one up tremendously, the user can't be sure if the next potion is going to be beneficial or not. In fact, none of the potions of the black market are beneficial.

After accepting a quest from the black market and bringing the artifact back to them and additionally using the potion, the good ending can't be triggered anymore. Since the potency of the potions are a constant, the ethical level will never reach 100, even if one does the rest of the quests with the archaeological institute, meaning one is left in the void with no ending in sight and is somewhat forced to proceed with the quests of the black market, not knowing what will happen after reaching an ethical level of 0. Doing quests from both the black market and the archaeological institute creates an imbalance and this is how the void is created. That's the punishment for the user for their actions. Although the choice was given, it doesn't mean one should consider it. After the user realizes they're stuck in a void, the only choice is to restart the level, which can be easily made via the pause screen. This design choice was made to empathize the desperateness a person might have after making questioning decisions and not having control over it after. In this case, the power is taken from them. After reaching an ethical level of 0, a cutscene is triggered, hearing the main character having a monologue about the consequences. After that, the user is respawned to a checkpoint and is allowed to try again. The game might have different choice paths, but is still determined when it comes to the right ending and that is the ethically correct way, which was the focus of this project.

Main Plugins

Voxel Plugin

Narrative Quest and Dialogue Editor (Quest system hasn't been used) - Thank you to Reuben Ward who provided the Plugin for free for educational purposes only.

Async Loading Screen

Replica

Game Guide

These are the steps that need to be taken, in order to have a full walkthrough in the game.

- Test out both ethic potions (black market + archaeological institute). By using them and maximizing/minimizing the ethical level, both endings will be triggered.
- Use the cheat shop to play around with the features
- Try picking all dialogue paths in order to know the whole script.
- Get attacked by the wolves
- The door's code is 950
- the key is on the grave
- The hidden artifact in the voxel level is in the hill.

Bugs

There are a few bugs, that could not be resolved:

- The digging animation is displayed, whenever R is pressed, although this should only have been possible in the Voxel level
- Sometimes, one has to press the desired key twice, in order to activate the desired action
- the MetaHumans are clipping through their clothes, no matter how much it has been adjusted
- Triggers overlap occasionally

Assets used outside of the projects bounds

Victorian house by massive-graphisme

https://sketchfab.com/3d-models/victorian-house-dbb02bf8f5ed4851b05dd97a1784a1fb

Greek House - Low Poly by MagicCGIStudios

https://sketchfab.com/3d-models/greek-house-low-poly-411367b95444425c9da2d081149c076d

Shop by Vadim Rychkov

https://sketchfab.com/3d-models/shop-a1242d77341545b59c50960395753f3a

Best graphics settings

These are the recommended graphics settings that have the best combination of good visual fidelity and performance.

DlSS is available for eligible RTX graphics cards. The use of DLSS is recommended for ultimate performance.

High-end PC

Display Resolution: Any
Framerate limit: Any
View Distance: Medium
Anti-Aliasing: Medium
Texture quality: High
Shadow quality: High (anything under high causes flickering in cinematic scenes)

Lower End-PC

Display Resolution: 1920x1080
Framerate limit: 30fps or 60fps
View Distance: Medium or Low
Anti-Aliasing: Medium or Low
Texture quality: Medium or Low
Shadow quality: Low or High

Nadjim Noori - Programming, world design, game structure

Learning Unreal Engine has been a challenge for itself, however, the knowledge about coding in general has increased and I'm grateful to call this an ability of mine now. The entire gameworlds and the game's logic have been designed by me. I haven't done it from scratch, there was a basis with brushify levels I extended on. The texture design on the landscape and the alignment of the trees should be an indicator for players, when they feel they can't keep track. Additionally, the dialogue system was embedded by me, including the whole script. Through Replica, which can be purchased for free by the Epic Games marketplace, I used generated AI voices for the dialogues and through level sequencers I created real time-cutscenes to add a change of perspective through the dialogues. I also textured the main artifacts and created an animal-ai, that is dynamic. The deers are the most advanced in the game, having a system that determines after a while, if the animal is tired from walking too much. If that's the case, the deers will take a break and sit down. The energy drainage and restorement is handled through Blueprints, which is also a big factor in the game. The BluePrint behavior of the wolves sense the user via pawn sensing and attack when on sight. A detection widget, also handled by BluePrints, is being filled up, when the wolves are aware. Performancewise, C++ would have been better, however, Blueprints are the main attraction of Unreal Engine, creating logic without writing code directly, since it's been handled through nodes. The blueprints I made create widgets, determine the ai behavior of the animals, trigger sounds and levels, create menu and pause screens, the character movements and animations, the detection behavior of the wolves in a specific scene that has an indicator, the door and box behaviors with keys and codes, the voxel digging logic and more little things, that don't need mentioning.

I used a lot of box triggers and blocking volumes, in order to lead the player to the specific area I wanted them to be in. Therefore, the game bounds are set, although the map looks much bigger. Voxel was a hard box to crack, therefore, only a tiny level is affected by it. Initially, the group had decided to find the artifacts through a mini-game system, which didn't make sense to me during my workload. For instance, having a world this dynamic, a Tic-Tac-Toe mini-game wouldn't have made sense to me, therefore I have integrated three sequences that have to be played in the main gaming world and are fully interactable with the character. Speaking of characters, it was important to me to design the MetaHumans in a diverse manner.

The main character uses the movements and animations of the Advanced Locomotion Character (ALS), which can be purchased for free in the Epic store. This is a very advanced system that contains the climbing, crouching and general movement system. Retargeting it to the main character was a huge inconvenience, since Unreal Engine itself does not have a standardized retargeting system for their own characters yet. Retargeting the character transfers all the moving components, such as the skeletons and bones, in order to have full access to a functioning character and not just a simple mesh.

The learning component about the Roman Empire was essential to me, hence why most dialogues contain crucial information about it, which also helps processing through the game. For instance, one listens about Roman pottery, not knowing this information is important to open the door with the code, since there is a number that needs to be memorized. In this playful manner, one can learn and continue with the game simultaneously.

Julia Haschke - Design

I was responsible for every 3D and 2D art for our game. I've listed the models below, you can check out the pictures and screenshots I took in the folder "Assets_Screenshots".

3D Models are made with Blender. The three main artifacts are mostly created in the sculpting tab of Blender by me. Using the sculpting tab in Blender has been difficult for me in the beginning, since I have not used it before. But to conclude, it has been the perfect solution to create models which look similar to the pictures I got, due to the possibility to shape every angle perfectly. Using this method has been time consuming (especially for the Rider's Mask and the Drachenstandarte) but I am very happy with the results).

2D Graphics are made with the App ProCreate on the Ipad. All of the 2D art is used as icons in the different stages of the game. All of the icons are the size 128x128 px. Sketches of the main character are not included in the game because the group decided to use meta humans instead. Designing the meta human has also been one of my tasks.

List of 3D models made by me:

- Cage Cup
- Drachenstandarte
- Rider's Mask
- Ceramic Vase
- Ceramic Jar
- Ceramic Urne
- Sharl 1
- Sharl 2
- Shovel
- Hammer
- Box

(Models which did not make it into the game and why)

- Cage Cup (different Style)
- Coin (texture is hand drawn with proCreate, could not be imported into the game)
- 3D Character Head/Body (group decided to use meta human instead)

List of 2D graphics made by me:

- Icon Heart
- Icon Potion
- Icon Hourglass
- Icon Coins
- Icon Shovel
- Icon Vase
- Icon Sharl
- Icon Rider's Mask
- Icon Drachenstandarte
- Icon Cage Cup

(Graphics which did not make it into the game and why)

- Sketch Main Character (group decided to use meta human instead)

Pascale Boisvert - Programming

My part of the project mostly had to do with the programming part of the game and the use of *unreal engine 5* to bring our idea into a playable project. As we decided to work with the new engine we knew we had a lot to learn. So, for the first few weeks of the project we watched tutorials, in order to be able to find our way around. We also built a few worlds and a few mechanics to a little better understand what blueprints were and how they worked. Once we could get around *unreal engine 5* we also already had an idea of the game features we would need and how much time we had to do those.

I was mostly focusing on the implementation of different game systems, like the inventory system, the buy and sell system from the NPCs as well as the possibility to open and close the journal. Another one of my tasks was to have a functioning quest-like system, where the player could accept and finish a quest to get the rewards from it. I also implemented the main game menu and the introduction scenes of the game. I then also added a basic ethical system, that changes according to the selling of the items or by using potions. I also managed to make an equipment system, where the character can hold the shovel and an animation is played when the player presses the dig button. Those were mostly achieved by following different tutorials on youtube as even with the basic knowledge, getting a good enough grasp to do everything on my own was too big of a project. However I could use the tutorials and adapt them to make them more logical to our project. Everything at that point was done using basic shapes and temporary models as we were waiting for the real 3D models to be made, as well as the 2D art.

Most of the work was done using *blueprints* logic, which closely follow the programming logic one would use when using c# or c++. It doesn't not required hard coding. The process however stays very programming-logical, where variables can be local or global, where functions can be called from different actors. It is also possible to find similar logic such as *if, if else, switch* and more. It does take a bit of "getting used to" to grasp the *blueprint* concept but once the similarities and differences are known, the work is made a bit easier. Problems can still arise, such as falsely connected noded

Once we had the models, all we had to do was exchange the temporary shapes by their real counterparts and also change the icons from the many items present in the game. This still took some time as every item had to be first individually listed, but for the 3D model, the use of *blueprints* made the job easier, as only the *blueprint* model had to be changed in order for all its instances to also be changed, once the game updated.

On top of the programming, I also was responsible for the planning of the project and trying to make sure the deadlines were met. I also had to keep an eye on when presentations were due and organize the weekly meeting that everyone could attend. I also kept a list of everything needed done and tried to see what could or could not be done in the time frames given.