Caius Adventures VR
Game Design Documentation

Caius Studios - cedric.membrez@gmail.com

December 2021

Contents

1 Design History

This section regroups the different version of the current game design documentation.

1.1 Design 0.0.1

The game started as a course project at the university of Fribourg, Switzerland. The adventure genre and the virtual reality technology were the starting point.

The aesthetic was a simple low polygons, cartoon-like, relatively bright environment for the beginning of the game. A dungeon was included with darker environment.

The player was equipped with only a sword.

The player was guided by an unseen fairy. She was the only NPC so to speak.

The player faced mainly one type of enemy: flying head skulls (normal size). The final boss of the only dungeon was a bigger flying head skull which had the ability to spawn tiny flying head skulls.

For the user interface, dialogues were used to introduce the fairy and her story. A start and pause menu was also introduced.

Due to technical problems near the end of the project, a mouse/keyboard version has been handed-in for the course university. However, the main technology will remain the virtual reality in the future versions.

1.2 Design 1.0.0

Building on top of the university project toward a personal project.

First steps are to clean the Unity project and the code base/structure. Re-introduce the virtual reality controller properly (adapt the user interface, the mechanics).

The game design documentation is reshaped to strengthen the story of the game, its design and content.

2 Game Overview

2.1 Philosophy

Caius Adventures VR is a first-person virtual reality video game. Dive into a distant world, and become a hero! Guided by the fairy who summoned you in this new world, you will gain knowledge, skills and strength to venture the horizon and meet new friends and foes. You will liberate the villagers from evil forces and give the freedom back to the inhabitants of this distant world.

2.2 Common questions

- What is the game?

  A first-person adventure game taking place in a distant fairy world. You immerse yourself into it with the use of a virtual reality standalone set.

- Why create this game?

  The creation of this game gives the unique opportunity to explore the capability of the virtual reality technology.

- Where does the game take place?

  We are in a distant world, similar to Earth at first sight. But many more types of inhabitants (elves, ...), spirits, monsters, and surprises await you.

- What do I control?

  You are, and therefore control, a young hero. You will control your weapons, where you go in the world, and which side quests you want to tackle along the main quest.

- What is the main focus?

  Go on an adventure, gain exceptional skills, and restore the freedom that every inhabitants deserve on this distant world.

- What's different?

  A rich/dense/vast/abundant and positive/colorful/bright/happy adventure with an immersive fighting and crafting environment through the virtual reality standalone set.

3 Feature Set

General features

- virtual reality standalone
- open world, adventure
- fights
- crafting to improve skills, weapons

Editor

- Unity
- Blender

Gameplay

- immerse the player through VR
- the player starts in a distant world, discover her surroundings, friends and foe, learn from higher spirits to respect the freedom of everyone
- let control the sword, the bow naturally with the two VR controllers
- let the player craft his own improvements, potions, ...
- let the player explore incredible nature and landscape
- let the player solve puzzles in dungeon and from villagers' quests/problems
- different races in addition to the fairy and the human villagers: elves, ...
- support the story with descriptions, documents, historical items, distant memories, ...
- let the player keep an important and rich inventory: clothes, weapons, food, potions, tools, story's documents, ...

4 Game characters

4.1 Overview

4.2 Creating a character

4.3 Enemies and monsters

5 Story

5.1 Quests

Main Quests It follows a specific order, each one unlock the next.

1. Intro/Talk with Fairy/Explore New Home/Village
2. Acquire Bow, Sword
3. Visit the Alchemist/Understand Health Points and Potions?
4.

Side Quests

- Learn crafting: learn how to make an arrow? learn how to make fire and create fire-arrow?
- Visit the dungeon nearby

- pick them from the **Message Board**?

6 Game Economy

6.1 Money system

The player is able to gain some coins (money) through quests, looping enemies, and objects (chest, box, etc.). These coins can be used to buy objects (potions, weapons, etc.) and access some parts in the game (pay a NPC an entry-right, pay for small game/amusement, etc.).

6.2 Player levels and points

The player has a health status (see below), as well as a stamina level of 100 points at the beginning of the game. As the player runs/sprints or fight (e.g. the use of a sword, a bow), instead of walking, the stamina level decreases. He can regain stamina points by remaining idle or simply walking. Potions can help as well. As the player progresses and trains, her stamina will increase.

Depending on her adventures, the player might be able to gain magic level. This will greatly improves her ability to fight and recover through specific spells.

In addition to health, stamina and magic levels, the player will have offensive and defensive points. Those are a result of the combination of her equipment: clothes/helmets, swords/bows/spears, and shields. So, better equipment or improving them will improve the offensive and defensive points.

Health point systems' base idea: the player starts with an initial health worth of 100 points. The health points decrease as the player gets hit by enemy or fall from some heights (e.g. high stairs in a dungeon) or onto something sharp (e.g. a trap with spikes).

Experimental/Future development ideas: could I add a fitness tracker (think Garmin watch, https://developer.garmin.com/) such that pulsation impact the health/stamina of the player? Do the pulsations vary enough to make a meaningful and realistic impact in the game? What other data would be valuable to transfer from a fitness tracker? Which fitness tracker to consider?

7 Aesthetic

7.1 Sound Effects

To add a 3D, make sure that: (1) The Camera GameObject in your Scene has an "Audio Listener" component, (2) Sounds that are part of your VR experience have their "Audio Source Spatial Blend" set to 3D

Static sound effect: (1) select the parent object, (2) add component "Audio Source" in the inspector, (3) go to your Assets>VR Beginner>Audio and drag a file into the 'AudioClip' property of your object'inspector.

Contact sound effect: in this tuto, there is custom system called SFXPlayer, which enables scripts to play SFX (special effects) sounds.

One feature of this Unity Project is a custom script to set sounds when an interactable object makes contact with another object. For example, this script will play a sound if the grabbable flashlight is dropped onto a table

Closed captions for these sound effects

Example of a Closed Caption System, with three scripts:

- CCDatabase, which associates a given audio clip with one or multiple lines of text and display timings for each line
- CCSource, which can be added to an object that has an Audio Source component (like the Fusebox) and enables closed captions for the object
- CCManager, which checks for audio sources with closed captions enabled in the Scene and displays the text when the user is in front of the object

In Project, right click to Create > ClosedCaption > CCDatabase. Then, you can drag and drop the same sound (as added to your object earlier) and add lines of text with time when to show them. Then, go to Hierarchy and select your "CCSystem", and change the "Database" in its inspector to what you just created.

Now, need to set the Fusebox GameObject as a source for closed captions in the Scene. Select the Fusebox in the hierarchy, add a component "CC Source". Then use the "CCCanvas" prefab (Project>Assets>VR Beginner>Prefabs>System) and drag it to the inspector CC source component' Canvas Prefab. (This canvas defines how the closed captions will appear. If you have more experience with UI development for Unity, you can customize this for your VR experience.) In the Contact Sound Player component, enable the Closed Captioned checkbox

7.2 Shader Graph and VFX Graph

Clearly not a priority for my project. These are from a Unity Tutorial 'Escape Room' and some advances/details suggestions.

With the artistic team: The team established their requirements as: A simplistic low-poly art style with just a few textures, to convey an engaging world but maintain good performance and optimization

Creating the majority of the shaders for the environment in Shader Graph https://docs.unity3d.com/Packages/com.unity.shadergraph@6.9/manual/index.html?_ga=2.93717277.1060644285.1616315442-972121481.1599717892, so they would be easily accessible to more advanced users who want to customize them

Creating more complex effects with the VFX Graph https://docs.unity3d.com/Packages/com.unity.visualeffectgraph@7.1/manual/index.html?_ga=2.93717277.1060644285.1616315442-972121481.1599717892, which the technical artist used to integrate effects (like the cauldron) into the Project

8 Mechanism

8.1 Actions

Walk:

- How can the player be aware of his surrounding? rotating too much → potential VR sickness. Can the Player have/gain **surrounding detection/radar/awareness skill**? then, you output on the UI some orange/red warning (like a car parking warning)

Run:

- Ability to move slightly faster throughout the world

Sword:

- swing
- hold, drop,

Bow:

- hold it, pull the cord, release the cord and throw an arrow
- Grab an arrow (from the back, from the ground) and fix it on the bow?

.

Axe:

- think of a big knife or axe to cut trees|branches
- a cut from the axe create an item: a piece of wood, a small branches
- combine the branch and the wood to start a fire, craft an arrow from the piece of wood, resell the wood to villagers (in exchange for another item, rewards)

8.2 Space

3D world.

8.2.1 Overview

8.2.2 World features

Contains plain, grassland, hills, mountains, caves, dungeons, buildings (houses, ranches, ...).

8.2.3 Physical world

overview:

key locations:

travel:

scale:

objects:

weather:

day and night:

time:

8.2.4 Rendering system

8.2.5 Camera

Left and right eyes camera, center eye camera.

Camera for UI, as a child of center eye camera.

8.2.6 Game engine

8.2.7 Lighting models

8.2.8 World layout

Is there a best approach toward Open World games?

Do you do the whole world in a giant scene, or do you split the world into multiple scenes and then load them as needed? and if so, How would you make sure the scenes connect seamlessly? Or is there some other method? (from https://www.reddit.com/r/Unity3D/

There are many packages on the Unity Asset Store (such as World Creator) that allow you to tile terrains. In fact, using a tool such as that one (there are many others, that's just the one I know), you can create a terrain from filters that could potentially tile across vast distances, and then place both vegetation and buildings (which can flatten the terrain as well).

You can also build such solutions yourself of course. But a tiling terrain and a world streaming system is a great start.

8.3 Object

8.3.1 Virtual Reality Character

Only the hands of the player are visible.

Should we consider a full body for more realism and better immersion?

8.3.2 Items

Parchment with quests on it at the Guild? or should we simply use some UI (i.e. use canvas instead of game object?)

Bottle: can be empty, can contain water, options, food/soup.

Rock: simple rock that can be thrown (for fun?, to distract enemy, to hit small/weak enemy, to break small things, ...)

Historic document: additional story elements given to the player for an increased knowledge of this new distant world.

8.3.3 Weapons

Sword: they will be of different material and power, giving the player different type of strike and different strength/striking power.

Shield: gives additional protection to the player. they will be of different material and resistance.

Bow and arrow:

Spear:

8.3.4 Non-Player-Character

Fairy: she is a child fairy, represented as an egg with wings flying around the player. She as summoned the player/young hero. At some point in the game, the egg will hatch and she will grow into an adult fairy after some time and experience through the game.

Villagers: they will be of different races and populate the different villages. They will have distinct skills and attached activities (farmers, guards, shops related to weapon-food-potion-shield-cloth, ...).

8.3.5 Villages, buildings, and nature

A **village** will be a set of different buildings and villagers, and specific nature (forest, grassland/plain, mountain, ...).

Guild: a unique building regrouping all adventurers (the player and other NPC-adventurers). In this building, a board quests (i.e. a message-board for the quests) displays all requests from villagers that the player can take upon as the main and side quests.

Houses: the hero's home, the alchemist's house, ...

Nature

- Forest: will be a set of trees.
- Trees: simple tree-generating within Terrain (GameObject) of Unity? or importing from Blender?
- Road: simple coloring of the Terrain (GameObject) in Unity?

Destroyable: a one-piece base GameObject and a Prefab representing the fragments of the former entire GameObject. A script on the base GameObject where the update() make the fragments appear and the base disappear on a specified event (e.g. collider or press of a button).

8.3.6 Enemies and monsters

What form? What characteristics?

To what does he react? When does he attack?

How much health does he have? How much damage does he inflict?

How add Reinforcement Learning in Unity for enemies?

8.4 Rules

8.4.1 Game over conditions

When the player's health points is zero, the player is game over and needs to be resurrected by the fairy (which cost stamina, magic points or time?).

8.4.2 Victory conditions

Destroy every dungeons, as well as the final game boss who rules over all those dungeons.

8.5 Skills

A good coordination of the controllers of the VR is expected. For example to aim with a bow, or strike with enough strength using a sword.

8.6 Chances

Not considered at the moment (December 2021).

9 Technology

9.1 Oculus Quest 2

The game is developed for an Oculus Quest 2 on a standalone basis. It contains a headset and two hand controllers.

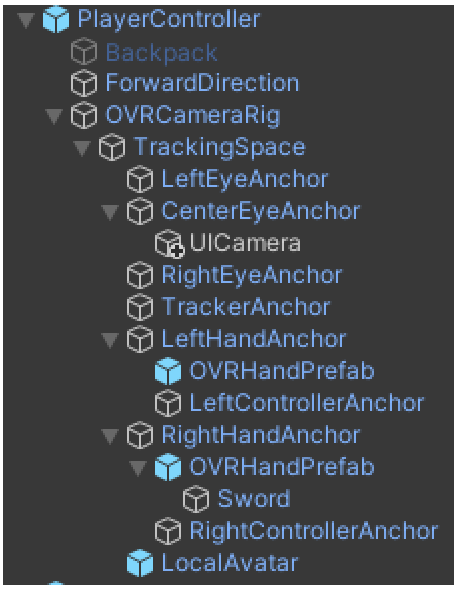
It has not been tested on other virtual reality sets, and it does not include any additional sensor.

9.2 Player Controller

In Unity, the player controller represents both the headset and the two hand controllers: there are camera for the left, right and central eyes, as well as controllers for the left and right hands.

The player controller used comes from the Oculus XR Plugin via the Package Manager in Unity.

The **OVRPlayerController** prefab in the Hierarchy with children:



```
PlayerController
  Backpack
  ForwardDirection
  OVRCameraRig
    TrackingSpace
      LeftEyeAnchor
      CenterEyeAnchor
        UICamera
      RightEyeAnchor
      TrackerAnchor
      LeftHandAnchor
        OVRHandPrefab
        LeftControllerAnchor
      RightHandAnchor
        OVRHandPrefab
          Sword
        RightControllerAnchor
      LocalAvatar
```

Figure 1: Unity Hierarchy: the GameObject components of the Player Controller

(temporary overview of the game: more graph/diagrams are available on https://lucid.app/lucidchart/5db705e2-54cb-40af-95b5-b2446241ebca/edit?invitationId=inv_3d4c1046-40d3-4632-b286-cf34d420f5ce)

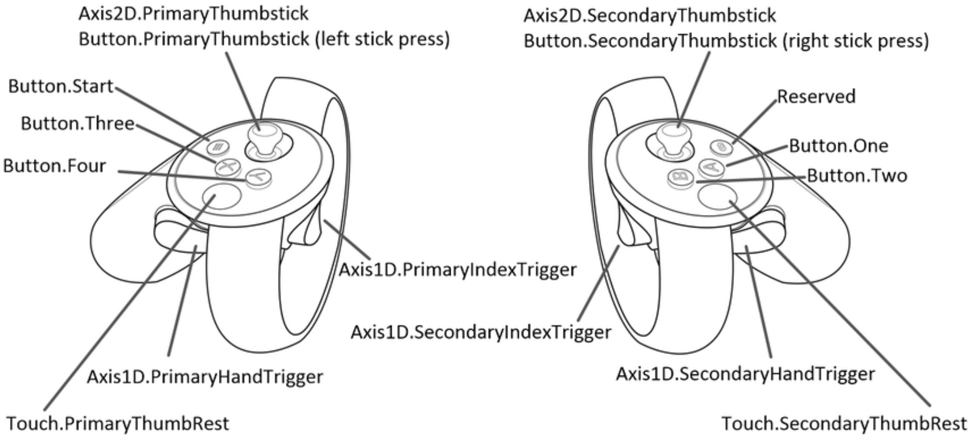

Axis2D.PrimaryThumbstick
Button.PrimaryThumbstick (left stick press)
Axis2D.SecondaryThumbstick
Button.SecondaryThumbstick (right stick press)
Button.Start
Button.Three
Button.Four
Reserved
Button.One
Button.Two
Axis1D.PrimaryIndexTrigger
Axis1D.SecondaryIndexTrigger
Axis1D.PrimaryHandTrigger
Axis1D.SecondaryHandTrigger
Touch.PrimaryThumbRest
Touch.SecondaryThumbRest

Figure 2: Oculus Controllers

Buttons, stick, and triggers:

- one: A
- two: B
- three: X
- four: Y
- Primary Thumbstick

- Secondary Thumbstick
- Axis1D: Index Trigger, Hand Trigger
- Axis2D: Thumbstick
- Touch: thumb rest?

9.3 Interactable Object

This is a more general type object and is applied to both **items** and **weapons**. Because we use virtual reality controllers, left and right hands, we want to make the interaction with objects as realistic as possible. We need to consider how to grab such object properly and with realism.

In the Unity tutorial, the "XR Grab Interactable component" is where you have a script already provided to control the behavior of the object when activated.

Through Project>Assets>VR Beginner>Prefabs>Tutorials>Objects, select one of them and drag it into the scene (e.g. the BubbleGun).

1. click on the parent in the Hierarchy ('BubbleGun'). you can also click on the little arrow to have a look at its children.
2. Go to the Inspector (with parent selected!) and add a componenet "XR Grab Interactable"
3. Under Interactable Events, "On Activate", click 'Add', drag from Hierarchy the child 'ObjectActivator' into the none field. and set function=ObjectActivator>activate()
4. This makes the interactable object work by triggering a specific event (for example, the bubble gun emitting bubbles) on the activation of the object.

Precise Grab points: the idea is to be able to precisely grab a GameObject to make it more authentic: e.g. the sword should be picked up in the middle, the quest paper should be picked-up properly (good orientation...), the bow should be hold by its grip, etc.

10 User Interface

10.1 UI and VR in general

- not specific to this game, but general notes -

UI development with the XR Interaction Toolkit:

1. A GameObject with the **XR Ray Interactor** component attached for each controller. It will cast a ray to interact with the UI - but the XR Direct Interactor used to grab objects is not compatible.
2. A normal **UI Canvas** GameObject, with an additional component called **Tracked Device Graphic Raycaster**
3. An **XR UI Module**

UI - Menu:

channel to communicate stream of data: parts of the screen, avatar and enemies, music and SFX.

dimensions of the channel: textual information, colors, font types/sizes. Using more dimensions -> reinforce information

Theme the interface. Simulate touch with sounds. Use metaphors. **TEST!!!!**

- Need to know always: ... immediate surrounding?
- Need to know from time to time: money, health, distant surrounding, current weapon/special item, number of ammunition (number of bombs, arrows, ...)
- Need to know occasionally: other inventory, other clothes, other weapons, map, details of the quest, ...

10.2 UI for Main menus

A main menu, where the player can start the game and adjust options will be rendered in separated scene.

At game over, the player will also go back to the main menu.

10.3 UI in game

These in-game are rendered on world space canvas, attached to the player controller.

The player will be using some inventory to keep track of her items. Some status are required to show her current conditions: health, stamina, etc.