Year 13

Nicolas Sabogal-Garcia

Year 13



Computing Project: VR Game

Index Contents:

[Analysis Checklist 2](#_Toc503258148)

[Problem Area 2](#_Toc503258149)

[Stakeholders 3](#_Toc503258150)

[Existing Solutions 5](#_Toc503258151)

[Essential Features & Computational Methods 7](#_Toc503258152)

[Limitations 8](#_Toc503258153)

[Requirements 8](#_Toc503258154)

[Success Criteria 10](#_Toc503258155)

[Design 11](#_Toc503258156)

[User Input & Menus 11](#_Toc503258157)

[Graphical user interface (GUI) 13](#_Toc503258158)

[Review: Flowchart 13](#_Toc503258159)

[Review: Structure Graph 13](#_Toc503258160)

[Review: Data Dictionary 14](#_Toc503258161)

[Review: Test Table 14](#_Toc503258162)

[Flowchart 15](#_Toc503258163)

[Structure Graph 16](#_Toc503258164)

[Data Dictionary 17](#_Toc503258165)

[Test Table 18](#_Toc503258166)

[Development 19](#_Toc503258167)

[Iterative development process 19](#_Toc503258168)

[Testing during development 22](#_Toc503258169)

[Summary Test Table 24](#_Toc503258170)

[Corrections of Errors 24](#_Toc503258171)

[Evaluation 24](#_Toc503258172)

[Final Testing 24](#_Toc503258173)

# Analysis Checklist

## Problem Area

The problem area that my project will cover will be allergies. People who are allergic to a pet or people who for some circumstance, can’t have a pet. They will ask for a solution using today’s technology. We can solve it by using computational methods, this is because today’s technology provides a lot of possibilities to cover this kind of problems that in other time will be impossible to solve.

My project will be based on the idea of using a computer to simulate a pet, with this the user will have a safe environment in which he will not be affected by any allergy. There are lots of forms to insert an input in a computer, we can use keyboard, mouse, touch screens, etc… The hardware to input the interaction with the software will be adapted to the user in order to make it more personal. This make the software be simpler to be interpreted by the user and it will make the solution even better.

For example, using a VR headset, we can create a virtual space in where it will be a room with a preselected pet that the user wants. He will not suffer the allergy because it isn’t real so it will solve the problem and some circumstances like the lack of space or life in a flat that doesn’t allow pets. In addition, thanks to the advanced headsets on the market it will look pretty real and it will create a good simulation satisfying the user needs.

Furthermore, by the use of function and variables I can create an iterative and interactive program which will entertain the user with the same idea again and again. This idea will be based in the simulation of having a dog in a virtual space and the possibility of interact with him in an iterative way. With iterative way I mean that the user will be able to feed the dog or play with him each day in order to take care of him.

The features of this game plus the features of the headset will make the simulation more intuitive and interactive. The fact that it won’t need a guide or tutorial will be a really good point of the game, because it will be like real life. If you want to take a stick and throw it, you just need to move your hand with the controller next to it and grab it and then as in real life you throw it by a physical movement. This makes the solution the best one to solve these problems. The only concerns about it will be the requirement of the solution in terms of software and hardware that we will discuss later on.

## Stakeholders

In this project I will have a client that requests a solution for the problem with the allergies of his son to animals. He asked for a game that will satisfy his son and because I have some experience with the new technology of Virtual Reality, I recommended that he used a VR space for the game and explained it would create a realistic experience that will be the best way to solve his son’s problem.

When I finished to explain how VR headset works and what implications will be required in order to do it, he gave me a positive answer to follow my recommendation. Then I started to ask some questions to his son to be able to cover the problem following the client needs. To make thing easy I will call them Dan and Timmy in order to protect their real identity. This was the conversation:

* Me: What is your favourite animal?
* Timmy: Birds because I’m not allergic to them.
* Me: Your dad told me that you want a dog but he refuses it because you are allergic to them, so you still want it?
* Timmy: Yes, sir but my dad said no
* Me: Your dad gives me a mission of creating a virtual dog for you
* Timmy: That will be incredible, I will be super happy if you do that sir.
* Me: Just to be sure, what do you prefer dogs or cats?
* Timmy: I prefer a dog! they are more friendly than cats
* Me: What kind of dog do you want? There is a big variety of dogs.
* Timmy: I don’t know about dogs because I can’t be close to them, so anyone you want sir.
* Me: Hmmm, your dad told me that he had a Boston Terrier called “Ted”
* Timmy: Yes, I had seen him in photos of my father childhood.
* Me: What do you think? Do you want the same dog that your father had?
* Timmy: Yes, sir, that will be fantastic!
* Me: And what do you want to do with the dog?
* Timmy: I will like to pet him or play with him with a stick or a ball, something on these lines. Ah! And I want to be able to feed him.
* Me: I will make your dog be able to do that and if you want any other thing just tell me.
* Timmy: Okay, if I think of anything special for the dog to do I will tell you
* Me: Then we have a deal. I will make you a fantastic friend
* Timmy: Thanks, I will love it.
* Me: Then that’s all for today we will meet again to see the progression of your little friend.

Once I finished the conversation with the son of my client, I wanted to consider some point with him, so I did a second interview with him to discuss the decisions of his son, the hardware required to create the simulation that I had in mind and some other aspects of the project. Here is the interview:

* Me: Hi sir, lets discuss some points now that I know what your son wants.
* Dan: Yes, of course. Ask me whatever you want
* Me: First, the project that I had in mind to cover your request is a pet simulator in virtual reality.
* Dan: Hmm you told me to do the game using Virtual reality but I still don’t really get how it works
* Me: This technology is basically a headset and 2 controllers tracked by red-light sensors. With the headset you will have a 360 view of the virtual terrain and with the controllers you will be able to manipulate this virtual environment with physical interactions. Let me show you.

-In this point I used my phone to show the client a video of how the HTC vive works and how they look. I usually use this video to explain how the VR technology works, is the best one that I had found.

Video: <https://youtu.be/qYfNzhLXYGc>

* Dan: This looks amazing, I really like your idea but tell me the requirements of your project, I know it will be expensive.
* Me: Yes, the requirements for this project is that you need one of the two main VR headset, the HTC Vive or the Oculus rift. I have the HTC Vive, so I will develop the project using it has reference but you can select between both option. The big change between them is the price and the quality. The HTC vive cost 700£ and the Oculus 400£.
* Dan: Wow, I see. These headsets are expensive but I still want to follow your idea, so I will accept these cost. Which one of them do you recommend me?
* Me: I will highly recommend you the HTC vive because is the one that I have and the one with better feedback for the community.
* Dan: Okay, I will follow your recommendation, any other requirements?
* Me: May be, Can I see the computer that you will use?

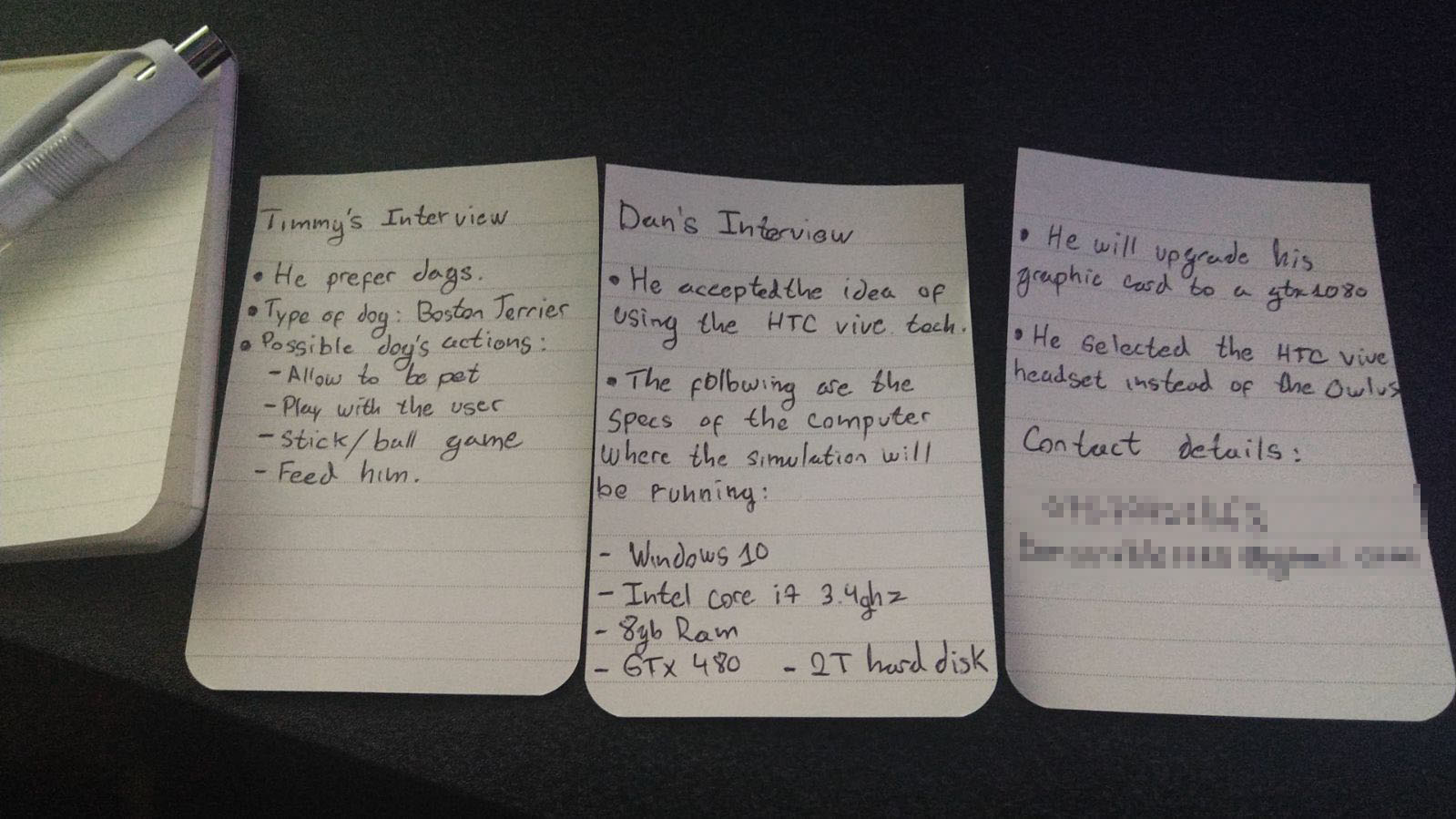
-Here, my client showed me his computer in order to check if his computer have the right specifications to use VR headsets. This was his specs:

1. Windows 10
2. Intel core i7 3.4ghz
3. 8gb Ram DDR3
4. GTX 480
5. 2T Hard drive

* Me: You have a really good computer here, but you will need to upgrade your graphic card to a one compatible with VR, I recommend the new gtx1080 but if you don’t want to waste more money, a gtx1060 will be fine.
* Dan: I will go for the gtx1080 anyways, I wanted to upgrade my computer since a while, so this is the best option.
* Me: That’s good, then this is all for today, I have all the information needed to start the project. Any question or comment to add?
* Dan: Hmmm, no if you don’t need any other thing from me I think we are done for today too. Thanks.
* Me: No problem, if you have any problem or anything to tell me just call me or send me a message.

-Here, we exchange numbers and emails to be in contact with each other and this ends our interview for the project.

Now I have all the information required to create it in a proper way. I wrote the main points that I should cover create the perfect simulation based on the kid’s references and fathers agrees. The following picture are my notes of the interview:



In summary, I will develop a solution by computational methods to solve the problem of be allergic to animals. My client will review the whole project and have an opinion over it to add ideas over the design of it and how it will fit the features necessary for his son with allergies. Also, based on how I will develop this project, it can have other public with different problems apart from the main one(allergies). So they will enjoy having a pet without the problems that a real one will give to them.

My son’s client will make use of the proposed solution by playing in a virtual space with a virtual pet I this case following my client’s suggestion it will be a dog. This will solve the problem of his allergic to dogs because the dog isn’t real but by using this computational method the simulation creates the sense of having a real dog in front of the user. This make the solution appropriate to the needs of my client because he will have a simulation game which cover the problem of allergies of his son by creating a virtual dog to play.

The stakeholders will be the people with allergies or circumstances that didn’t allow them to have a pet or just people that don’t want a pet in real life but they want a kind of virtual pet to look after. The solution will cover all the needs of this group, they will have a virtual pet in a virtual space that doesn’t conflict with the real world

## Existing Solutions

There are couple of solutions for this problem already on the market. The idea of having a virtual pet is old. The first recorded virtual pet is in a game called Dogz. With the introduction of the Tamagotchis and digimons in 1996 and 1997. The idea of having a virtual pet was very popular and since then the market of virtual pets always updates as the hardware updates. So new games of pets come out each year.

What will make my solution better than the ones already on the market is that I will be programing it in a new area of the technology that for me will be the future of the industry of simulations/games the Virtual Reality (VR). This tech is so new and there aren’t a lot of pet simulation games so it will make the solution even better and unique. The only ones that have the same concept that my solution are the simulation pets from portable consoles like “Nintendogs”. But because the way of how these simulations are playing by just holding the console and press buttons to do actions, it can’t be compared with a VR version that will be more immersive. The user will literally be touching the pet and playing with it with physical movements. It will make the difference.

Furthermore, before I propose this idea of making a virtual pet with VR for my computing project, I did a huge research about the hardware and software required and also about the difficulty of programing for this new tech headset. I have some experience about how they work in term of a normal user not a developer. The hardware was simple, the user will need a VR headset, there are two main options HTC Vive or Oculus rift. Both are good headset but the one wining by having better tracking and better feedback for the community, that use both of these headset is HTC Vive. But the bad point of it is the price is higher than the price of the Oculus rift. The Oculus cost 400£ when the HTC Vive cost 750£. Apart from the point of having better specifications that only have an impact in the experience, you can develop for both in the same way so there isn’t a big difference between HTC Vive and Oculus in terms of developing games.

What I found while I was researching for the developing of games for VR by amateurs, is that there are two main game engine to use: Unity and Unreal engine. When I hear about the developing of VR games in Unity, I was happy because I had some experience with the game engine some I will be able to apply it in the future if I choose Unity. The other option will be Unreal engine other game engine more complex that Unity and with a public more professional not like unity that have a mainly public of amateurs on this industry. For these reason and as the main one my experience with Unity, I expected to do the project in Unity.

Then I started to see tutorials and mini-tutorial-project made by the community, to see if the level of difficulty. What I found was that the developing of game in VR have a huge different approach in comparison with a normal game, the movement and the camera of the user are already coded or in better words they are given by the red-light sensors tracking the headset and the controllers, so these things were one of the main parts when you are developing a normal game, giving this I was thinking that they give you a lot of the work done by using your real movement as input already stated. Taking this as a good point that make my project less difficult and other examples of scripts coded for interaction in VR, I decided to present this idea of making a pet simulation in Unity using the HTC Vive, as my project.

References:

<https://www.vrstatus.com/news/unreal-engine-vs-unity.html> ( Best game engine)

<https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/> (VR headsets)

<https://en.wikipedia.org/wiki/Digital_pet> (The concept of digital pets)

<https://en.wikipedia.org/wiki/Tamagotchi> (Big one: Tamagotchi)

<https://en.wikipedia.org/wiki/Digital_Monster_(virtual_pet)> (Popular one: Digimon)

<https://www.youtube.com/watch?v=9x-oXMK0L6Y> (Evolution of virtual pets)

## Essential Features & Computational Methods

The essential features of the pet VR game simulator are simple, because VR create a space that looks like real life. We need to implement the feeling of being there by adding the right features like the movement of the pet and how he reacts to the user, the interactions with it depending on the type of pet, the fact that you need to feed him every day as a real pet and take care of it. Different mini-games with the pet make the game interactive and interesting in order to avoid the users get bored. Here is the list of the features that the simulation must have:

* The pet must be a dog and will be a Boston Terrier. This is to cover the client points and needs that we spoke in the interview.
* An A.I. (Artificial intelligence): We will need an A.I for the pet to make it looks real and have his own reactions. This A.I will be based in the reactions and interactions from real animals.
* A familiar and friendly environment: The way how the room and the other scenarios in which our user/player will need to be familiar like a garden or a typical room make it more comfortable to the user to spend his time there.
* A minimalistic interface: The point of our solution is to create a realistic simulation so the interface must be simpler in order to avoid the user to be distracted or take out of the realistic experience.
* Relaxing O.S.T. (Open Source Track): I will use an open source track that’s is free to use as you want. It will be used to relax the user with this he will be more comfortable and will enjoy the experience.
* Based on my notes from the interview, the dog’s interactions will be:

1. Allow to be pet.
2. Play with the user.
3. Stick/Ball Game
4. Feed him.

How/Why is the Problem Solvable by Computational Methods?

Based on these features, I can explain my project in a more methodical way in terms of a computational approach. I will use iterative function of decisions to be able to base the output of the dog in the decision made by the user, an example of this is the decision of the taking an object and throws/leave it or the decision of trying to pet again the pet when he resists to be petted. The user will need to understand the dog by experience, he will need to use try and error to know the best way to play with the dog, this will create a heuristic system.

Furthermore, it must be iterative in the way for the program to be able to repeat the same action until the user wants to do another action and keep/store a record of this recursion as a variable. This variable will be used to create another one that will store the happiness of the dog, I will call it “Confidence”.

Based on this variable the dog will be able of doing new output from the decisions made by the user, for example, if he try to pet the dog with 3 of confidence, the dog will try to resist but if he try to pet him having 5 of confidence then the dog will allow the user to pet him. This happiness/confidence will be rewarded by the recursion of action by the user to the dog. Using this iterative approach, we can clearly see the use of Pipelining, the output of one process is the input of another.

In addition, by using performance modelling I designed a test criteria based in the interview and client’s opinion to see if a program/ task is successful or not to solve the problem. The results of this modelling were that the program will cover all the client’s needs and will solve the main problem of his son by giving him a virtual pet which is anti-allergies.

## Limitations

One of the limitations of my project is that you will need a big space to move around in the real and in the virtual world, the space required recommended is 3mx1.9m, so it will limit people that don’t have a big space to use the headset. The good thing is that in the case of the HTC Vive it comes with a mode chair that allows you to just stay in a chair to be able to play and you will have a pointer to teleport to the position that you want to be. It takes out a little bit the realism of the simulation but is a good alternative if you want to still play without much space.

Another Limitation is that the hardware required to support a VR headset and the headset itself is very expensive. Normally if you want to do develop a simulation game you will use the pc as unique platform without using any expensive headset. It is the cheap way because everybody as a computer in this days. They will only need the software, so they don’t need to buy a headset between 400-700£ and having already a computer enough powerful to be able of using these headsets that cost between 1000-1200£. Then, this will be one limitation for user with a low budget to expend in this type of project, but for me and my client is fine, we already discuss about it.

The last limitation will be the possibility of epilepsy episodes while you are playing the simulation. Some report from VR users and doctors recommend not to use VR headset if you have epilepsy. This limitation will not affect my client because once I found this problem with the VR technology I called my client to ask if his son suffer of this and he said that his son doesn’t have any problem with his vision, so we will be fine.

## Requirements

In order to play the simulation, you need a VR headset as Oculus Rift or HTC Vive and a hardware that can support a VR headset and the game specs, the minimum requirements for the software to work are the following:

|  |  |
| --- | --- |
| **Hardware Requirements** | **Justification** |
| **Headset:** HTC Vive or Oculus Rift | This is the hardware in which the simulation is based so without it is impossible to play the simulation |
| **GPU:** Nvidea GeForce GTX 970, AMD Radeon R9 290 or equivalent/better. | This is the minimum CPU processer power to display a double output screen with high resolution of the VR environment in the headset. |
| **CPU:** Intel i5-4590, AMD FX 8350 or equivalent/better. | This is the minimum GPU processor power to run 3D environment using the Unity engine and render the textures and terrain. |
| **RAM:** 4 GB or more. | This is the minimum RAM memory to store the temporal files of the game and the normal operative system in second plane files. |
| **Video Output:** HDMI 1.4, DisplayPort 1.2 or newer. | This is a required port to connect the HTC headset to the graphic card |
| **USB Port:** 1xUSB 2.0 or better port | This is a required port to connect the HTC headset to the computer |
| **Space:** At least 5gigabytes (GB) of available space on the hard drive. | This is the minimum space required to store the game |

|  |  |
| --- | --- |
| **Software Requirements** | **Justification** |
| **Operating System:** Windows 7 or later | This is the minimum operative system that the HTC software accept |
| Steam and SteamVR installed | This is the software used to control, set up and use the HTC Vive/ Oculus rift |
| Direct X 9 or later | This is a set of APIs that allows the use of multimedia and graphics accelerator features more efficiently. |

Vive requires a minimum play area of 1.5m by 2m, or 5ft by 6.5ft. Some games require much more, up to 3m by 3m. The two lighthouses that track your movement cannot be more than 5m (16ft) apart. They each require a mains plug within easy reach.

My client has a home-build computer running windows 8 and a mac pro running Mac OS X but he accepted to upgrade and use the home-build pc, so there isn’t any need of including the Mac equivalent of the requirements. Furthermore, this decision is because VR headset are better supported in windows rather than in Mac OS, and Valve/Oculus highly recommend to use windows. Also, the following list are the new specs of my client’s computer after he upgrade it.

1. Windows 10
2. Intel core i7 3.4ghz
3. 8gb Ram DDR3
4. GTX 1080
5. 2T Hard drive & 500gb SSD

## Success Criteria

In order to create a success solution using the idea of a pet simulation in VR. I will need to cover the following main points that I created based on my research and in the client’s interview:

* The dog’s interactions with the user should follow the suggestions given in the interview. These are: Be able to feed/pet the dog and be able to play with him e.g. stick/ball game.
* My client accepted my idea of developing this project based on the VR technology. Therefore, the final product must be as realistic as possible. To do this the simulation must have:

1. An immersive scene which prompts the user to explore and test things.
2. A minimalistic interface and tutorials. This idea of using simple elements with little embellishment to create an interface will help to avoid the user from the immersion.
3. A peaceful soundtrack to create a relaxing environment and avoid the user to stress while he is trying new things with the dog.

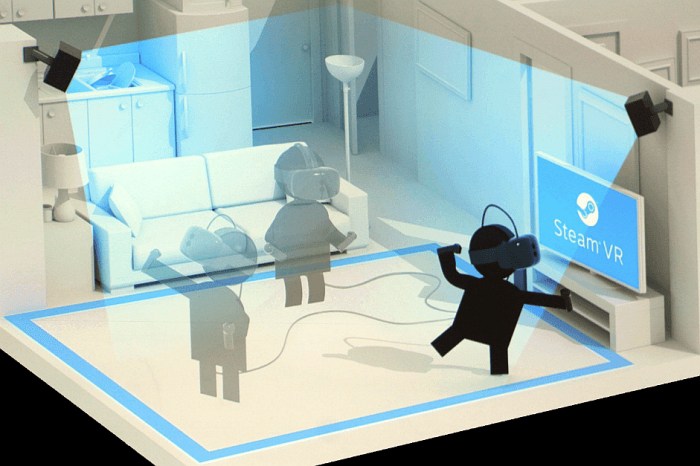
* The 3D model of the dog must have the characteristics of a Boston Terrier which was the childhood’s dog of my client and his son wants the same type of dog that his father.
* The final product must run under the requirements provided

This success criteria is keeping you back. If you sort that out along with the other comments I have made (identifying programming language), there is no reason for you not to be on 9/10

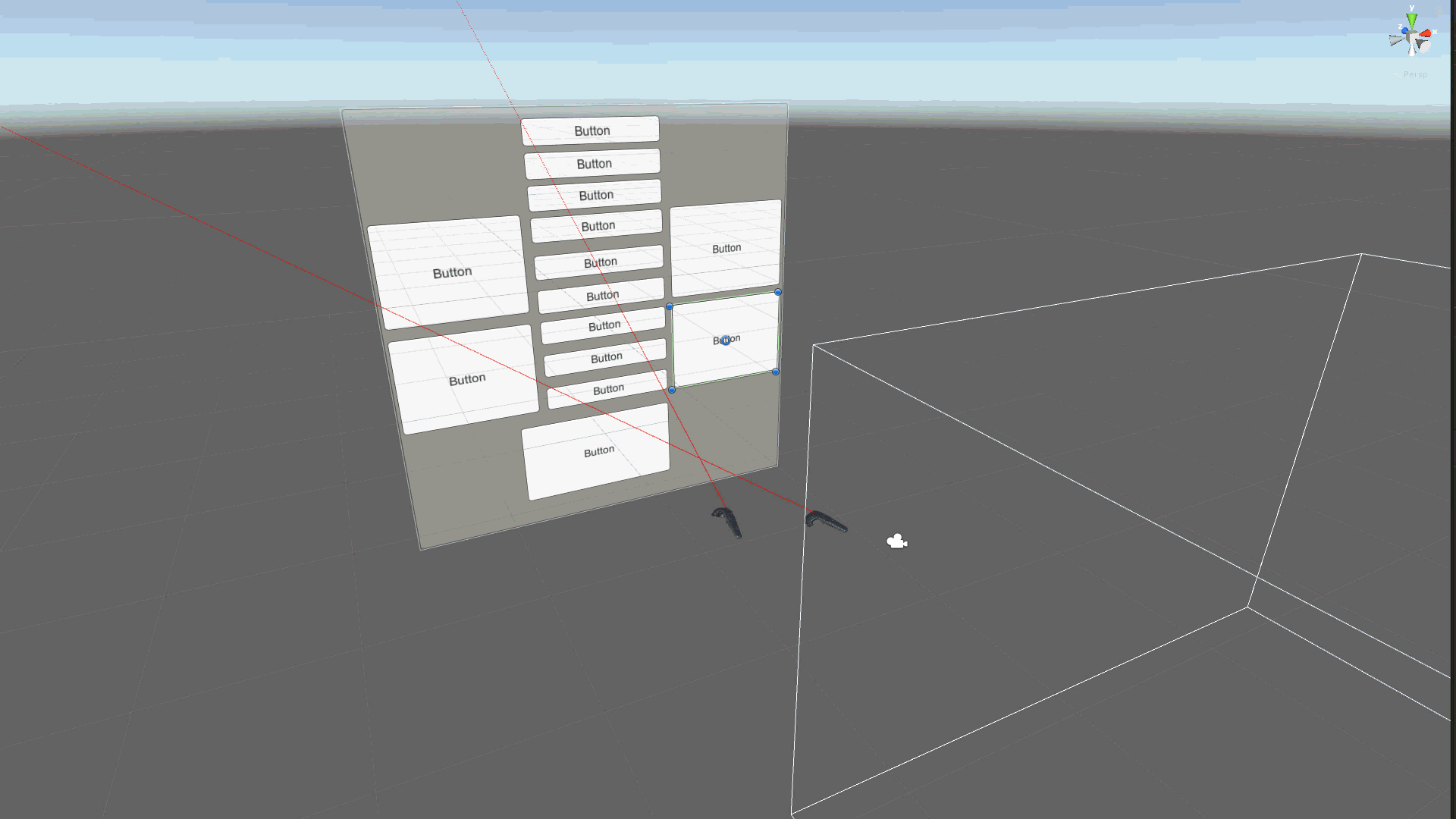
# Design

## User Input & Menus

The user’s input of my program is his physical movement recorded by the two red light sensor of the HTC Vive, while the user is using the two controllers and the headset of it. It will track every single movement of the controllers as the hands of the user and the headset as his head in the virtual space. This is in order to create a model in VR that is following the user’s movement.

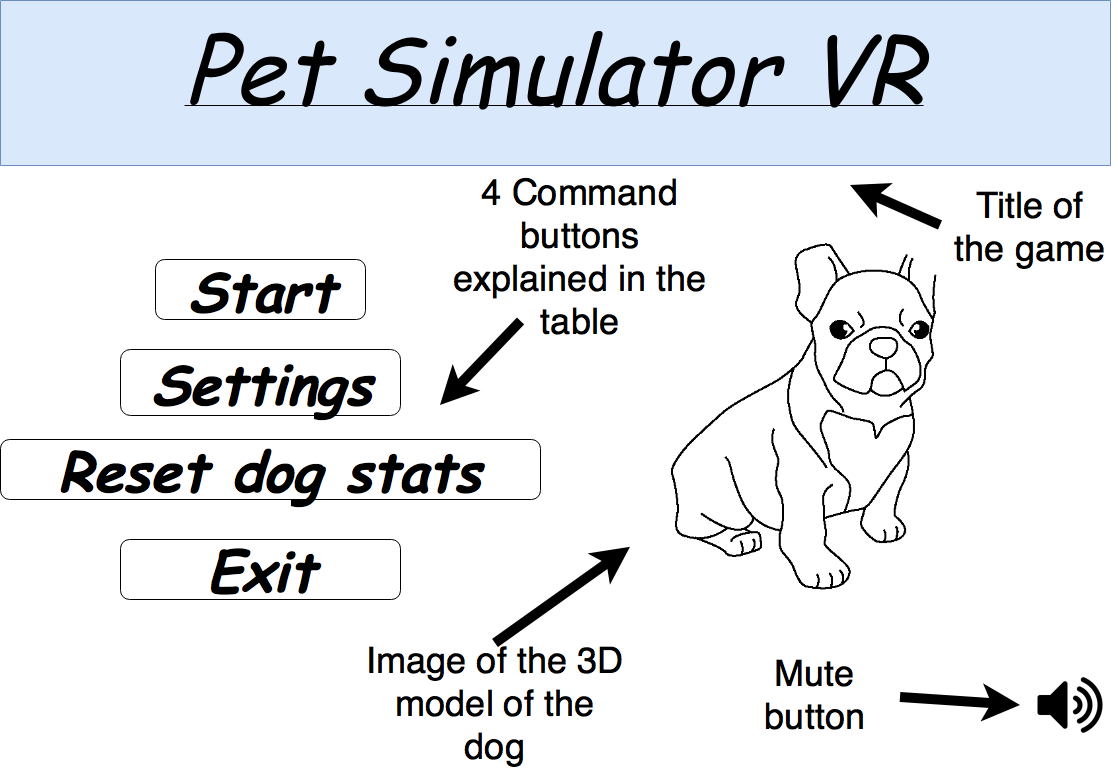


Given that the user’s input is his physical movement I will create a menu that respond to this input. It will be minimalistic in order to prevent the user to be distracted from the immersion. The use of a pointer will be implemented to select the options in the menus. This pointer is a long red line attach to the end of the controller so when you point at a button on the menu with the controller you can see the line hitting the button that you are pointing at. This idea is to give the user a feedback of which button he is selecting and made it easy so do so.

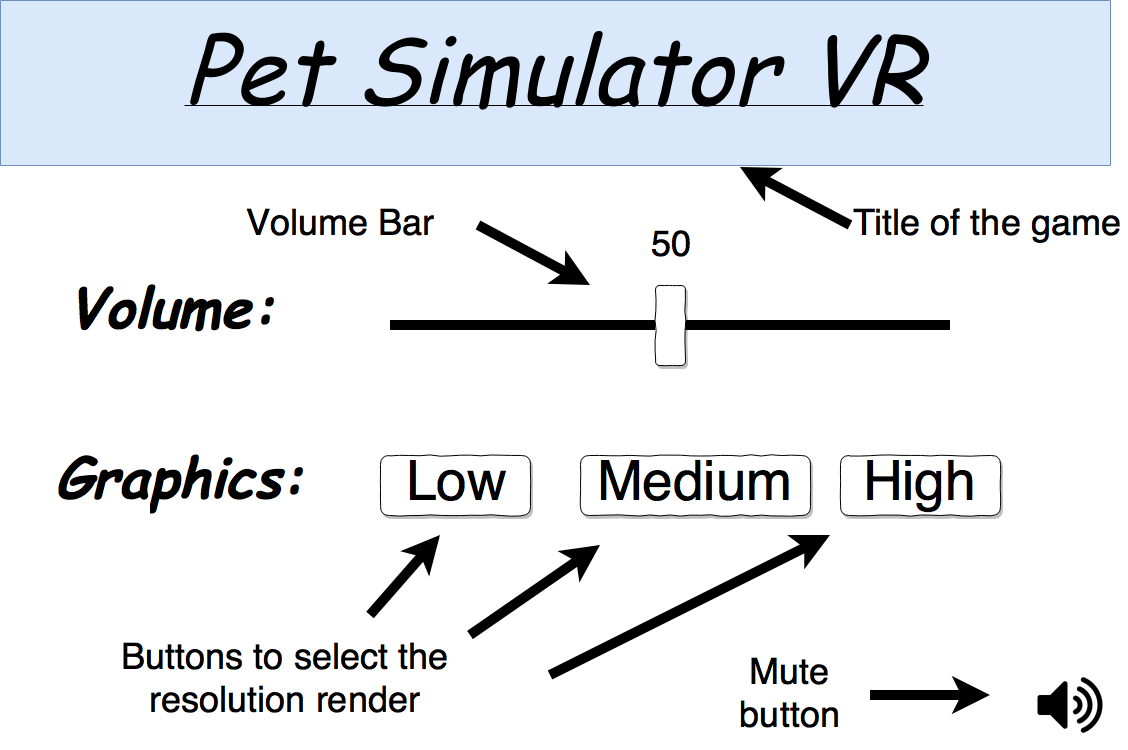


The main menu will be simple as I remarked before, it will be displayed every time the game is launched and it will contain simple and basic options. The following tables have all the option that my project should have:

|  |  |  |
| --- | --- | --- |
|  | **Main Menu Table** |  |
| **Option Name** | **Action** | **Why is this action required?** |
| Start | This option will start the game with the last game save. | This is needed to start the actual game when the user wants to. |
| Configuration | This option will open another menu with a different set of option | This is needed to keep the main menu minimalistic without a lot of option that will distract the user |
| Reset dog stats | This option will reset the stats related with the dog. So if the user wants to start from 0 he can select this option | This is needed to reset all the variables that the game store while the user is progressing with the dog |
| Exit | This option will close the game | This is needed because there isn’t other way to exit from the game rather than do it manually in the computer by pressing the X in the top right. |
| Mute/unmute | This option will mute the game | This is needed to give more options to the user. This is a quick way of mute the game without going to configuration and level down manually the percentage of volume |



|  |  |  |
| --- | --- | --- |
|  | **Configuration Menu Table** |  |
| **Option Name** | **Action** | **Why is this action required?** |
| Volume | This will have a bar to select the volume to a scale of 0 to 100 | This is needed to let the user modify the volume as he wants |
| Graphics | This will have 3 sub buttons: Low, medium and high. Each one will change the resolution of the game so it will improve the performance by lowering the quality of the image | This is needed to solve problems of performance or to increase quality of the graphics |



## Graphical user interface (GUI)

I will follow my idea of a minimalistic effect in all the menus and information given to create the most immersive experience possible. In order to do so, I will just add two items to the GUI of the user that will be always displayed. It will be a bar and a clock. The bar will have a text next to it with the word “Confidence” and as the word indicates, this bar will show the amount of confidence with the dog using squares that fills the bar up to the maximum. The clock is to show the user the real time to prevent expending to much in the game and don’t lost the sense of the real time.

N:\My Pictures\GUI.png

## Review: Flowchart

My flowchart is based on 4 types of modules: inputs, process, decisions and variables. Each one of these are used to represent an action on my project. First we have two process modules to synchronise the red-light sensor with the controllers and headset, then the input module will get the physical movement of the user tracked with the sensor as the input of the program and by using decisions modules each of the different input will have a different output process module. Furthermore, some of the decisions can be affected based on the variable module called confidence. Also, some of the output process of the decisions will increase or decrease the number stored in the variable. This flowchart will be recursive until the user want to exit the game. So by looking at this flowchart I can get a better understanding of the steps and stages of each part of my project and how it can be decomposed in smaller stages.

## Review: Structure Graph

My structure graph is based on the idea of having different process and decision and the fact of state each of the input and the output given by each process. An example of this is when the processes “set confidence of the dog” or “get input of the user” only give outputs variables which are “confidence” and “user input” respectively. In other cases, we can see for example that for the main body (the centre), we use 2 input variables: confidence and user input. To get 3 output variables: confidence, Dog’s response 1, 2 and 3. Furthermore we use at the end the dog’s responses 1,2 and 3 as inputs variables for the process “get final action” to get 1 main variable called “Final action” used as the main output of the entire graph. So by looking this type of structure graph I will have a better understanding of the use of the variables in my project. This will be so useful in the development part where I will need to set all these variables and process

## Review: Data Dictionary

The data dictionary table is based on the possible variables that I will use to create the program. These variables are simple just by using my structure graph that have every single interaction with them, I created a table with the characteristics of each of the variables that I will use to create the program. This characteristic will cover:

|  |  |
| --- | --- |
| Field Name | This will be the temporal/possible name assigned to the variable |
| Field Type | This will be the type of variable, it can be an integer, string, array, function, etc.. |
| Field Size | This will be the possible size of the variable, for example the variable confidence will be in a range to 1-10 only where the actions list will be a constant list of 5 elements |
| Example Data | This will be an example of the data that will be stored in the variable. |

Also, apart from these characteristics the table will have a validation section to know the valid data that each variable can store and a description of the what data is inside. Furthermore, this table is needed to identify each of the variables that I will use to create the simulation and have a track and an idea of how to set each variable during the iterative development of the solution.

## Review: Test Table

The test table is used to record the possible test on the data that I can doing during the in iterative development of the solution. This table have an explanation of what is going to be test, how the test will be carry out, the possible variables, the expected result and the type of test. Also, because my inputs are always related with the dog animations I created another column to specify the dog animation that should be happening during each test. The purpose of these test is to find new problems that shouldn’t be there and to test if every single thing is working as it was programed to.

## Flowchart

## Structure Graph

## Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Example Data** | **Validation** | **Description** |
| Confidence | Integer | 1-10 | “5” | Not Negative and between 0-10 | A variable that storage the confidence of the dog with the user. |
| User input | String | 20 | “Pet dog” | From the list: “Action list” | A variable that storage the user’s input as one of the possible actions from a list. |
| Actions list | Array | 5 | “Take an object” | N/A | A list used to storage all the possible actions of the user |
| User input receptor | Function | N/A | “Throws object” | From the list: “Action list” | A function used to get the user movement and set it as one of the possible action from the “Action list” and then storage it in the “User input” variable. |
| Dog Ignores | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of be walking around ignoring the user. |
| Dog lets the user pet | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of walking to the user and letting him to pet his body. |
| Dog plays with the user | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of be playing with the user . |
| Dog takes the object | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of taking the object that the user throws. |
| Dog walk around | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of walking around. |
| Dog waits for the user | Function | N/A | N/A | N/A | A function used to turn on the animation for the dog of be standing quite waiting for the user. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **What are we going to test?** | **Test Data example** | **Value of the variable confidence?** | **What is expected to happen?** | **Dog Animation** | **Type of test** |
| If the confidence is 0 when we start of the game. | User looks at the confidence bar. | 0 | The bar is 0 when the game start. | Dog walking around | Typical |
| If the user presses the trigger button next to an item | User presses trigger button | N/A | The user will take the object in his virtual hand | Dog gets excited for the item | Typical |
| If the user presses the any other button next to an item | User presses pad buttons | N/A | The item will stay there | Dog walking around | Erroneous |
| If the user tries to throw the item to the dog | User throws the object | 0-3 | The dog will ignore the user | Dog ignores the user | Typical |
| If the user tries to throw the item to the dog | User throws the object | 4-10 | The dog will take the object | Dog walks to the object and take it. | Typical |
| If the user tries to pet to the dog | User tries to pet the dog | 0-5 | The dog will ignore the user | Dog ignores the user | Typical |
| If the user tries to pet to the dog | User tries to pet the dog | 6-10 | The dog will let the user to pet him | Dog lets the user to pet him | Typical |
| If the user tries to play with the dog | User tries to play with the dog | 0-3 | The dog will ignore the user | Dog ignores the user | Typical |
| If the user tries to play with the dog | User tries to play with the dog | 4-10 | The dog will play with the user | Dog plays with the user | Typical |
| If the user takes an object | User takes an object | N/A | The dog will get exited | Dog gets excited for the item | Typical |

## Test Table

Two things you need to get on the top band.

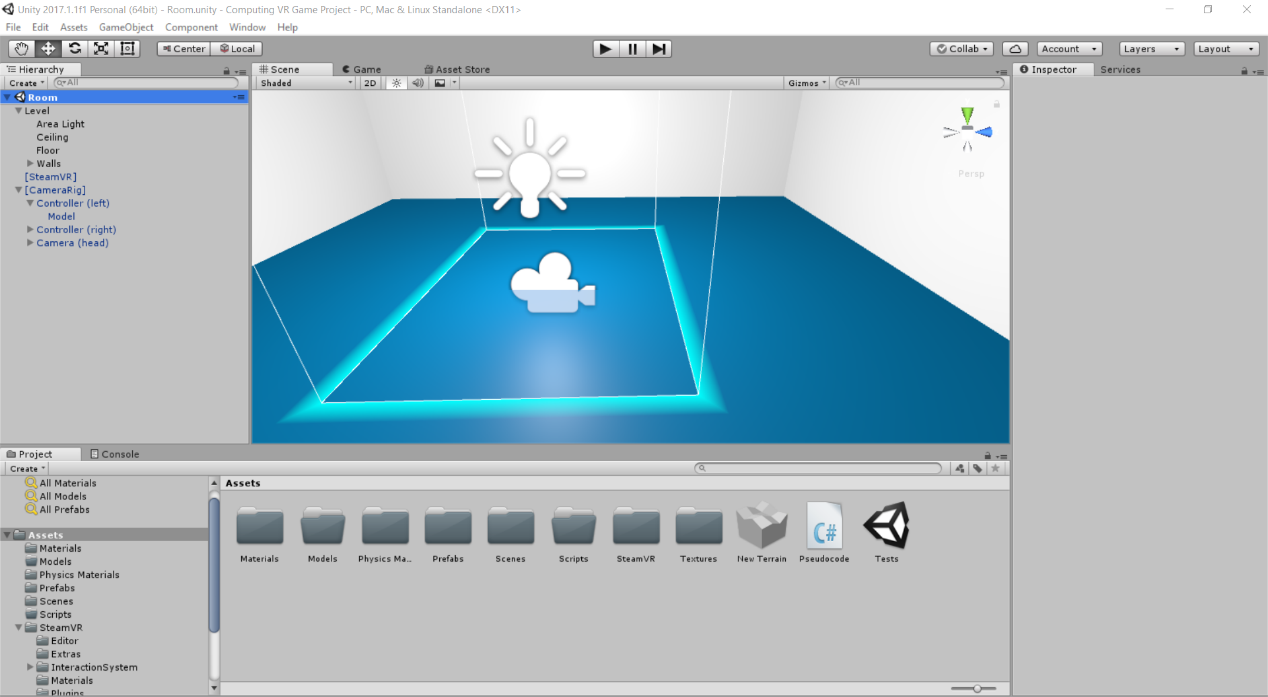
You need further algorithms to explain your program step by step. This should be done in pseudocode. You can have a look at the algorithms you have already in the design and the code what you have created. What you need to do is to make sure that what follows the design is not a surprise, hence the need for more detailed algorithms in pseudocode.

Next thing is linking your test data with your success criteria and justifying how the test data and test will show that you’ve met your success criteria.

# Development

## Iterative development process

First, before I start coding I need to setup the Unity’s project and the HTC Vive. To do so, I created a new 3D project on Unity called “Computing VR Game” and to setup the HTC Vive I used a guide ([http://academyofvr.com/intro-vive-development-introduction-setup/)](http://academyofvr.com/intro-vive-development-introduction-setup/)%20) to download the proper assets used to get the input from the red-light sensors of the headset and the controllers. After having all the necessary complements to starts making my project, I created a room using 6 modified cubes to work as walls and ceiling. Once the room was set, I added the camera ring of the headset in order for the HTC Vive to work. It is the player’s point of view, so if I run this early template I should be able to see the room and walk around with the headset.



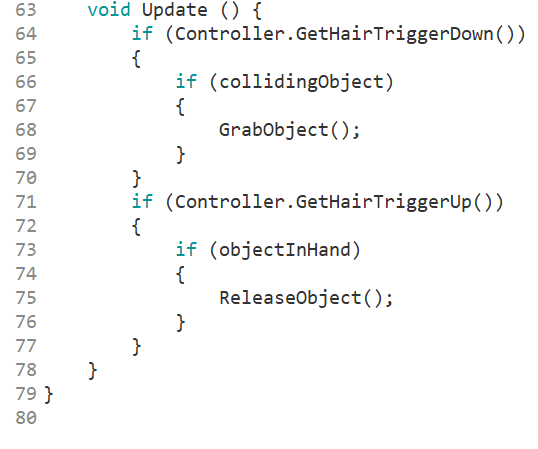
Once I tested if the headset work as it should be, I need to test if the controllers are giving the correct input from each of their buttons. To do so I need to do a big research about the function of every single script that SteamVR plugin provide to work with HTC Vive in Unity. Each of this scripts are used to control the virtual environment using the headset, controllers and sensors.

Then because I want to test the controllers’ inputs, I’m going to be focus on just two of the whole set of provided scrips. They will be the Steam\_TrackedObject and the Steam\_Controller. This two scripts are used to set the controllers inputs recorded by the sensor and the actual buttons on the controller to certain variables in order to work with them through the entire developing of the game. Using these two scrips I created a new one to use the inputs of the controllers recorded by the red light sensors and stored temporally in the Steam\_Controller script to show them in a debug log console. To do so, I needed to recall the two mention scrips in my new script to inherent some of their variables. Then I needed to create some if statements to set the output of the debug console, based on what has been pressed.



Then, I started to think about what I need in order to reach my goal project, the basics behind it. One of the main features of my wanted program is the possibility of the interaction with non-real objects using the controller. For example, throwing a ball with one of the controller. Also this script is required because if I try to grab an object with the controller with my actual code, the object will stay in the same place because there aren’t interactions between them. To do this I created a new script that allow the user to grab object with a collector box by pressing the trigger of the controller.



Once all required methods were correctly stated, I create the main method that trigger the actions, the update () method. it has 4 if statement, they are used to check if the trigger is pressed or is released. If one of this two option is true, it will use another if statement in order to check if the controller’s collider and the object’s collider are touching or if there is an object already attached to the controller’s collider. Then when the second if statement is true it will run the GrabObject() method or the ReleaseObject() method to grab an object or to release it

## Testing during development

In the following section I will test every single important part of my project while I’m developing it, so it can be modifying in an early stage of the code. Furthermore, because my project is based on the idea of wearing a headset to play it, I will add some pictures of the expected result of each test in game.

Test 1:

•The first test that I did was if the camera added in the room works with the headset and if the sensors recorded the movement of the headset and the controllers.

-Result: The headset gives the correct image of the room that I created and I can see the correct respond in terms of virtual position of both controllers. I can see them and they follow the movement of the real controllers in the virtual space.

Test 2:

•Once I setup the first script to grab an object I did a test if all was correctly programmed. It must grab an object by pressing the trigger and leave it by unpressed the trigger:



-Result: As we see both methods used to grab an object by pressing the trigger and leave it by unpressed, works as they should be.

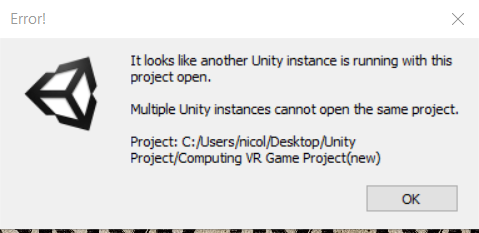
## Summary Test Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test N°** | **What are we going to test?** | **What is expected to happen?** | **Wanted result** | **Any error?** |
| 001 | If the headset give the right view of the room using the camera right added. | While I’m using the headset I should be able to see the room and walk around | Yes | No |
| 002 | Grabing an object by pressing the trigger and leave it by unpressed it | While I’m using the controller I should be able to grab objects and throws or leave them | Yes | No |

## Corrections of Errors

Error 1

• This error occurred while I was trying to open my unity’s project in my laptop. Unity didn’t allow me to open my project, it said that I had open multiple unity instances, here the error:



Solution: I ask a friend that use unity a lot, he said that I needed to delete the file UnityLockFile in the temp folder. Then I did it and once I relaunched unity with my project to see if the problem occurs I could see how my problem was solved.

# Evaluation

## Final Testing