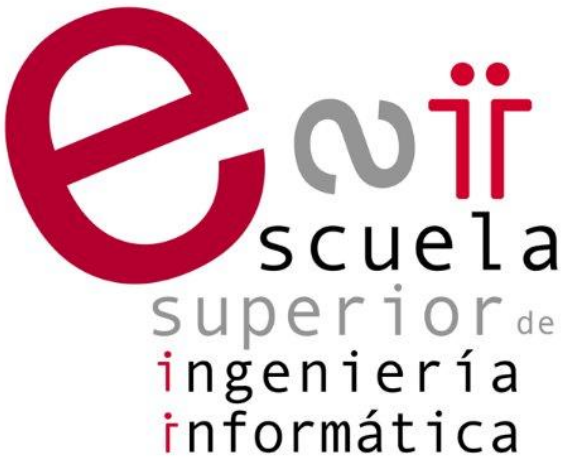
STEAMVR INTERACTIONS_EXAMPLE SCENE

Óscar Gómez Monedero



25 DE ABRIL DE 2019

ESIIAB - UCLM

Videogames and virtual reality

Contenido

Introduction

In this exercice, I'm going to create a minigame for HTC Vive in Unity3D. In the subject, the use of the plug-in "SteamVR" is encouraged, because is easy to use and free in Unity3D. So that, The exercise says to create a mini-game with “Circular Drive”, “Lineal Drive”, “Arrow” or “Throwable” actions of SteamVR. This actions are implemented in “Interactions_Example” scene. This is done so that we adapt to the environment of Unity and VR

Idea

In my case, I’m going go to implemented a “Circular Drive” action in a minigame. Some ideas that I thought were:

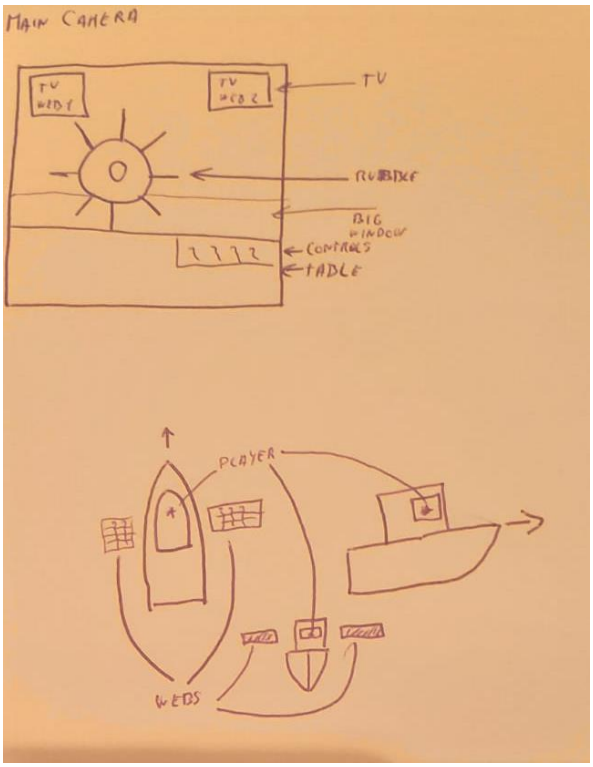
- Drive a vehicle
- Control a NPC that I can run in a 2D dimensión
- Handle heavy equipment that is based on turning on a fixed point

Finally, I chose a first option when I saw a film of pirates. The first idea for the game was make a pirate boat, and attack other objects. Later, the idea was changing until you reach a boat that collects ítems

Concept

In the game, you are a captain of a rescue boat and you have a one misión, recolect the barrels that other boat lost in middle of the ocean. Then, you will control a boat, with two webs that you need it for pick up the ítems.

Appearance



Controls

The first and most important control is a “Circular Drive”, this type of control are implemented in a rudder for change the direction. Also, the boat have a “Lineal Drive” for modify the speed and two bottons, that we will use for actívate the different webs.

How I need use the HTC Vive, the player interact with the world throught the controllers and the VR helmet.

When the player need interact with the virtual world, he will touch the ítem with the controller and he will press the trigger of the controller (if the object has to be taken, rubber and velocity controler) or he will only touch the ítem (bottons).

The player can’t move in a 3d model, beyond the limits of the cabin, but as the virtual reality system is not able to stop the user without hindering the experience, it can go through the walls and do what he wants, but these cases will not be contemplated.

Behaviour

As the game will be developed to learn how to use the system, the user should only start the application. Later, the player will se the rubber, and the principal table where will be the rest of controls.

In this position, the player can play with the controls, and he will discover what each one does. The player, by instinct or curiosity, will catch a barrel.

As the game has been developed to learn the mechanics of the device and to manage them, it is not necessary to implement more things. If the game were developed for entertainment, we should set a more developed objective, limits (time, for example), or systems of puntulación or levels.

The game

External asserts

As the game is plannificated, we need search the neccesary elements for make it.

The fist asset is a “SteamVR”, that have the neccesary components for use the HTC Vive. The other asset is “Fast Buoyancy” by Nathan Gauër that implements the models of boats, barrels, and the physics of the water and waves. This asset is free but have copyright.

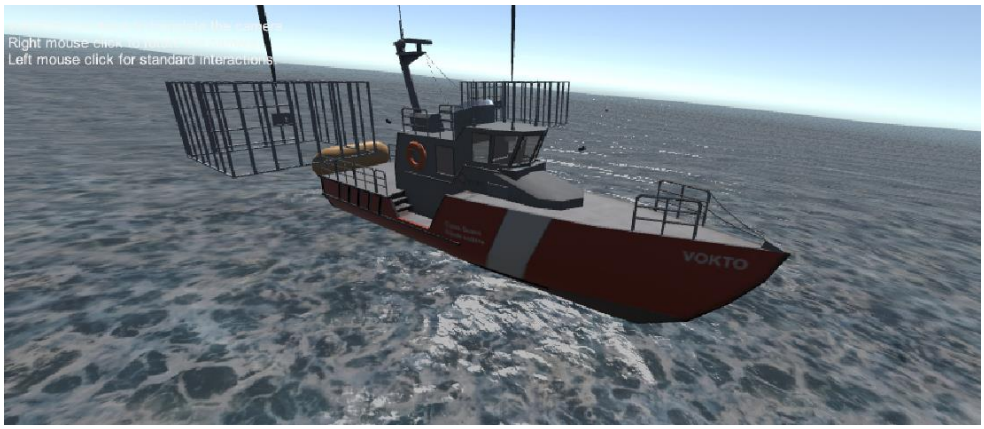
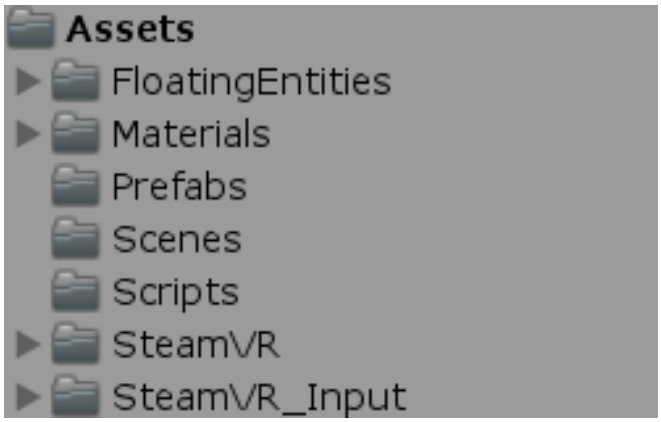


Ilustración 1 Boat in game

Structure

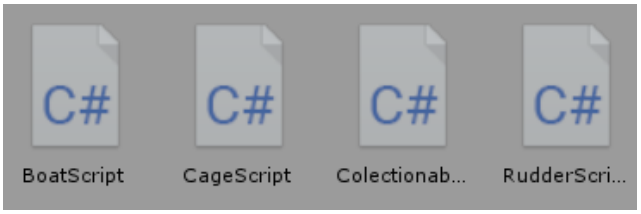
The proyect has been structured It has been structured in this way:



The assets that I have imported (FloatingEntities, SteamVR and SteamVR_Input) are in a principal folder, later, we can see different folders, that I use for save a different resources.

The main scene is in SteamVR -> InteractionSystem -> Samples, and click in a InteractionSystem scene. The reason for this is to facilitate the game, since this scenario contains the elements necessary for its use already configured

Scripts



I make a 4 scripts, for a differents elements in the game.

The first script, the BoatScript, serves to manage the speed of the boat, so you need to know the status of the lever and send over the control script of the ship (which is implemented with the asset, the name of this script is PropellerBoats). We can see that the boat have 4 states, still, reverse, slow forward and fast forward.

The second script, is the controller for the webs, that inicialy was cages. This script serves to manage the download and collection of the web.

The third takes care to know if the web hits an object, and put it in the back of the boat.

The last is the rudder controller, that deals with knowing the position of the rudder, and moving the boat as it is. To do this, look at the result of circular drive, which is found in the auto-generated script "LinearMaping" in the ship's gameobjet.