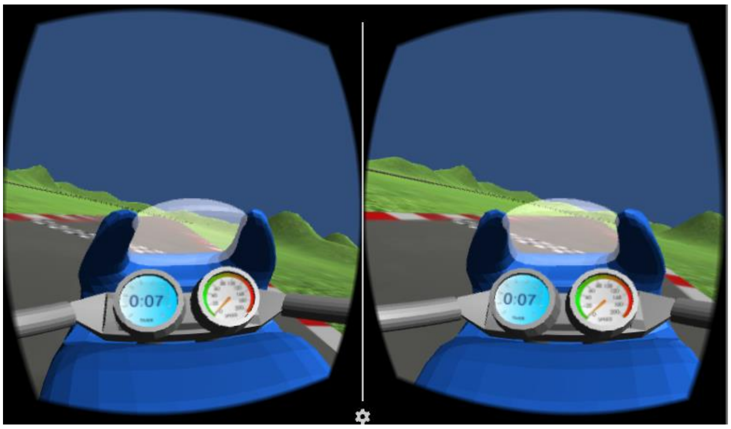
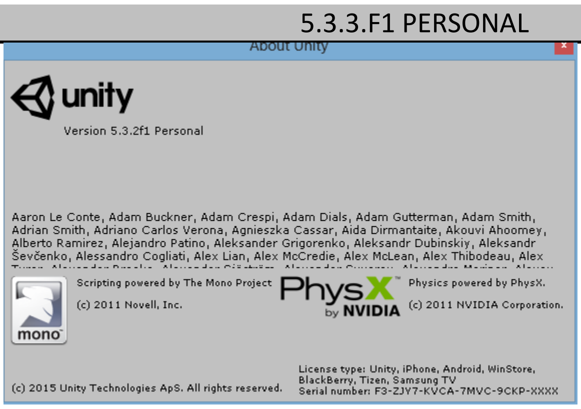
Cardboard VR Motor Bike Controller



1. GENERAL INFORMATION

DATE OF DOCUMENT	17/04/2016
NAME OF THE PROJECT	Cardboard VR Motor Bike Controller
AUTHOR	Michael Soler
UNITY VERSION	5.3.3.F1 PERSONAL



Version 5.3.2f1 Personal

Aaron Le Conte, Adam Buckner, Adam Crespi, Adam Dials, Adam Gutterman, Adam Smith, Adrian Smith, Adriano Carlos Verona, Agnieszka Cassar, Aida Dirmantaite, Akouvi Ahoomey, Alberto Ramirez, Alejandro Patino, Aleksander Grigorenko, Aleksandr Dubinskiy, Aleksandr Ševčenko, Alessandro Cogliati, Alex Lian, Alex McCredie, Alex McLean, Alex Thibodeau, Alex

Scripting powered by The Mono Project

(c) 2011 Novell, Inc.

Physics powered by PhysX.

(c) 2011 NVIDIA Corporation.

(c) 2015 Unity Technologies ApS. All rights reserved.

License type: Unity, iPhone, Android, WinStore, BlackBerry, Tizen, Samsung TV
Serial number: F3-ZJY7-KVCA-7MVC-9CKP-XXXX

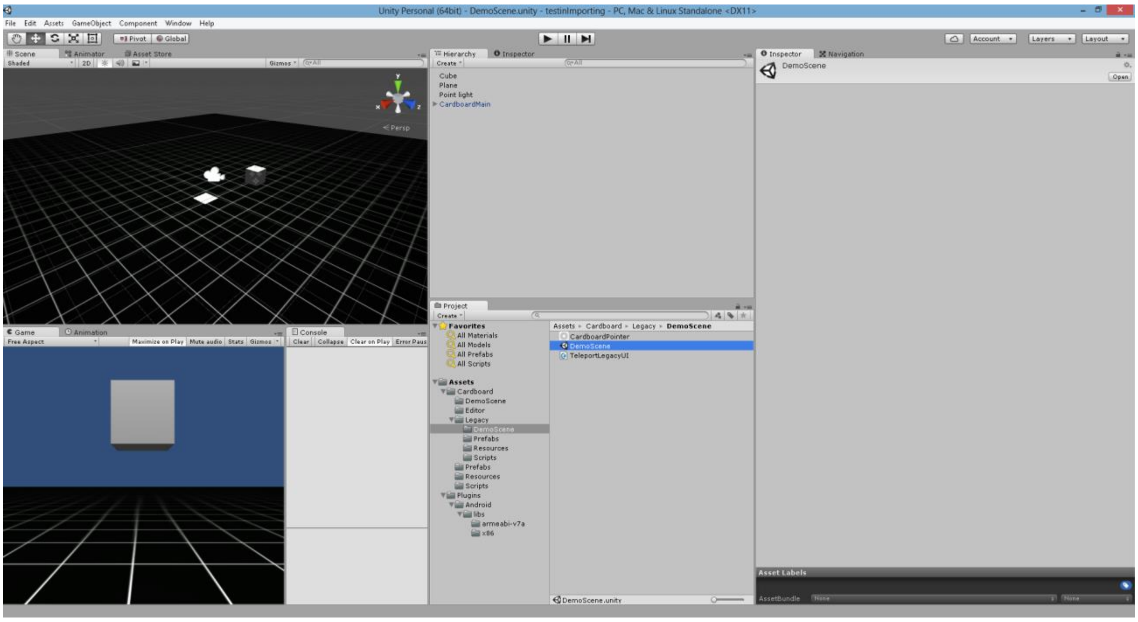
Index

2. IMPORTING INFORMATION

This package works with the “google cardboard” for UNITY that must be downloaded first using the following link:

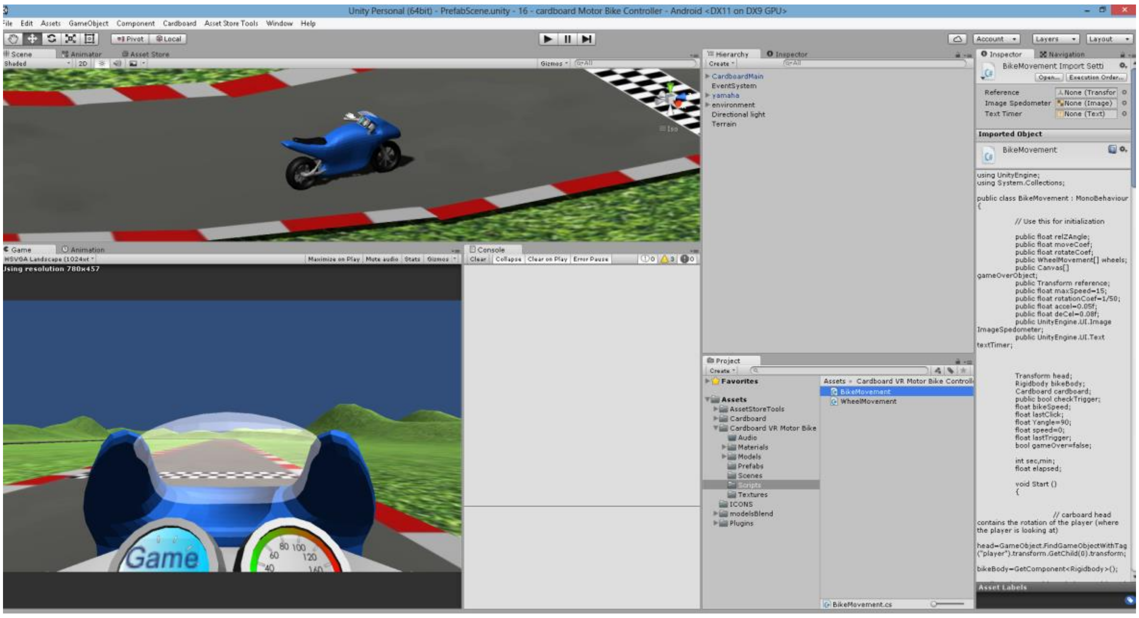
https://developers.google.com/cardboard/unity/?hl=en

Once downloaded and imported to unity, your project should look like this:



Then, import our package to the project, which will leave you the following folder configuration:

The gaze input collider must be disabled. There is no need to change collider or other game objects.



Unity Personal (64bit) - PrefabScene.unity - 16 - cardboard Motor Bike Controller - Android <DX11 on DX9 GPU>

HSVGA Landscape (1024x(

Using resolution 780x457

CardboardMain
EventSystem
yamaha
environment
Directional light
Terrain

Favorites

Assets
AssetStoreTools
Cardboard
Cardboard VR Motor Bike
Audio
Materials
Models
Prefabs
Scenes
Scripts
Textures
ICONS
modelsBlend
Plugins

Assets » Cardboard VR Motor Bike Controll

BikeMovement
WheelMovement

BikeMovement.cs

BikeMovement Import Setti

Reference	None (Transfor
Image Spedometer	None (Image)
Text Timer	None (Text)

Imported Object

BikeMovement

```
using UnityEngine;
using System.Collections;

public class BikeMovement : MonoBehaviour
{

            // Use this for initialization

            public float relZAngle;
            public float moveCoef;
            public float rotateCoef;
            public WheelMovement[] wheels;
            public Canvas[]
gameOverObject;
            public Transform reference;
            public float maxSpeed=15;
            public float rotationCoef=1/50;
            public float accel=0.05f;
            public float deCel=0.08f;
            public UnityEngine.UI.Image
ImageSpedometer;
            public UnityEngine.UI.Text
textTimer;



            Transform head;
            Rigidbody bikeBody;
            Cardboard cardboard;
            public bool checkTrigger;
            float bikeSpeed;
            float lastClick;
            float Yangle=90;
            float speed=0;
            float lastTrigger;
            bool gameOver=false;

            int sec,min;
            float elapsed;

            void Start ()
            {


                        // carboard head
contains the rotation of the player (where
the player is looking at)

head=GameObject.FindGameObjectWithTag
("player").transform.GetChild(0).transform;

bikeBody=GetComponent<Rigidbody>();
```

3. PROJECT DESCRIPTION

This package allows the user to ride a motor bike in a circuit using a VR headset. The movement of the bike is controlled with the rolling angle (moving the head to the right or left shoulder). The speed of the motorbike is displayed on the panel, as well as the lap time.

This package includes:

- 3D models of the motor bike, the environment. All simple but realistic.
- Textures and materials of the objects.
- Game manager which controls the evolution of the game and restarts it.
- Player script to control the bike.
- A main scene with the basic racing game.

The cardboard/car movement integrations are summarized in the following lines:

First, the angle of the head in the Z direction is obtained. Using this angle, the rotation and inclination of the bike is achieved.

4. LAYERS, TAGS AND COLLIDERS

LAYERS

All objects are placed on the default layer.

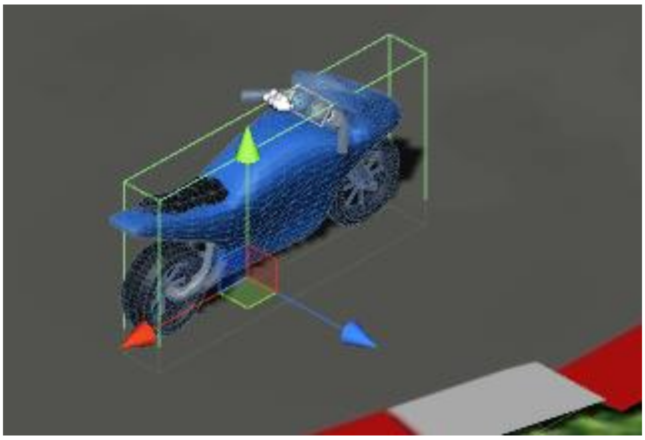
TAGS:

TAG	description
player	Player’s gameobject with cardboard
finishLine	It is used to reset the timmer
environment	Buildings/other things.
Road	Each part of the tracks.

*Other gameobjects are untagged.

COLLIDERS

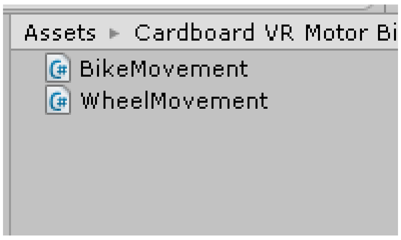
This game uses a trigger to determine whether or not the bike is on track.



Check if the “gaze pointer cursor” collider is disabled:

It is important to import the cardboard package correctly, and check if these colliders are working properly.

5. SCRIPTING INFORMATION



Assets ▸ Cardboard VR Motor Bi

BikeMovement

WheelMovement

We explain each script with some detail in the following table:

- **bikeMovement:**

  This script controls the movement of the motor bike. It considers the rotation of the head to apply movement to the bike. It uses the trigger function/click button to advance.

IMPORTANT VARIABLES
public float relZAngle; → it is used to store the rotation value of the head. **public WheelMovement[] wheels;** → both wheel script. **public Transform reference;** → it is used as the objective position for lerp motion of the cardboard **public float maxSpeed=15;** → it is used to set the maximum speed; **public float rotationCoef=1/50;** → it is used to set the rolling speed; **public float accel=0.05f;** → it is used to set the variation of speed (acceleration); **public float deCel=0.08f;** →it is opposite to "accel". **public UnityEngine.UI.Image ImageSpedometer;** → this is the arrow image for the speedometer. **public UnityEngine.UI.Text textTimer;** → this is the text used for the timer.
IMPORTANT FUNCTIONS
void moveWheels()→ it rotates the wheels. **void getTimer()**→ sets the timer value. **void restart()**→it restarts the game (called when the player gets off tracks.) **void OnTriggerStay(Collider col)** → used to chek if the player is still on track.

- **wheel:**

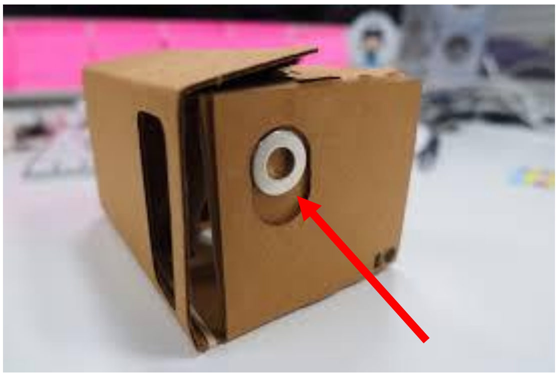
  Is used to move the wheels to simulate the movement of the car.

IMPORTANT VARIABLES
float speed→ rotating speed of the wheel. **float rotAngle**→ rotating position of the wheel. **Transform parent**→ the container for the "clone" objects.
IMPORTANT FUNCTIONS

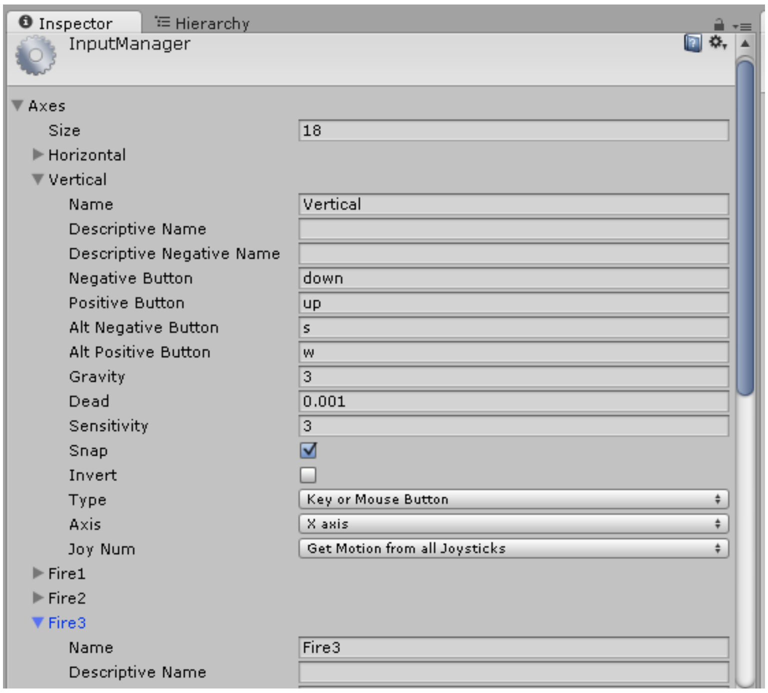
6.MOVEMENT AND SETUP ON ANDROID DEVICE

When testing the program in the android device you need to use the google cardboard glasses. To move the car you will have to press the magnetic trigger. The line of code that is used for the movement is:

```
        if(cardboard.CardboardTriggered==true || Input.GetButton ("Fire1")==true)

                {

                        if(         checkTrigger==true)

                        {

                                    checkTrigger=false;

                        }

                        else

                        {

                                    checkTrigger=true;

                        }

                }
```



Please check your input module to match the following:



InputManager

▼ Axes	
Size	18
▶ Horizontal	
▼ Vertical	
Name	Vertical
Descriptive Name	
Descriptive Negative Name	
Negative Button	down
Positive Button	up
Alt Negative Button	s
Alt Positive Button	w
Gravity	3
Dead	0.001
Sensitivity	3
Snap	☑
Invert	☐
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks
▶ Fire1	
▶ Fire2	
▼ Fire3	
Name	Fire3
Descriptive Name	