**Year 4 Interactive Media Development**

**Assignment 2**

**Martin Futas – X00111325**

**Approach taken;**

1. **Obtaining Assets.**

As this game is being developed via Unity 3D I’ve decided to obtain models created in Blender as they are imported natively

<https://docs.unity3d.com/Manual/HOWTO-ImportObjectBlender.html>

To obtain the assets I’ve googled for freeware blender pre-made models.

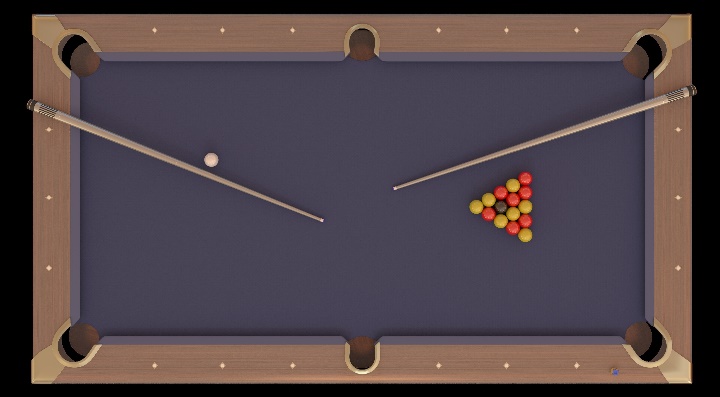
This ultimately lead me to website called cgtrader where I’ve found freeware American Pool Table model together with cue models.

<https://www.cgtrader.com/free-3d-models/furniture/table/american-pool>

As the table consisted of different parts it was perfect for assigning different materials to cushions and cloth.

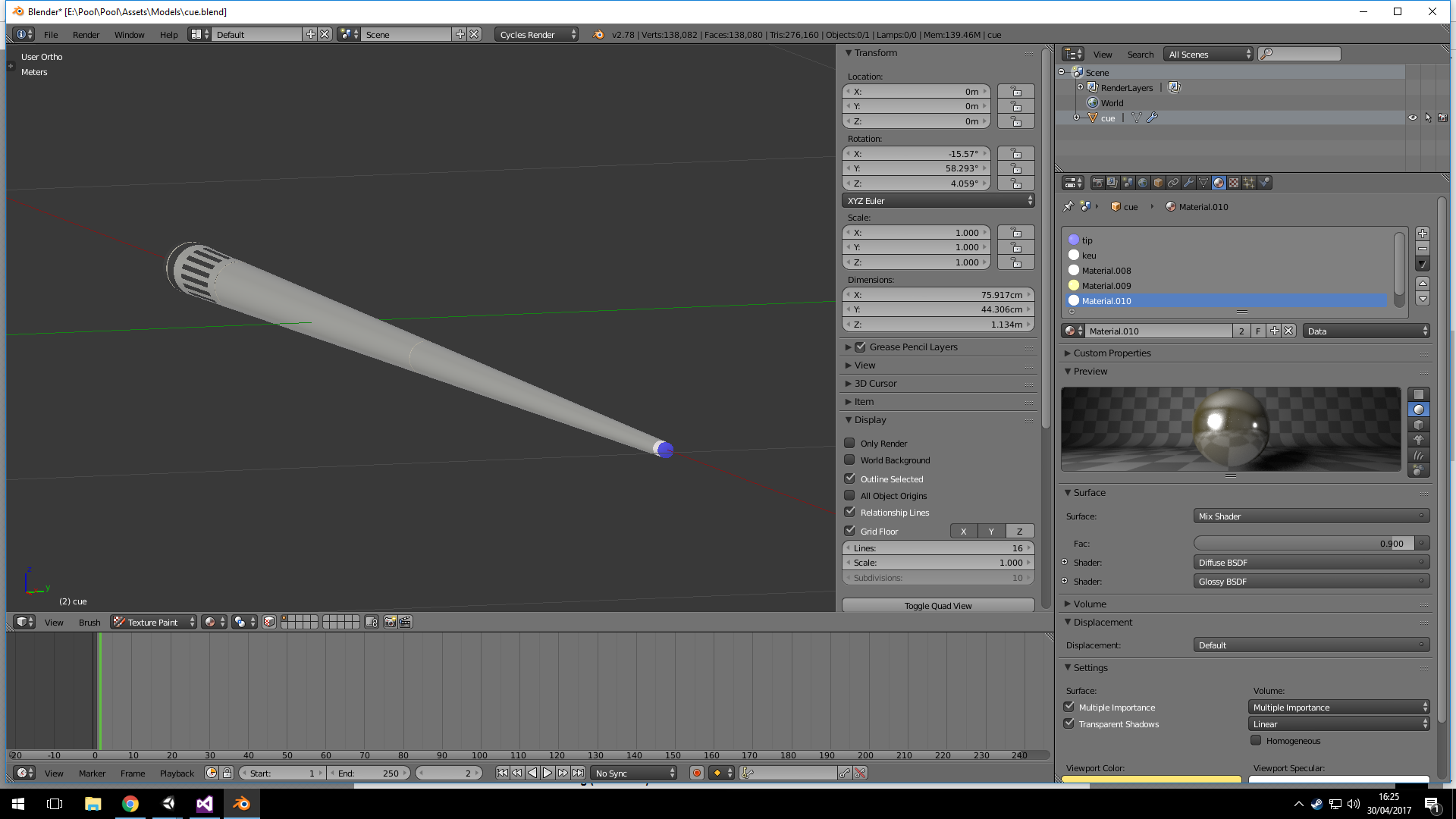
Separate model was also used for the ball models, however this was not an necessary step.

<https://www.cgtrader.com/free-3d-models/sports/equipment/biiard-balls-rigged-for-bouncing>



1. **Editing the model**

As the cue from American Pool Table has been pre-positioned I had to export the cue model into a new Blender file where I recalculated it’s physical properties, set centre of gravity and re-assigned it’s axis so that the pool cue could be used as an physically interactive property used by player to strike the cue ball.



Pool table also had to be edited to flip textures which were facing upside down (Backface culling) Textures were changed to create a more traditional looking pool table also.

1. **Creating the gameplay area**

The play area is rather simple, but contains all elements required. A spot light is placed above everything, including camera angles. It is centered on to the table, lighting up the playable area and casting shadows on cushions and sides of table, table pockets, cue stick and balls.

4 walls and floor are added in order to make player feel as if he was inside interior of a bar, this helps the light to bring out the important aspect of game which is the pool table play area which is in centre of the room. A cue stick is leaning against the pool table.

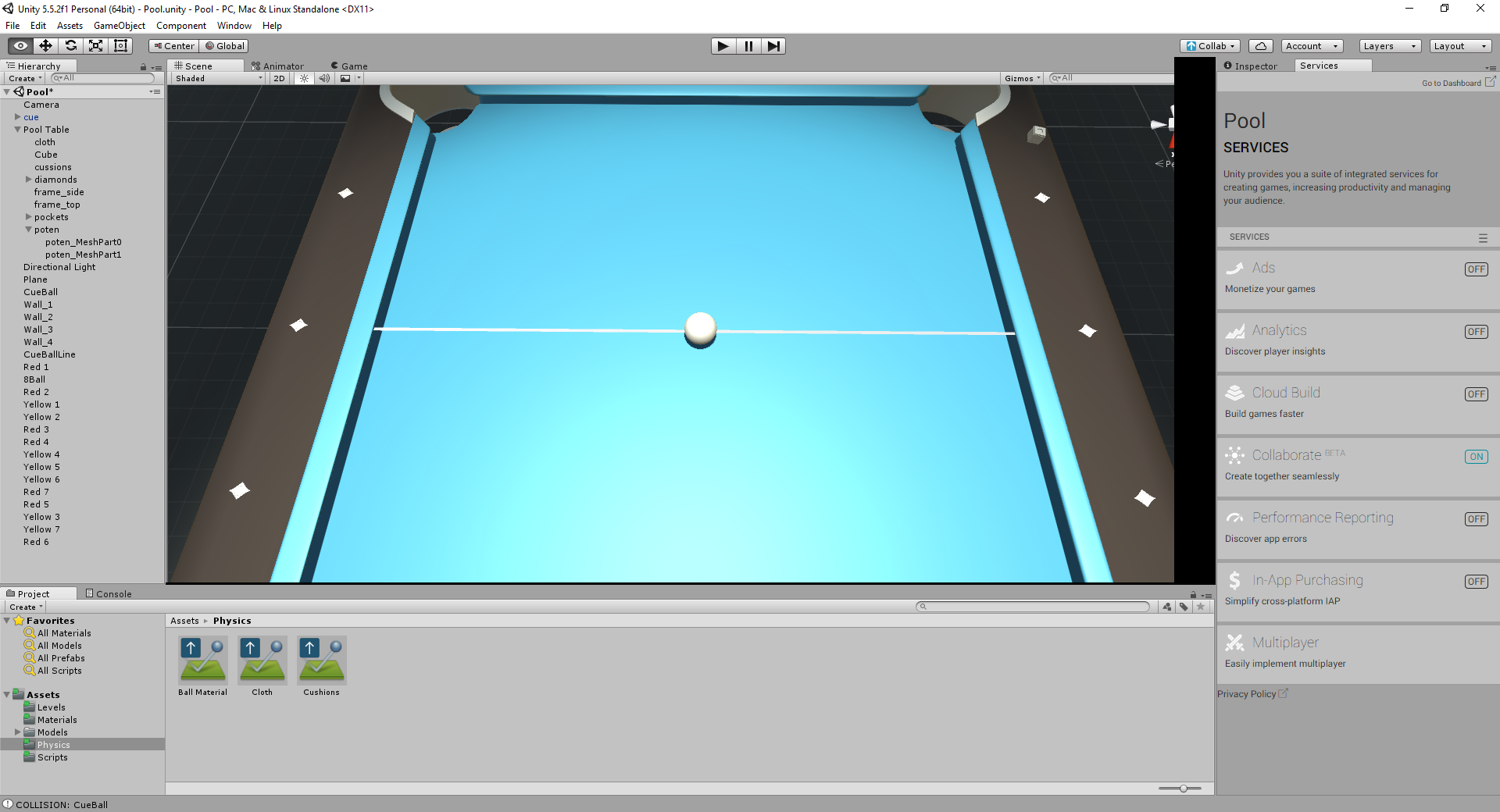
Tournament blue colour was chosen for cushions and cloth. #2483B1FF

Non-moving parts are declared static, to avoid unwanted movement.

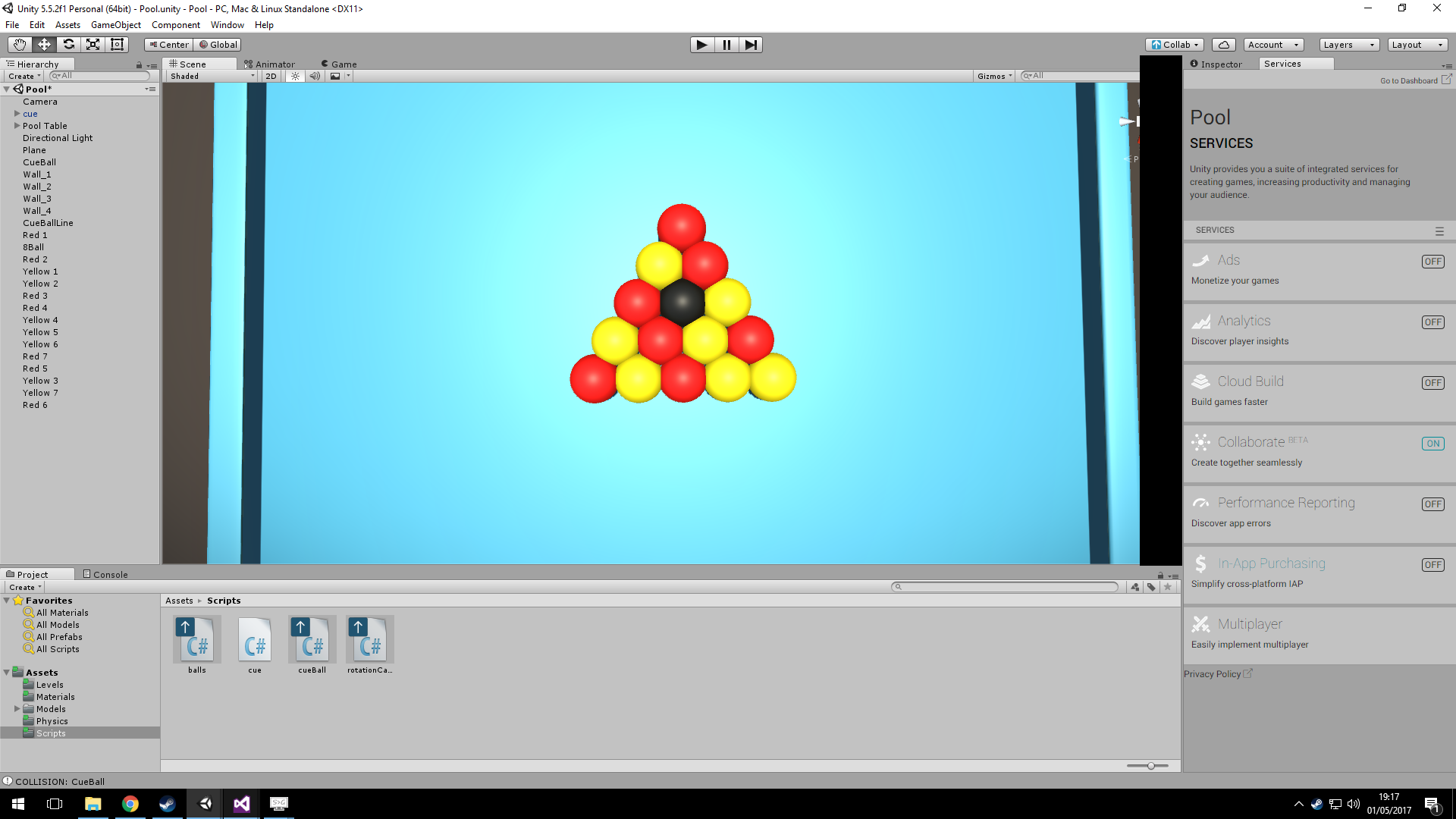
1. **Positioning game assets**

I started off by creating a line renderer between the 2nd diamond on both sides of table, this marks our starting line for cueball from which the player shoots for first time, or where the ball will gets respawned on foul shot.

Afterwards, I placed a cue ball on the line, centred on the table.



Next step was to add the formation of 15 balls between the first and second diamond on the opposite side of the table. I’ve chosen colour scheme of UK 9-ball pool (red for one player, yellow for the other)



One of the challenges was ball positioning to represent better accuracy of physics, if balls were to close to each other, they didne break accurately, therefore gaps had to be added. Once positioned, it was time to start tweaking the gameplay portion of the game.

1. **Scripting game movement**

I’ve decided that the best way one can play pool is by utilising the mouse as this brings a certain level of realism to the player, for example the player can only aim cue as well in real life as well as he can move mouse in real life.

The players cue stick orbits around the cueball together with camera. Player can toggle between different views using the “V” key. The second view is a top-down view showing the whole table statically, giving an impression that the game is 2D and providing overall picture of game state.

Once the player finds their desired firing position, the firing of projectile is initiated and player pulls back mouse while holding mouse button. Upon releasing the mouse button a force equivalent to distance of cue from ball is applied to rigidbody of cueball, sending it in direction of camera (therefore the direction of cue).

The speed of all balls in the game is controlled by script, this is to ensure that balls on pool table don’t roll around forever, it was quite a challenge to find a sweet spot to ensure that player isn’t waiting around for too long and that the balls act naturally, however some compromises had to be made as Unity isn’t able to simulate real life physics.



The above snippet ensures that ball stops when its rigidbodie’s square magnitude reaches below 0.005.

Targeting/shooting system was based off pool game by Sergio P. (referenced in this document), however it was heavily modified.

1. **Physics**

One of the biggest challenges was getting the game physics right to reflect physics of the real pool game. Countless of hours have been invested in research and implementation and a lot of failed experiments have been conducted. However I was eventually able to reproduce physics similar to real pool game.

Pool Table Cloth; this is the main “plane” of the play area where balls roll on. No bounciness applied to this, only dynamic and static friction, to slow balls down as they roll.

Balls; no friction whatsoever, the reason for this is so that when balls hit cushions they bounce off them as in real game, applying friction makes them stop upon impact with cushions. Bounciness set to 80%, any more bounciness causes balls to jump around the table/off table.

First implementation of physics ([.gif](https://media.giphy.com/media/xUA7b0QWG7pLGmqbu0/giphy.gif))

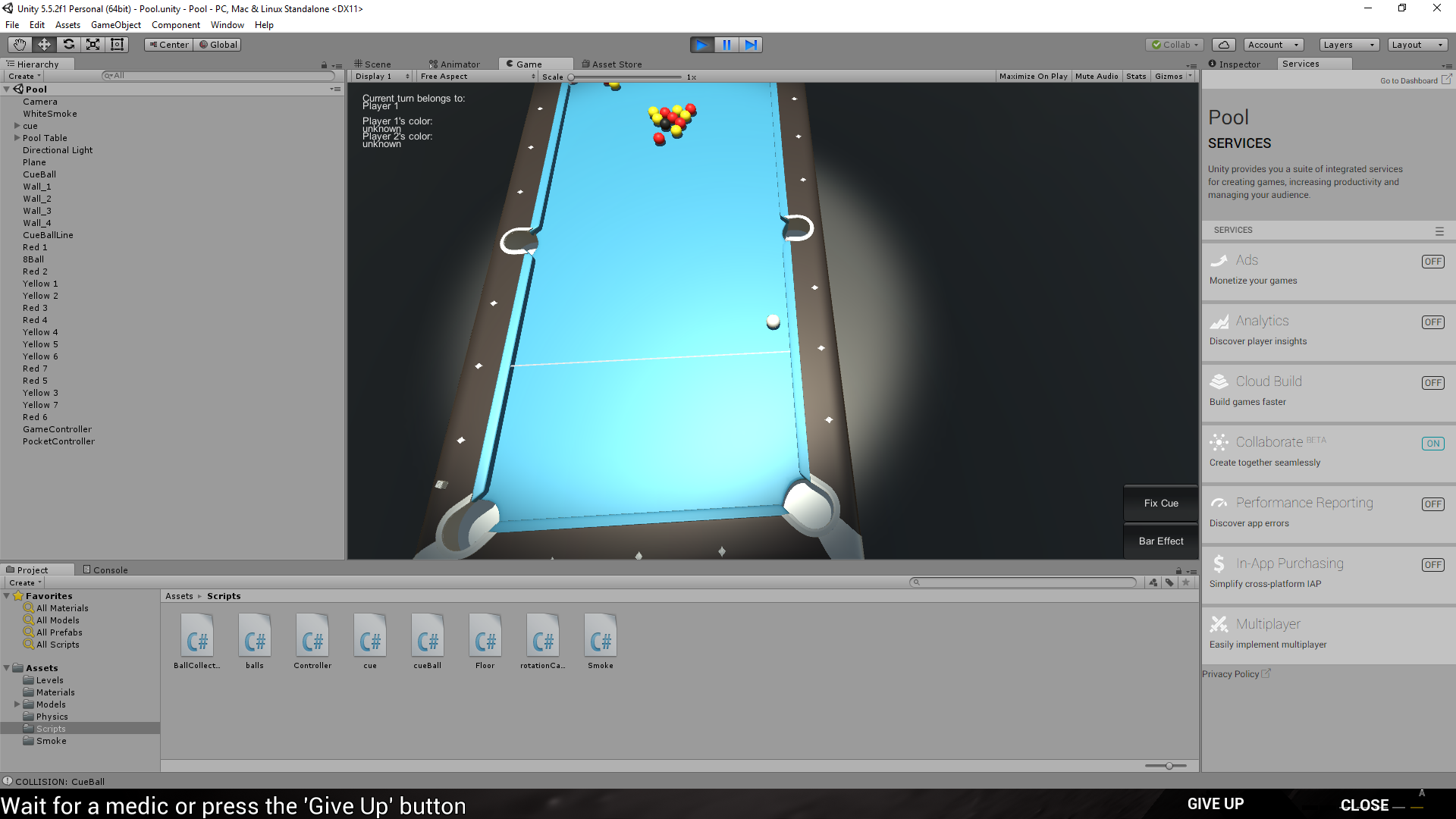
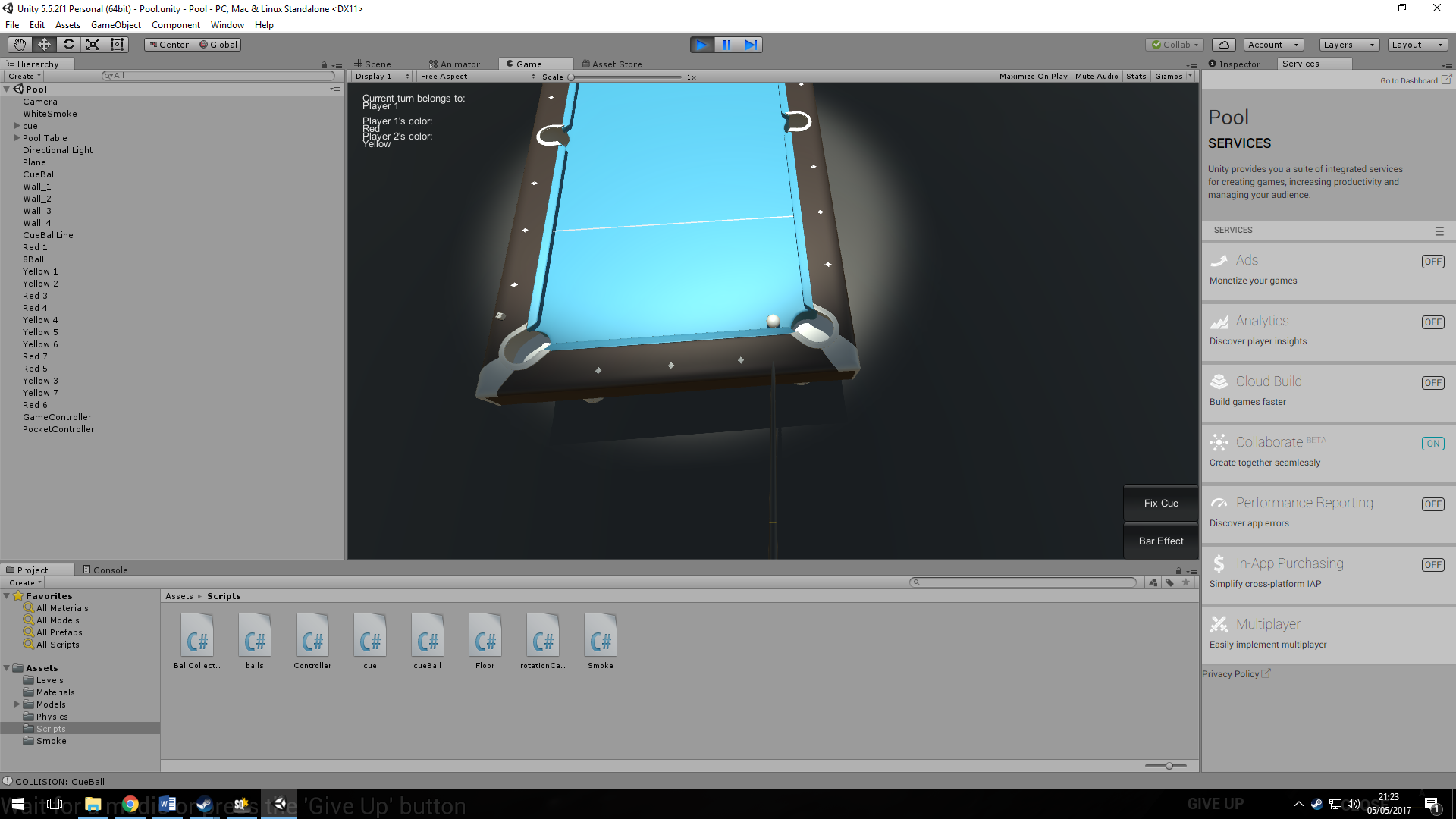
Upon tuning down some of the values to make the physics interaction less “explosive”, I’ve obtained these results ([.gif](https://media.giphy.com/media/3og0ItRwJnGa1btEGc/giphy.gif))

After re-balancing physics, and maximising physics calculations for unity engine via physics setup. Adding a little bit of drag to balls and fine tuning vast majority of all combinations of physics settings, I’ve arrived to this representation of realistic physics ([.gif](https://media.giphy.com/media/3oKIP8R3bV9zSGzXJC/giphy.gif))

Therefore, the physics met required standards for pool;

1. Balls roll and lose momentum as they move around
2. Balls have their weight but bounce properties also
3. Cushions allow balls to bounce off them
4. Pockets are physical, allow balls to fall in and despawn.
5. Table cloth has “felt” physical properties.
6. Pool Cue physically hits cue ball applying force to it.
7. Balls can seldom bounce off table, just as in real life.
8. **Gameplay**

In order to provide the players with gameplay, game rules had to be coded and implemented. Game play starts with player 1, when first coloured ball gets potted that ball colour gets assigned to user who potted it, and other colour gets assigned to the other player. There is a player switch which gets called depending if a player fouled or not.

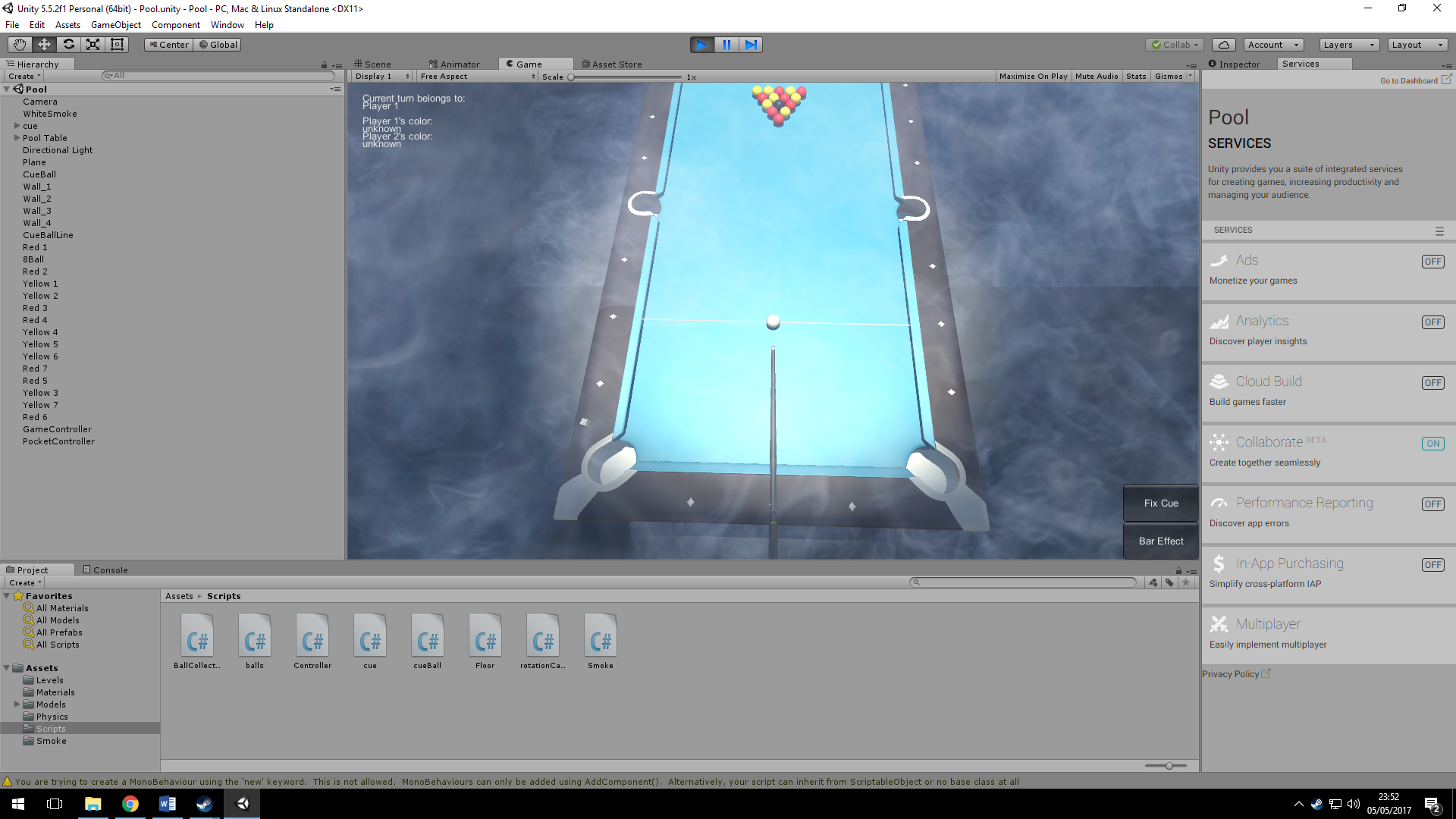
For this reason a simple UI had to be implemented, It informs the users of whose turn it is, and which ball colour belongs to which player. Player wins if he pots all of his/hers assigned balls and the 8ball.

If a player pots the 8ball before he is supposed to, the other player wins, upon win/lose a message is displayed on the screen informing players of which player won and gives them the option to restart the game, for this reason cursor also re-appears on screen.



1. **Extra Features**

As pool tables are typically found in bars/pubs I thought it would be a nice feature to add the atmosphere of these in game, as optional addition. The player is given an option to trigger the smoke effect via a button on bottom right of the screen. This triggers smoke for duration of 5 minutes using Unity’s particle engine.



1. **Game Features**

The game can be played by 2 players, players share a pool cue. The pool cue interacts via physics, therefore it can happen that player misses cue ball and drops the cue, at which point he can return cue to original position. When player fouls, next player gets turn, if he pots opponents ball then opponent gets turn. If 8 Ball is potted, the active player loses. If a ball goes outside of playing area (on the floor) The ball is returned back onto the “spot” of table. When a player pots all of his assigned balls and an 8 ball, that player is victorious. Extra feature includes an smoke effect which lasts for 5 minutes. Users can use UI to see whose turn it is according to game rules. When game is won/lost, players get option to restart the level.

**Checklist**

The game:

Create a two player game, allowing each (human) player to take turns according to the rules

**1. The game should implement the physics of the real game (30%):**

a. Ball direction and momentum depends on the force and direction of the cue and cue ball.

b. Collisions between balls

c. Collisions between balls and the table edges (cushions).

**2. Game play – the player who pots all of their coloured balls + the black ball wins (30%):**

a. Allow player to choose direction and force of cue, allowing a ball to be targeted.

b. Player loses if black ball potted before the colour balls

c. Players take turns.

3. Lighting should be from above the table (10%)

4. Use appropriate camera position(s) to show gameplay (20%)

5. An extra surprise or interesting feature (10%)

**References**

Sergio P. A Pool game based on Unity3D, DOI:

<https://github.com/Sergio-P/Pool>

Unity Forums for countless of topics, ranging from physics to UI, DOI:

<https://forum.unity3d.com/>