

Subject

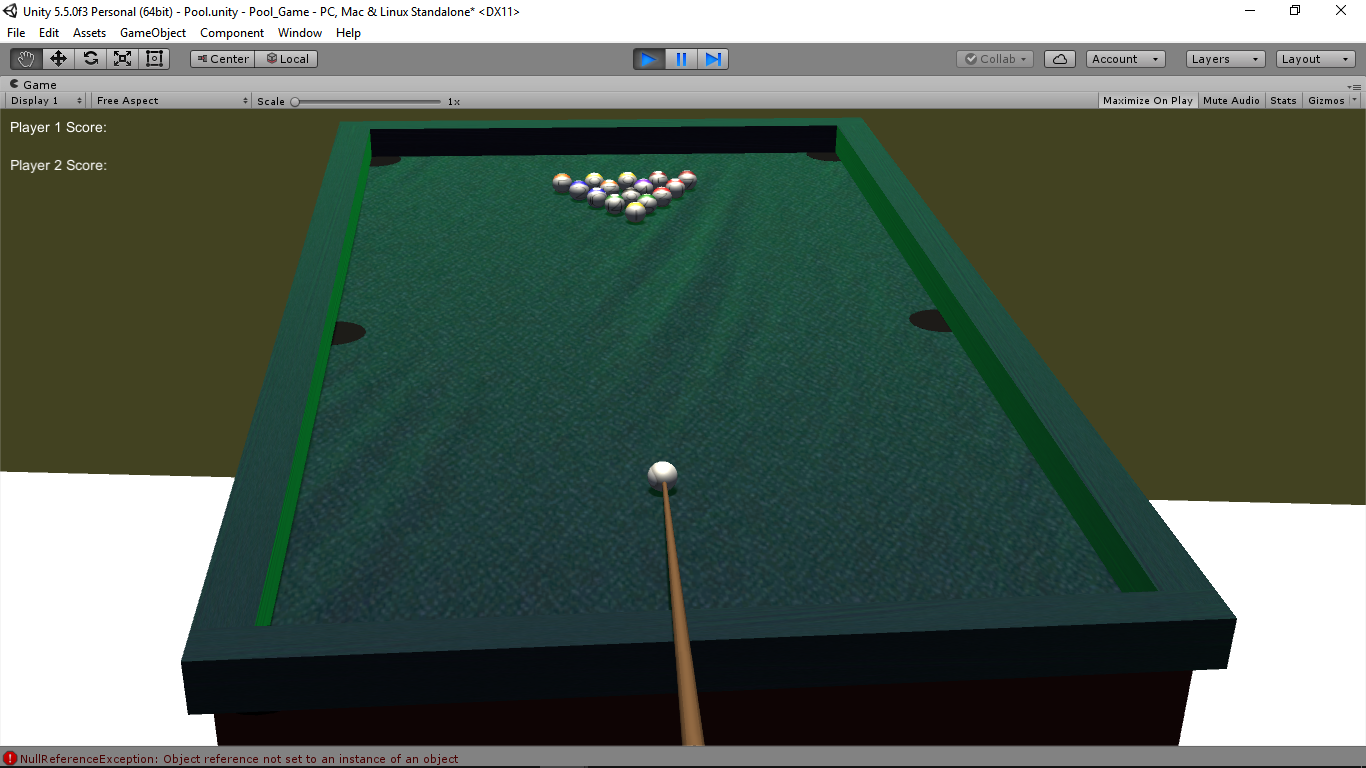
Interactive Media Development

Topic

CA2 – 2 Player Pool Game in Unity 3D

Students

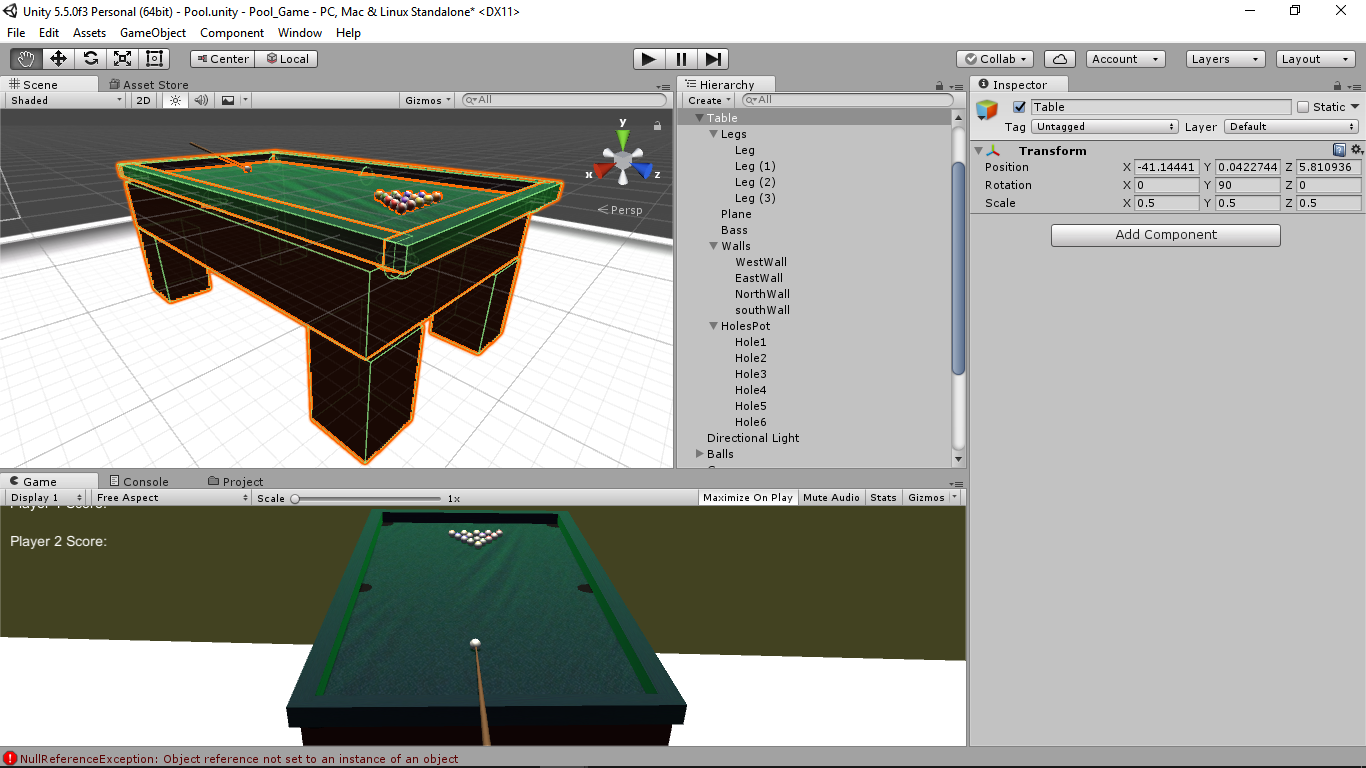
Thomas Murray X00105013



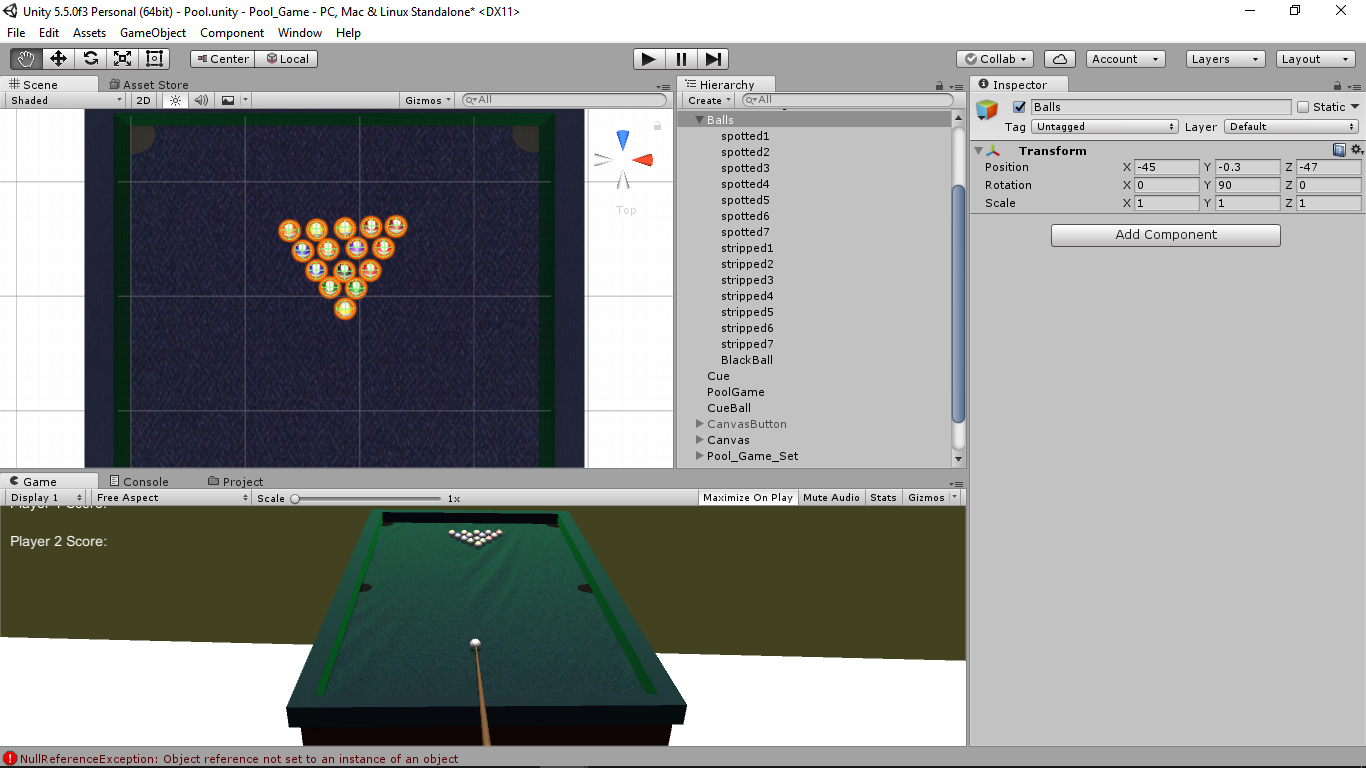
**Approach taken**

For this project I wanted to create a 3D pool game played from the users perspective.

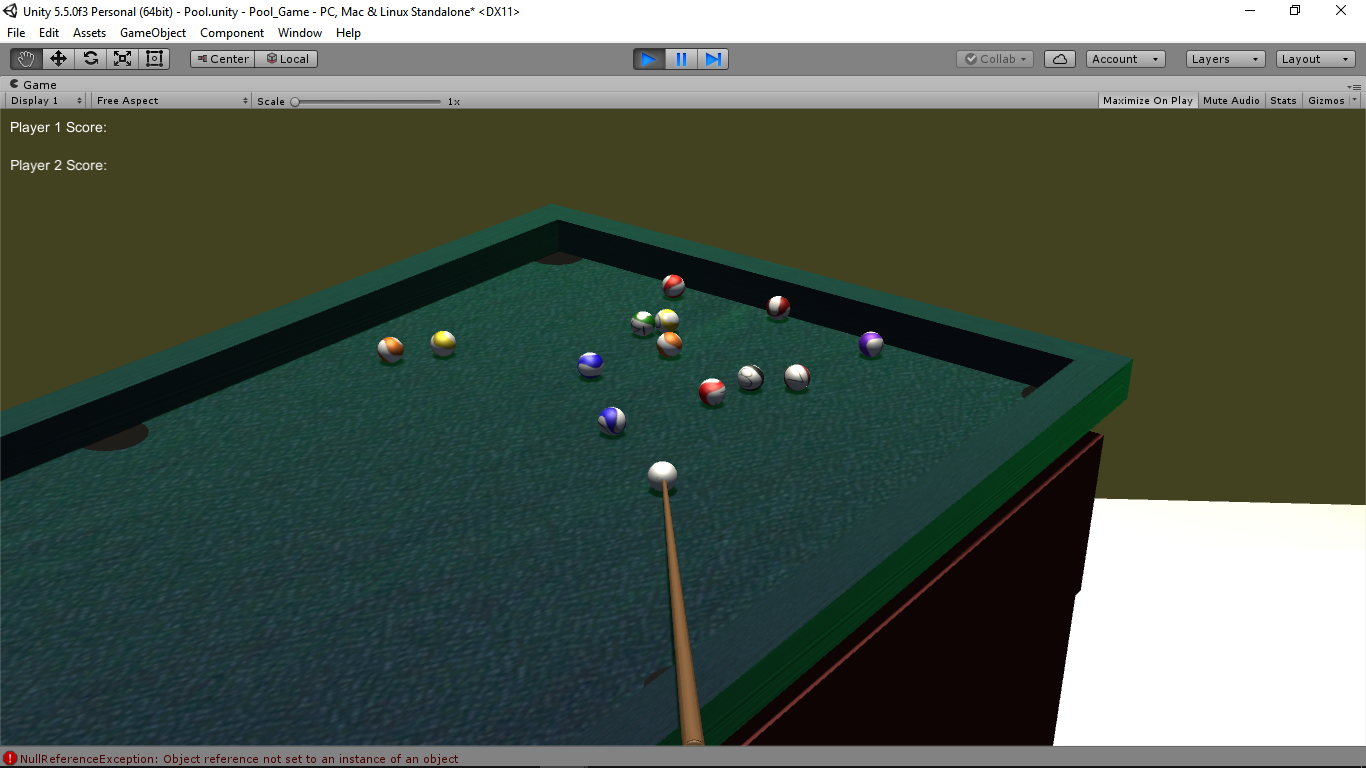
* First I had to provide a room to place the table.
* My next step was to create a table to be played on. This was done by building each part of the table separately and putting them all together. I did this because I wanted to use as little pre built assets as possible.



* The pool balls were my next step. This I did by creating each ball individually and the coating them in an asset I managed to get online.



* My next step was to create a cue that an turn 360 degrees around the cue ball and disappear and reappear.



**game feature**

* Cue Action
* Ball destruction
* Ball physics (gravity, friction)
* Directional light
* Repositioning of potted cue ball
* Camera movement

To add force to the ball hold down the mouse button anywhere on the screen, the longer you hold the harder it hits. Cue direction is inputted via the left/right direction keys.

**Challenges**

There were a lot of challenges with this project and I tried to do as much as I could without any help as this is the type of work I would like to try out because of its creative aspect but the biggest challenges are listed below.

The biggest challenges making this game was to get the:

* Cue movement
* Camera positioning
* Ball destruction
* Pocket mechanics
* General positioning (camera, balls, cue and table)
* Implementing the scoring
* Two player implementation
* Winner output
* Game rules

**Solutions**

Some of these problems I overcame using trial and error and the others I just ran out of time but feel I was very close to fixing. The biggest problem I ran into was with getting the game to switch from player to player. This I feel I could have solved if I had of focused all of my time on this project instead of having to jump between this and the fourth year project (I enjoyed doing this more) but the pressure on the other was very strong.

This is the stage of the dual player I was working when my time ran out.

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

using System;

/\*

\* IMD CA2 Pool Game

\* Thomas Murray

\* X00115013

\*/

public class Potted : MonoBehaviour {

public static bool ballOrder;

public static bool white, playerOne, playerTwo;

public static int playerOneScore, playerTwoScore;

public GameObject score1;

public GameObject score2;

public GameObject winner;

private string result;

private GameObject cueBall;

private bool potted;

private bool ball1, ball2, ball3, ball4, ball5, ball6, ball7, ball8, ball9, ball10, ball11, ball12, ball13, ball14, ball15;

private bool[] ballsLeft = new bool[] { true, true, true, true, true, true, true, true, true, true, true, true, true, true };

// Use this for initialization

void Start () {

white = true;

playerOne = true;

playerTwo = false;

playerOneScore = 0;

playerTwoScore = 0;

potted = false;

}

void Update()

{

setCountTextp1();

setCountTextp2();

var cueBallmotion = cueBall.GetComponent<Rigidbody>();

//Change player as soon as the cue ball stops

if (cueBallmotion.velocity == Vector3.zero)

{

if (playerOne == true)

{

playerTwo = true;

playerOne = false;

}

if (playerTwo == true)

{

playerOne = true;

playerTwo = false;

}

}

}

//Getters / Setters

public bool player1Get()

{

return playerOne;

}

public bool player2Get()

{

return playerTwo;

}

public void player1Set(bool stat)

{

playerOne = stat;

}

public void player2Set(bool stat)

{

playerTwo = stat;

}

public bool pottedGet()

{

return potted;

}

public void pottedSet(bool pot)

{

potted = pot;

}

//Balls potted

void OnCollisionEnter(Collision col)

{

GameObject temp;

Rigidbody rb;

if (col.gameObject.name == "spotted1")

{

Debug.Log("player 1 "+playerOne);

Debug.Log("player 2 " + playerTwo);

Destroy(col.gameObject);

ballsLeft[0] = false;

}

else if (col.gameObject.name == "spotted2")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[1] = false;

}

else if (col.gameObject.name == "spotted3")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[2] = false;

}

else if (col.gameObject.name == "spotted4")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[3] = false;

}

else if (col.gameObject.name == "spotted5")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[4] = false;

}

else if (col.gameObject.name == "spotted6")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[5] = false;

}

else if (col.gameObject.name == "spotted7")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[6] = false;

}

else if (col.gameObject.name == "stripped1")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[7] = false;

}

else if (col.gameObject.name == "stripped2")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[8] = false;

}

else if (col.gameObject.name == "stripped3")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[9] = false;

}

else if (col.gameObject.name == "stripped4")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[10] = false;

}

else if (col.gameObject.name == "stripped5")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[11] = false;

}

else if (col.gameObject.name == "stripped6")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[12] = false;

}

else if (col.gameObject.name == "stripped7")

{

changePlayer();

potted = true;

Destroy(col.gameObject);

ballsLeft[13] = false;

}

else if (col.gameObject.name == "BlackBall")

{

for (int i = 0; i < 14; i++)

{

if (ballsLeft[i] == true)

{

gameOverBlackBall();

}

}

if (playerOne == true)

{

playerOneScore += 1;

}

if (playerTwo == true)

{

playerTwoScore += 1;

}

gameOverScore();

Destroy(col.gameObject);

}

//cue ball potted

else if (col.gameObject.name == "CueBall")

{

temp = GameObject.Find("CueBall");

rb = temp.GetComponent<Rigidbody>();

StartCoroutine(MyMethod());

temp.transform.position = new Vector3(-40.16f, -0.6f, -21.65f);

rb.velocity = Vector3.zero;

if (playerOne == true)

{

playerOneScore -= 1;

playerOne = false;

playerTwo = true;

}

else if (playerTwo == true)

{

playerTwoScore -= 1;

playerTwo = false;

playerOne = true;

}

}

}

void setCountTextp1()

{

var score1st = score1.GetComponent<UnityEngine.UI.Text>();

score1st.text = String.Format("Player 1 Score: {0}", playerOneScore);

}

void setCountTextp2()

{

var score2nd = score2.GetComponent<UnityEngine.UI.Text>();

score2nd.text = String.Format("Player 2 Score: {0}", playerTwoScore);

}

//If black is potted before all balls are cleared

public void gameOverBlackBall()

{

if (playerOne == false)

{

result = string.Format("The winner is Player 1");

}

if (playerTwo == false)

{

result = string.Format("The winner is Player 2");

}

var text = winner.GetComponentInChildren<UnityEngine.UI.Text>();

text.text = result;

winner.GetComponent<Canvas>().enabled = true;

}

//Finish game

public void gameOverScore()

{

if (playerOneScore > playerTwoScore)

{

result = string.Format("The winner is Player 1");

}

else if (playerOneScore < playerTwoScore)

{

result = string.Format("The winner is Player 2");

}

else if (playerOneScore == playerTwoScore)

{

result = string.Format("It's a draw");

}

var win = winner.GetComponentInChildren<UnityEngine.UI.Text>();

win.text = result;

winner.GetComponent<Canvas>().enabled = true;

}

//Change player

public void changePlayer()

{

if (playerOne == true)

{

playerOneScore += 1;

playerOne = true;

playerTwo = false;

}

if (playerTwo == true)

{

playerTwoScore += 1;

playerTwo = true;

playerOne = false;

}

}

IEnumerator MyMethod()

{

Debug.Log("Before Waiting 5 seconds");

yield return new WaitForSeconds(5);

Debug.Log("After Waiting 5 Seconds");

yield break;

}

}

I will continue to work on this project as I really did enjoy the creative aspect. I could get the player switch working only when the white ball is potted but ran out of time to fully investigate this but I will.

**Bugs that need a restart**

* If the cue touches a pot it can affect the cue position when the cue ball is repositioned
* The cue ball can get stuck spinning at the side cushion causing the cue to not reappear

**What I would have liked to add and will**

* Second inset camera showing a top view
* Better textures to the table base
* Proper holes in the table to take away the need to destroy the objects
* Decorate the surroundings
* Put an actual light over the table
* UI controls to stop and start games
* Sound effects
* Power and strike in game UI which I did start but ran out of time

**References and citations**

* I got help with the Cue movement from <https://github.com/fgrehm/pucrs-unity3d-pool>