



# **R&D PROJECT REPORT**

## **“ROLL A BALL”**

**A 3D INTENSE TREASURE QUEST GAME**

**BY:**

**SAI VAMSI KRISHNA YERRAMSETTI**

**S1032599**



## Introduction:

**“Roll A Ball” is a 3D game where the player aka the “ball” has to find 7 treasures in a complex maze without touching the walls. He/She will use the arrows on the keyboard to navigate the ball through the maze. The slightest touch to the wall and the player loses the game.**

**I had created this game for a game development competition last month.**

**This game was developed on the Unity Platform and I used C# to code the game.**

**This game is designed for toddlers and kids as a pass time game but it was also developed for people who face mental disorders or finger/limb related disorders. It helps people with the above problems as an activity to help build up concentration and could be used in their treatment.**

## Description :

### *Property Focus.*

There are three main properties/components in the game.

- The maze/walls
- The ball
- The treasure chests

I programmed the ball to go at a certain speed and designed a complex maze where 7 treasure chests are hidden. The walls are programmed to detect any impact from the ball. The camera follows the ball from a small elevated height and turns whenever and however the ball turns. Whenever the ball picks up a treasure chest the treasure chest disappears and the counter is updated.

### *Product justification:*

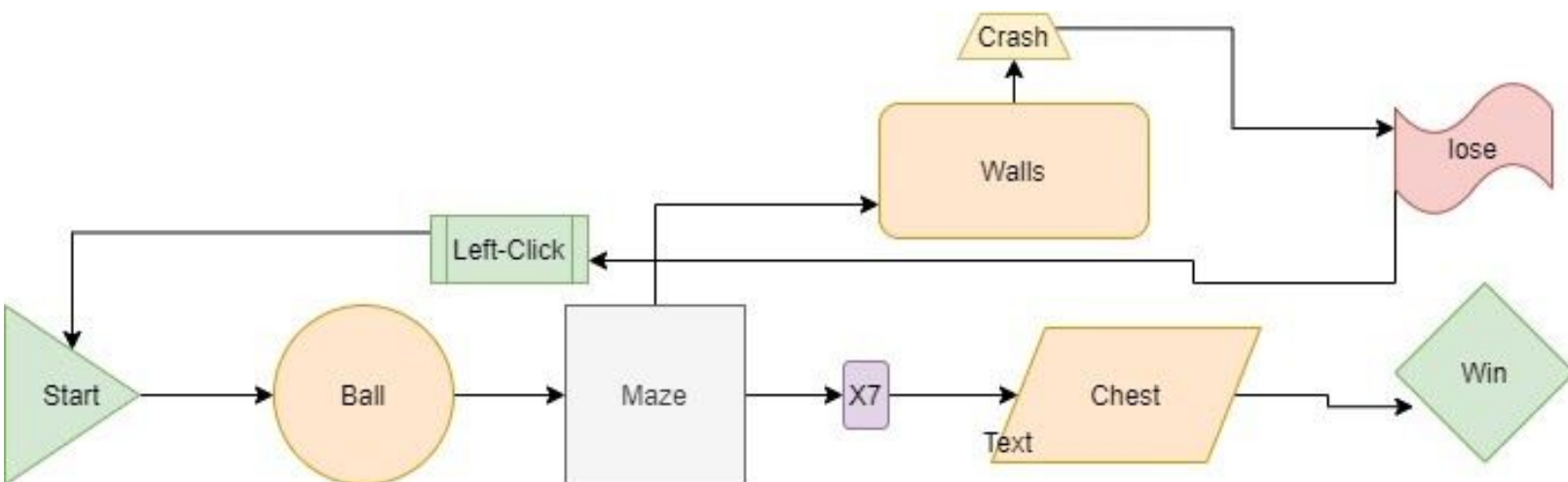
I believe this product is more than just a kids game. I relate to this game personally because it has slightly helped my aging grandfather have better control of his hands. He played the game several times and was kept telling how he has a finer grip at things especially at his fingertips, moreover, he says his concentration has improved.

I feel this game is not only challenging and fun but has a wide set of audiences. I am able to say all of this because I tested this game among my friends and family and they all gave similar feedback.

There are a lot of similar products like this however the design of my maze is unique. The concept of the game is simple, but it is a challenging one compared to other systems on the market.

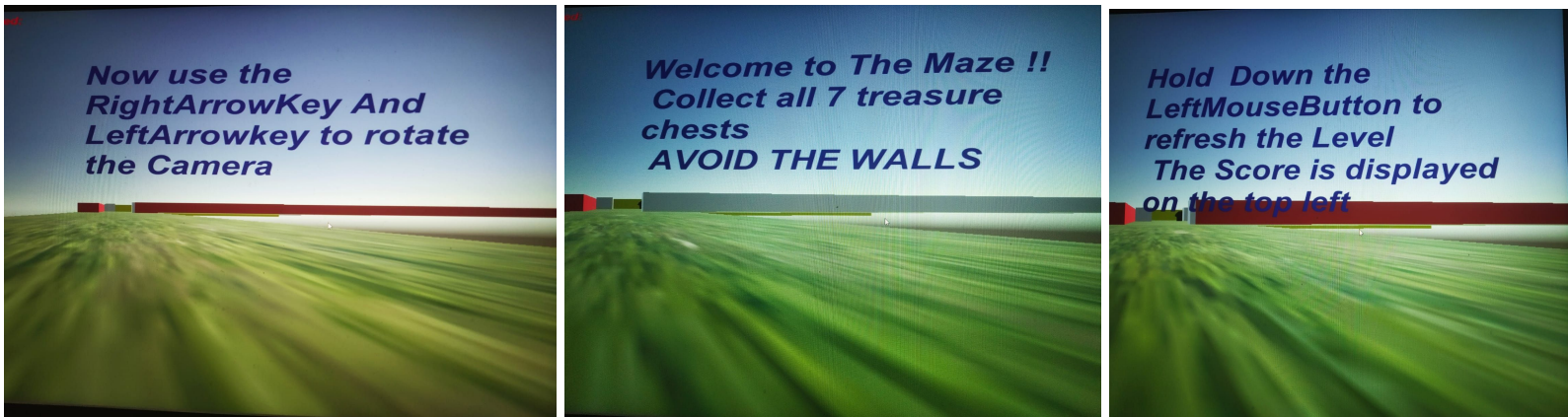
**Specifications:**

**Game Flow Structure:**



## Design:

### *Global design:*



There are three main properties/components in the game.

- The maze/walls
- The ball
- The Camera

### Ball:

I programmed the ball to go at a certain speed and designed a complex maze where 7 treasure chests are hidden.

(the code explanation for this is basically the controls, to be honest, there is no ball it's just the camera following the controls of the user.)

### Walls:

The walls are programmed to detect any impact from the ball.

**I had color-coded the walls(which i designed via unity) so that the player does not get lost all the time. The following code sets up the walls and takes care of collisions.**

using UnityEngine;

```
public class MazeCellEdge : MonoBehaviour {

    public MazeCell cell, otherCell;

    public MazeDirection direction;
}

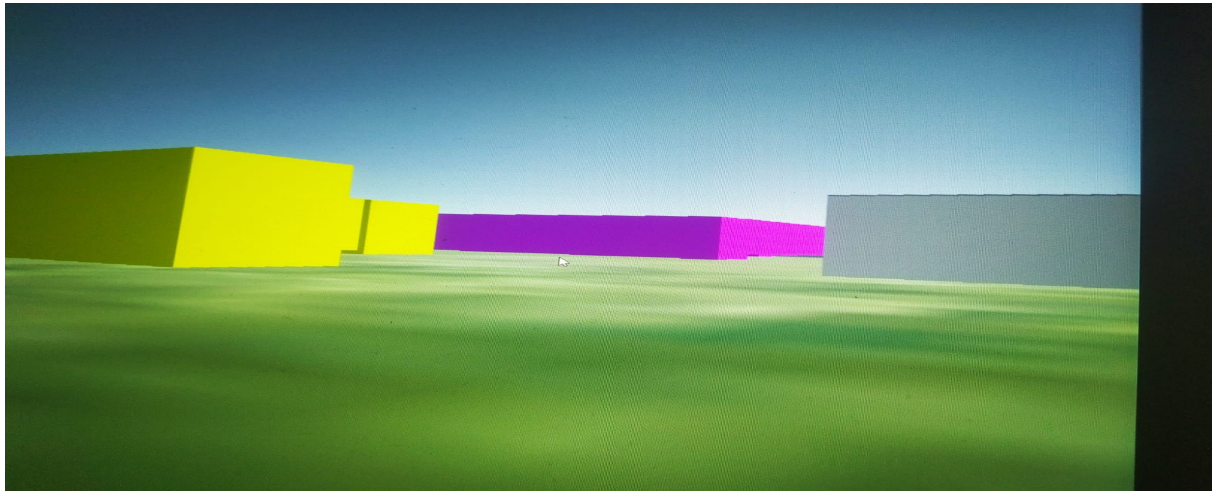
public void Initialize (MazeCell cell, MazeCell otherCell, MazeDirection direction) {
    this.cell = cell;
    this.otherCell = otherCell;
    this.direction = direction;
    cell.SetEdge(direction, this);
    transform.parent = cell.transform;
    transform.localPosition = Vector3.zero;
}

private MazeCellEdge[] edges = new MazeCellEdge[MazeDirections.Count];

public MazeCellEdge GetEdge (MazeDirection direction) {
    return edges[(int)direction];
}
```

```
}
```

```
public void SetEdge (MazeDirection direction, MazeCellEdge edge) {  
    edges[(int)direction] = edge;  
}
```



### Camera:

The camera follows the ball from a small elevated height and turns whenever and however the ball turns.

### Controls:

The arrow keypad on the keyboard controls the ball and its movement. The following code explains how the keys are used.

```
private MazeDirection currentDirection;
```

```
private void Look (MazeDirection direction) {  
    transform.localRotation = direction.ToRotation();  
    currentDirection = direction;  
}  
  
private void Update () {  
    if (Input.GetKeyDown(KeyCode.UpArrow) ||  
Input.GetKeyDown(KeyCode.UpArrow)) {  
        Move(MazeDirection.North);  
    }  
    else if (Input.GetKeyDown(KeyCode.RightArrow) ||  
Input.GetKeyDown(KeyCode.RightArrow)) {  
        Move(MazeDirection.East);  
    }  
    else if (Input.GetKeyDown(KeyCode.DownArrow) ||  
Input.GetKeyDown(KeyCode.DownArrow)) {  
        Move(MazeDirection.South);  
    }  
    else if (Input.GetKeyDown(KeyCode.LeftArrow) ||  
Input.GetKeyDown(KeyCode.LeftArrow)) {  
        Move(MazeDirection.West);  
    }  
}
```



## Chest:

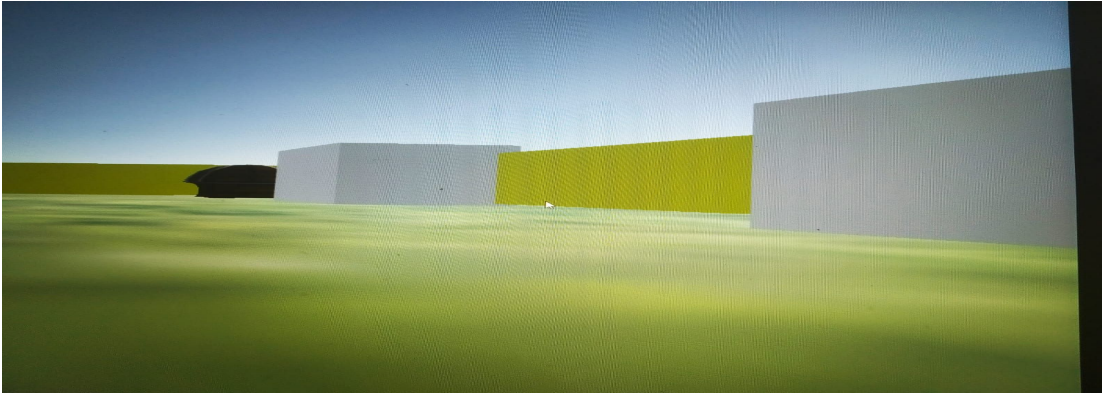
**Whenever the ball picks up a treasure chest the treasure chest disappears and the counter is updated.**

**The following code inputs the chests :**

```
public MazeChest ChestPrefab;

[Range(0f, 1f)]
public float ChestProbability;

private void CreatePassage (MazeCell cell, MazeCell otherCell, MazeDirection
direction) {
    MazePassage prefab = Random.value < ChestProbability ? ChestPrefab :
passagePrefab;
    MazePassage passage = Instantiate(prefab) as MazePassage;
    passage.Initialize(cell, otherCell, direction);
    passage = Instantiate(prefab) as MazePassage;
    passage.Initialize(otherCell, cell, direction.GetOpposite());
}
```



### Counter:

**The counter on top of the screen indicates how many chests you have collected.**

```
public class Counter: MonoBehaviour {
```

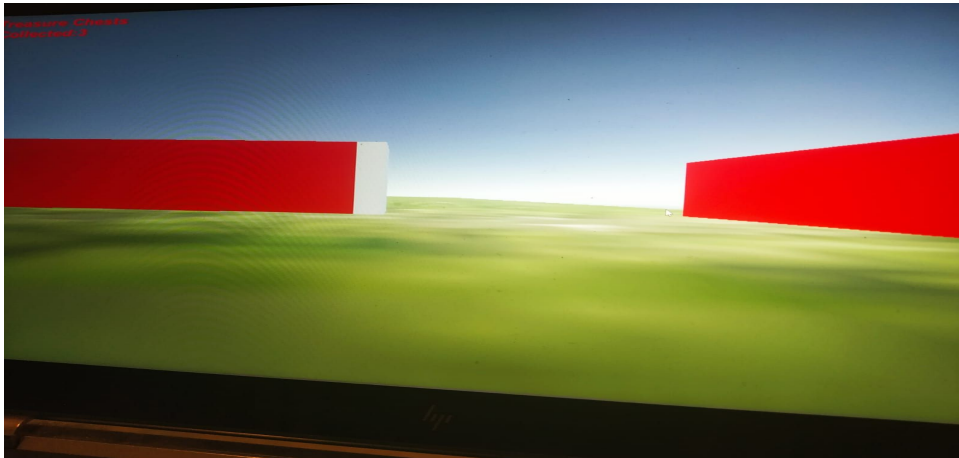
```
    Int count=0;
```

```
    count++;
```

```
    If count==7;
```

```
    dispWinner();
```

```
}
```



### Reset:

**When the player hits the wall the ball goes into an indefinite loop and the camera starts spinning indicating he has lost the game. When this happens the players must just LeftClick to reset.**

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class GameManager : MonoBehaviour {
```

```
    private void Start () {
```

```
        BeginGame();
```

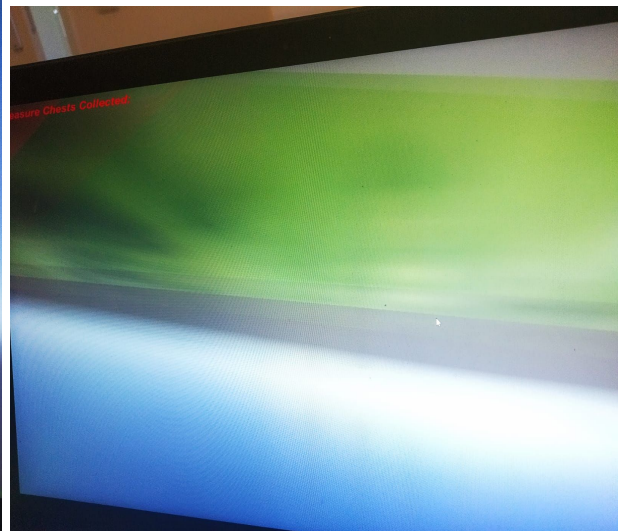
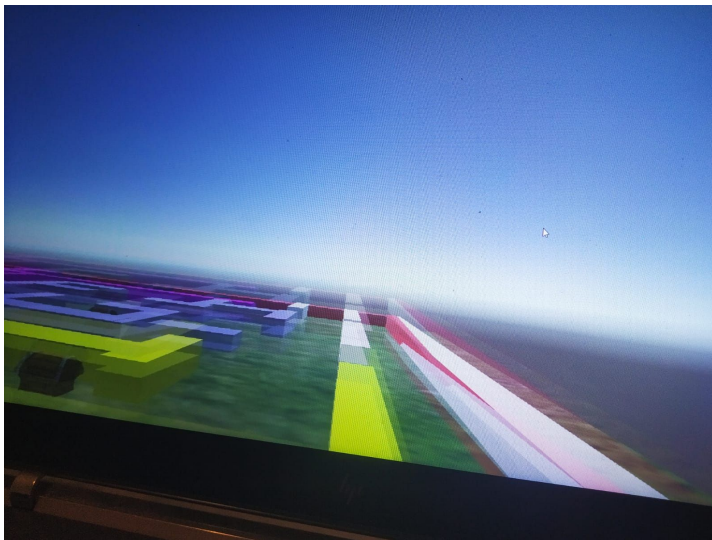
```
    }
```

```
    private void Update () {
```

```
        if (Input.GetKeyDown(KeyCode.LeftClick)) {  
            RestartGame();  
        }  
    }  
  
    private void BeginGame () {}  
  
    private void RestartGame () {}  
}
```

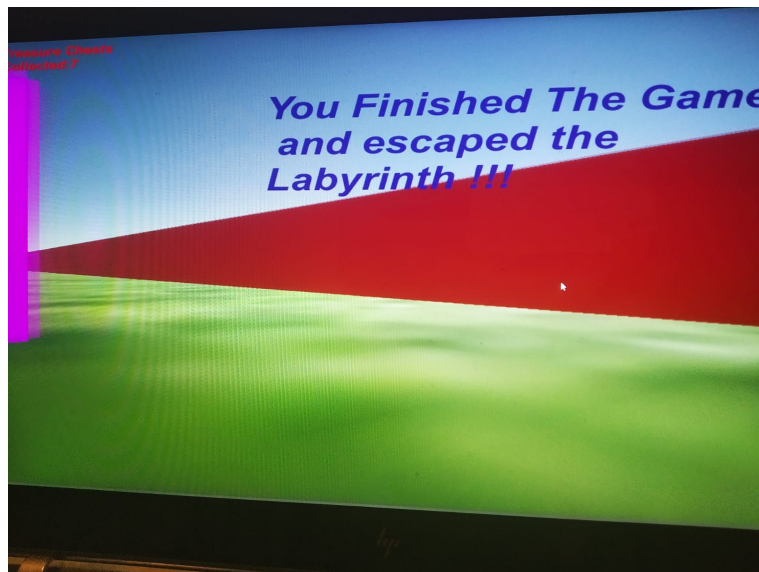
### Losing:

For this i did not use any C# i just adjusted the nature of the wall components to “camera free “ . So when the user touches the wall he just spins and has to reset.



### Winning :

When the user has finished the game by collecting all the chests a message will pop up indicating he has won the game(the counter reaches 7), he can then click R and play the game again basically using the reset option.



### *Detailed design:*

I used unity to create the physical structure of the maze. So the size and color of the maze + the sky and ground were all of the unity's template textures which I used. I programmed the camera and movement of the ball in the ball class and the chests in the chests class. I mainly used if statements and combined each class with the Unity set of Physics. So the speed of the camera and rate of rotation and etc were are set via unity. If you wish to see all of this then you must download unity and play the game there. All my code and numbers are more readable there. I used the arrows

keys and left-click button only, these were set via Unity. Through C# I established most of my classes and if statements there. I also used unity to track the movement. So when the players start from the original position and move forward the at each concurrent spot a new message pops up, this was done via unity.

## CODE ANALYSIS AND EXPLANATION:

I could not export the code file from unity hence I gave a detailed explanation of all the major components as suggested by my TA.

I explained all the important parts of the project above.

Unity is a platform that helped design the maze and basically the appearance.

For each component that I defined in Unity, I used C# to program it . Like the walls and chests and the movement(the code is above). The size vectors and shapes of all of the objects you see were made via the unity platform. Even the text and its size and font which is displayed is all using unity.

### *Design justification:*

I believe the design and code of the game are clear and can easily be interpreted by a coder who has experience with Unity and C#.

We do not encounter any glitches and the treasure counting system works perfectly. All the major components including the camera all work according to the blueprint. The code is well readable and the characteristics of each class are well aligned and self-explanatory.

## Evaluation :

I am really happy with my project and had a lot of fun building it.

The only issue I faced was the occasional glitching of the wall when the ball passes by. I feel it was caused by the number of blocks I used to make some walls.

I learned a lot about the language C# itself and how to use it.

Programming the camera and the ball were the most difficult parts but it worked out in the end. Unity is a really fun and amazing platform to work on.

In the future, I thought I could advance the game by making it a multilevel maze with more chests to find and better movement of the ball.

Important note:

I have coded this project two months ago at the very start of this course. In the by, you will see a few other names in this project as well. These people were not at ALL involved in the development/coding/making of this project at all. They presented this project which I made at a game development competition in India hence I included their name. I thought it would be unethical if I removed their name hence I kept it in, however, I repeat they had NO involvement in the making of the is a project I have written and signed statements from the 3 other mentioned people as proof. I took these precautions and am explaining my self to ensure the TAs don't misunderstand the authenticity of my work.

The only help I got from this project was from a few youtube videos, which I have linked below.

<https://www.youtube.com/watch?v=L8sfSBLBRGY>

<https://www.youtube.com/watch?v=ma2QGWI064Y>

<https://www.youtube.com/watch?v=5fZnwmcdAR8>