**REPORT ON**

“Software Engineering : Mini Project ”

# Bachelor of Technology

## CSE Dept.



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPUR TECHNICAL UNIVERSITY**

**March, 2023**

**Team Name: CYPHER AUGMENTORS**

**Team Members:**

**Reagan Sagolsem (2001CS0221)**

**Moirangthem Daniel singh(2001CS0225)**

**Barlin Chongtham(2001CS0237)**

**Meinam Ashaad(2001CS0217)**

## 

## ACKNOWLEDGEMENT

We take this opportunity to bestow our deep sense of gratitude and indebtedness to  **Sir Hanjabam Saratchandra , Head of Department Computer Science and Engineering, Manipur Technical University , for his co-ordination in extending every possible facility for the completion of this project.**

for their valuable and continuous encouragement.

We are really grateful because we managed to complete our Game “Arithmetic Destination” within the given time. The assignment cannot be completed without the effort and co-operation from group members , Daniel , Barlin , Reagan and Ashad .

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| Sl. No | Contents | Page No |
| 1 | Chapter 1: Introduction  - Abstract   * Team Intro | 1  2 |
|  | Chapter 2: List of Hardware & Software used   * Project design/ functionalities | 3 |
| 3 | Chapter 3:Implementation details  - Project codes | 4-16 |
| 4 | Chapter 4: Conclusion | 17 |

# CHAPTER 1

# **Team Introduction**

**Introducing the team of four who came together to create a new game. First up, we have our coder, Reagan , who brought the game to life with his programming skills. His dedication and attention to detail helped ensure that the game runs smoothly and is highly enjoyable to play.**

**Assisting in the creation of the game were two level designers, Barlin and Ashad. These two individuals worked tirelessly to create challenging and engaging levels that keep players on their toes. They used their creative skills to design intricate layouts that test players' problem-solving abilities while keeping them entertained.**

**Last but not least, we have our user interface (UI) expert, Daniel. Their keen eye for design and intuitive user interface made the game visually stunning and easy to navigate. With their help, players will have a seamless and immersive experience as they play the game.**

1

**Introduction**

**Abstract**

Arithmetic Destination is the name of the game , it is a small size 2D tile game where the player has to move our cute character towards the portal. The player has to use the calculator based buttons to instruct the character to move using arithmetic skills.

**Objectives**

The objective of our game is to help players improve their basic arithmetic skills and speed thinking

2

**CHAPTER 2**

List of Hardware & Software used

|  |  |
| --- | --- |
| **USERS** | **DEVELOPER** |
| System Specification :  Android | System Specification:  i3-11th Gen, 256 GB SSD and above  Cooler Pad, Keyboard, Mouse |
| Language used:  C# | Softwares :   * Photoshop * VS code * Sketchbook * Unity Game Engine |

3

**Chapter 3**

**Implementation Details**

**Visual Studio Code**

**This is the software where we designed our main game code for android version of our 2D game**

C#

The programming language that we are using for this game is C# it is a language that was developed by Microsoft and runs on .NET Framework.

Photoshop

The assets and UI were designed and resized using photoshop and exported as png file format.

**Sketchbook**

The characters and its animation were design using this application.

**Unity Game Engine**

The software where we managed and made our 2D game which were combined it into a project

**4**

**PROJECT CODES**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using TMPro;

using UnityEngine.Tilemaps;

using UnityEngine.SceneManagement;

public class MovementCalculation : MonoBehaviour

{

//UI management

[SerializeField] private GameObject gameOverMenuUI;

[SerializeField] private GameObject levelCompleteMenuUI;

//calculator

[SerializeField] private int maxDigits = 4;

[SerializeField] private TextMeshProUGUI displayText;

[SerializeField] private Text resultText;

[SerializeField] private Text detector;

//character animation

[SerializeField] private Animator anim;

//character sound

[SerializeField] private AudioSource moveSound;

[SerializeField] private AudioSource deathSound;

5

[SerializeField] private AudioSource stubDeathSound;

[SerializeField] private AudioSource finishSound;

//board

[SerializeField] private Tilemap tilemap;

//direction detector

[SerializeField] private Text upTilesText;

[SerializeField] private Text rightTilesText;

[SerializeField] private Text downTilesText;

[SerializeField] private Text leftTilesText;

//timer

[SerializeField] private float timeRemaining = 60;

[SerializeField] private Text timerText;

[SerializeField] private float startBlink = 5f;

[SerializeField] private float blinkSpeed = 0.5f;

[SerializeField] private Outline outline;

private bool timerIsRunning = true;

private float firstNumber;

private float secondNumber;

private float result;

private string operation;

private Vector3Int currentTile;

private bool enterToggle = false;

private float tileSize = 1f;

6

private static bool gameOver = false;

private static bool levelComplete = false;

private Transform[,] tiles;

private void Start()

{

Clear();

UpdateCurrentTile();

}

void Update()

{

UpdateCurrentTile();

//count tiles

CountTiles(new Vector3Int(0, 1, 0), upTilesText); // up

CountTiles(new Vector3Int(1, 0, 0), rightTilesText); // right

CountTiles(new Vector3Int(0, -1, 0), downTilesText); // down

CountTiles(new Vector3Int(-1, 0, 0), leftTilesText); // left

//tume start

if (timerIsRunning)

{

timeRemaining -= Time.deltaTime;

if (timeRemaining <= 0)

{

timeRemaining = 0;

timerIsRunning = false;

7

DeathMenu();

}

UpdateTimerText();

if (timeRemaining <= startBlink)

{

StartCoroutine(Blink());

}

}

}

private void DeathMenu()

{

gameOver = true;

gameOverMenuUI.SetActive(true);

timerIsRunning = false;

}

private void LevelCompleteMenu()

{

levelComplete = true;

levelCompleteMenuUI.SetActive(true);

}

private void OffBool()

{

anim.SetBool("move", false);

}

private void UpdateCurrentTile()

{

currentTile = tilemap.WorldToCell(transform.position); 8

}

private IEnumerator Blink()

{

outline.enabled = true;

while (timerIsRunning && timeRemaining <= startBlink)

{

outline.enabled = !outline.enabled;

yield return new WaitForSeconds(blinkSpeed);

}

outline.enabled = true;

}

//update time

void UpdateTimerText()

{

int minutes = Mathf.FloorToInt(timeRemaining / 60);

int seconds = Mathf.FloorToInt(timeRemaining % 60);

int milliseconds = Mathf.FloorToInt((timeRemaining - Mathf.FloorToInt(timeRemaining)) \* 99);

timerText.text = string.Format("{0:00}:{1:00}", seconds, milliseconds);

}

private void CountTiles(Vector3Int direction, Text text)

{

int count = 0;

Vector3Int targetTile = currentTile + direction;

while (tilemap.HasTile(targetTile))

{

Collider2D[] colliders = 9 Physics2D.OverlapCircleAll(tilemap.GetCellCenterWorld(targetTile), tileSize / 2);

foreach (Collider2D collider in colliders)

{

if (collider.CompareTag("Wall"))

{

text.text = count.ToString();

return;

}

}

count++;

targetTile += direction;

}

text.text = count.ToString();

}

private void DChange()

{

enterToggle = !enterToggle;

if (enterToggle)

{

detector.text = "Y";

}

else

{

detector.text = "X";

}

}

public void Move(Vector3Int direction) 10

{

moveSound.Play();

Vector3Int targetTile = currentTile + direction;

Collider2D[] colliders = Physics2D.OverlapCircleAll(tilemap.GetCellCenterWorld(targetTile), tileSize / 2);

foreach (Collider2D collider in colliders)

{

if (collider.CompareTag("Wall") || collider.CompareTag("Water"))

{

return;

}

}

Vector3 start = transform.position;

Vector3 end = tilemap.GetCellCenterWorld(targetTile);

RaycastHit2D[] hits = Physics2D.LinecastAll(start, end);

foreach (RaycastHit2D hit in hits)

{

if (hit.collider.CompareTag("Wall") || hit.collider.CompareTag("Water"))

{

return;

}

if (hit.collider.CompareTag("Trap"))

{

timerIsRunning = false;

anim.SetTrigger("death");

deathSound.Play();

}

if (hit.collider.CompareTag("TrapStub"))

{ 11

timerIsRunning = false;

anim.SetTrigger("deathStub");

//stubDeathSound.Play();

}

if (hit.collider.CompareTag("Finish"))

{

timerIsRunning = false;

finishSound.Play();

Invoke("LevelCompleteMenu", 2f);

}

}

if (tilemap.HasTile(targetTile))

{

transform.position = tilemap.GetCellCenterWorld(targetTile);

currentTile = targetTile;

}

}

public void Clear()

{

firstNumber = 0;

secondNumber = 0;

result = 0;

operation = "";

displayText.text = "0";

resultText.text = "0";

}

public void Delete()

{//error in delete firstNumber 12

int length = displayText.text.Length;

if (length > 0)

{

displayText.text = displayText.text.Substring(0, length - 1);

if (operation == "")

{

firstNumber = Mathf.Floor(firstNumber / 10);

resultText.text = firstNumber.ToString();

}

else

{

secondNumber = Mathf.Floor(secondNumber / 10);

result = CalculateResult();

resultText.text = Mathf.Ceil(result).ToString();

}

}

}

public void Number(float number)

{

if (operation == "")

{

if (Mathf.Floor(Mathf.Log10(firstNumber) + 1) < maxDigits)

{

firstNumber = firstNumber \* 10 + number;

}

displayText.text = firstNumber.ToString();

resultText.text = firstNumber.ToString();

}

Else 13

{

if (Mathf.Floor(Mathf.Log10(secondNumber) + 1) < maxDigits)

{

secondNumber = secondNumber \* 10 + number;

}

displayText.text = firstNumber.ToString() + operation + secondNumber.ToString();

result = CalculateResult();

resultText.text = Mathf.Ceil(result).ToString();

}

}

public void Operation(string op)

{

operation = op;

displayText.text = firstNumber.ToString() + operation + secondNumber.ToString();

}

public void Equal()

{

int result = (int)CalculateResult(); ;

if (firstNumber == 0 && secondNumber == 0 && operation == "")

{

DChange();

return;

}

if (operation == "" || firstNumber == 0 && secondNumber == 0)

{

displayText.text = "0";

return; 14

}

resultText.text = Mathf.Ceil(result).ToString();

displayText.text = firstNumber.ToString() + operation + secondNumber.ToString() + "=" + Mathf.Ceil(result).ToString();

firstNumber = 0;

secondNumber = 0;

operation = "";

if (enterToggle)

{

anim.SetBool("move", true);

Move(new Vector3Int(0, result, 0)); // move down

}

else

{

anim.SetBool("move", true);

Move(new Vector3Int(result, 0, 0)); // move right

}

}

private float CalculateResult()

{

switch (operation)

{

case "+":

return firstNumber + secondNumber;

case "-":

return firstNumber - secondNumber;

case "×":

return firstNumber \* secondNumber; 15

case "÷":

if (secondNumber == 0) // prevent dividing by zero

{

return 0;

}

else

{

return firstNumber / secondNumber;

}

default:

return 0;

}

}

}

16

**CONCLUSION**

As a conclusion, we can say that this project was a great experience. Thanks to this project, we acquired deeper knowledge concerning our technical skills but we also personally benefited. Currently Game are part of one of our daily lifes , and developing it is also one of the most popular challenge for by developers worldwide. If we surf internet we can see millions of websites about various 2D , 3D games. we learn to live in a different environment .Indeed, we grew more independent in work and also in everyday life. we realized that we could do more things than we thought like learning new things from each other.There are huge opportunities available for the students who want to work in this field. Many private and public organizations hire game developers for their work . With the rapid advent of online industry, the demand of Game development professionals is increasing and this has created a huge job opportunity for the aspirants in the upcoming days. Also an experienced person in this field can also work as a freelancer; there are many online companies which provide online projects to the individuals.

17