Using Virtual Reality for visualizing theoretical operating system concepts

A PROJECT REPORT

Submitted by

T.SAIPREM [RA1911003010974]
JATIN ARORA[RA1911003010976]

Under the Guidance of

Dr. M.KOWSIGAN
Assistant Professor, Department of Networking and Communications

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in INTERNET OF THINGS

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603 203

MAY 2022

Department of Networking and Communications
SRM Institute of Science & Technology
Own Work Declaration Form

Degree/ Course : **B. TECH/CSE-CORE**

Student Name : **T.SAIPREM/ JATIN ARORA**

Registration Number: **RA1911003010974/ RA1911003010976**

Title of Work : **Using Virtual Reality for visualizing theoretical operating system concepts**

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

T.SAIPREM [RA1911003010974]
JATIN ARORA[RA1911003010976]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B. Tech project report titled **"Using Virtual Reality for visualizing theoretical operating system concepts"**

This is the bonafide work of **Mr. T.SAIPREM and Mr. JATIN ARORA,** who carried out the project work under our supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE

Dr. M.KOWSIGAN
SUPERVISOR
Assistant Professor
Department of Computing technologies

SIGNATURE

DR. M.Pushpalatha
HEAD OF THE DEPARTMENT
Professor
Department of Computing technologies

Signature of the Internal Examiner

Signature of the External Examiner

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C.Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr. K.Annapurani Panaiyappan** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Program Coordinators **Dr. S.Suresh,** Professor and Panel Head **Dr. P Balamurgan,** Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**, Dr. K.C. Prabu Shankar,** Assistant professor, Department of Computer Science and engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. M.Kowsigan,**Assistant professor, Department of Computing technologies, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. He provided me with the freedom and support to explore the research topics of my interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Department of Computing technologies staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support, and encouragement.

T.SAIPREM

JATIN ARORA

ABSTRACT

Engaging learning environments for studying complex materials in higher education are difficult to implement because the content is mostly theoretical and difficult to visualize. This issue motivated the use of Mobile Augmented Reality in this study. The goal of this study is to design and develop a mobile augmented reality application that can enhance student's perceptions of the operating system course. The application was developed using three levels of knowledge (understanding, applying and analyzing) and included the integration of video, audio, graphic and textual information. This application is intended to be a good introduction to the complex material in your chosen course and is extendable to different levels of knowledge and subjects.

TABLE OF CONTENTS

CHAPTER NO.	CHAPTER NAME	PG NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	x
1.	**INTRODUCTION**	**1**
	1.1 OVERVIEW	1
	1.2 SCOPE	1
	1.3 RESEARCH OBJECTIVES	2
	1.4 PROBLEM STATEMENT	2
	1.5 MOTIVATION	3
	1.6 MODEL OBJECTIVES	3
	1.7 INNOVATION IDEA	3
	1.8 PURPOSE OF THE PROJECT	3
2.	**LITERATURE SURVEY**	**4**
	2.1 HOUSE PRICE DETERMINATION	4
	2.2 ISSUES OFR THE GENERAL PUBLIC	4
	2.3 HOW REGRESSION CAN BE USEFUL	4
	2.4 SIMPLE LINEAR REGRESSION	5
	2.5 DATASET RELIABILITY	5
	2.6 GENERAL FEATURES	5
	2.7 GRAPHICAL ANALYSIS	6
	2.8 ECONOMIC FACTORS	6
	2.9 OUTLIERS	7
	2.10 K FOLD CROSS VALIDATION	7
3.	**PROPOSED METHODOLOGY**	**8**
	3.1 OVERVIEW	8
	3.2 LINEAR REGRESSION	8
	3.3 REASONS TO USE	10
	3.4 DISADVANTAGES	10
4.	**MODULES AND IMPLEMENTATION**	**10**

	4.1 BLOCK DIAGRAM	11
	4.2 SEQUENCE DIAGRAM	12
	4.3 MODULES UNDERTAKEN	13
	4.4 IMPLEMENTATION	15
	4.5 FEATURE ENGINEERING	18
	4.6 OUTLIER REMOVAL	23
5.	**RESULTS AND DISCUSSION**	
	5.1 BEST MODEL	40
	5.2 FINAL OUTPUT	42
6.	**CONCLUSION AND FUTURE ENHANCEMENTS**	44
	REFERENCES	46
	APPENDIX	47
	PLAIGARISM REPORT	52
	PAPER PUBLICATION	53

LIST OF FIGURES

FIGURE NO.	FIGURE NAME

LIST OF ABBREVIATIONS

ABBREVIATIONS	Expansion
W.R. T	With Respect To
P.D	Pandas
SNS	Seaborn

ML	Machine Learning
AI	Artificial Intelligence
VS	Versus
DF	Data Frame

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Some of the most important content in operating system course is CPU scheduling, synchronization, and deadlocks. These are theoretical concepts that present a major challenge in teaching students. A series of integrated educational simulators were developed to support the teaching and learning of operating systems and were used to introduce students to modern operating system concepts. Virtual Reality (VR), a new technology of increasing interest in the all the industry, it is a potential technique that can be used by students to study complex material. VR can be used to better explain complex concepts by overlaying relevant information. Therefore, the goal of this research is to design and develop a virtual augmented reality application that can improve student's perception of selected material in operating system courses.

1.2 SCOPE

The goal of this research is to design and develop a Virtual reality application that can improve student's perception of selected material in operating system courses. In order to identify the effectiveness of VR in learning selected topics of operating system. We have divided the research into 4 parts.

1st part is to understand the virtual and learning object.

2nd part is to design the application.

3rd part is to developed the application.

4th part is to implement the developed application.

1.3 RESEARCH OBJECTIVES

Hence, we need to build a model which can do the job for us. In order to start acting upon our idea, we need to be clear about the general idea and the process which we're going to follow to reach the desired goal. So, picking up is the suitable choice to move forward with the thought idea. Next comes what algorithm we're going to use to solve the target problem. There are various algorithms in machine learning ranging from simple to highly complex ones. We need the one which is easy to apply and understand and most importantly gives us precise results. For our target problem, we decided to use the Simple Linear Regression Algorithm.

1.4 PROBLEM STATEMENT

- In this project, we will develop and create a game which will help students to play and understand scheduling algorithm.

1.5 MOTIVATION

To us the motivation behind this project is that finally we could see how all the OS topics we have studied is applied in real life. It's not only interesting but also very useful how we can use our game to bring out something that would be useful to the general public.

1.6 MODEL OBJECTIVES

- Helping students to learn the OS topics with the help of this game
- After reading any concept we can't remember it for a long time, so with the help of our game a student can remember the concept for a long time.

1.7 INNOVATION IDEA

We have the knowledge and implemented the knowledge in a pratical format. We have revolutinise the idea of studying and teaching an concept. Traditional way of learning was teaching the concept directly to student but we have tried to implement it in the way of game through which student will have interest to learn the topic.

1.8 PURPOSE OF THE PROJECT

The idea behind the project is to developed a game using unity software. This project aims to apply scheduling algorithms to get a visual understanding of the OS concept.

CHAPTER 2

LITERATURE SURVEY

2.1 TYPE OF LEARNING

2.1.1 GAME BASED LEARNING

In game-based learning, course content is presented to learners while they are engaged in game playing. The primary goal of this approach is to improve learner motivation .

The game itself is not the main focus of the course; its use is only part of the teaching strategy. For example, educational games can be integrated with cooperative learning. When students are shown an abundance of colorful screen designs coupled with sound effects and instructional materials, they are motivated to learn, and they learn better. Researchers have applied game-based learning in a variety of educational settings.

In one study, researchers found that students who played a video game designed to improve their reading comprehension skills learned more than those who did not play at all. In another study, researchers found that when students were taught about an unfamiliar subject using an educational video game instead of traditional methods, they retained more information than those who did not play at all.

2.1.2 COOPERATIVE LEARNING

Cooperative learning is a type of group learning that emphasizes mutual dependency and cooperation. In performing each learning task, group members work together to achieve the same learning goal. There are various notions of

cooperative learning, with some groups referring to cooperative learning as discussion and peer assistance, while others refer to collaborative work or cooperative problem-solving. Despite this divergence in definitions, the general consensus is that for a group to achieve success with cooperative learning, members should be engaged in group discussion, member interaction, and peer assistance.

2.1.3 Game-Based Learning by Digital Design

Researchers can add the element of game playing to a course to increase learners' motivation and attract their attention, thereby helping them focus. It is not easy to design a game that is both integrated with existing course content and interesting. In this study, we propose some possible approaches for designing online games for learning purposes. First, we present a conceptual framework of game playing that helps us understand why people like playing games. We also explain how game play serves as a motivator for learners. Next, we discuss how game design can be used to increase learning motivation and interest in learning tasks. Finally, we present some methods for developing online games that are appealing to learners.

2.2 EXPERIMENTS CONDUCTED BY Chung Yuan Christian University.

This experiment was conducted on a class of students at Chung Yuan Christian University in the Department of Information and Computer Engineering. The participants were 128 students who took the Operating Systems course in the Spring semester of 2011. The two classes naturally became the experimental group and the control group, with 63 participants in the experimental group and 65 in the control group. The participants' learning motivation, process, and achievement were evaluated both before and after the experiment through a

questionnaire designed by Pintrich et al. that assesses three dimensions of learner attitude. Wilke explains that only the first dimension of attitude pertains to learner motivation; this is how well people feel about their own achievements or performance.

2.3 GENERAL FEATURES

Bin-Shyan Jong et al, have written a paper which determines that apart from the general features that we have considered like the type of graphics is user-friendly and the game shouldn't have many actions. The game shouldn't be too much graphic intense so it can be played in all type of laptop. The game should be able to play in platforms. In this study, an online game was developed to enable students to learn cooperatively. The findings indicate that students' desire to win the game motivates them to learn from online course materials before they play, which in turn can enable them to achieve better learning outcomes.

The study was conducted with a sample of 148 undergraduate students in the University of Santo Tomas (UST). The participants were selected through convenience sampling and were asked to complete a questionnaire about their motivation for participating in cooperative learning activities. The data regarding their participation in cooperative learning activities were collected from a survey administered via an online survey tool.

The results show that most of the students prefer cooperative learning activities over individualized instruction because it allows them to learn at their own pace and also gives them an opportunity to interact with other classmates. Although most of them enjoy working together, some of them believe that it is difficult for some learners who are not good at collaborating with others.

2.4 DESIGNING ANALYSIS

In this study, the effectiveness of CKO was evaluated through a pretest–posttest research design. The study was intended to meet two objectives: (a) to provide a template for designing knowledge management simulation games, and (b) to determine the effectiveness of CKO through an empirical study which involved 32 final-year Business Studies students reading an elective module entitled Knowledge Management Systems in an institute of higher education in Singapore.

The findings confirmed that CKO was a viable and effective instructional tool for imparting knowledge to the participants. In addition, the scores obtained from CKO had a moderating effect on the participants' attitude towards the subject matter.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 OVERVIEW

Couples iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model It provides the potential for rapid development of an increasingly more complete version of the software. Using spiral, software

developed in a series of evolutionary releases. Early iteration, the release might be on paper or prototype. Later iteration, more complete version of the software.

The spiral model is a way to develop software by repeatedly improving interim versions until they reach a desired state. The first step is to create a prototype of an idea and test it using customers, users and other stakeholders who are involved in building the product. Then after finding out what they want in a solution they will start building it using their requirements documentation which will help you build your product faster than before because you have already got their feedbacks about how they want your product to look like and work like. Finally once you have reached your desired state then you can start testing again with new users/stakeholders and get feedbacks from them about how would this solution look like for them? So this way we can iterate our solution until we reach our goal

3.2 SPIRAL MODEL

The spiral model has two phases: an early-iteration phase and a later-iteration phase. In early iteration, we create a prototype for our software with paper or other materials. Then, we test it out with users to get feedback on how it works and what they think about it. This is called "iteration 0" because it's our first iteration of the product. Later on, we add more features to our prototype and release it as "Iteration 1". Then, as we continue working on iterating our prototype, we eventually end up with "Iteration N", where N is equal to the number of iterations (1..N) that have been completed by then.

3.3 REASONS TO USE SIMPLE LINEAR REGRESSION

• Our project is based on 4 segments.

• So we decide to work on segments wise and make risks ,planning and model on each segment and then shift to second segment and so on.

• Our project will loop in different parts of spiral.

• Risk Analysis was not available in other models.

• This is the reason for choosing spiral model

3.4 DISADVANTAGES

• Cost will be high compare to other SDLC models

• It will take more time as it is divided into 4 segments

• It is much more complex than other SDLC models.

CHAPTER 4

MODULES AND IMPLEMENTATION

4.1 FLOW CHART

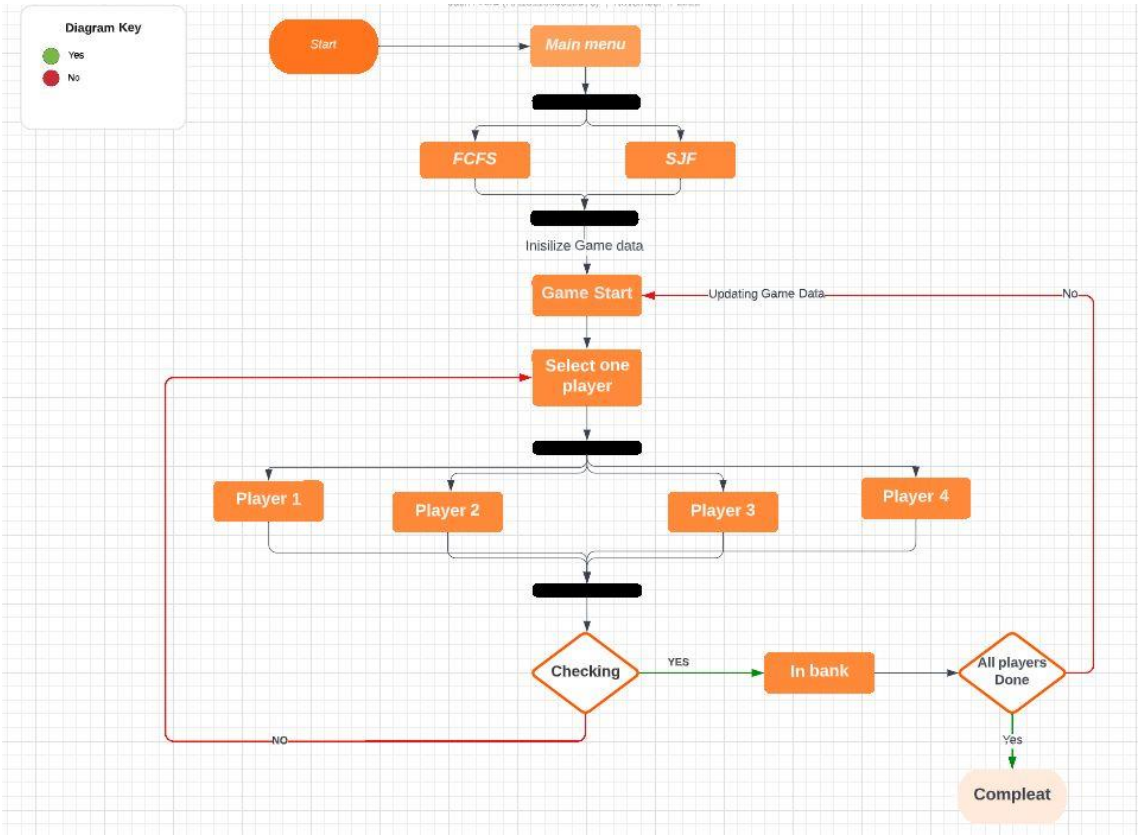

Figure no- 4.1

In the above flow chart, it is represented the working flow of thr game.

1. Main menu: In main menu player can select which algorithm he wants to play and read the instructions for the following algorithms.
2. Fork (FCFS, SJF): after selecting an algorthm game starts.
3. Game Start: Game data is insilize or updated game data is inserted for playing game.
4. Select Player: User will have options to select from 4 players with different arrival time and burst time.
5. Fork (Player_1, PLayer_2, Player_3, Player_4):
    a. Player_1: Arrival Time: 0, Burst Time: 2
    b. Player_2: Arrival Time: 1, Burst Time: 3
    c. Player_3: Arrival Time: 3, Burst Time: 6
    d. Player_4: Arrival Time: 4, Burst Time: 1
6. Condition: If the selected player is correct according to the algorithm selected before the player is allowed to go inside the bank, otherwise player will go to select player stage.
7. Condition: If all the player have went inside bank the game terminate, else game data will be updated and send back to Game start stage and whole game start again.

4.2 MODULES UNDERTAKEN

The modules undertaken as a part of this project are as follows:

1. **Player movement**
2. **FIFO logic**
3. **SJF logic**
4. **Camera controller**
5. **Animators**
6. **Player Selection**
7. **Game data**

4.2.1 PLAYER MOVEMENT-

- There are multiplayer movement in the game such as player walking, hand shake, salute etc.
- For walking we have two types of player

- For player in the city we are moving the player with the arrow keys for the movement.
- For the player inside the bank we are moving the player with the help of w,a,s,d keys for the player movement and for the direction we have use mouse.

- The player can walk through out the city and the bank and roam all the places

42 lines (33 sloc) | 1.03 KB

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
    public CharacterController controller;
    public Animator anim;
    public float speed = 0.1f;
    private Vector3 moveDirection;
    private Vector3 velocity;

    [SerializeField] private bool isGrounded;
    [SerializeField] private float groundCheckDistance;
    [SerializeField] private LayerMask groundMask;
    [SerializeField] private float gravity;

    // Start is called before the first frame update
    void Start()
    {
        anim  = GetComponentInChildren<Animator>();
    }

    // Update is called once per frame
    void Update()
    {
        float x = Input.GetAxis("Horizontal");
        float z = Input.GetAxis("Vertical");

        Vector3 move = transform.right * x + transform.forward * z;
        if(move * speed != Vector3.zero){
            anim.SetBool("isWaking", true);
        }else{
           anim.SetBool("isWaking", false);
        }



        controller.Move(move * speed * Time.deltaTime);

    }
}
```

Figure no- 4.2

- For the hand shake part, we have use a invisible collider which consist of a trigger. Whenever a player collides the trigger wall the script run and the particular action is performed.

30 lines (28 sloc) | 777 Bytes

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class handShakCollider : MonoBehaviour
{
    [SerializeField] public Animator player;
    [SerializeField] public Animator manager;
    public GameObject errorText;
    void Start()
    {
        errorText.SetActive(false);
    }
    private void OnTriggerEnter(Collider Player){
        player.SetBool("isSalute",true);
        manager.SetBool("isSalute",true);
        errorText.SetActive(true);
        StartCoroutine("WaitFor5ec");
    }
    private void OnTriggerExit(Collider Player){
        player.SetBool("isSalute",false);
        manager.SetBool("isSalute",false);
    }
    IEnumerator WaitFor5ec()
    {
       yield return new WaitForSeconds(5);

       errorText.SetActive(false);
    }
}
```

Figure no- 4.3

- For the salute part here also we have used a invisible collider wall which consist of a trigger. When ever a player collide the trigger wall the script run and the particular action is perform.

```
17 lines (13 sloc)  |  357 Bytes
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class triggerSelute : MonoBehaviour
{
    [SerializeField] public Animator anim;


    private void OnTriggerEnter(Collider Player){
        anim.SetBool("isSalute",true);
    }

    private void OnTriggerExit(Collider Player){
        anim.SetBool("isSalute",false);
    }
}
```

Figure no- 4.4

4.2.2 FIFO LOGIC

- FCFS is the simplest scheduling algorithm. There is a single rule; schedule the first process to arrive, and let it to run to completion after completion of the first process will executed it will be done in a order.
- In the game we have took four person in a particular order the fcfs will be implemented in the order, while the entry of the first person all other person entry is not allowed. After the exit of first person only the second person is allow and this continue for all the person.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;

public class fifo : MonoBehaviour
{
    public GameObject player1;
    public GameObject player2;
    public GameObject player3;
    public GameObject player4;

    public TMP_Text uiText;
    public GameObject errorText;


    //[SerializeField] public fcfsInfo data;
    //[SerializeField] public ganndata chart;
    [SerializeField] public gameData data;

    void Start()
    {
        uiText.text = "FIFO Current Time: " + data.currentTime;
        errorText.SetActive(false);

    }

    // Update is called once per frame
    private void OnTriggerEnter(Collider Player)
    {
        if(data.fifo){
        if (Player.gameObject.tag == "Player")
        {
            if (Player.gameObject.name == "player1" && data.check1)
            {
                data.check1 = false;
                data.check2 = true;

                data.currentTime = 2;
                data.waitTime = 0;

                //setRun.run = false;
                uiText.text = data.currentTime.ToString();
```

Figure no- 4.5

```
47
48                  SceneManager.LoadScene(2);
49              }
50              else if (Player.gameObject.name == "player2" && data.check2)
51              {
52
53
54                  data.currentTime = 5;
55                  data.waitTime = 1;
56                  data.check2 = false;
57                  data.check3 = true;
58
59                  //setRun.run = false;
60                   uiText.text = data.currentTime.ToString();
61                  SceneManager.LoadScene(2);
62              }
63              else if (Player.gameObject.name == "player3" && data.check3)
64              {
65
66                  data.currentTime = 8;
67                  data.waitTime = 2;
68                  data.check3 = false;
69                  data.check4 = true;
70                  //setRun.run = false;
71                   uiText.text = data.currentTime.ToString();
72                  SceneManager.LoadScene(2);
73              }
74              else if (Player.gameObject.name == "player4" && data.check4)
75              {
76
77                   data.currentTime = 9;
78                  data.waitTime = 6;
79                   data.check4 = false;
80                  //data.check1 = true;
81                  //setRun.run = false;
82                   uiText.text = data.currentTime.ToString();
83                  SceneManager.LoadScene(2);
84                   //check2 = true;
85              }
```

Figure no- 4.6

4.2.3 SJF LOGIC

- In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.
- In this also we have applied the same logic as applied in FCFS with an additional feature to check which person has the least burst time and allowing that person to perform the task.

```
97          }else if(data.sjf){
98                  if (Player.gameObject.tag == "Player")
99          {
100             if (Player.gameObject.name == "player1" && data.check1)
101             {
102                 data.check1 = false;
103                 data.check2 = true;
104 
105                 data.currentTime = 2;
106                 data.waitTime = 0;
107 
108                 //setRun.run = false;
109                 uiText.text = data.currentTime.ToString();
110 
111 
112                 SceneManager.LoadScene(2);
113             }
114             else if (Player.gameObject.name == "player2" && data.check2)
115             {
116 
117 
118                 data.currentTime = 5;
119                 data.waitTime = 1;
120                 data.check2 = false;
121                 data.check3 = true;
122 
123                 //setRun.run = false;
124                  uiText.text = data.currentTime.ToString();
125                 SceneManager.LoadScene(2);
126             }
127             else if (data.check3 && Player.gameObject.name == "player4")
128             {
129 
130                 data.currentTime = 6;
131                 data.waitTime = 1;
132                 data.check3 = false;
133                 data.check4 = true;
134                 //setRun.run = false;
135                  uiText.text = data.currentTime.ToString();
136                 SceneManager.LoadScene(2);
137             }
138             else if (data.check4 && Player.gameObject.name == "player3" && Player.gameObject.name != "player4")
```

Figure no- 4.7

```
            {

                data.currentTime = 9;
                data.waitTime = 3;
                data.check4 = false;

                //data.check1 = true;
                //setRun.run = false;
                uiText.text = data.currentTime.ToString();
                SceneManager.LoadScene(2);
                //check2 = true;
            }
            else
            {
                errorText.SetActive(true);
                StartCoroutine("WaitFor5ec");
                //if(Player.gameObject.name == "player1"){
                //      player1.GetComponent<moveBack>().enabled = true;
                //      player1.GetComponent<PlayerMovement>().enabled = false;
                //}
            }

        }
        }
    }

    IEnumerator WaitFor5ec()
    {
       yield return new WaitForSeconds(5);

       errorText.SetActive(false);
    }
}
```

Figure no- 4.8

4.2.4 Camera Controller

- We have multiple cameras in our game
    - Main camera- It’s use to see the whole game.
    - Dolly camera- This camera is use to show the cinematic view of the whole city in which we have developed the game.
    - Player camera- This camera is to show us the first person view through which we can get the whole city view.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CamSwitch : MonoBehaviour
{
    public GameObject player1;
    public GameObject player2;
    public GameObject player3;
    public GameObject player4;
    public GameObject player0;
    public GameObject dollyCamera;
    public GameObject movingPlayer;
    public GameObject movingCamera;
    public GameObject uiText;


    // Start is called before the first frame update
    void Start()
    {

        player0.SetActive(false);
        dollyCamera.SetActive(true);
        movingCamera.SetActive(false);
        movingPlayer.GetComponent<basicMovement>().enabled = false;
        //player1.GetComponent<PlayerMovement>().enabled = false;
        //player2.GetComponent<PlayerMovement>().enabled = false;
        //player3.GetComponent<PlayerMovement>().enabled = false;
        //player4.GetComponent<PlayerMovement>().enabled = false;
        player1.GetComponent<outsidePlayerMovement>().enabled = false;
        player2.GetComponent<outsidePlayerMovement>().enabled = false;
        player3.GetComponent<outsidePlayerMovement>().enabled = false;
        player4.GetComponent<outsidePlayerMovement>().enabled = false;
        //uiText.SetActive(true);
    }

```

Figure no- 4.9

```
119         if (Input.GetButtonDown("5Key"))
120         {
121             movingCamera.SetActive(true);
122             movingPlayer.GetComponent<basicMovement>().enabled = true;
123             player0.SetActive(false);
124             player1.GetComponent<PlayerMovement>().enabled = false;
125             player2.GetComponent<PlayerMovement>().enabled = false;
126             player3.GetComponent<PlayerMovement>().enabled = false;
127             player4.GetComponent<PlayerMovement>().enabled = false;
128 
129             player1.GetComponent<outsidePlayerMovement>().enabled = false;
130             player2.GetComponent<outsidePlayerMovement>().enabled = false;
131             player3.GetComponent<outsidePlayerMovement>().enabled = false;
132             player4.GetComponent<outsidePlayerMovement>().enabled = false;
133             //uiText.SetActive(false);
134         }
135 
136         if(Input.GetButtonDown("Cancel")){
137            Application.Quit();
138            Debug.Log("Quit");
139         }
140         if(Input.GetButtonDown("continue")){
141           player0.SetActive(true);
142           dollyCamera.SetActive(false);
143         }
144     }
145 }
```

Figure no- 4.10

4.2.5 Animators

- Every game incomplete without any animations
- For giving the animations we have developed our own animator design as shown below

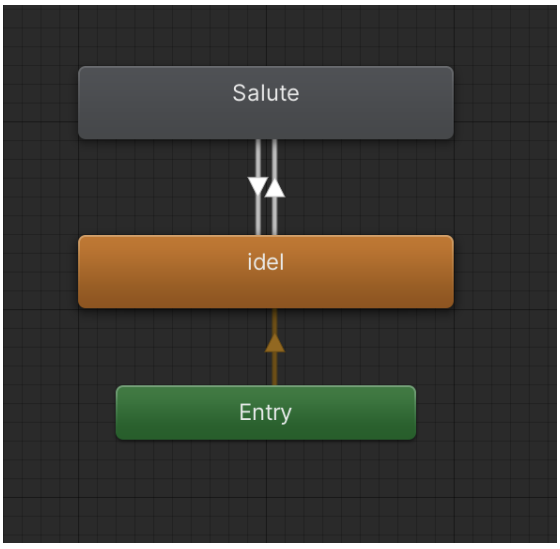

Figure no- 4.11

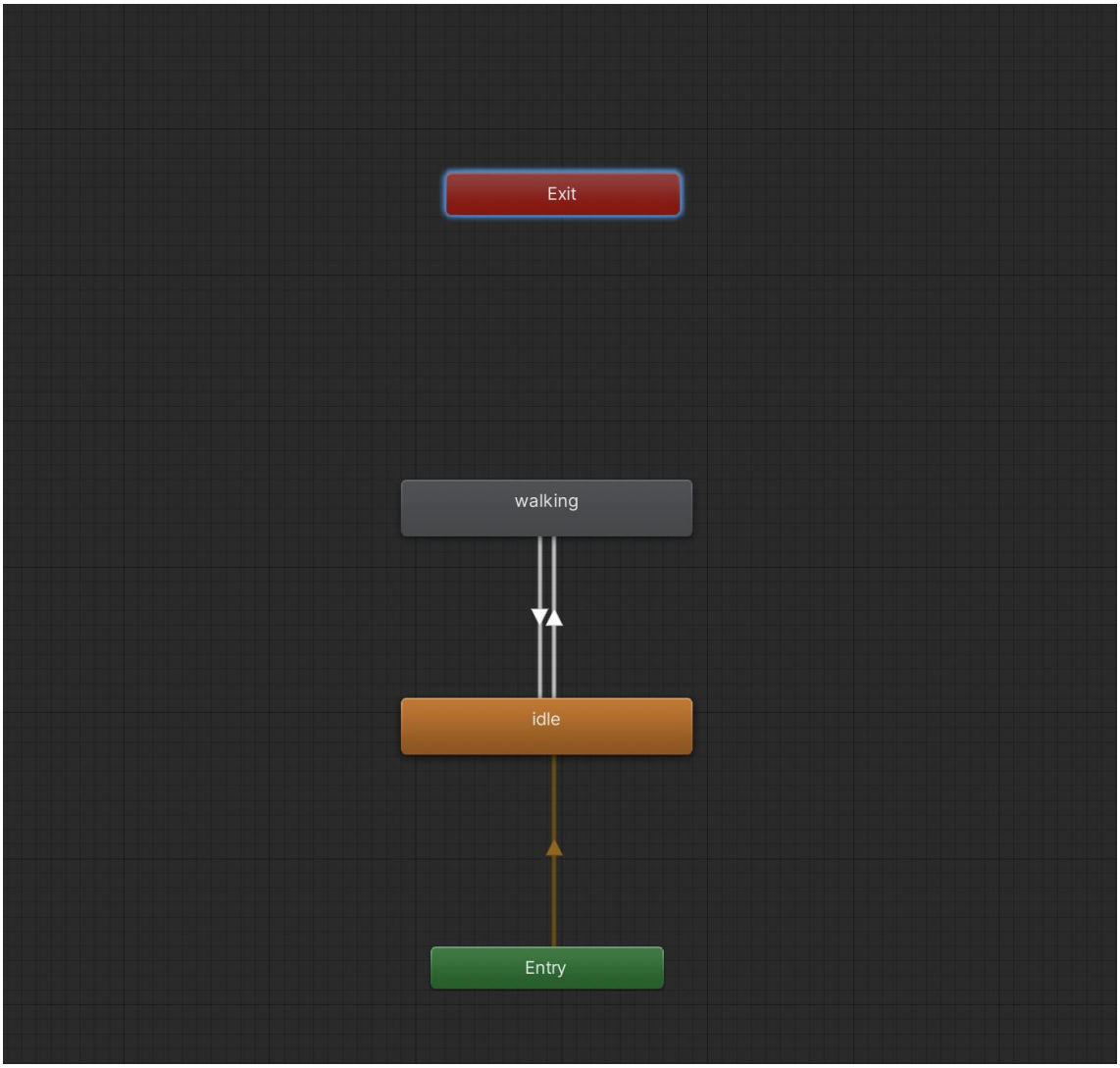

Figure no- 4.12

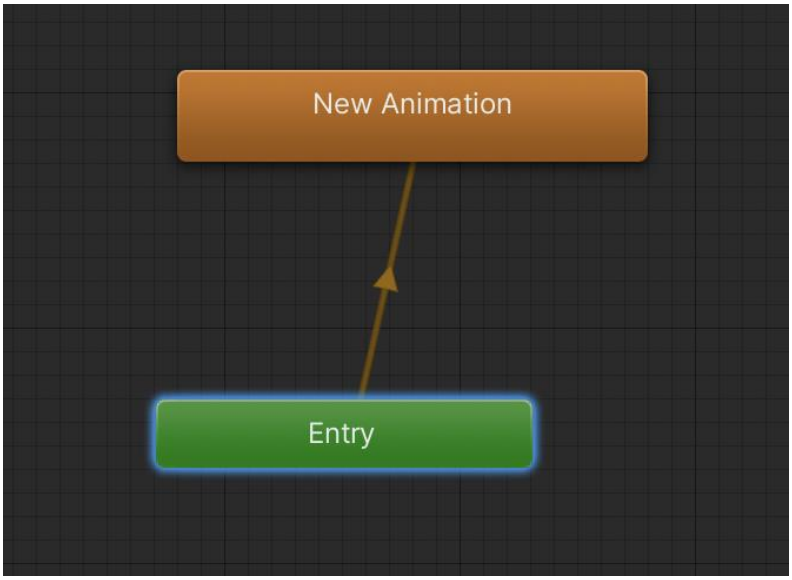

Figure no- 4.13

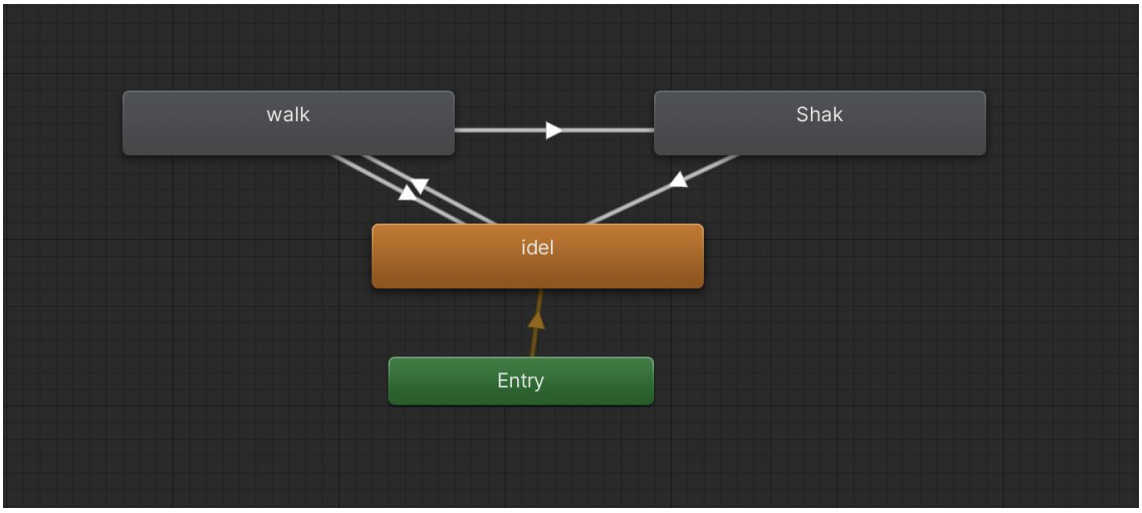

Figure no- 4.14

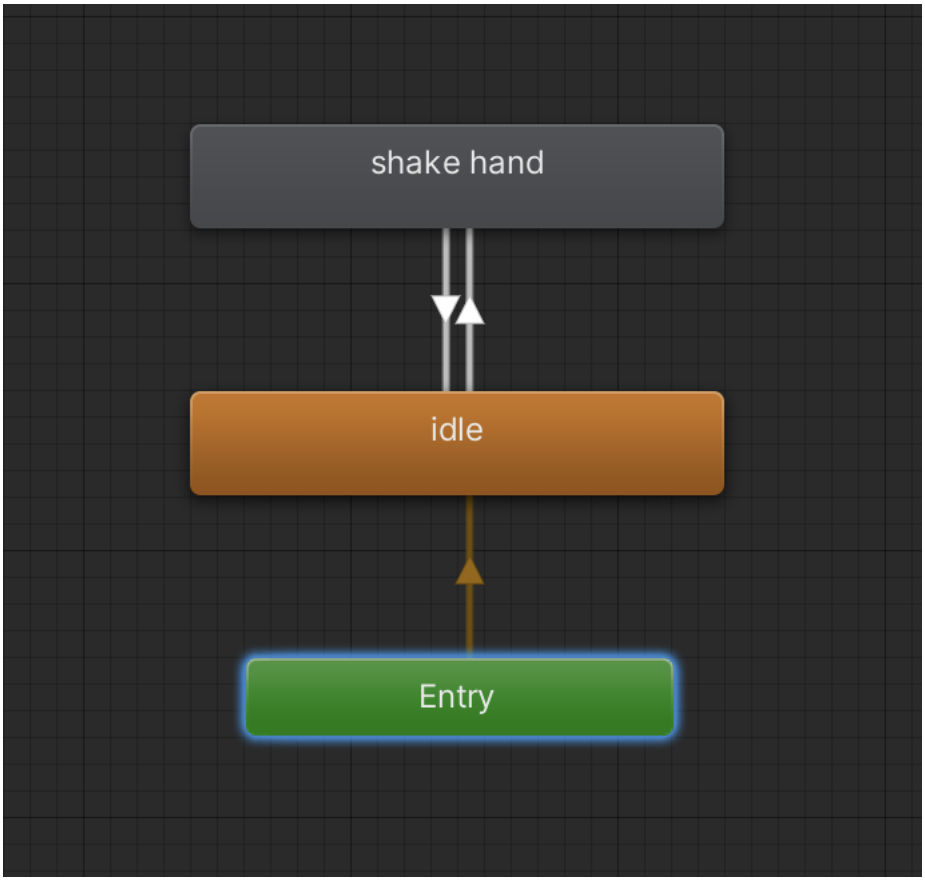

Figure no- 4.14

4.2.6 PLAYER SELECTION

- There are total 4 players in the game so for the FCFS logic we have select only one person so we have use the player selection script to select the player.
- We have given four keys for four players each key select the particular player and only that player movement.

Figure no- 4.14

```
37      // Update is called once per frame
38      void Update()
39      {
40
41          if (Input.GetButtonDown("0Key"))
42          {
43              player0.SetActive(true);
44              movingCamera.SetActive(false);
45              movingPlayer.GetComponent<basicMovement>().enabled =false;
46              //player1.GetComponent<PlayerMovement>().enabled = false;
47              //player2.GetComponent<PlayerMovement>().enabled = false;
48              //player3.GetComponent<PlayerMovement>().enabled = false;
49              //player4.GetComponent<PlayerMovement>().enabled = false;
50
51              player1.GetComponent<outsidePlayerMovement>().enabled = false;
52              player2.GetComponent<outsidePlayerMovement>().enabled = false;
53              player3.GetComponent<outsidePlayerMovement>().enabled = false;
54              player4.GetComponent<outsidePlayerMovement>().enabled = false;
55              //uiText.SetActive(true);
56
57          }
58          if (Input.GetButtonDown("1Key"))
59          {
60              movingCamera.SetActive(false);
61              movingPlayer.GetComponent<basicMovement>().enabled =false;
62              player1.GetComponent<PlayerMovement>().enabled = true;
63              player2.GetComponent<PlayerMovement>().enabled = false;
64              player3.GetComponent<PlayerMovement>().enabled = false;
65              player4.GetComponent<PlayerMovement>().enabled = false;
66
67              player1.GetComponent<outsidePlayerMovement>().enabled = true;
68              player2.GetComponent<outsidePlayerMovement>().enabled = false;
69              player3.GetComponent<outsidePlayerMovement>().enabled = false;
70              player4.GetComponent<outsidePlayerMovement>().enabled = false;
71              //uiText.SetActive(false);
72
73          }
74          if (Input.GetButtonDown("2Key"))
75          {
76              movingCamera.SetActive(false);
77              movingPlayer.GetComponent<basicMovement>().enabled = false;
78              player1.GetComponent<PlayerMovement>().enabled = false;
79              player2.GetComponent<PlayerMovement>().enabled = true;
80              player3.GetComponent<PlayerMovement>().enabled = false;
81              player4.GetComponent<PlayerMovement>().enabled = false;
```

Figure no- 4.15

```
        if (Input.GetButtonDown("4Key"))
        {
            movingCamera.SetActive(false);
            movingPlayer.GetComponent<basicMovement>().enabled = false;
            player1.GetComponent<PlayerMovement>().enabled = false;
            player2.GetComponent<PlayerMovement>().enabled = false;
            player3.GetComponent<PlayerMovement>().enabled = false;
            player4.GetComponent<PlayerMovement>().enabled = true;

            player1.GetComponent<outsidePlayerMovement>().enabled = false;
            player2.GetComponent<outsidePlayerMovement>().enabled = false;
            player3.GetComponent<outsidePlayerMovement>().enabled = false;
            player4.GetComponent<outsidePlayerMovement>().enabled = true;
            //uiText.SetActive(false);
        }
        if (Input.GetButtonDown("5Key"))
        {
            movingCamera.SetActive(true);
            movingPlayer.GetComponent<basicMovement>().enabled = true;
            player0.SetActive(false);
            player1.GetComponent<PlayerMovement>().enabled = false;
            player2.GetComponent<PlayerMovement>().enabled = false;
            player3.GetComponent<PlayerMovement>().enabled = false;
            player4.GetComponent<PlayerMovement>().enabled = false;

            player1.GetComponent<outsidePlayerMovement>().enabled = false;
            player2.GetComponent<outsidePlayerMovement>().enabled = false;
            player3.GetComponent<outsidePlayerMovement>().enabled = false;
            player4.GetComponent<outsidePlayerMovement>().enabled = false;
            //uiText.SetActive(false);
        }

        if(Input.GetButtonDown("Cancel")){
           Application.Quit();
           Debug.Log("Quit");
        }
        if(Input.GetButtonDown("continue")){
          player0.SetActive(true);
          dollyCamera.SetActive(false);
        }
    }
}
```

Figure no- 4.16

4.2.7 GAME DATA

- This tries to keep the game consistent across all the scene.

- We have use a scriptable object for saving and updating the game data.
- With the help of this while the player changes the scene there is no error.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu(fileName = "gameData", menuName = "data")]
public class gameData : ScriptableObject
{
    public bool check1 = true;
    public bool check2 = false;
    public bool check3 = false;
    public bool check4 = false;

    public int currentTime=0;
    public int waitTime=0;

    public bool fifo = true;
    public bool sjf = false;
}
```

Figure no- 4.17

CHAPTER 5

RESULTS AND DISCUSSION

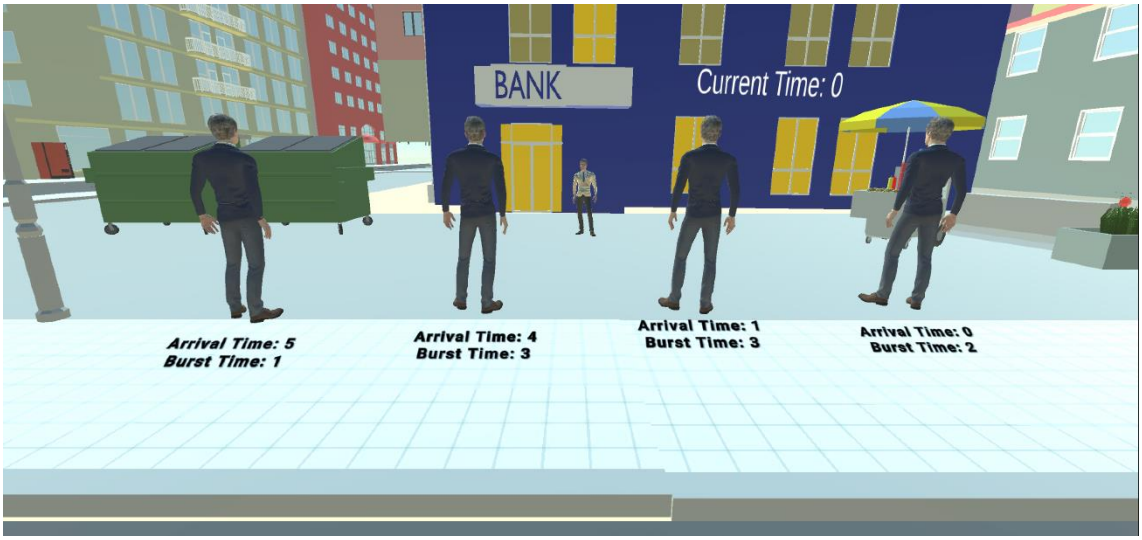

Figure no- 5.1

- A city has been implemented inside the Unity 3D software
- There is a bank, hospital, police station, resident house, buildings, petrol pump, roads, footpath and other important building.

Figure no- 5.2

- There are 4 persons standing outside the bank with their arrival time and burst time.

- A guard is there to check no person enter the bank without their turn and sent that player back.

Figure no- 5.3

- Interior designing of the bank has been done. There is a manager cabin, clerk, ATM machine, Washroom, and other important areas.
- Player can move in the bank freely using WASD keys and mouse for looking around.

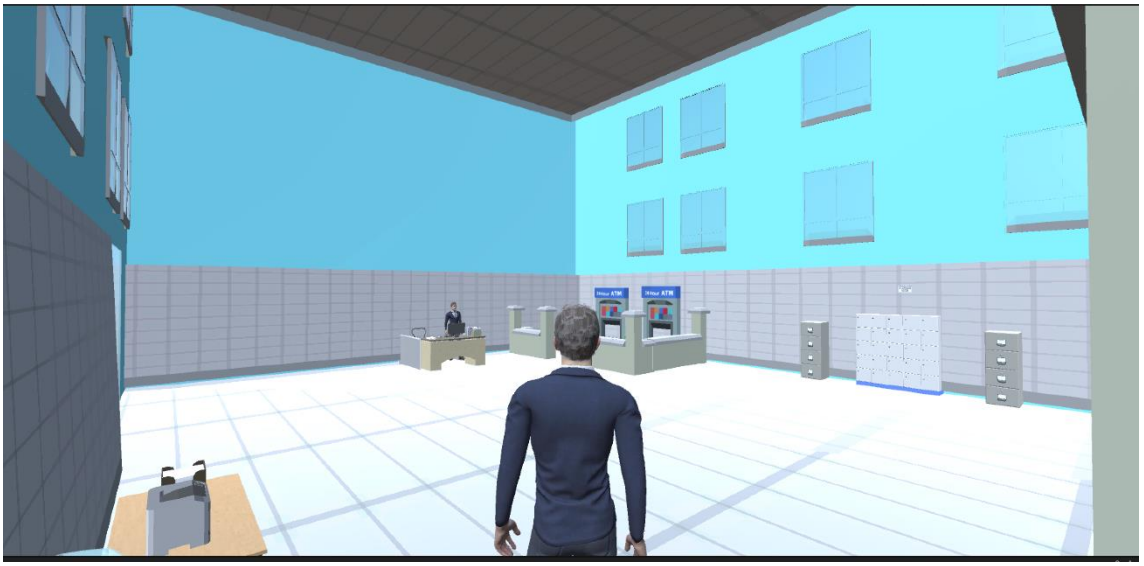

Figure no- 5.4

Figure no- 5.5

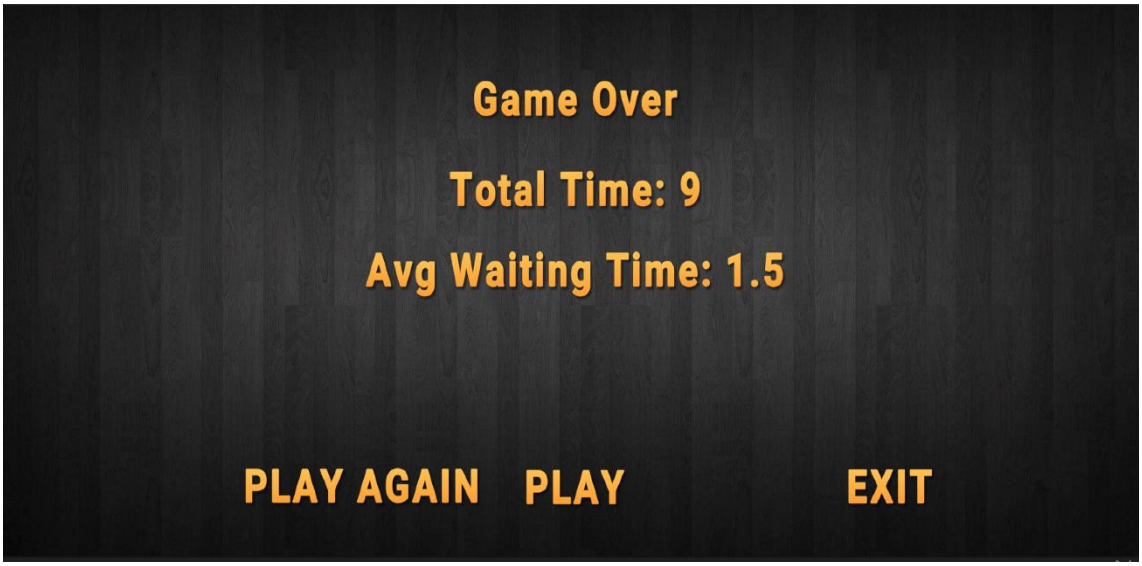
Game Over
Total Time: 9
Avg Waiting Time: 1.5
PLAY AGAIN
PLAY
EXIT

Figure no- 5.6

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

Game banded learning is very effective. It helps students learn the course material in a way that they are both interested in and engaged with. Peer interaction games are also fun, which makes it engaging for the students.

The game incentivizes cooperation, which is good for learning because it gives students an incentive to work together when they wouldn't naturally do so. And it also gives students an opportunity to work through their own problems and figure out solutions for them before being forced to do so by a peer group or another group of players.

The game also provides a way for students to practice social skills, which is important because most people don't think about how important those skills are until they need them but can no longer use them due to lack of practice or ability. Peer interaction games provide an opportunity for this practice, which can help prepare people for real-life situations where they may need those social skills. I think this pedagogy has improved student motivation and performance on all three measures: learning outcomes (which we evaluate), satisfaction with pedagogy (which we evaluate), and student performance (which we monitor). In conclusion, it is clear from our participant feedback that game-based cooperative learning can stimulate active learning. The experimental group were more willing to spend time on learning activities, and this shows that game-based cooperative learning can stimulate active learning. Many control group members relied on others to answer questions, which is only natural for these students.

We have successfully implemented the two sheduling algorithms FCFS and SJF.

6.2 Future Enhancements

- We have to implement Longest Job First, Shortest Remaining Job First, Round Robin.
- We will implement Virtual Reality and Argument Reality for better understanding.
- These games are very simple and students may lose intrust in them when the play them a lot. So, for that we have to implement better story version for our games.
- Gaming industry has a lot of potential and if we are able to use those in teaching industry then it will resolubilize the teaching.
- More Operating system concept can we implement in future.

- In the future, we hope to apply this method to different classes and with different types of participants in order to further develop the game and its context. We will also continue to improve the visual and sound effects which can make the game more engaging for students.
- We would like to explore ways to integrate more dynamic pictures and skills into the game so that it motivates students even more.

REFERENCES

[1] B. D. Coller and M. J. Scott, "Effectiveness of using a video game to teach a course in mechanical engineering," Comput. Educ., vol. 53, no. 3, pp. 900–912, 2009.

[2] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation," Comput. Educ., vol. 52, no. 1, pp. 1–12, 2009.

[3] J. Robertson and C. Howells, "Computer game design: Opportunities for successful learning," Comput. Educ., vol. 50, no. 2, pp. 559–578, 2008.

[4] N. Vos, H. Van Der Meijden, and E. Denessen, "Effects of constructing versus playing an educational game on student motivation and deep learning strategy use," Comput. Educ., vol. 56, no. 1, pp. 127–137, 2011.

[5] H. C. Jiau, J. C. Chen, and K. F. Ssu, “Enhancing self-motivation in learning programming using game-based simulation and metrics,” IEEE Trans. Educ., vol. 52, no. 4, pp. 555–562, Nov. 2009.

[6] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz, “A multi-national, multi-institutional study of assessment of program-ming skills of first-year CS students,” ACM SIGCSE Bull., vol. 33, no. 4, pp. 125–180, 2001.

[7] T. Hainey, T. M. Connolly, M. Stansfield, and E. A. Boyle, “Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level,” Comput. Educ., vol. 56, no. 1, pp. 21–35, 2011.

[8] E. Y. Leng, W. Z. W. Ali, R. Mahmud, and R. Baki, “Computer games development experience and appreciative learning approach for creative process enhancement,” Comput. Educ., vol. 55, no. 3, pp. 1131–1144, 2010.

[9] L. M. Miller, C. I. Chang, S. Wang, M. E. Beier, and Y. Klisch, “Learning and motivational impacts of a multimedia science game,” Comput.Educ., vol. 57, no. 1, pp. 1425–1433, 2011.

[10] I. L. Beale, P. M. Kato, V. M. Marin-Bowling, N. Guthrie, and S. W.

Cole, “Improvement in cancer-related knowledge following use of a

psychoeducational video game for adolescents and young adults with cancer,” J. Adolescent Health, vol. 41, no. 3, pp. 263–270, 2007.

[11] A. Y. K. Chua, “The design and implementation of a simulation game

for teaching knowledge management,” J. Amer. Soc. Inf. Sci. Technol.,

vol. 56, no. 11, pp. 1207–1216, 2005.

[12] B. C. Nelson, “Exploring the use of individualized reflective guid-

ance in and educational multiuser virtual environment,” J. Sci. Educ.Technol., vol. 16, no. 1, pp. 83–97, 2007.

CODE

Outside player movement-

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class outsidePlayerMovement : MonoBehaviour

{

    public Animator player;

    //public CharacterController controller;

    public float speed = 0.1f;

    public float rotationSpeed = 100f;

    float xRotation = 0f;
```

```
Vector3 Vec;

// Start is called before the first frame update

void Start()

{

   //anim  = GetComponentInChildren<Animator>();

}

// Update is called once per frame

void Update()

{

   xRotation = Mathf.Clamp(xRotation, -90f, 90f)

   if (Input.GetKey(KeyCode.UpArrow))

   {

      this.transform.Translate(Vector3.forward * Time.deltaTime);

   }

   if (Input.GetKey(KeyCode.DownArrow))

   {

      this.transform.Translate(Vector3.back * Time.deltaTime);

   }

   if (Input.GetKey(KeyCode.LeftArrow))

   {
```

```
            this.transform.Rotate(Vector3.up, -5);
        }
        if (Input.GetKey(KeyCode.RightArrow))
        {
            this.transform.Rotate(Vector3.up, 5);
        }
    }
}
```