**ARTIFICIAL INTELLIGENCE BASED**

**MULTIPLAYER SNAKEGAME**

**A MINIPROJECT REPORT**

***Submitted by***

**DEEPAK RAJAN S (160501039)**

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**SRI VENKATESWARA COLLEGE OF ENGINEERING**

**SRIPERUMBUDUR Tk – 602 117**

**ANNA UNIVERSITY: CHENNAI 600 025**

**OCTOBER 2019**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**ARTIFICAIL INTELIGENCE BASED MULTIPLAYER SNAKEGAME**” is the bonafide work of “**DEEPAK RAJANS (160501039)**”, who carried out the project work under my supervision.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

Submitted for the project Viva-Voce Examination held on …………………

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**ABSTRACT**

The purpose of this project is to make the player compete with each other on a multiplayer **snake game** and make the computer reflect the moves that the players have used in their previous games .The model is demonstrated in browser so this game is developed using the basic **SSE** server push mechanism and record the player move in a one on one battle with each and tries to use the previous player’s move. The computer uses the shortest path algorithm to reach the target without having self destructive moves and tries to escape from the opponent’s attacking moves by predicting it before.

**ACKNOWLEDGEMENT**

I thank my Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout my study in this college.

I express my sincere thanks to **Dr.R. Anitha,** Head of the Department, **Computer Science and Engineering** for her encouragement accorded to carry this project.

I am also thankful to my project coordinators **Dr.R.Jayabhaduri, Associate Professor and Dr.N.M.Balamurugan, Associate Professor** for their continual support and assistance.

I thank my family and friends for their support and encouragement throughout the course of my graduate studies.

**DEEPAK RAJAN S**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **iii** |
|  | **LIST OF FIGURES** | **x** |
|  | **LIST OF ABBREVIATION** | **xii** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 MULTIPLAYER SNAKE GAME | 2 |
|  | 1.3    CHARACTERISTICS OF THE  MULTIPLAYER SNAKEGAME | 3 |
|  | 1.4    SERVER PUSH FOR MECHANISM  FOR MULTIPLAYER GAME | 4 |
|  | 1.5    PREDICTION FOR BOT | 5 |
|  | 1.5.1 Database for Prediction Model | 5 |
| **2** | **LITERATURE REVIEW** | **6** |
|  | 2.1 SLITHER.IO | 6 |
|  | 2.1.1 Development | 6 |
|  | 2.1.2 Advantages and Disadvantages | 8 |
|  | 2.1.3 Screenshots | 8 |
|  | 2.2 AGAR.IO | 9 |
|  | 2.2.1 Development | 9 |
|  | 2.2.2 Advantages and Disadvantages | 9 |
|  | 2.2.3 Screenshots | 11 |
| **3** | **PROPOSED WORK** | **12** |
|  | 3.1 RECOMMENDED METHOD | 12 |
|  | 3.1.1 Advantages | 12 |
|  | 3.2 PROPOSED ARCHITECTURE  DIAGRAM | 13 |
|  | 3.3 DESCRIPTION OF ARCHITECTURE  DIAGRAM | 14 |
|  | 3.3.1 Participants | 14 |
|  | 3.3.2 Live User Listing | 18 |
|  | 3.3.3 Request Handler | 18 |
|  | 3.3.4 Game board Manager | 18 |
|  | 3.3.5 Data Collector | 19 |
|  | 3.3.6 Predictor | 20 |
|  | 3.3.7 User Move Database | 21 |
| **4** | **SYSTEM REQUIREMENTS**  4.1 HARDWARE REQUIREMENTS | **23**  23 |
|  | 4.2 SOFTWARE REQUIREMENTS | 23 |
| **5** | **IMPLEMENTATION MODULES** | **24** |
|  | **5.1** MODULES TO BE IMPLEMENTED | **24** |
|  | 5.1.1 LOGIN SERVICE | 24 |
|  | 5.1.2 LIVE USER LISTER SERVICE | 25 |
|  | 5.1.3 GAME CONTROLLER  MODULE | 26 |
|  | 5.1.4 USER SNAKE CONTROLLE | 27 |
|  | 5.1.5 GAME SYNCHRONIZATION | 31 |
|  | 5.1.6 PREDICTION MODULE | 35 |
|  | **SNAPSHOTS OF MODULES** | **36** |
|  | 6.1 LOGIN PAGE | 36 |
| **6** | 6.2 GETTING ID | 37 |
|  | 6.3 GETTING LIVE USERS | 37 |
|  | 6.4 SENDING REQUEST | 38 |
|  | 6.5 RECIEVING REQUEST | 38 |
|  | 6.6 CALIBRATING GAME | 39 |
|  | 6.7 STARTING GAME | 39 |
|  | 6.8 ENDING GAME | 40 |
| **7** | **CONCLUSION AND FUTURE WORK** | **41** |
|  | **REFERENCES** | **41** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 2.1 | Slither.io screenshot 1 | 8 |
| 2.2 | Slither.io screenshot 2 | 8 |
| 2.3 | Agar.io screenshot 1 | 11 |
| 2.4 | Agar.io screenshot 1 | 11 |
| 3.1 | Architecture diagram | 13 |
| 3.2 | Comparison of server push mechanisms | 15 |
| 3.3 | AJAX Architecture | 16 |
| 3.4 | snake JSON Code | 20 |
| 3.5 | Database structure | 22 |
| 5.1 | Code snippet live user listing | 25 |
| 5.2 | Code snippet live user listing client | 26 |
| 5.3 | Flow of getting ID | 27 |
| 5.4 | Snake control client side | 28 |
| 5.5 | Add body part code snippet | 29 |
| 5.6 | Remove body parts code snippet | 30 |
| 5.7 | Flo Flow for data collection | 31 |
| 5.8 | Model 1 for synchronization | 33 |
| 5.9 | Model 2 for synchronization | 34 |
| 5.10 | Timestamp calibration | 35 |
| 6.1 | Login Page | 37 |
| 6.2 | Getting ID | 38 |
| 6.3 | Getting Live Users | 38 |
| 6.4 | Sending request | 39 |
| 6.5 | Receiving request | 39 |
| 6.6 | Timestamp calibration screen | 40 |
| 6.7 | Starting the game | 40 |
| 6.8 | Game End | 41 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ABBREVIATION**

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| SSE | Server Sent Events |
| AJAX | Asynchronous JavaScript and XML |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |