GAMEZONE

Online gaming is hugely popular with children and young people. From sport related games to mission based games and quests inspiring users to complete challenges, interactive games cater for a wide range of interests, and can enable users to link up and play together.

Most games now have an online element to them, allowing users to take part in leader boards, join group games or chat to others. Internet connectivity in a game adds a new opportunity for gamers as it allows players to find and play against, or with, other players. These may be their friends or family members or even other users in the game from around the world. The main aim of this project is to develop a game for entertainment purpose. The project will have 4 different games like "hangman","quizgame","tic tac toe" and "snake game".

GAME ZONE shall provide user with option to select the game. User will be able to switch between different games. Every time user quit any game it will be redirected to the main.

**Tic Tac Toe Game**

This game is very popular amongst all of us and even fun to build as a Python project. It is a two-player game and consists of a nine-square grid. Each player chooses their move and with O or X and marks their square one at each chance. The player who succeeds in making their marks all in one line whether diagonally, horizontally, or vertically wins. The challenge for the other player is to block the game for their opponent and also to make their chain.

**Hangman Game**

Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player loses. The game also ends if the player correctly identifies all the letters of the missing word.

**Snake Game**

In the game of Snake, the player uses the arrow keys to move a "snake" around the board. As the snake finds food, it eats the food, and thereby grows larger. The game ends when the snake either moves off the screen or moves into itself. The goal is to make the snake as large as possible before that happens.

**Quiz Game**

In the quiz game the player wil be known about rules of the game and the start the game. There will be total of 15 questions and each question has some amount, if answered correctly user can get them. User will have a chance of taking helpline once The games ends when user answered all questions or if the lives are over user will receive half of the amount scored.

**Folder Structure**

|  |  |
| --- | --- |
| **Folder** | **Description** |
| 1\_Requirements | Documents detailing requirements and research |
| 2\_Architecture | Documents specifying design details |
| 3\_Implementation | All code and documentation |
| 4\_TestplanAndOutput | Documents with test plan and procedures. |
| 5\_Report | Complete report of the project. |
| 6\_ImagesAndVideos | Output images and videos. |

**Requirements**

## User Requirements

Since the project consists of multiple games, the intended users are gamers and programmers.

## Functional Requirement

1. GAME ZONE shall provide user with option to select the multiple game.
2. User will be able to switch between different games.
3. Every time user quit the any game it will be redirected to the main.

Non-Functional Requirements

1. We require Windows along with some libraries for our project.
2. Game Zone shall be portable.

## High Level Requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Status** |
| HR\_01 | Operating System (Windows 10) | Implemented |
| HR\_02 | python language | Implemented |
| HR\_03 | User shall be able to play tic-tac-toe game | Implemented |
| HR\_04 | User shall be able to play quiz game | Implemented |
| HR\_05 | User shall be able to play snake game | Implemented |
| HR\_06 | User shall be able to play Hangman game | Implemented |

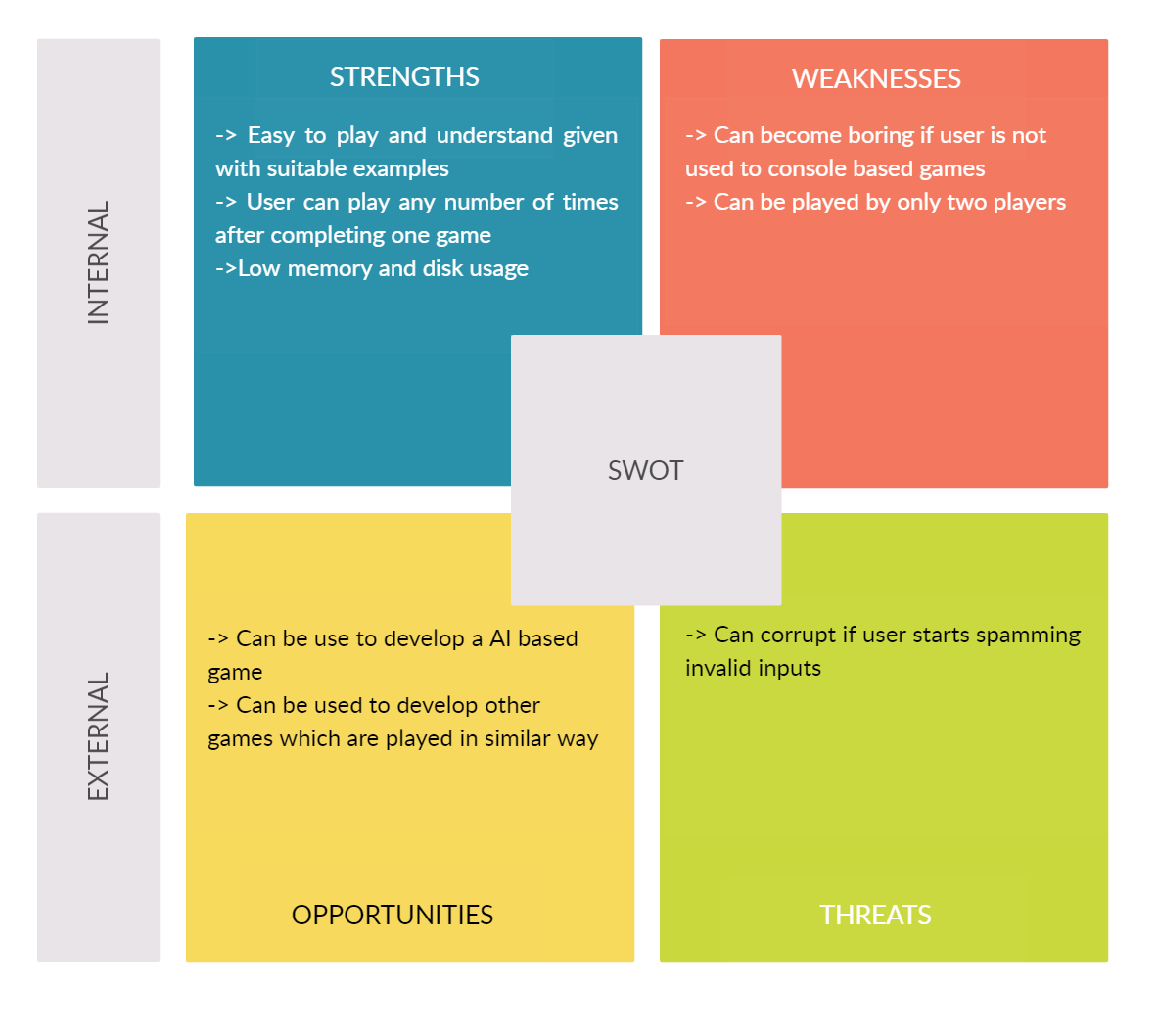
## Low Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **High level ID** | **Status** |
| LR\_01 | A 3x3 grid is displayed for tic-tac-toe | HR\_03 | Implemented |
| LR\_02 | A player and a computer, one called as X and the other as O | HR\_03 | Implemented |
| LR\_03 | Alternate chances are given to the player and computer | HR\_03 | Implemented |
| LR\_04 | After every move an updated version of the grid is displayed | HR\_03 | Implemented |
| LR\_05 | When the game ends the final and updated grid is displayed with the winner's name | HR\_03 | Implemented |
| LR\_06 | Rules will be displayed for quiz game | HR\_04 | Implemented |
| LR\_07 | Lives and helpline will be displayed and updated along with each question if used | HR\_04 | Implemented |
| LR\_08 | After each question the status of that question is displayed whether correct or incorrect along with the score | HR\_04 | Implemented |
| LR\_09 | After the lives are over half of the amount is given to user from what they have scored | HR\_04 | Implemented |
| LR\_10 | when all questions are answered, game ends and final amount will be given to user | HR\_04 | Implemented |
| LR\_11 | Displaying of Snake game output window | HR\_05 | Implemented |
| LR\_12 | Creation of snake and movement of the snake | HR\_05 | Implemented |
| LR\_13 | Game over when snake hit the boundaries | HR\_05 | Implemented |
| LR\_14 | Adding the food for the snake | HR\_05 | Implemented |
| LR\_15 | Increasing the length of the snake when snake comsume the food | HR\_05 | Implemented |
| LR\_16 | Displaying the score in the Snake game output window | HR\_05 | Implemented |
| LR\_17 | Adding quit and continue option to the user | HR\_05 | Implemented |
| LR\_18 | Hangman: A secret word is generated | HR\_06 | Implemented |
| LR\_19 | User is a given a chance to choose any alphabet | HR\_06 | Implemented |
| LR\_20 | If the all the letters are guessed correctly user wins | HR\_06 | Implemented |
| LR\_21 | If the the letters are not guessed correctly after certain attempts user loses ang=d hangman is seen | HR\_06 | Implemented |
| LR\_22 | Adding yes or no to play the game again | HR\_06 | Implemented |

**Cost and Features**

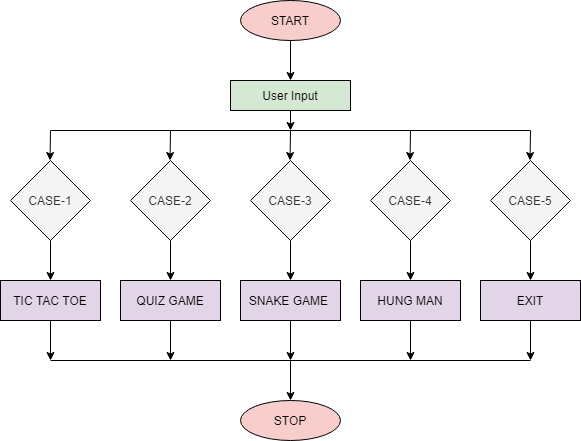
|  |  |  |
| --- | --- | --- |
| **Time** | **Feature** | **Cost** |
| Past | Offline mode of playing games | Earlier cost is less just once we need to buy the game in physical form once |
| Present | Most games can be played online with friends/family | Now we need to pay for internet and also some games are paid and may require in-game purchases |
| Future | Online games with advanced features such that we feel we are playing offline | Cost will be higher for playing games with advanced graphics |

**SWOT Analysis**

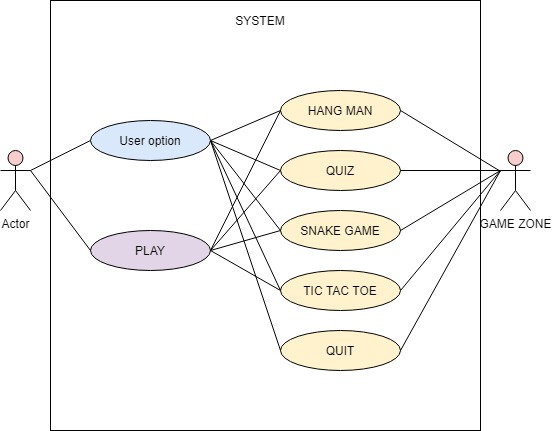


**Design**

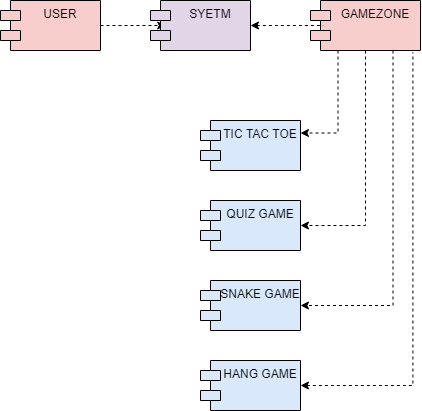
Game Zone Flowchart



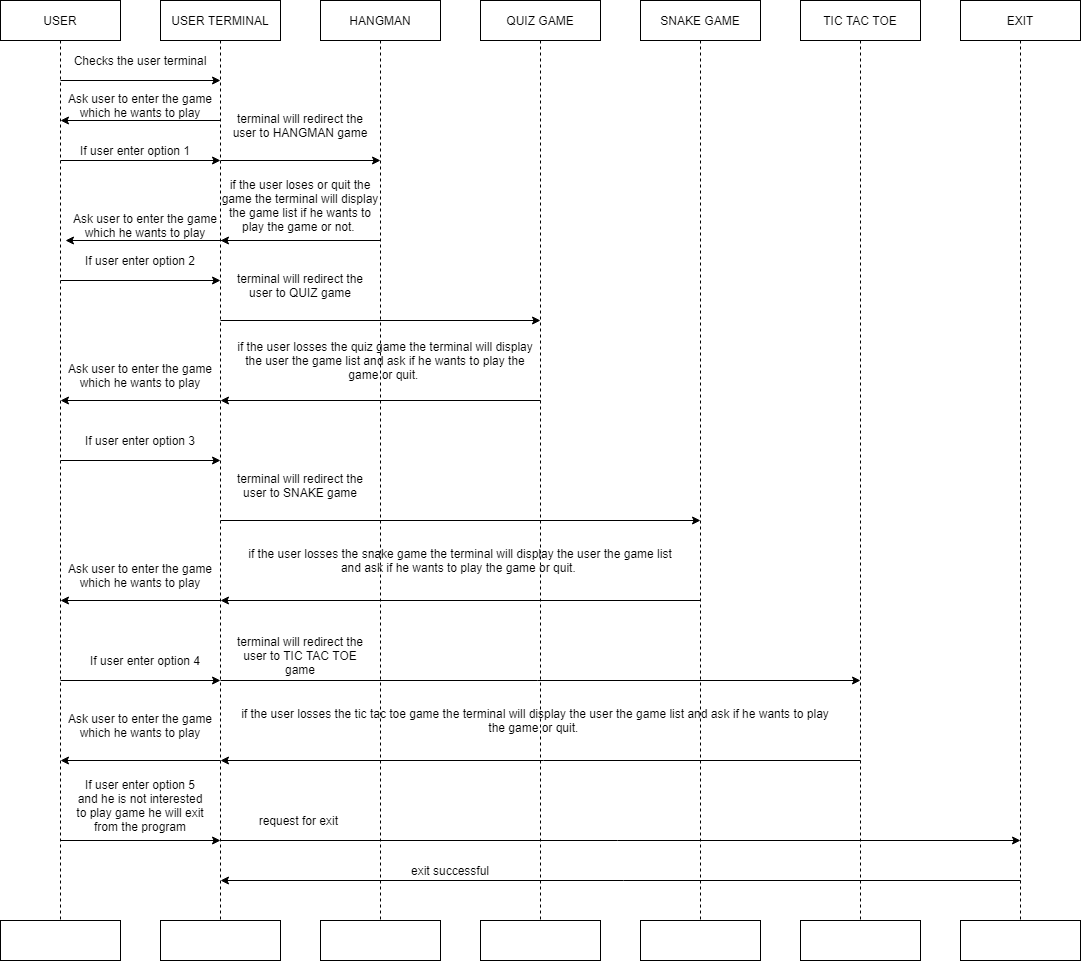
Game Zone Use case Diagram



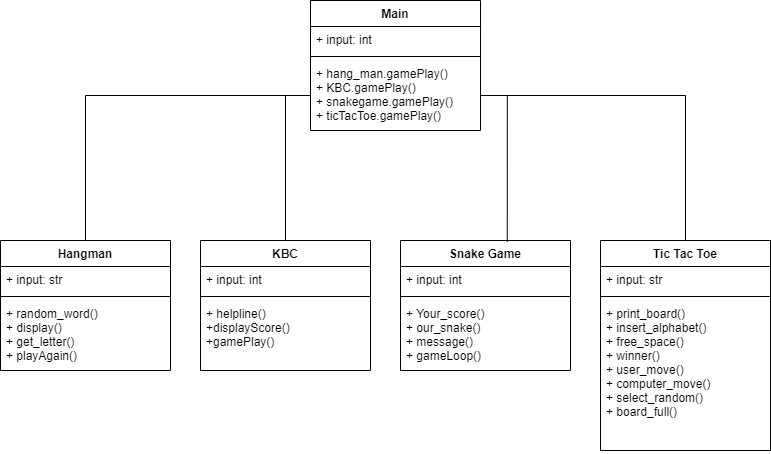
Game Zone Component Diagram



Game Zone Sequence Diagram



Game Zone Class Diagram



**Test Plan**

High Level Test Plan

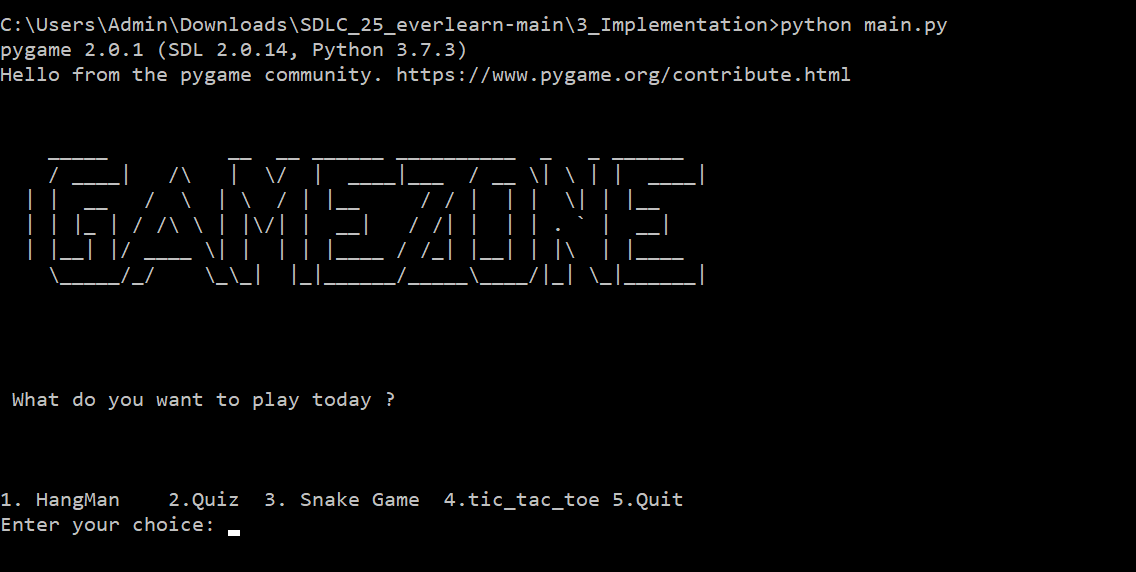
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Status** | **Test Type** |
| HR01 | System Compatibility | System requirements | Environment ready | PASS | Scenario |
| HR02 | Random word Selected | Key press | A Word selected to guess from a set of words | PASS | Technical |
| HR03 | User chooses a letter | Input letter | user guessed a letter | PASS | Technical |
| HR04 | User wins if all letters are guessed correctly | Input letter | 'User wins' message shown | PASS | Technical |
| HR05 | User loses if all letters are not guessed correctly after some attempts | Input letter | 'User loses' message shown | PASS | Technical |
| HR06 | Rules for quiz | Key press | Quiz rules displayed to the user | PASS | Technical |
| HR07 | Status of question answered | User option | Correct or Incorrect answer with amount displayed | PASS | Technical |
| HR08 | Check if quiz game over | User option | 'Game finished' message shown with final amount obtained | PASS | Technical |
| HR09 | Grid for TicTacToe | Key press | Grid displayed for the game | PASS | Technical |
| HR10 | User and Computer play | Key press | User and computer play alternatively with X and O | PASS | Technical |
| HR11 | Tictactoe Game complete | Key press | Game ends with the winner name displayed | PASS | Technical |
| HR12 | Snake GUI | Key press | GUI opened properly | PASS | Technical |
| HR13 | Snake movement | Key press | Snake moved in correct direction | PASS | Technical |
| HR14 | Snake growth | Key press | Length of snake increased when hits the cube | PASS | Technical |
| HR15 | Snake game complete | Key press | 'Game completed' message with the score shown | PASS | Technical |

Low Level Test Plan

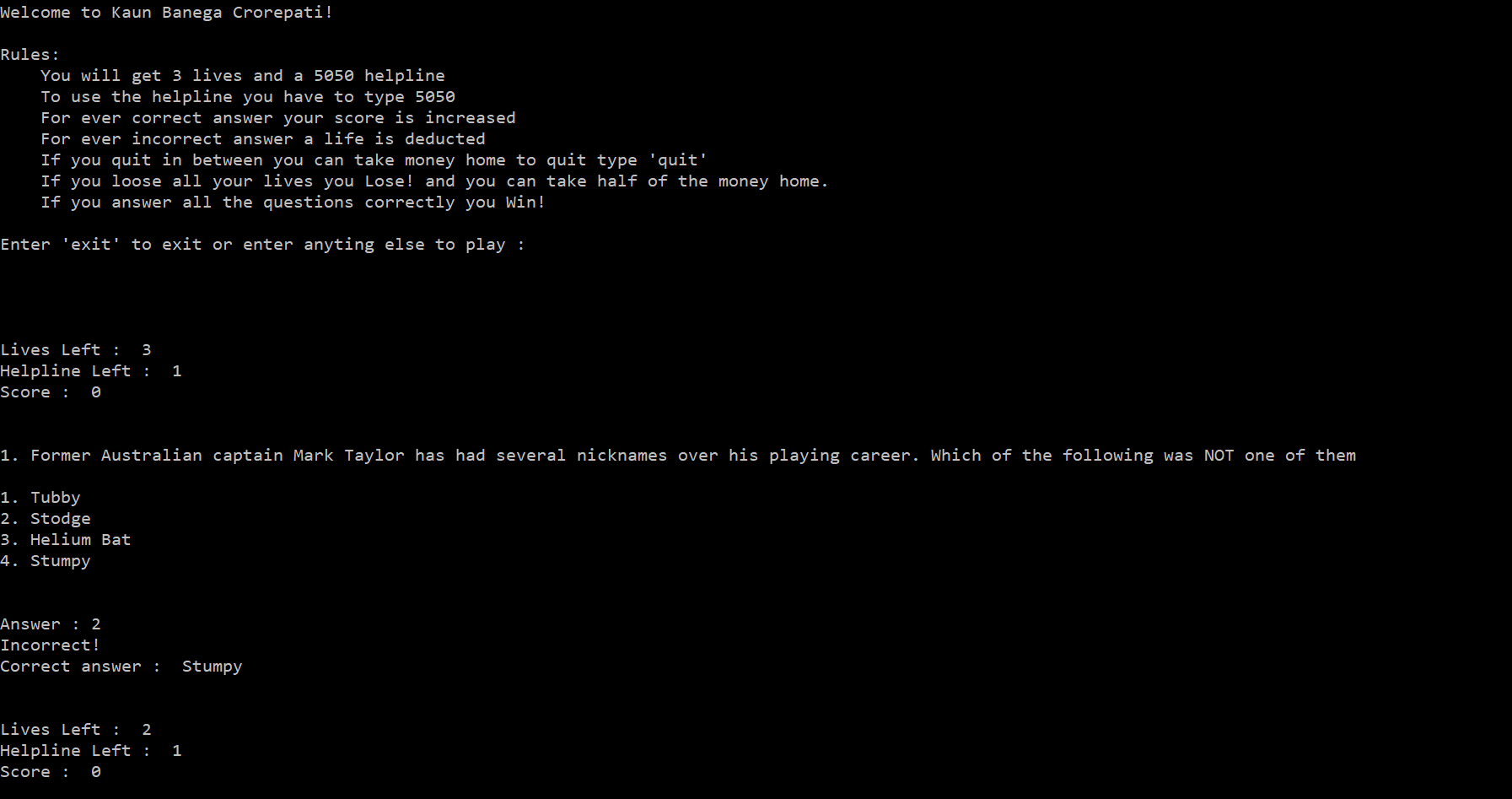
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **HLT ID** | **Description** | **Expected Input** | **Expected Output** | **Status** | **Test Type** |
| LR01 | HR01 | Requirements setup | System setup | Setup Success | PASS | Scenario |
| LR02 | HR03 | Choose correct letter | Input letter | Part of word formed with the letter | PASS | Requirement |
| LR03 | HR03 | Choose wrong letter | Input letter | A part of hangman formed | PASS | Requirement |
| LR04 | HR05 | Check number of attempts | Key press | Maximum number of attempts reached and hangman formed completely | PASS | Requirement |
| LR05 | HR05 | User play again | Key press | User continue or quit playing | PASS | Requirement |
| LR06 | HR06 | Rules for quiz | Key press | User accepts or rejects the rules | PASS | Requirement |
| LR07 | HR07 | Quiz question status | User option | Displays correct or incorrect option with scores | PASS | Requirement |
| LR08 | HR07 | Lives status | User option | Displays the lives and helpline left | PASS | Requirement |
| LR09 | HR08 | All lives are over | Key press | 'Game over' message with final amount scored | PASS | Requirement |
| LR10 | HR08 | All questions are done | Key press | 'Game over' message with final amount scored | PASS | Requirement |
| LR11 | HR08 | User play again | Key press | User continue or quit playing | PASS | Requirement |
| LR12 | HR09 | Tictactoe grid designed | Key press | A empty 3x3 grid created for tictactoe | PASS | Requirement |
| LR13 | HR09 | Grid update | key press | The grid updated and displayed after every move | PASS | Requirement |
| LR14 | HR10 | User play | Number from positions 1-9 which is remaining | X placed at user given position | PASS | Requirement |
| LR15 | HR10 | Computer play | Number from positions 1-9 which is remaining | O placed at selected position | PASS | Requirement |
| LR16 | HR11 | User wins | Key press | 'User won' message shown | PASS | Requirement |
| LR17 | HR11 | Computer wins | Key press | 'Computer won' message shown | PASS | Requirement |
| LR18 | HR11 | User play again | Key press | User continue or quit playing | PASS | Requirement |
| LR19 | HR12 | GUI design | Length and width for the frame | GUI frame created | PASS | Requirement |
| LR20 | HR13 | Snake head coordinate change | Arrow keys for movement | Snake movement in the specified direction | PASS | Requirement |
| LR21 | HR14 | Snake reaches maximum length | Key press | 'User wins' message | PASS | Requirement |
| LR22 | HR15 | Snake hits itself | Key press | 'Game over' message with scores | PASS | Requirement |
| LR23 | HR15 | Snake hits wall | Key press | 'Game over' message with scores | PASS | Requirement |
| LR24 | HR15 | User play again | Key press | User continue or quit playing | PASS | Requirement |

Output

Main Menu

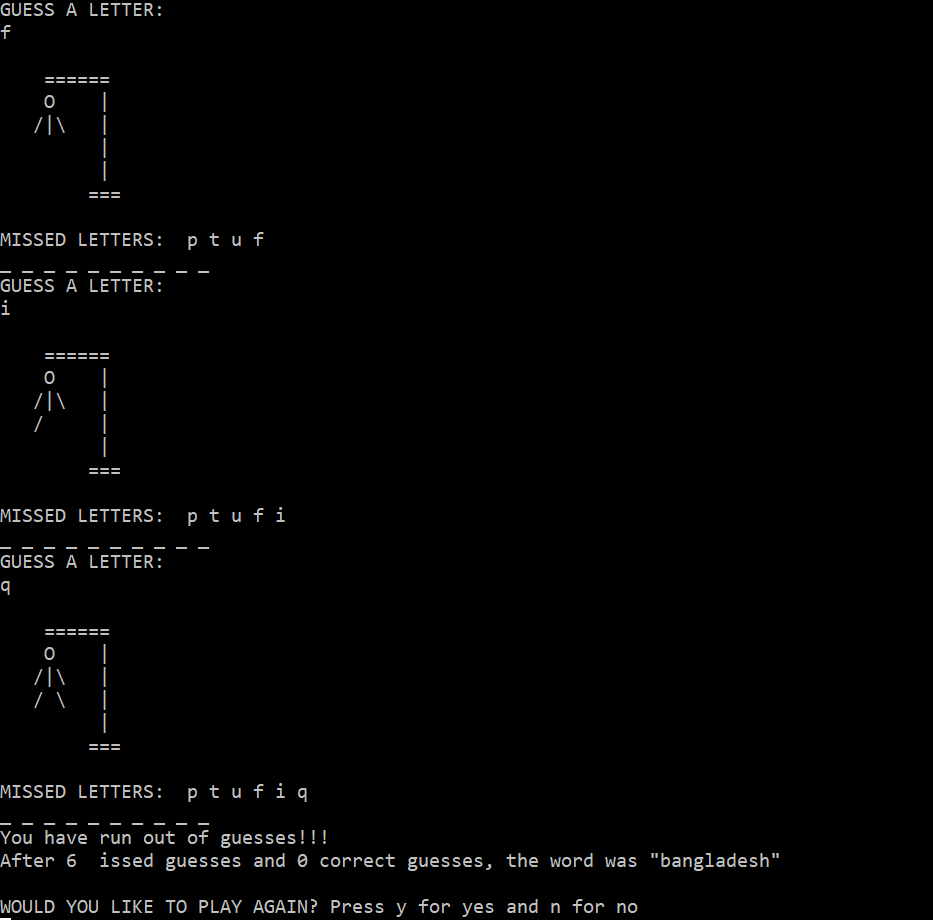


Quiz Game

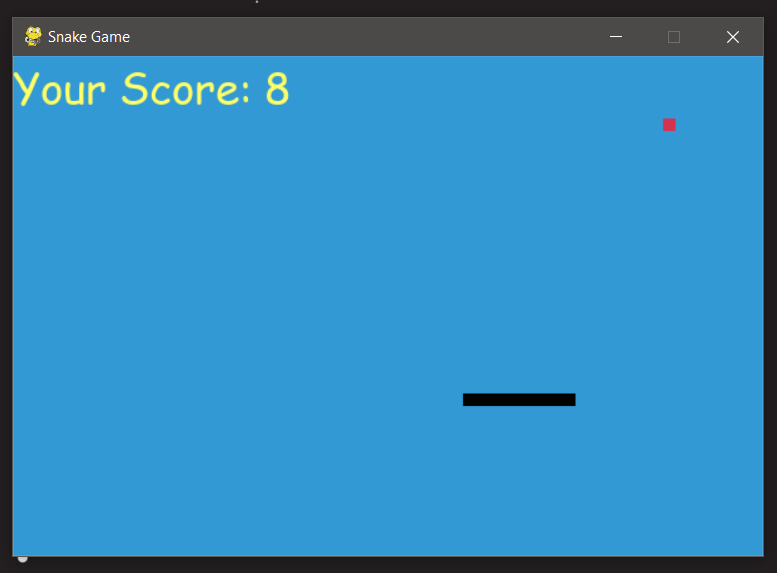


Hangman Game





Snake Game



Tic Tac Toe Game

