**Enemy Hunter**



Session: 2023 – 2027

**Submitted by:**

Muhammad Talha 2023-CS-12

**Supervised by:**

Dr. Awais Hassan

&

Mam Maida Mirza

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

***Contents***

[**Rubrics:…………………………………………………………………………………………………**11 2](#_Toc153871005)

[**Short Description** 3](#_Toc153871006)

[**Player** 3](#_Toc153871007)

[**Enemies** 3](#_Toc153871008)

[**Game Objects Description** 3](#_Toc153871009)

[**Rules & Interactions** 4](#_Toc153871010)

[**Goal of the Game** 4](#_Toc153871011)

[**Wireframes** 4](#_Toc153871012)

[*Figure 2: Main game* 5](#_Toc153871013)

[*Figure 3: Options menu* 6](#_Toc153871014)

[*Figure 4: Loading Screen* 6](#_Toc153871015)

[*Figure 6:Win Screen* 7](#_Toc153871016)

[*Figure 5: Game over* 7](#_Toc153871017)

[**Data Structures** 7](#_Toc153871018)

[**Function Prototypes** 8](#_Toc153871019)

[**Complete code** 9](#_Toc153871020)

[**Rubrics:** 10](#_Toc153871021)

# **Short Description**

Once there was a Bird who has to live peacefully on the Earth, But ongoing he was tracked into the trap of some bad Enemies that tried to kill him and tore him into pieces. Now he has to save himself from them.He started fight against them knowing that if he would not win then he will not stay alive. Let’s see what happen next. **Game Characters Description**

## **Player**

There is a one-player name Quack.

**Quack:**  
Quack is the main character of the story, a bird who is determined and brave. Throughout the game, he uses his skills to overcome obstacles and escape from dangers. Despite the challenges he faces, Quack never gives up and uses his courage and intelligence to find his way back home.

**Enemies**  
There are 3 enemies in the game.

**Piller:**  
Piller is a relentless enemy in the game, who can only moves vertically on the game. But he can fire projectiles at the player. Its movement is paired with its deadly accuracy. The player will need to be quick to avoid Piller shots.

**Stone:**  
Stone is a powerful enemy who patrols Horizontally on the ground and attacks the player on contact. Despite its inability to fire bullets, Stone’s brute strength makes him a formidable foe. Its massive size and slow movement make it difficult to avoid.

**Grace:**  
Grace is a formidable enemy that can move diagonally on the trap. He can also fire bullets. His accuracy and quickness make it a dangerous threat to the player. Grace’s bullets are highly destructive and difficult to dodge.

# **Game Objects Description**

Following are the Objects in the Game

**Player bullets:**

There will be player bullets represented by ‘o’. When a player’s bullet will hit the enemy the score will increase and the enemy’s health will decrease according to the applied conditions. **Enemy bullets:**

There will be enemy bullets like Piller fires ‘=’ , Stone fires ‘e’ and Grace fires ‘x’. When the enemy bullets interact with the player its health will be reduced.

# **Rules & Interactions**

Quack must have wings to avoid the bullets of enemies . If he touches a enemy bullet he will lose a life. If he touches an enemy or his enemy’s bullets touch him, he will also lose a life. He can add a life to his current lives by eating life pallets. The maximum life limit is three. If he collects a score pallet his score will increase by 10 from the current score. He can also fire bullets for a certain time by collecting shot pallets.

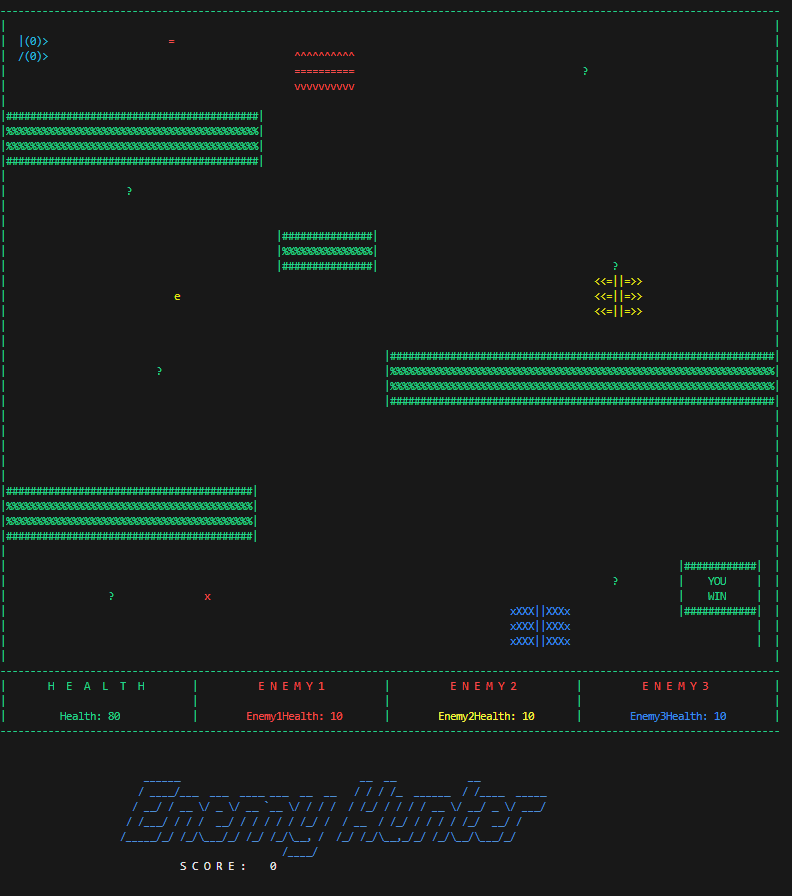
# **Goal of the Game**

The goal of the game is to overcome and beat each enemy one by one and make all the enemies dead and make a way back to earth.

# **Wireframes**



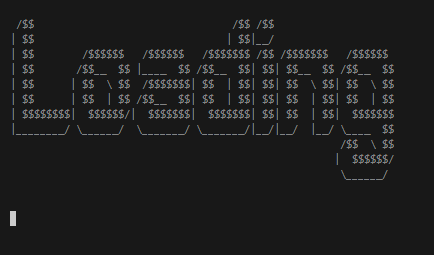
*Figure 1: Game menu*



## *Figure 2: Main game*



## *Figure 3: Options menu*



## *Figure 4: Loading Screen*



## *Figure 6:Win Screen*



## *Figure 5: Game over*

# **Data Structures**

* /// Array for Fire
* char fires [10000];
* int fireX [10000];
* int fireY [10000];
* bool isFireActive [10000];
* int fireCount = 0;
* /////For Enemies
* bool moveHorizontal = false; // for change of direction
* bool moveVertical = false; // for change of direction
* bool moveDiagnol = false; // for change of direction
* bool moveHorizontal2 = false; // for change of direction
* ///// Variables for Player & Enemy
* int eX1 = 49, eY1 = 4;///E1
* int eX2 = 2, eY2 = 20;//E2
* int eX3 = 98, eY3 = 27;//E3
* int pX = 3, pY = 2;//Player
* int e1Bx = 0, e1By = 0;
* bool e1Fired = false;
* int e2Bx = 0, e2By = 0;
* bool e2Fired = false;
* int e3Bx = 0, e3By = 0;
* bool e3Fired = false;

# **Function Prototypes**

* //Header
* string header();
* void inGameHeader();
* void loading();
* void gameOver();
* void Win();
* void instructionHeader();
* void hideCursor();
* /// Maze
* void printMaze();
* //Player print
* void printPlayer();
* //erase
* void erasePlayer();
* ///Player Movement
* void movePlayerLeft();
* void movePlayerRight();
* void movePlayerUp();
* void movePlayerDown();
* /////Enemy Print
* void printEnemy1();
* void printEnemy2();
* void printEnemy3();
* void printEnemy4();
* /////Enemy Erase
* void eraseEnemy1();
* void eraseEnemy2();
* void eraseEnemy3();
* void eraseEnemy4();
* /////Move Enemy
* void moveEnemy1();
* void moveEnemy2();
* void moveEnemy3();
* void moveEnemy4();
* ///check Bullet Hits Enemy
* void checkIfFireHitsEnemy(int [], int []);
* void checkIfFireHitsPlayer(int Enemy1FireX[], int Enemy1FireY[]);
* ///
* void reduceEnemy1Health();
* void reduceEnemy2Health();
* void reduceEnemy3Health();
* void reduceplayerHealth();
* /// Updates the score
* void updatescore();
* string setColor(unsigned short color);
* void gotoxy(int, int);
* char getCharAtxy(short int,short int);
* /// Health Player & Enemy
* void health();
* void enemyHealth();
* ////Create Fire
* void generateFire();
* //Erase Fire
* void eraseFire(int fireX,int fireY);
* //Bullet Go Inactive
* void bulletGoInactive(int idx);
* //Print Fire
* void printFire(int fireX,int fireY);
* // Delete Fire from Array
* void deleteFireFromArray(int);
* ///move fire
* void moveFire(int fireX[],int fireY[]);
* ////
* void e1FIREkrdo();
* void e1MoveFIRE();
* void e2FIREkrdo();
* void e2MoveFIRE();
* void e3FIREkrdo();
* void e3MoveFIRE();
* /// Score Increment
* void addScore();

# **Complete code**

# **Rubrics:**

**Student Reg. No. : 2023-CS-12**   **Student Name.**  Muhammad Talha

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A-Extensive Evidence** | **B-Convincing Evidence** | **C-Limited Evidence** | **D-No Evidence** |
| Documentation Formatting **Grade:** | All the documentation meets all the criteria. | Documentation is well formatted but some of the criteria is not fulfilled. | Documentation is required a lot of improvement. | Documentation is not Available |
| **Documentation Formatting Criteria:** In **Binder**, **Title** Page, **Header**-Footers, Font **Style**, Font **Size** all are all consistence and according to given **guidelines**. Project **Poster** is professionally design and well presented | | | | |
| Documentation Contents  **Grade:** | Documentation includes all of the criteria. | Documentation meet more than 80% of the criteria given. | Documentation meet more than 50% of the criteria. | When the documentation meet less than 50% of the criteria. |
| **Documentation Contents Criteria:** **Title** Page - **Table** of Contents - Project **Short Description and Story Writing of Game** - **Game Characters** Description - **Rules** & Interactions - **Goal** of the Game **- Screenshot** of the Game - **Data Structures** Used in the Game - **Functions** Prototype - **Full Code** | | | | |
| Project Complexity  **Grade:** | Project has at least 1 Player and 3 enemies. Proper use of gotoxy() function. Health system, Firing System and lives decreasing system. | Project complexity meet 80% criteria given in extensive evidence | Project complexity meet 50% criteria given in extensive evidence | Project complexity meet less than 50% criteria given in extensive evidence |
| Randomness  **Grade:** | Objects are produced randomly in the game. | meet more than 80% of the criteria given. | meet more than 50% of the criteria given. | Objects are appearing in the same pattern |
| Code Style  **Grade:** | All Code style criteria is followed | All code style criteria followed but some improvements required | lot of improvements required in coding style. | **Did not follow** code style, |
| **Code Style Criteria:**  Consistent code style. Code is well indented. Variable and Function names are well defined.  White Spaces are well used. Comments are added. | | | | |
| Code Documentation Mapping  **Grade:** | Code and documentation is synchronized. | Code and documentation does not synchronized at **some** places | Code and documentation does not synchronized at **many** places | Code and documentation **does not** synchronized. |
| Idea Novelty and Creativity  **Grade:** | Idea is unique of the game | Idea is merged by combining other different games | Same idea as a previous game | Could not implement the existing game idea. |
| Data Structure (2D Arrays)  **Grade:** | Data structure is sufficient for the project requirements | Data Structure is sufficient but require improvement to meet project requirements. | Data structure is not sufficient and need a lot of improvement | Data Structure is not properly identified and declared. |
| File Handling  **Grade:** | Game maze is loaded and the updated maze is stored in the file | Game maze is loaded and partial data is stored in the file. | Game maze is just loaded but the updated game configuration is not stored in the maze. | Project do not contain file handling |
| Modularity  **Grade:** | Meet all Modularity criteria | Meet all Modularity criteria but at some places it is missing | Do not sufficiently meet the modularity criteria. | No modularity or very minimum modularity. |
| **Modularity criteria:** Functions are defined for each major feature. Functions are independent (identify from parameter list and return types)- There is no global variable defined. Arrays and variables are passed as parameters to the functions. Functions exhibit single responsibility principle. | | | | |
| Screen flickering  **Grade:** | There is no Screen flickering. | Maze is not flickering but the characters are flickering at great speed | Flickering is done at lot of places | Screen is flickering at all places |
| Presentation and Demo  **Grade:** | Presentation and Demo was 100% working | Presentation and Demo require some improvements | Presentation and Demo require a lot of improvements | Presentation was not ok and Demo was not working |
| Student Understanding with the Code.  **Grade:** | Student has complete understanding how the code is working and knows the concept. | Student has good understand but some place he does not know the concepts | Student has a very little understand and lack the major concepts. | Student does not have any level of understanding of the code. |

|  |  |
| --- | --- |
| **Checked by:** | Click or tap here to enter text. |