**Hungry Shark**

**Group Members: Hassan Rasheed  
 Sikander Azam  
Class: BSCS-5B**

**Programming Project Report**

**Table of Contents**

1. **Abstract……………………………………………….**
2. **Concepts …………………………………………….**
3. **Snapshots of Code and Explanation…….**
4. **Class Diagram ……………………………………..**

**Abstract:**

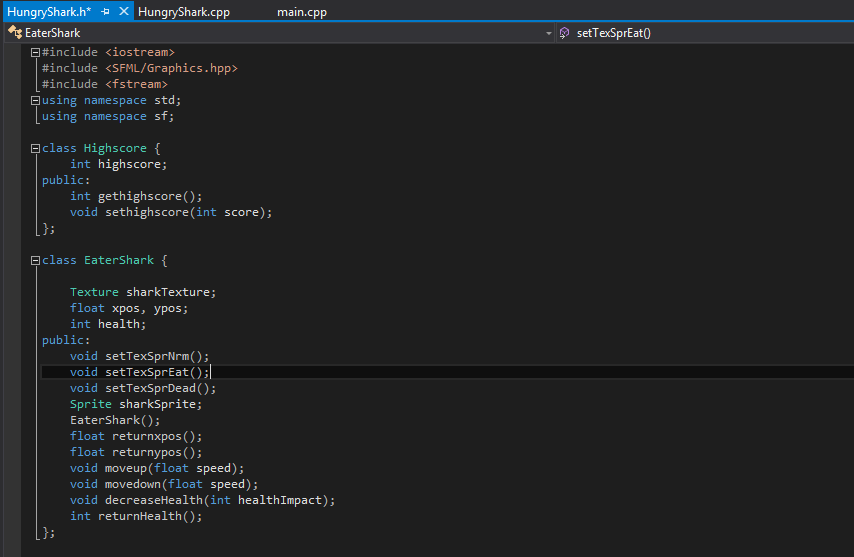
This Project is basically a fish eating game made using C++ object oriented concepts. The graphics used are of SFML graphics library. Many small fishes move from right to left of the window and there is a Shark which we move up and down on the left end of the window. If small fishes touch the shark, it eats them. The score increases if the shark eats the fish, it decreases if the shark misses. Each small fish has its own health impact. The game   
goes on until health bar of shark reaches zero!

**Concepts:**

* **Object Oriented Concepts (Classes)**
* **Setter & Getter functions**
* **Constructor & Destructor**
* **Inheritance**
* **Multiple Inheritance**
* **Diamond Problem**
* **Virtual Functions**
* **Abstract Classes**
* **Pure Virtual Functions**
* **Constant Class Members**
* **File Handling**
* **Dynamic Memory Allocation**

**Screenshots of Code:**

**Header File:**

****

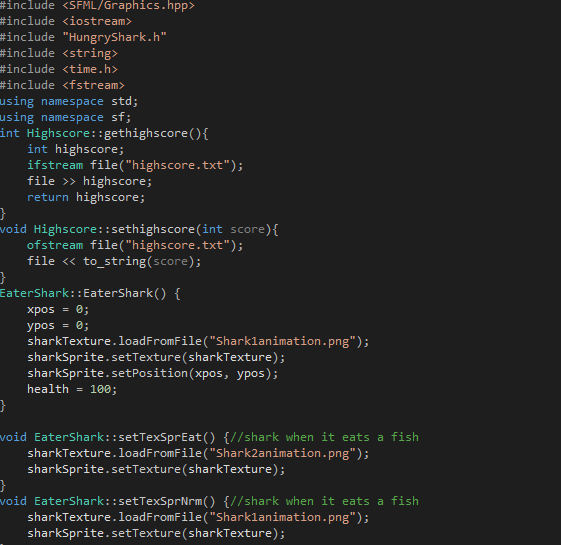
****

****

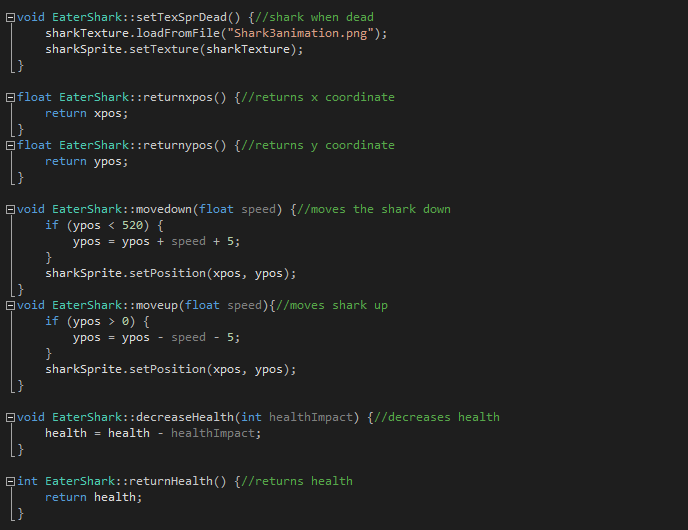
**This is our header file and contains all the declarations of the functions used in this project.**

* **The Highscore class is used to store the highest score made by a player in the game, this data is stored in a text file in the project folder using file handling.**
* **Class EaterShark has all the members and functions of the Shark used in our game eating the small fishes. It includes functions to move the fish up and down, decreasing its health and creating the sprite (image) of the shark.**
* **The class Fish is the base class of all the small fishes that the shark eats. It includes the general data for all the small fishes such as their movement function, sprite declaration, texture declaration and position coordinates declaration.**
* **The greenfish, redfish and yellow fish classes are inherited classes of Fish class.**
* **The Main class contains the speed and score functions used to increase score and speed of fishes as the game progresses.**
* **Maingame() is our main function of the game and has all the graphics (SFML) liked with our classes.**

**Source File:**

****

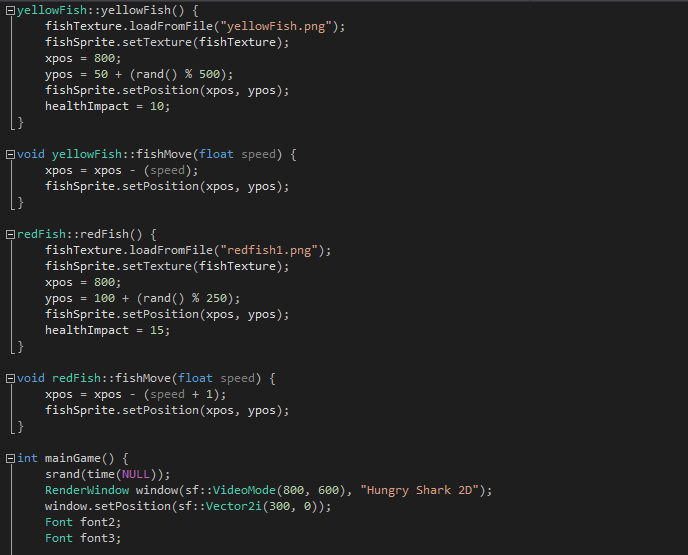
* **The Source file has all our functions defined.**
* **Gethighscore reads the high score forms the text file saved in the project folder.**
* **Sethighscore writes takes the high score from game (if made) and stores it in the text file.**
* **EaterShark() is the constructor for the shark object and initializes it’s position ,texture, sprite and health.**



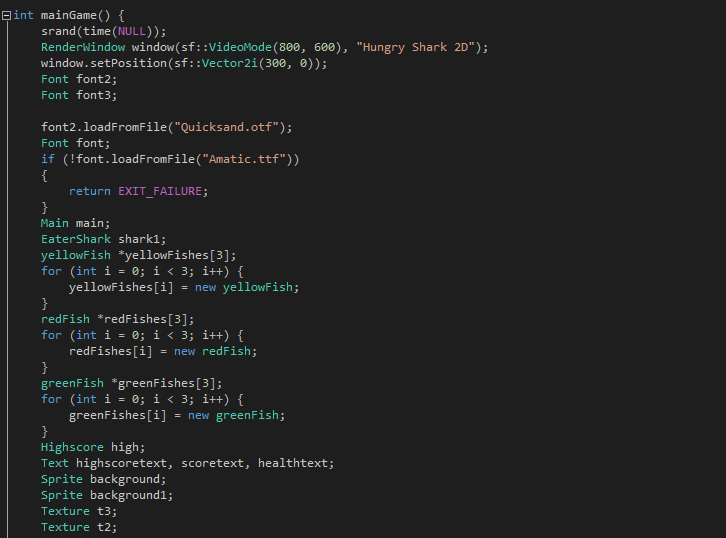
* **Movedown function moves the shark down the window.**
* **Moveup moves it up the window.**
* **Decrease health takes healthimpact of the small fish as argument and decreases sharks health if it misses the small fishes.**

****

* **Main() is the constructor for main class object and sets initial score and speed.**
* **ScoreIncrement increases the score as the shark eats small fishes, it takes fished health impact as argument.**
* **The constructors of the small (yellow, green, red) fishes create sprites for the small fishes, sets their initial positions and health impacts.**
* **The fishmove function moves the small fishes across the window from left to right.**

****

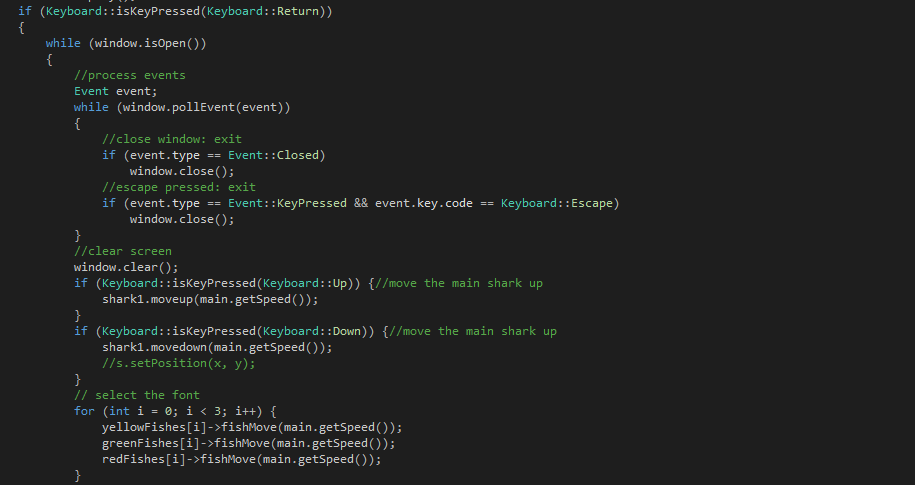
* **The mainGame() function is the function with all the SFML and C++ linked together.**

****

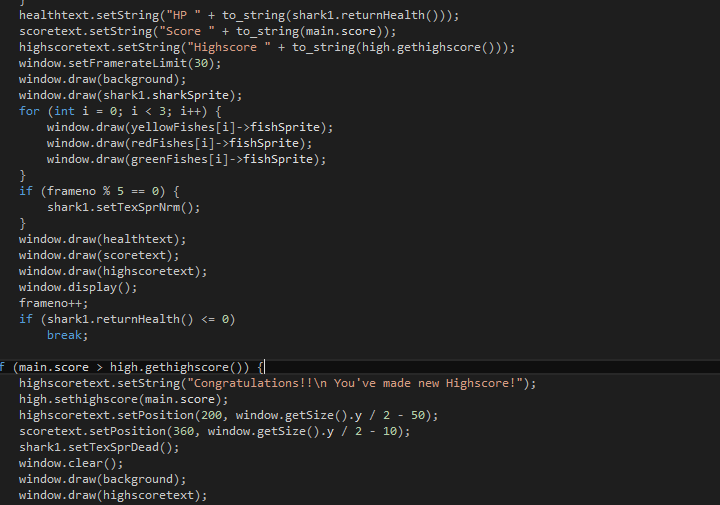
* **Fishes are given y-coordinate on the window using srand, rand() functions.**
* **Objects of the classes are made and used to call functions.**
* **For loops are used for creating array of small fishes.**
* **Fishes are created using dynamic memory allocation.**
* **Background sprites are included.**

****

* **The texts such as highscore, fish health are showed on the window by these function above.**

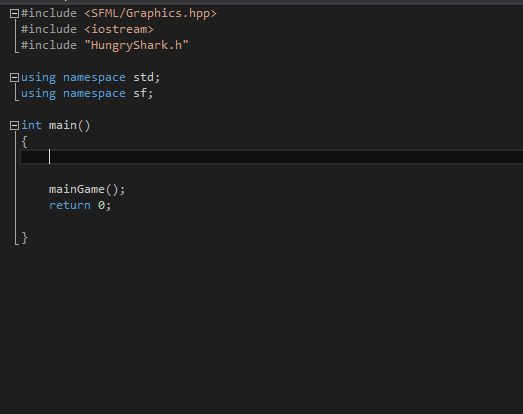
****

* **Instructions from the Keyboard are read using the iskeypressed function in SFML.**
* **Fishmove function takes speed of fishes as argument and moves the array across the window.**

****

* **All the sprites created are drawn in the above code lines using the draw() function in the SFML library.**

**Main:**

****

* **Our main function is pretty clean, it only calls the mainGame function in the source file.**

**Class Diagram:**

Class Main  
Members;  
speed;  
score;  
Functions:  
Main();  
increaseSpeed();  
returnScore();  
scoreIncrement()  
getSpeed();

Class Eater\_Shark  
Members:   
SizeofShark  
-SharkSprite;  
- Position  
- Speed  
- Health  
Functions:   
setTexSprNrm();  
setTexSprEat();  
setTexSprDead();  
returnxpos();  
returnypos();  
moveup(float speed);  
movedown(float speed);  
decreaseHealth(); returnHealth();

Class Fish  
Members:  
-Speed  
-Position  
-Heathimpact  
-Position  
-Fish Sprite  
-Fishtexture  
Functions:  
fishMove();  
healthDecrease();

Class RedFish: Public Fish  
Members:  
//inherited

Functions:  
Fishmove;

Class YellowFish: Public Fish  
Members:  
//Inherited  
  
Functions:  
Fishmove;

Class GreenFish Public: Fish  
Members:  
//inherited

Functions:  
Fishmove;