Basics of Programming-2: Project SNAKE GAME:

NAME: Chaitanya Arora
NEPTUN CODE: BWTFX8

Developer DOCUMENTATION

INDEX

Name: Chaitanya Arora Neptun Code: BWTFX8

Introduction:

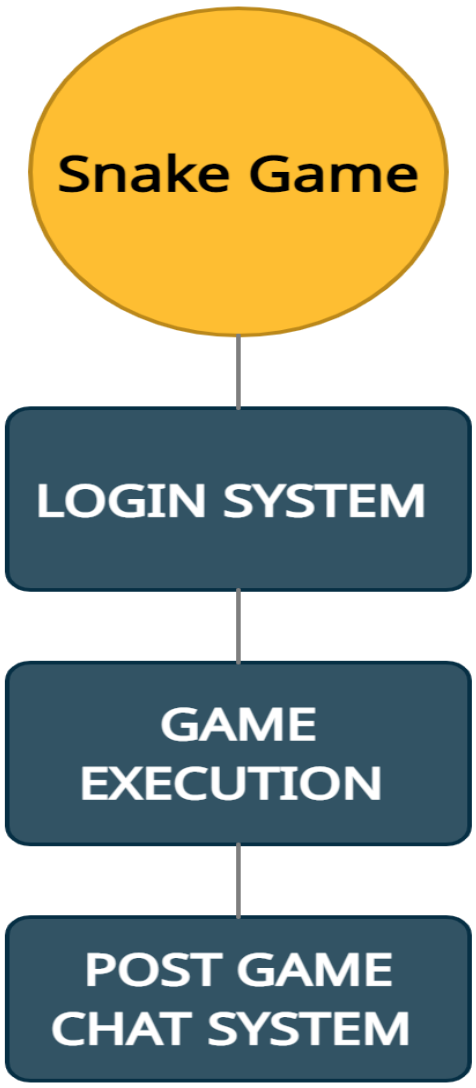
Welcome to the snake game developed in the C++ language. The project is divided into three parts as specified in the last submission.

1. Login System:
2. Game
3. Post-Game:

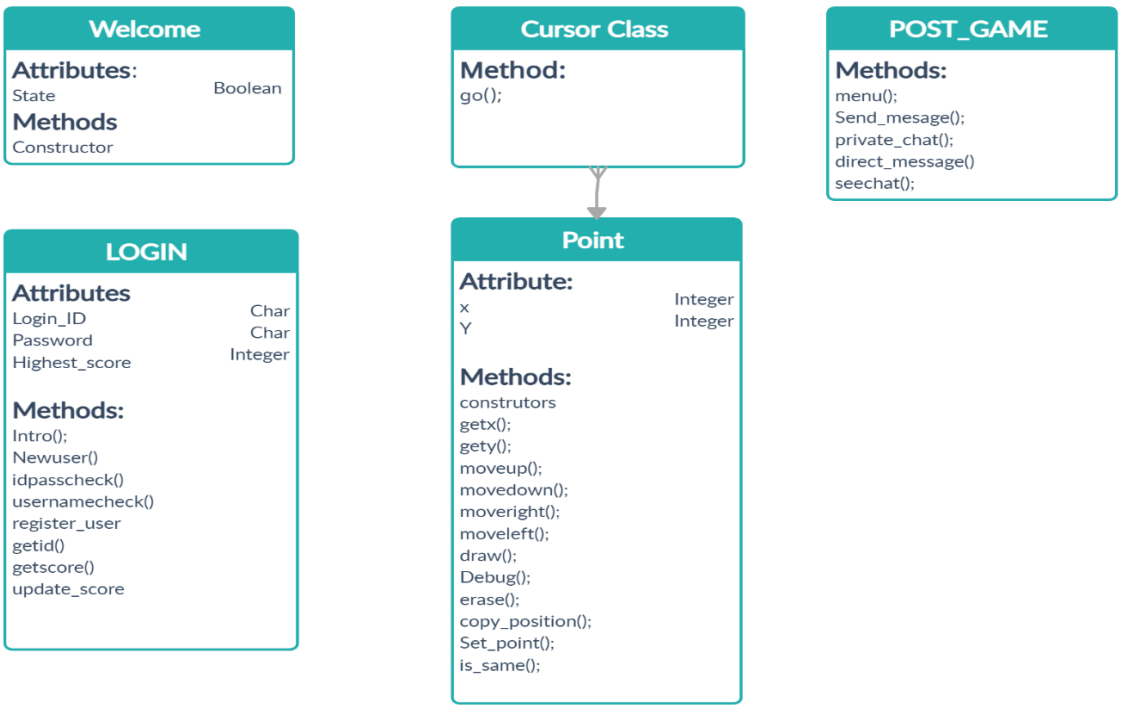
This project regressive use of Object oriented program , exception handling, file handling (binary as well as text file) , dynamic memory allocation, inheritance.

The code is cleanly divided in the three parts as mentioned above with separate .h and .cpp files for each of them.

The code is well guided with comments throughout the project.



Name: Chaitanya Arora Neptun Code: BWTFX8



Login System:

This is the first and the foremost part that the user will be able to see when he runs the program, He will be able to see menu driven program which asks the user if he is a new user or and existing user or if he just wants to play the game on the guest mode:

Implementation:

We created a welcome class which required a bool parameter consisting of a parameterized constructor. Which would print the welcome screen:
For the login system I created a class login which consisted of three attributes the login ID, password, both of character type and the user highest score as an integer in the private section.

This class had set of methods which did all the tasks:

1. int intro()

   This method is required to print the menu for the user, whether he is an existing user or not. The unique part in my code is that I have used while (1) loops as they are infinite loops they can be put to a good use, I wrote my menu driven program In the while(1) loop followed by if else conditions. So here If the user picks a wrong choice he/she will be presented with the options again so that we can choose the right one to move forward.

Name: Chaitanya Arora Neptun Code: BWTFX8

2. Void Existing_user()

This method is the follow up method after the user's choice to move forward as an existing user. This itself contains a menu driven program in the while(1) loop where we ask the user for his/her login id and password. This method triggers the other methods in the class namely : "Idpass_check(id,pass)".

3. Bool idpass_check(const char* id, const char* pass)

This function takes two parameters id and password and returns a Boolean value. This main idea behind creating this function is to check the id and password of the user in our local file of all users. This function opens the encoded (not readable) binary file in the system and reads the user's login id and password and logins in the system if the credentials are correct.

4. New user login : `void newuser();`

This function is the new user login it requires the user to set up a login id and password, since our id is our primary key. This method calls the other methods in the class i.e. "bool username_check()" This function also opens the file and checks if the username is present in the file if positive then we ask the user to choose another login Id as it is already present in our database and we can't have people with the same username.

Main Snake Game:

This is the second part where the actual snake game is made, after the user successfully logged in the system, we start the game. Before diving straight how the game work.
Let us discuss the logic how the game is made. I first thought that how will the snake be made? What will it look like? After doing good amount of research, I decided to have a snake which will have a many '*' which will move behind it.

I decided to make a point first and then the snake because if we refine the game as we see it, a snake is first a point and then many of them combine to form the snake's body. This made me first create a point class which consists of attributes and methods:
Attributes: x, y coordinates in the screen
Methods: constructors, parameterized constructor, getters, setters, movement functions, draw, erase function, copy position.

Name: Chaitanya Arora Neptun Code: BWTFX8

Detailed Description of the class:

Attribute:

I have created two integer x and y representing the x and y axis coordinate.
Methods:

Default constructor:

This basically makes use of the initialization list and assigns 10 units to both x and y.

Parameterized constructor:

This takes in two values as the parameter and assigns them to x and y.

Getter setter functions:

These are used to assign values to the variables or get the value of same variables which are private attributes of the class and cannot be accessed outside. So, we get them through a method which is the public section of the class.

Movement functions:

These are a combination of 4 functions which control the movement of the point, like by incrementing or decrementing the x and y attribute. Here I have also added a feature that if the value is greater than 100 then the snake comes from the other side.

Draw Function:

This is used to basically draw the point.

Class Snake:

Here is where I thought that the snake is made up of points so one of it's attribute should be a pointer of the type class point as here to execute dynamic memory allocation the snake's length will be dynamically allocated, as the snake is also an array of dots.

Attributes:

Size, high_score,current_score, direction, some Boolean variables and object of class point which will be the target that the snake will chase.

Method:

Name: Chaitanya Arora Neptun Code: BWTFX8

Constructor:

This basically initializes all attributes and dynamically allocate the first dot which will be visible to the user. And shows the dot for the target at a random location.

Getter, Setter:

To get high score of the user and to set the high score of the user.

Add dot:

This function is basically used to increase the value of the array and create a dot dynamically, this should be called when the snake eats the target.

TurnUp/TurnDown/TurnLeft/TurnRight:

This function assigns direction ‘w’ to the snake similarly for down left and right.

Move:

This is the major function which is exactly the master in the program, This involves firstly a condition which will check if the user is active, Active implies whether the user is playing or lost,
User Inactive(lost): then we display his score and whether it was highest or not and we ask him if he wants to play more or exit.

If the user is still active: Then we have to execute some tasks:

- Follow the direction based on input
- Eat the target
- Increase Its length
- Produce new target
- Check case of self collison
- Making each dot follow the snake’s point

In order to execute each of these, I added few of them in the move function as they will be needed every time the snake moves.

To Follow the direction based on input we have a set of if then else by which we call the required function.

For eating the target the logic I have used that the x and y coordinate will be same, so the target’s x and y coordinate will match with the same.

Name: Chaitanya Arora Neptun Code: BWTFX8

To increase the snakes length we do the same condition if it has eaten the target then a new point will be created dynamically.

Producing new target will be done randomly:

To make the points follow the snake we use a loop to draw the points.

Game_play:
This is basically the thing that will work first when the user is executing the game: it calls the move function and the required functions for directions based on the direction.

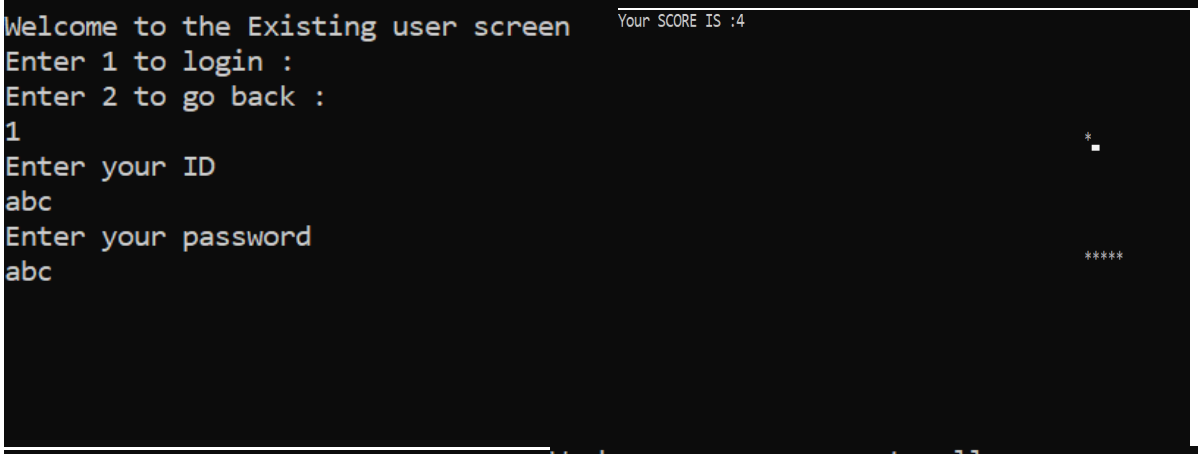
Post-Game:

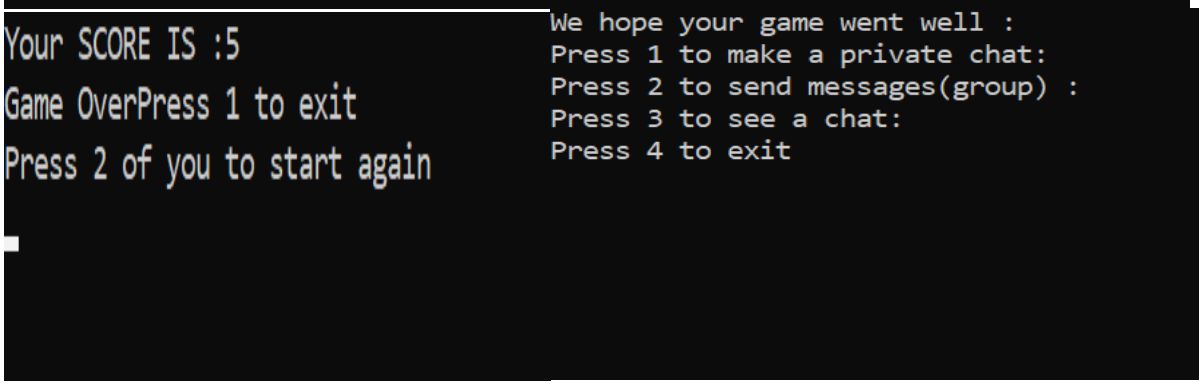
The post-game feature is used to create a chat system for the user-to-user interaction in the platform. For this I have made the use of file handling specifically text files to execute this, we provide the user a menu driven program to ask him if he wants to chat privately with a user or a normal group chat or see a chat. Based on his input we will execute the required function

Sending a private message to the user involves some kind of security so I have used a 6 letter confidential code that the user needs to write which will be the lock to his chat there we ask him whom is it exactly that he wants chat with, and check if he/she is a valid user. After verifying the user and the confidential code we take the user's message which will be written in the file.

Similarly in terms of group chat we need to ask the user a unique filename which will be our key field which the user can enter his or her message.

Screenshots of Project:





Name: Chaitanya Arora Neptun Code: BWTFX8

Name: Chaitanya Arora Neptun Code: BWTFX8