SNAKE

Kevin Dimarco

Brandon Sheppard

**Project Overview:**

The purpose of this application is to create an online multiplayer version of the classic video game “Snake”. Users are able to host their own match with friends, or to find a publicly hosted game match within our servers.

The main objective of the snake game is simply to stay alive longer than the other players. You move around a 2D grid, collecting food which grows your snake’s tail. Touching a wall or a snake’s tail (including your own) will eliminate you from the game. The last snake standing wins!

We are using ASP.NET to host our game server. The server will be responsible for handling all of the data across our game. SignalR will help us achieve asynchronous communication between the client and the server using Websockets requests. We will be using Microsoft SQL Server to hold our player information, and local text-based logs to hold large amounts of game data on a weekly rotation.

**Functional Requirements / Use Cases:**

Login:

1. Enter website
2. Click Login
3. Enter information
   1. Username
   2. Password
4. Click Login
5. Repeat until login was successful.

Register:

1. Enter Website
2. Click Login
3. Click Sign Up
4. Enter information
   1. Username
   2. Password
   3. Email
5. Repeat until account is valid.
6. User is registered

Reset Password

1. Enter Website
2. Click Login
3. Click Forgot Password
4. Enter email
5. Click forgot password
6. Link to reset password sent to email

Host Game:

1. Generates code to send to friends to join game
2. Input Information
   1. Choose map
   2. Choose rounds
   3. Choose player colour

(Once 2+ people are in game) Start Game button appears

1. Wait for host to start game
2. Spawn Snakes
3. Wait until Snakes < 2, or less than 2 players are in the game
4. Show Winner
5. Repeat until Round Cap is reached
6. Kick Players from game.

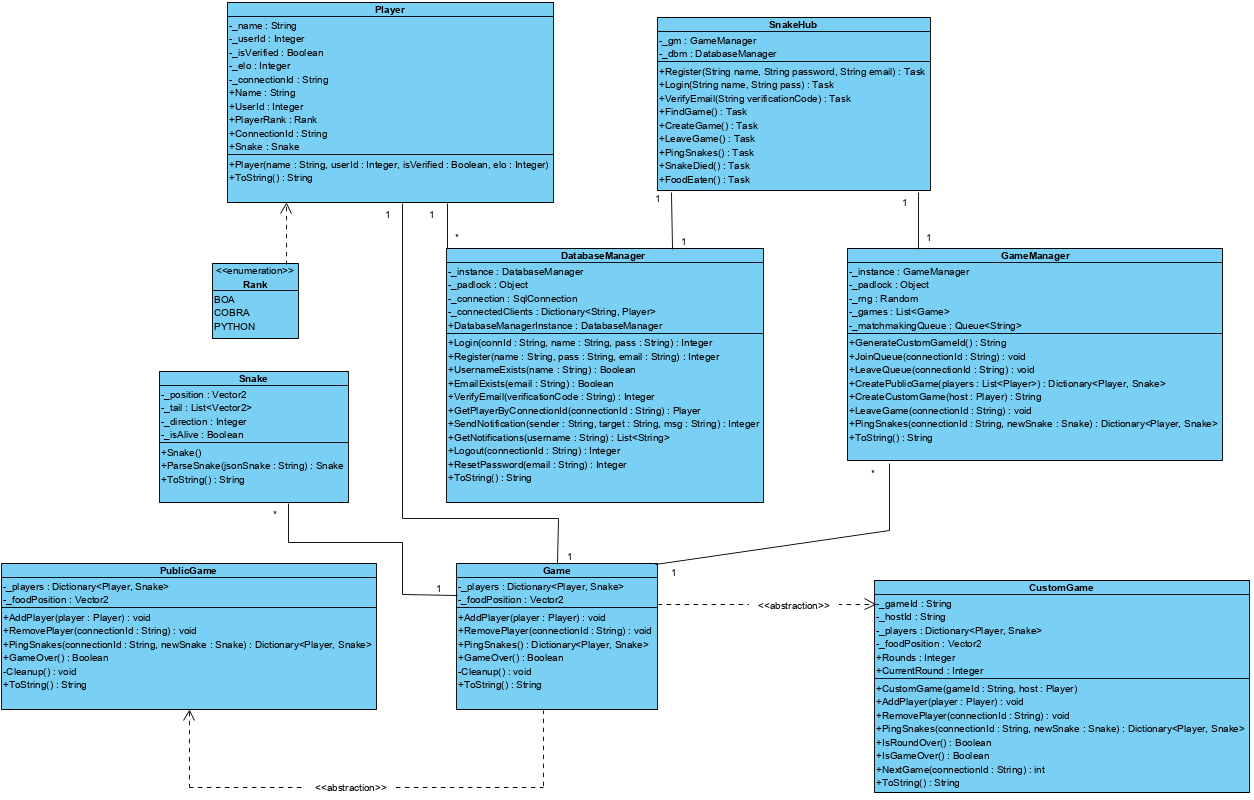
Join Game:

1. Use code given by host to join game
2. Choose player colour
3. Wait for host to press Start Game button
4. Game Loop
   1. Spawn Snakes
   2. Wait until Snakes < 2
   3. Show Winner
5. Repeat until round max is reached.
6. Kick Players from game.

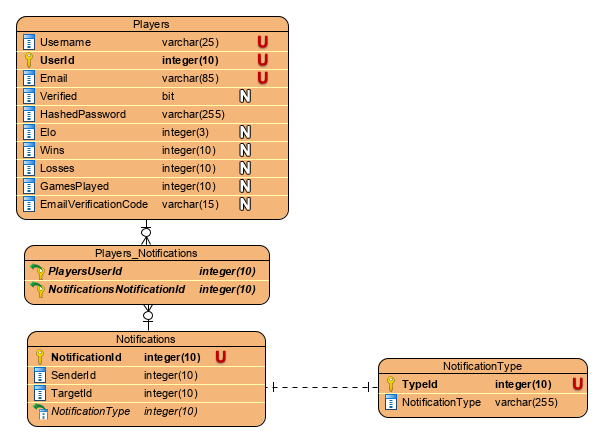
Find Game:

1. Click find game for server to generate game for you
2. Choose player colour
3. Wait for server to Start Game
4. Game Loop
   1. Spawn Snakes
   2. Wait until Snakes < 2
   3. Show Winner
5. Kick Players from game.

**Class Design:**

****

**Database Design:**



**Work Assignments:**

Game Loop (Javascript) (Brandon)

Client Communication (Javascript) (Kevin)

Server Communication (C#)

Game Logic

Game Manager (Brandon)

Game (Brandon)

Custom Game (Brandon)

Public Game (Kevin)

Players (Kevin)

Snake (Kevin)

Hubs

SnakeHub (Brandon)

Database

DatabaseConnection (Kevin)

**Project Milestones and Deliverables:**

Project Proposal:

By this point we have finished mapping out the basics of this project. We know the use cases and functional requirements needed for this application to run. We have an idea for the class and database designs. We have also figured out how we are going to divide the workload among the group members

Entity Classes:

By this stage in the project the c# classes will be completed along with the backend logic for the game. The classes will be ready to implement the database and present data to the UI.

UI Design and DB Access:

At this point the front-end will be completed, including game functions however will not be functional until the web services have been designed.

Web Service and Integration Testing:

By this point, we want to have the game implemented using SignalR for client communication.

Full Presentation:

This is the final step after the game is completely finished and working without errors. We will give an in-depth look at how this game functions and the logic behind it. We will show a real gameplay run through and offer the class to give the program a try in competing against their friends in our game of Snake.