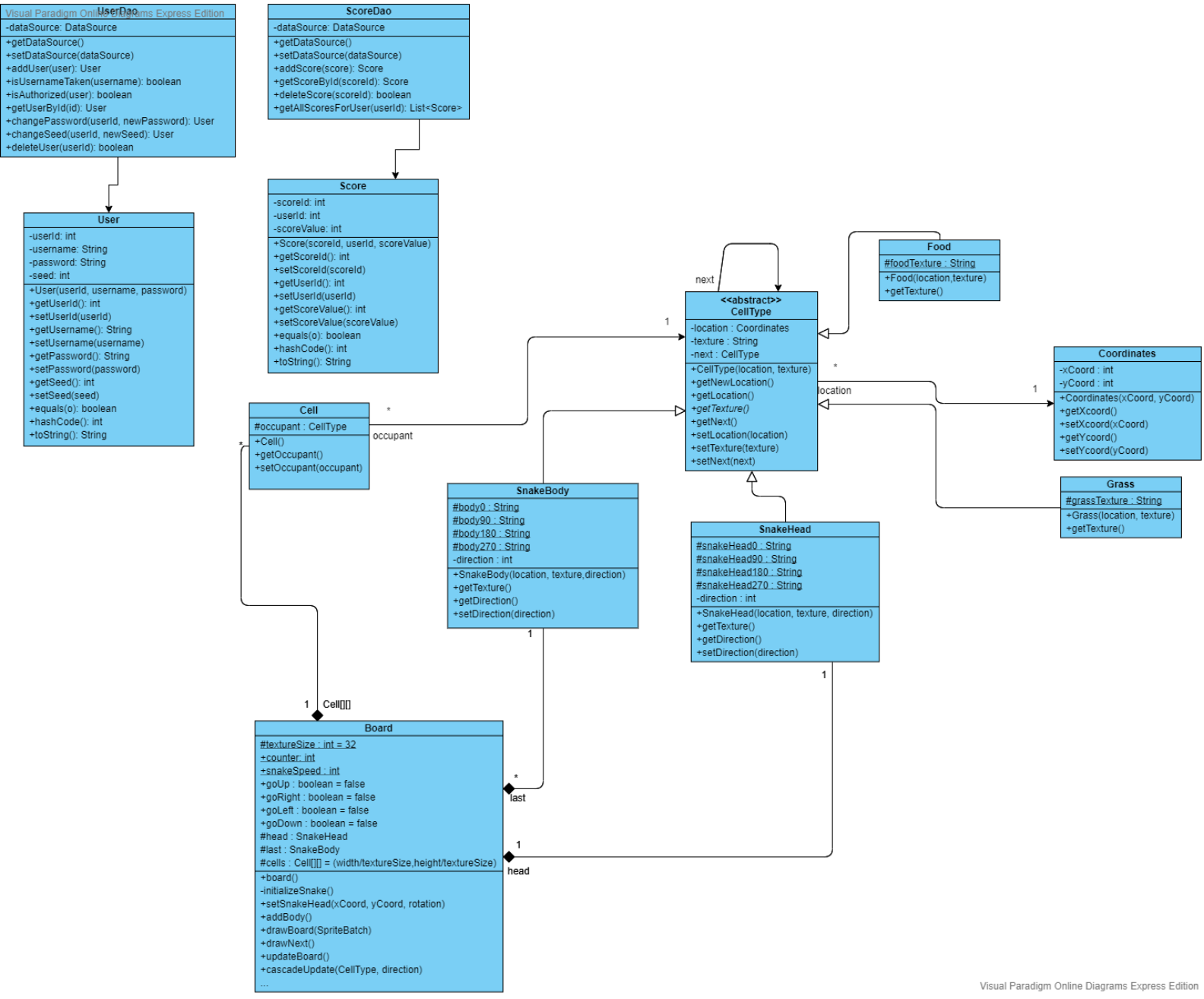
2019/12/05

Assignment 2: Group-Snake-5

Task 1:



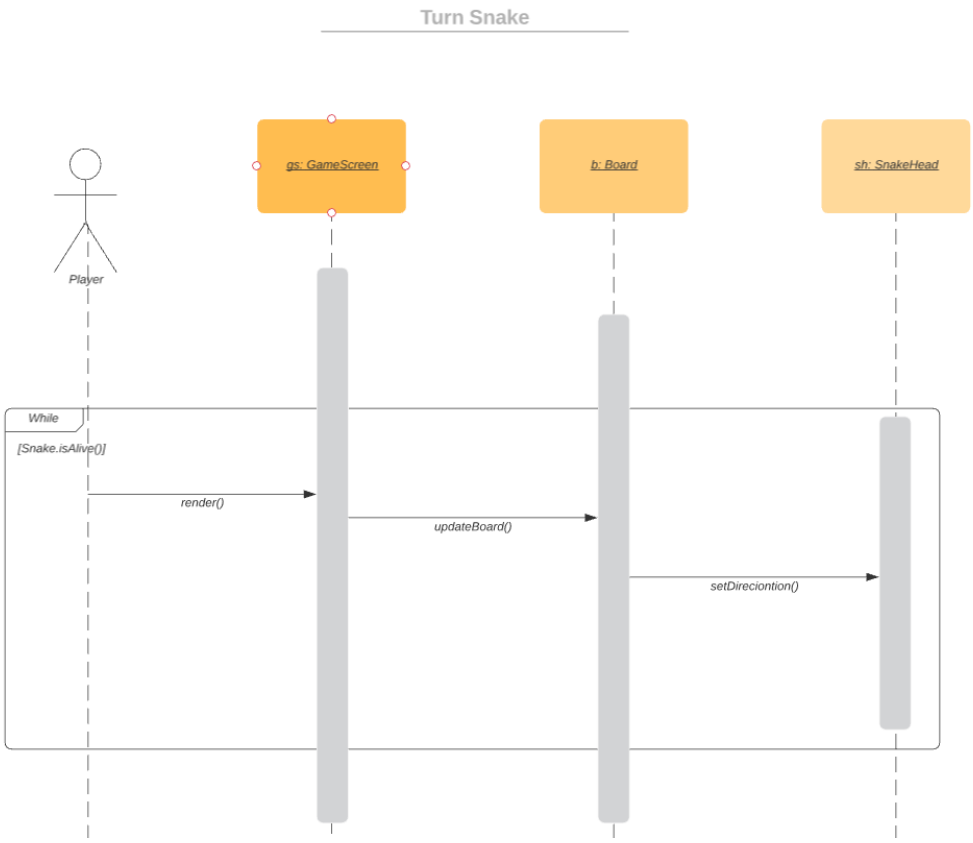
The reason we chose the following classes is as follows: To be able to play a game there must be a board that keeps track of all the cells and what occupants are exactly on there. This is needed to check for collisions between the snake and other objects. The occupants all share some common attributes, that is why we decided to go with the inheritance structure. To conveniently store the location of the occupants we introduced the Coordinates class that will store the x and y coordinates.

To interact with the database we decided to make a separate User and Score class, the score class is used to keep track of all the scores a user has achieved. The user class is used for authentication and identifying records that are stored in the database. For a nice interface we decided to implement the DAO classes which make it easy to interact with the database.

2019/12/05

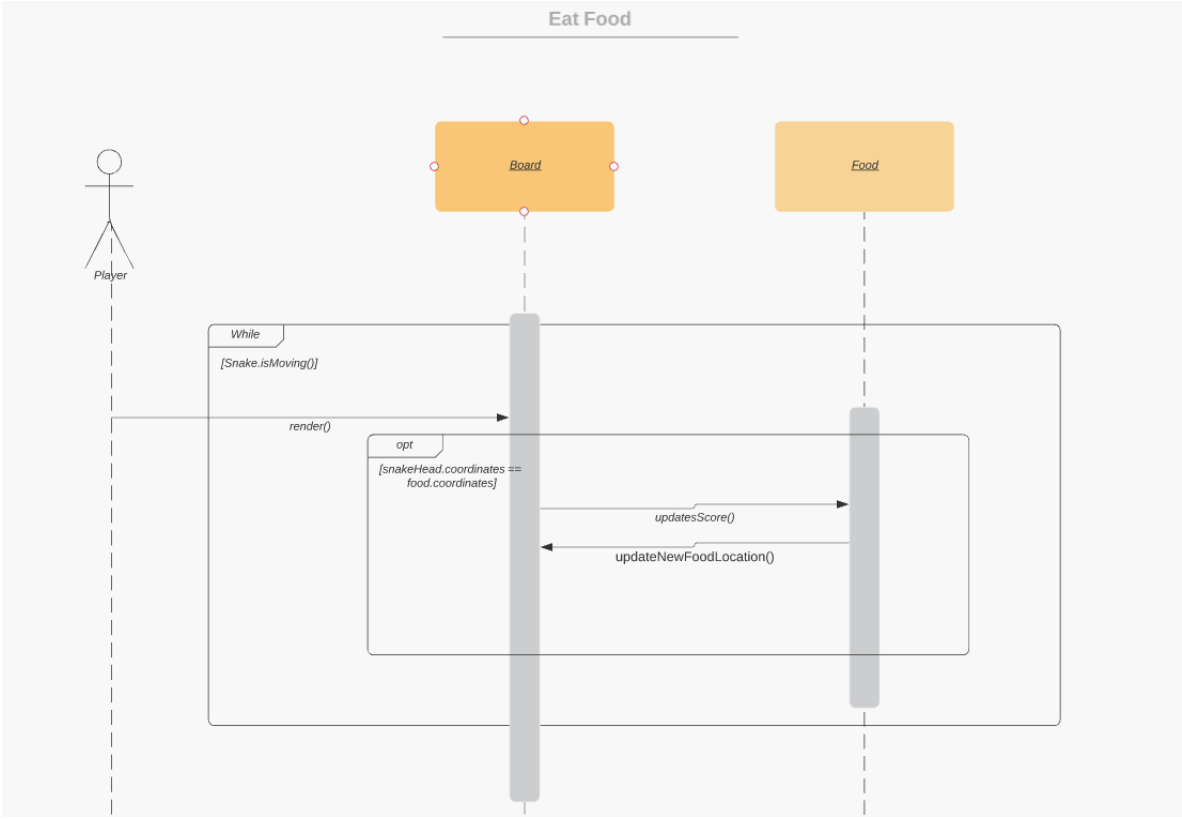
Task 2:

Turn snake sequence diagram:



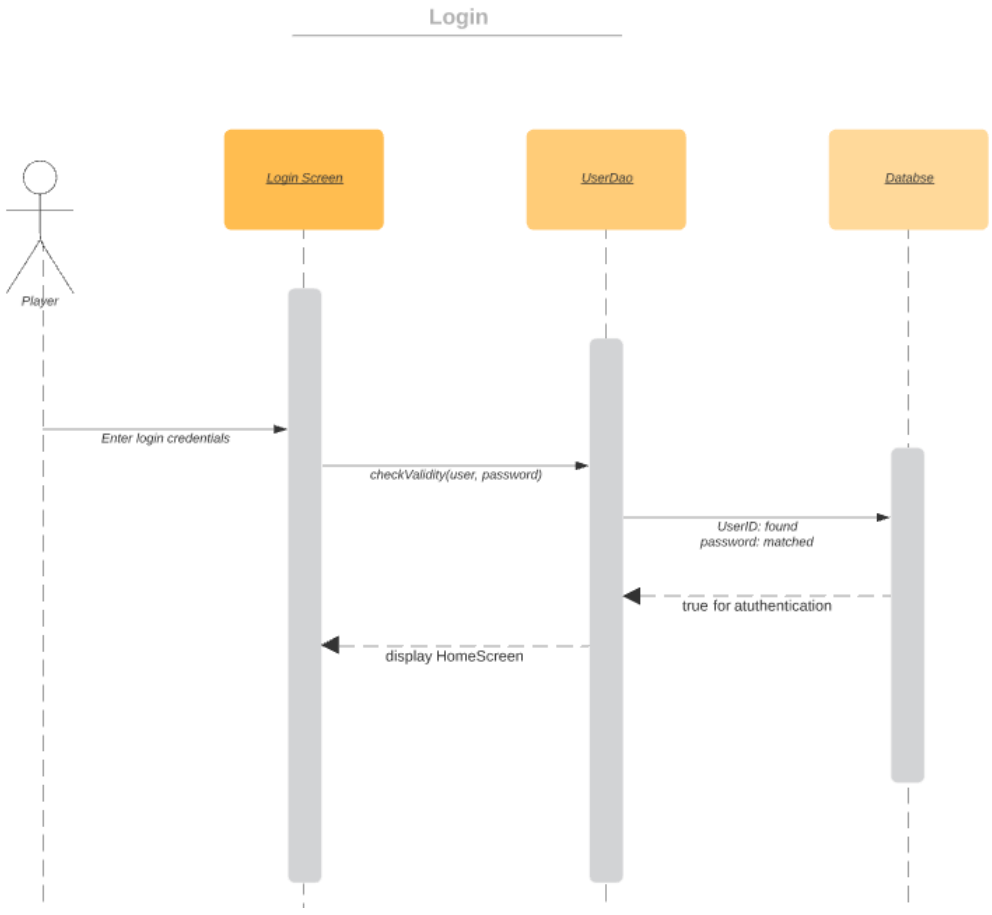
While the game is in play, meaning the snake has not collided itself or the walls, every frame we update the board with the new positions of objects in it. These methods are all void so they do not return anything.

Eat food sequence diagram:



Here the diagram is also active while the snake is alive. We render the board every frame and update the location of the food only if the snakes head coordinates are the same as the food coordinates that are currently on the board.

Login sequence diagram:



When logging in the user enters his credentials in the login screen, which are then forwarded to the data access object class of user to check for validity. The method connects to the database and checks if a user with such ID exists and compares the passwords. If they match a request to change screen to Home Screen is made.