For our 2D game, we made a game similar to agar.io in which players control a circle and consume smaller circles to become larger.

Rules:

1. The controlled circle can increase in size by coming into contact with smaller circles and absorbing them.
2. Smaller and larger circles move about the screen independently of the controlled circle.
3. If a larger circle comes into contact with the controlled circle, the game starts over with the controlled circle at a small size that will remain the same each time.
4. All circles move at the same maximum speed.

The four controls are the W, A, S, and D keys to move the player circle.

The way by which the player understands the game state is the size of the player circle relative to the other circles on the screen.

The GUI interface lists how many smaller circles the player has consumed for that round, and how many total rounds the player has gone through (i.e. how many times they have been consumed).

The player wins when they have consumed all 50 circles that spawn at the beginning of the round. When this happens, a victory screen will display with how large the player circle is. The lose condition is when the player circle is consumed by a larger one. A lose screen then displays, also showing how large the player circle was at the time of loss.

The classes include the player’s circle and the randomly generated enemy circles. Player is a subclass of Circle since it includes all the same functionality, plus the controls WASD for direction. Functions of Circle allow the object to consume smaller circles and to be destroyed by larger circles upon contact, and to move about randomly.

Jack implemented the motion and consumption functionality in the beginning. Braxton then wrote win/lose conditions. Jade created the GUI and worked with Braxton on finishing the win/lose screens. Braxton then implemented the restart game and pause screen functionalities.