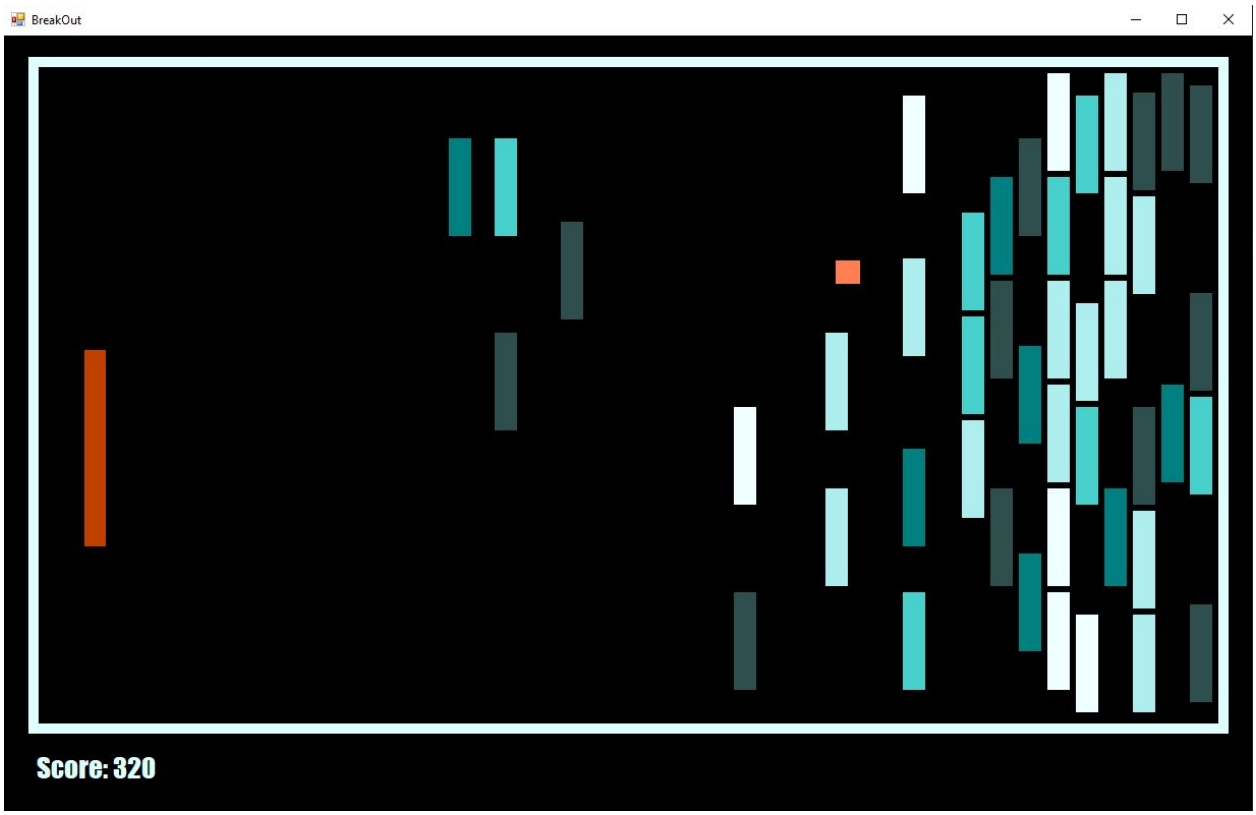
Jennifer Halder
Noelle Lansford
Dr. Delano
ITM 2300-01
28 Jan 2020

C# Topic Deliverable:

Overview:

Breakout is an existing computer game that we have recreated. In this game you hit a ball with a paddle in order to hit and clear bricks. Like pinball or pong, if you miss the return of the ball with your paddle you lose the game. You gain points by hitting the bricks and the objective is to clear all of the bricks. Our game will have one level, as well as a score and life tally at the bottom.

Our Game:



Screenshot Changes:

- We changed some of the colors, thickened the walls, and removed the lives section since you now only get one life.

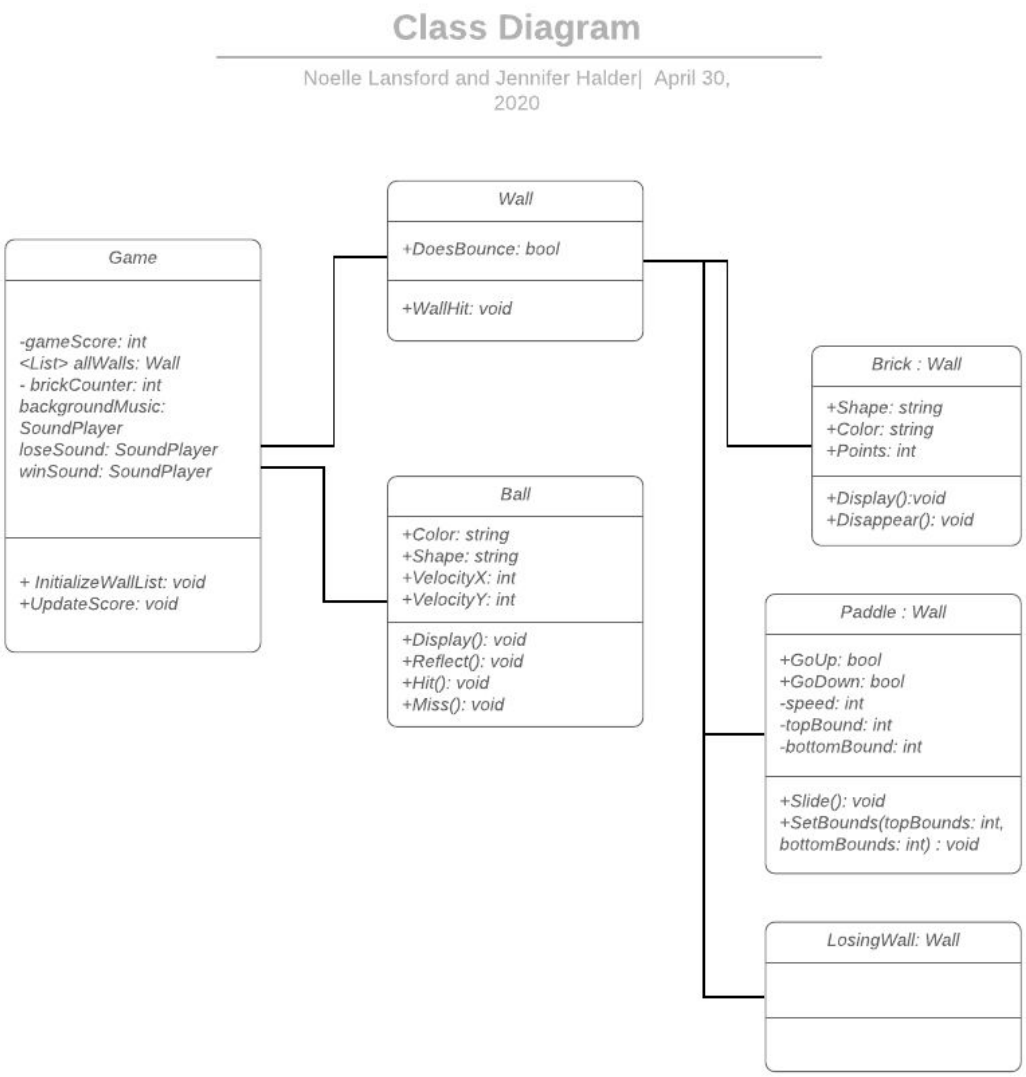
Rules:

- You must change the direction of the ball by hitting it against the paddle and/or wall.
- You can only knock out bricks by hitting them with the ball.
- You gain points and clear bricks by hitting the bricks with the ball.
- Different colored bricks have different values of points. You can track this score in the left bottom corner and try to beat your high score.
- You lose when the player's paddle misses the ball's rebound.
- You win the game by clearing all of the bricks within 3 turns.

Classes:

1. Wall - *Noelle/Jennifer*
2. *LosingWall - Noelle/Jennifer*
3. Brick - *Noelle/Jennifer*
4. Paddle - *Noelle/Jennifer*
5. Ball - *Noelle/Jennifer*
6. Game - *Jennifer*

Classes Diagram:



Reviewer Comments:
We decided to omit the lives section and only allow the player to have one life. The class diagram has changed as we have used inheritance and polymorphism to create our game. Everything besides the Game and Ball class are a wall (AKA a thing that should be bounced off of). Then, each more specific kind of wall has its own class as well and inherits from Wall.

Advanced Topic:

- Polymorphism - *Chapter 12*

Sources:

"Breakout." *Play the Breakout Game*, The Simple Arcade, thesimplearcade.com/play/breakout.html, last accessed 28 January 2020.

"C# Polymorphism with Examples." *Tutlane*, www.tutlane.com/tutorial/csharp/csharp-polymorphism.