Team 2 Executive Report: Atari Breakout

**Introduction**

For the final project of our Introduction to Computers and Programming class, our professor assigned a final project to our entire class. The goal of the project was to create our own modified version of Atari Breakout. Each student in the class was assigned to a group of three members. Each group was assigned to a client that they needed to work with, as well, introducing us to the realities of the software industry.

The goal of our version of Atari Breakout is to reach the top of the screen with a ball as quickly as possible, minimizing the number of bricks destroyed in the process. The player has the ability to control a paddle -- the paddle is used to direct the ball towards the bricks. The ball will bounce off of the paddle, the bricks, and the sides of the screen, but not the top and bottom of the screen. The player will lose of a life if they allow the ball to hit the bottom of the screen -- however, if they cause the ball to hit the top of the screen, they progress to the next level. With every level, comes an extra row of bricks. The player will be allowed to keep playing until they run out of lives.

**Team**

Each team member had their own role to play in the project. Ferris Esposito primarily worked on the main game, while Valeria De Lira and Imani Breckenridge-Peete mostly worked on designing classes. Valeria also reformatted many of the images used in our game. Ferris’ major strength was his ability to help us overcome our struggles, lead the team, and employ object oriented design principles -- however, he could have made better use of his time during meeting. Valeria’s major strength was her ability to design classes and modify images -- however, she could have contributed more to the main game. Imani’s major strength was her ability to design classes, as well, and her ability to come up with good ideas, but she also could have made a larger contribution to the main game. Each member, however, did their fair share of work (much of Valeria’s and Imani’s contributions were through the discussion of how to approach the implementation of most of the game’s functionalities).

**Features Implemented**

There are many features that we incorporated into our game. We will briefly discuss each of them in this section. At the start of each game, a row of bricks is generated and displayed onto the screen. A paddle is also displayed at the bottom of the screen -- the paddle can also be moved using the left and right arrow keys. At the start of every game, a ball randomly appears in between the bricks and the paddle with a random initial direction. Whenever the ball hits the walls, paddle, or bricks, the ball bounces off of them. The angle the ball bounces off of the paddle depends on where the ball hits it. When a brick is hit by the ball, it now long appears on the screen. When the ceiling is hit by the ball, the player is taken to a new screen that indicates that they have completed the level. When the ball hits the bottom of the screen, an image indicated the game has ended is displayed onto the screen.

After a level is completed, the game waits for the player to decide to continue playing. Each consecutive level generates an extra row of bricks. If the ball is reset, the game doesn’t start until the player decides to start. The player has a score that is determined by how fast they complete the level, and how many bricks remained after they completed the level. The more bricks that remain, and the less time (in seconds) elapsed after every level yield in a higher score. After every game, the player is asked to enter their initials. If they obtained a high score, their score is written to a file that contains the top ten scores. In the main menu, the player has the option to start the game, view the high scores, view the instructions, and exit the game. The score and number of lives of the player are displayed onto the screen as they play. Also, the time elapsed during each level is also displayed onto the screen. The ball and paddle reset, and the player loses a life whenever the ball hits the bottom of the screen.

* Row of bricks display at the start of each game
* Paddle is displayed at beginning of every game
* Ball is randomly placed at beginning of every game
* Ball bounces off of bricks, paddle, and walls
* Indication of level completed or level failed when the ball hits the top or bottom of the screen
* Change of angle the ball bounces off of the paddle depending on where it hit it
* Brick is removed when it is hit by the ball
* Player can control the paddle with the left and right arrow keys
* Level progression
* Game scoring
* High score functionality
* Main menu that allows player to start the game, view instructions, view highscores, and exit the game
* Detects when the player should lose a life, and when the game should end
* Lives, score, and timer are displayed during the game
* The game records the player’s initials and score (if it is a high score)

**Beyond the Standard Features**

We added three features beyond the standard features of Atari Breakout. Firstly, we added functionality that allows the player to choose the difficulty they will play at. The difficulty controls the speed of the ball, and the number of lives the player has. A screen that prompts the user to click an image that corresponds with their desired difficulty is displayed before the start of every game. The higher the difficulty, the faster the ball and the less lives the player has. The three difficulties are easy, medium, and hard.

The second feature that we added was sound. At the beginning of every game, three sound files are loaded. Whenever the ball hits a wall, the paddle, or a brick it makes a sound that resembles a smacking noise. Furthermore, whenever a player fails to prevent the ball from reaching the bottom of the screen, a sad song is briefly played. In addition, whenever the player completes a level, a song of triumph is played.

The last feature that we added allows the player to choose from an assortment of bricks. The brick that they choose is the brick that will appear on the screen during the game. Each brick is displayed onto a screen that prompts the user to choose one before the start of each game. The game does not continue until the user clicks on the brick image that they want. The appearance of the brick will correspond to the image that they have chosen.

**Software Development Process**

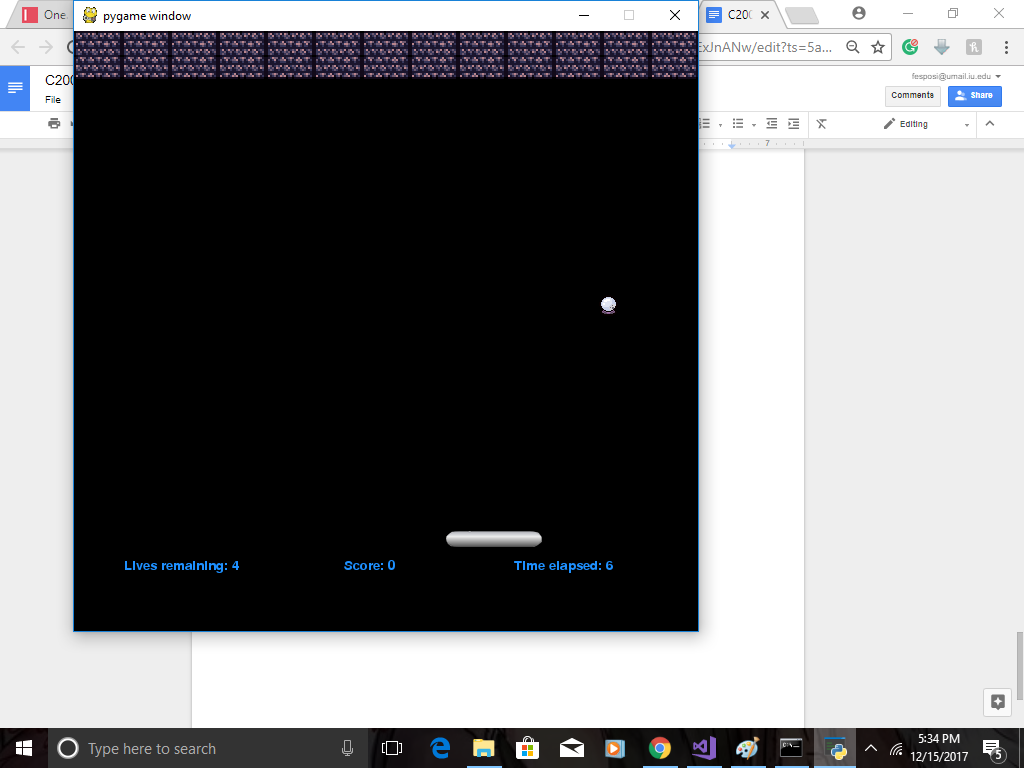
Having no experience with object oriented programming, initially, our group was unable to begin the development of the game for some time. In spite of our struggles, however, we were able to produce a complete game. Computer programmers rarely improve without difficulties -- we have grown a lot from this experience, not only as computer programmers, but also as people. Going into this project, we did not realize how import and powerful object oriented design really is. This project would have taken so much longer, and our game would have been far less efficient without it. The skills we have acquired from this experience will follow us throughout the rest of our careers. The most challenging aspect of working as a team was splitting up the tasks. Some tasks were difficult to complete without code that was assigned to another group member -- as a result, we often found ourselves working at separate times opposed to the same time. With better collaboration and planning, the software development process would have gone more smoothly.

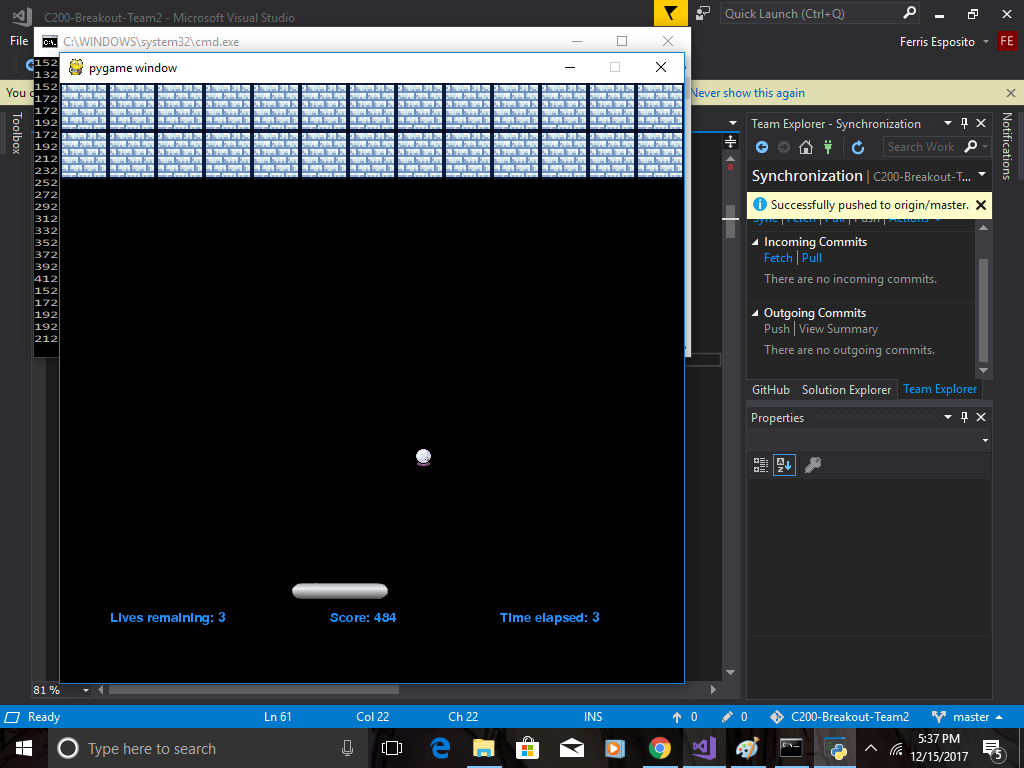
**Future Work**

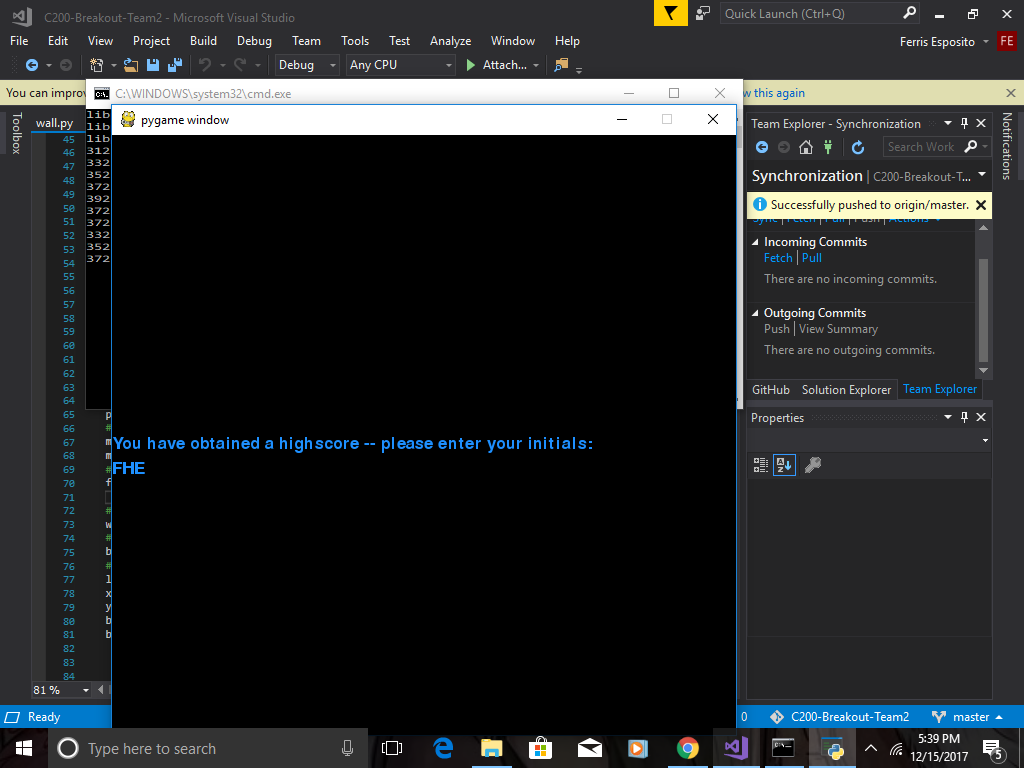
If we could add extra features in the future, we would add bricks with special features and power-ups. We would design new types of brick classes that are subclasses of the main brick class. They would have properties that make them harder to destroy, change the properties of the ball (such as speed), increase the life of the player, etc. Furthermore, we would make it so power ups randomly appear on the screen. When the ball comes into contact with a power up, its properties would change (for example, its size or speed).

**Images**

****

****

****

****

**Code Listing**

* main.py
* ball.py
* brick.py
* continuee.py
* exit.py
* GoBack.py
* highscore.py
* instructions.py
* level.py
* paddle.py
* player.py
* resume.py
* score.py
* start.py
* text.py
* timer.py
* wall.py