

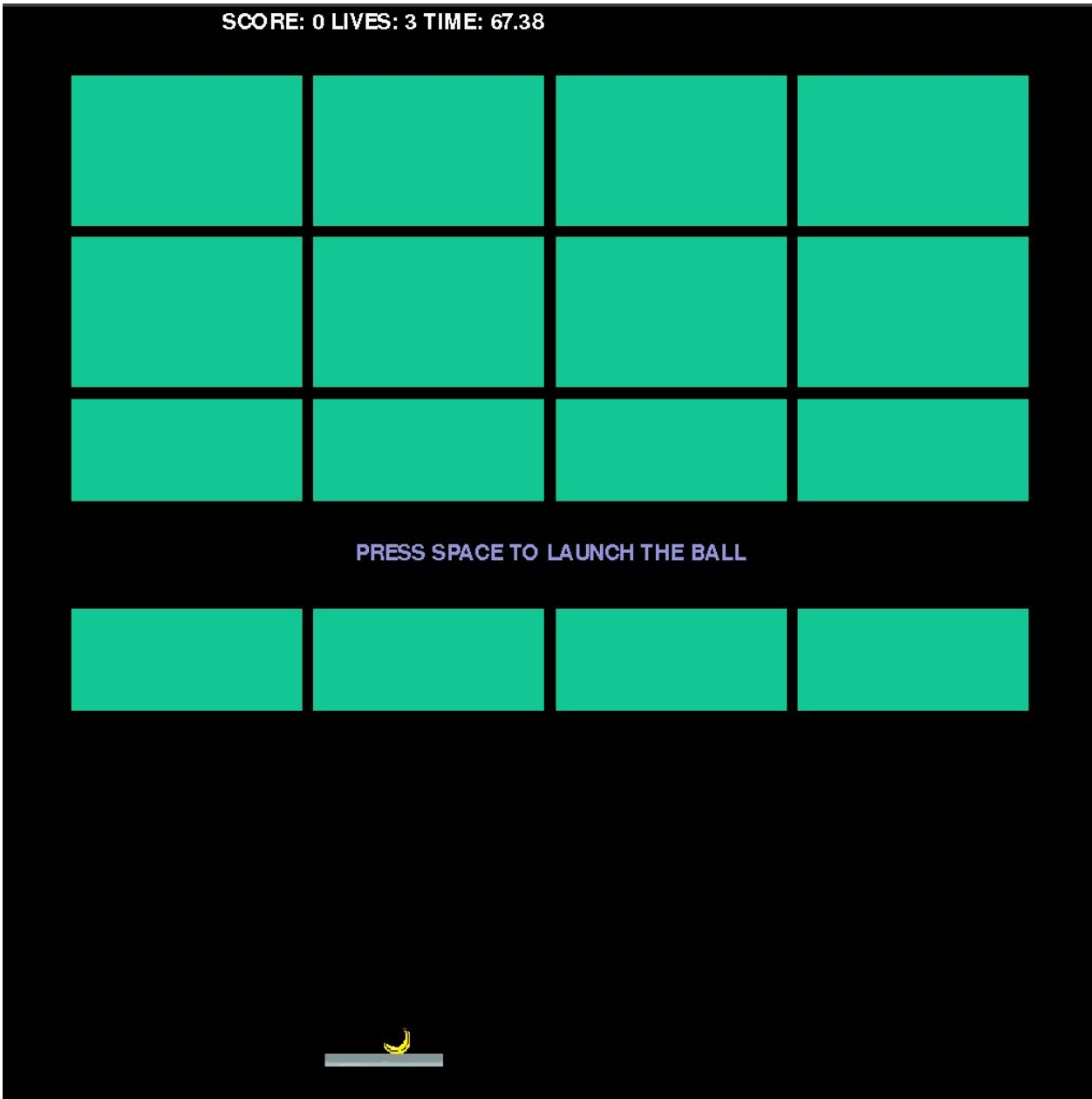
Abstract

This project was inspired by “Breakout” and the old classic, “Brick breaker” to create “Banana Breaker.” We built a brick breaker game that has several levels, power ups, sounds, sprites, and a story. Banana Breaker is 10 levels of brick breaking, with memes and magic interlaced.

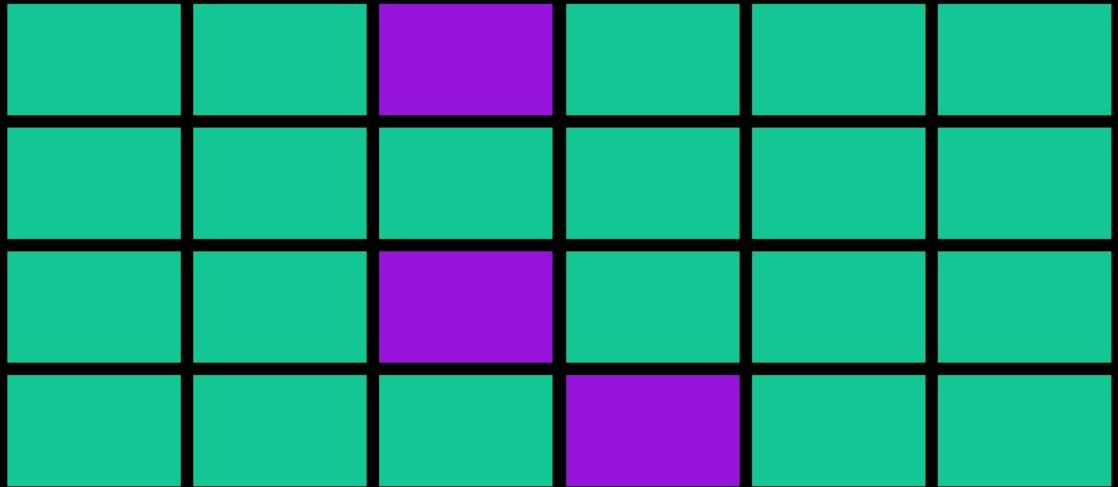
Results

Our final result was indeed a breakout/brick breaker game. We added on several features to improve the brick breaker game. The first was the ability to parameterize the amount of bricks in the game. We were able to set how many bricks we were able to create, and it would divide by the width of the screen so that the bricks would always be able to fit. Each level would increase the amount of bricks, until a max of 12 x 12. We fixed collisions with the walls and paddle so that it would bounce in an appropriate fashion. The ball bounces at the angle it hits the paddle now instead of directly 90

degrees.

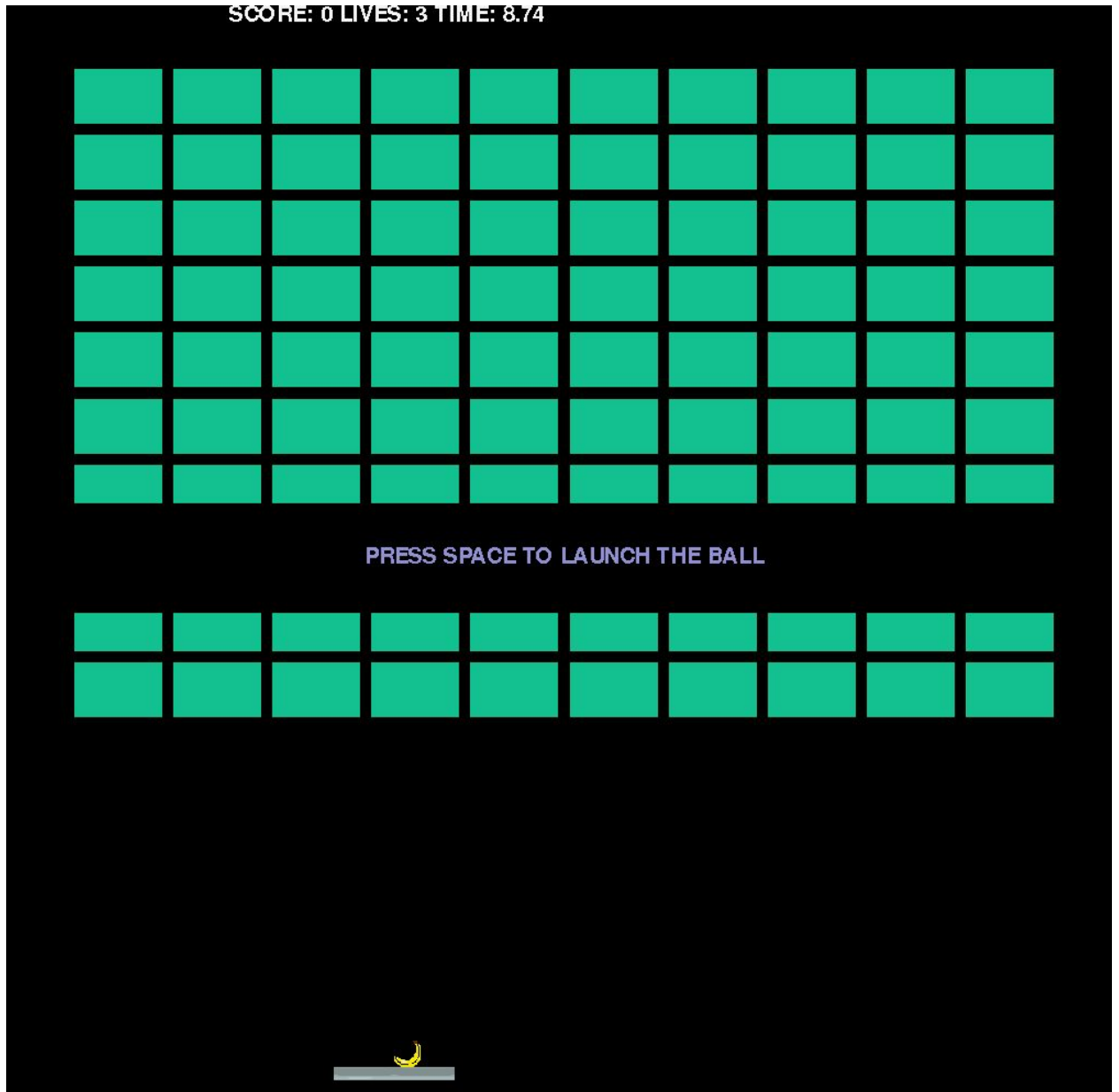


SCORE: 0 LIVES: 3 TIME: 6.28



PRESS SPACE TO LAUNCH THE BALL





Our next big step was to add power-ups, sprites and sound to give the game a more authentic and aesthetic feel. The ball became a banana, because why not? When given certain power-ups, it became a flaming banana, because once again...why not? Purple bricks were two-hit bricks that we added as more challenge. In regards to sound, whenever the ball hits a paddle or the

brick, it makes a *bonk* noise. Also as small Easter egg, we changed the text on the window for each level.

Implementation

For our implementation, we started off with a breakout clone that we found online. We used that as a base for our code and began to improve on it. Soon, we realized that the code was not formatted in a great way, and began to move constants into its own page, had code for sprite sheet extraction, a main code page where we initially improved on the brick breaker game, and a levels page which from we ran the program. To test sprites and sound and other funky things we wanted to add, we used another simple game from online and added to that, which we called testOne.

The constants file is just a bunch of global constants which we would import into all our code. The spritesheet_functions was also just imported into the Brickbreaker for us to use the sprites. The function from which we ran the game from, levels, was just different classes which we imported Brickbreaker and changed values.

```

144 class Level10(Bricka):
145     def __init__(self):
146         Bricka.__init__(self)
147     def init_game(self):
148         Bricka.init_game(self)
149         self.start_message = "BONUS ROUND:  NO HELP, NO HOPE, NO PROBLEM"
150     def create_bricks(self):
151         self.power_chance = .0
152         self.double_chance = 1
153         self.xbrickNum = 12
154         self.ybrickNum = 12
155         Bricka.create_bricks(self)
156
157 #class Level3(Game):
158 #    print('blah')
159
160 if __name__ == "__main__":
161     Level1().run()
162     Level2().run()
163     Level3().run()
164     Level4().run()
165     Level5().run()
166     Level6().run()
167     Level7().run()
168     Level8().run()
169     Level9().run()
170     Level10().run()

```

Reflection

From a process point of view, what went well? What could you improve? Other possible reflection topics: Was your project appropriately scoped? Did you have a good plan for unit testing? How will you use what you learned going forward? What do you wish you knew before you started that would have helped you succeed?

Also discuss your team process in your reflection. How did you plan to divide the work (e.g. split by class, always pair program together, etc.) and how did it actually happen? Were there any issues that arose while working together, and how did you address them? What would you do differently next time?

We started off the project with a major brainstorming session. We came up with a ridiculous amount of ideas and wanted to implement all of them, but we did make sure to put on the breaks and agree that we would start simple, and then work our way up.

