Suika Write-up:

"You may be able to make-up some lost points by writing a 1-2 page writeup detailing what issues you wish you could've solved with your faulty solution."

I'd like to preface that I did not choose to implement my own physics solution (and wildly fail) because of my programming hubris- I have little faith in my skills as it is. It's mostly because I thought that if I could nail the physics system by using Box2D, the project would require less overall state coding than Pac-Man and thus be faster and easier.

I, and seemingly everyone else I know who attempted this project, was unable to get Box2D to work, and instead used a Verlet physics implementation, which is French for "doing the whole physics system yourself," apparently.

Existing issue number 1: The "hanging" collision. Rather than keeping a ball enclosed inside the square, balls hang off the bottom. Currently, the program correctly creates a ball, checks the closest point to the closest edge of the rectangle, and when it overlaps with that point (when the distance is less than the radius), it pushes back with an opposite force so that the ball stays in place. This technically works, but on the wrong side of the wall. It will only let the ball go past the wall for as long as the ball is, meaning that larger balls stick out farther than small balls do. I can't understand this issue- it must be simple and small, but as much as I flip around greater than/less than signs and negate distances and radii, nothing seems to fix it. Even though this technically adds a false bottom to my suica game and would usually mean that it becomes impossible to win/beat, I was able to add a win condition to create a full experience. Also, the random explosions (mentioned later) help to make it very quite possible to lose.

My suspicion is that I used the "ball-to-rectangle" collision theory, which checks if a ball from the outside is coming into a rectangle. But after shoddily implementing it, I realized that I wanted the exact opposite- a ball on the inside of a rectangle going outside. My solution was take

the main if statement that tracks if the ball collided on the outside, and just stick all my ball force movement code into the “else” clause. It was hacky, and yet even after re-writing the system to track the collision from the inside of the rectangle, nothing was effective as that solution.

Some other issues arise out of this main issue. For one, the corners of the box behave as if though they are rounded, and so balls roll off from the side. This can lead to balls hitting each other at an angle, rolling into a wall, and like Sonic, spin-dashing up and into the game over zone. My hypothesis is that because it’s hanging off the edge, it must be registering the simultaneous input of a horizontal wall and a vertical wall incorrectly.

Another issue is the sudden explosions in force that will happen to balls when they are near other balls when they combine. If two grapes turn into a tangerine, if there was a cherry between those two, the increased size of the orange will rocket the cherry up into the blast zone. The problem seems obvious: I need to check for objects and push them out of the way before instantiating the new fruit, but I unfortunately didn’t have time for that.

I also didn’t get time to add my home-made art assets, but that’s not really a coding problem as it is a timing problem. Thanks for listening.