Aidan Wadin - Comp 521 Assignment 2

RUNNING THE ASSIGNMENT:
My assignment was done in html/css/javascript.
To run, extract the folder, and open "index.html" in your browser and you should be good to go.

CONTROLS:
Space bar is fire, left/down arrows moves the cannon angle down, while right/up moves it up.
If the cannon is about to tilt out of the game window, you will be alerted, and the cannon will be moved back into the game area.

NOTES:
1. This was a difficult assignment. I made a main array of points and an array of "sticks". The entire physics system acts on ONLY the points (as verlet integration is a point based system). The sticks are simply constraints to hold points together. All physics are done using verlet calculations.
2. In terms of libraries or outsourced material, the perlin noise generators is a combination of information found on the internet. But ultimately, this entire project was made from scratch by yours truly :)
3. The wind is randomly changed approximately every two seconds. It only affects cannonballs.
4. Please note, sometimes ghosts do not react to cannonball collisions, this is due to the slow frame rate of my update function, and an issue with the javascript overall. Please go easy on me. Javascript is not exactly optimized for things of this manner (when used on a html canvas like this) so please be gentle, I only learned about its lack of optimization after getting 3/4 of the way through.
5. Stonehedge is solid, and balls bounce off it as desired.
6. The cloud will simply wrap around to the other side of the screen if it reaches a boundary.
7. My code is structured in the following way:

First is the setup for the javascript canvas

Then I set up most of the variables I will be using (some variables are declared later, right before the function they are used for readability sake).

Then I have all of my main constructors (cannon, ghosts etc).

Next comes my perlin generator functions and supporting functions for perlin.

After that is all of my supporting functions, to be used to help draw objects mainly or manage the wind.

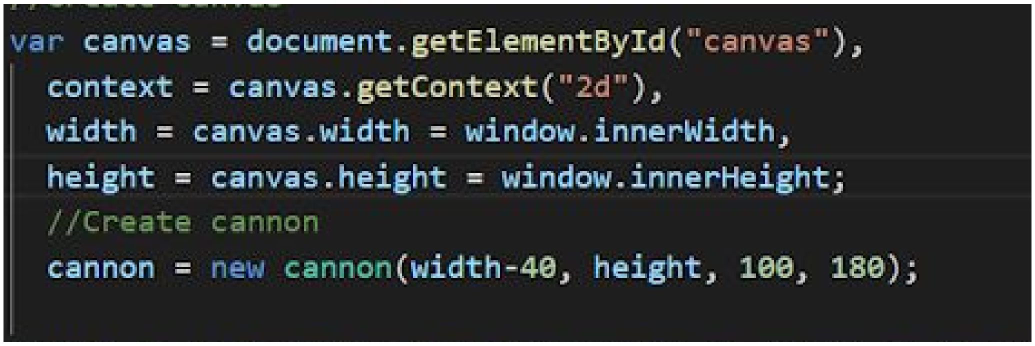
Finally comes my support point and stick updating functions. These functions are made to keep the points acting the way they are supposed to. They both manage the physics of the game at every frame and make sure everything is displayed correctly.

Then comes the actual updating and rendering methods, so that each frame, the points and sticks are updated in both their positions and where they are being displayed.

Finally I call update(), to start the game, and generate the starting 4 ghosts.

8. Please note, when the ghosts are hit with a cannon ball, they take part of its velocity (taking the full velocity of the cannonball causes them to glitch out). If they go off the left hand side of

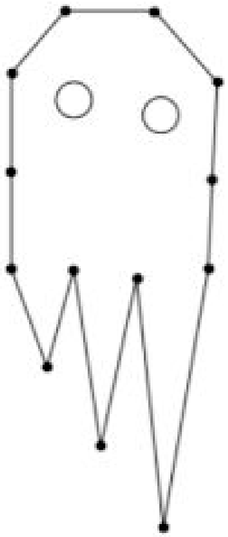
the screen, they will come back, only when they cross the right hand side of the screen do they despawn and generate a new ghost.
9. If things are placed oddly on your screen when you run it, it may be due to the fact that the canvas fits into the window of your browser, so it might change based on that. If, for some reason, things are working very oddly, change:



```
var canvas = document.getElementById("canvas"),
  context = canvas.getContext("2d"),
  width = canvas.width = window.innerWidth,
  height = canvas.height = window.innerHeight;
  //Create cannon
  cannon = new cannon(width-40, height, 100, 180);
```

Width = canvas.width = 1500;
Height = canvas.height = 700;

I worked very hard on this, as I wanted to see if it could be done in javascript as opposed to Unity.



Ghost Description:

I have each ghost consisting of 13 points, where its eyes are points, but slightly larger. They are generated randomly on the left side of the stones. They rise up to about height/2 of the canvas, and then float to the right side of the screen (unless hit with a cannon). If they reach the right side, they despawn and a new ghost is created. If they are hit, the point closest to the collision takes on the new velocity of the cannonball and drags the rest of the points with it backwards. The ghost retains its shape by having invisible support beams (sticks) within the ghost, so it remains rigid, but also has quite a bit of floppiness to it. These sticks will have the property of “hidden”.

Thank you!!