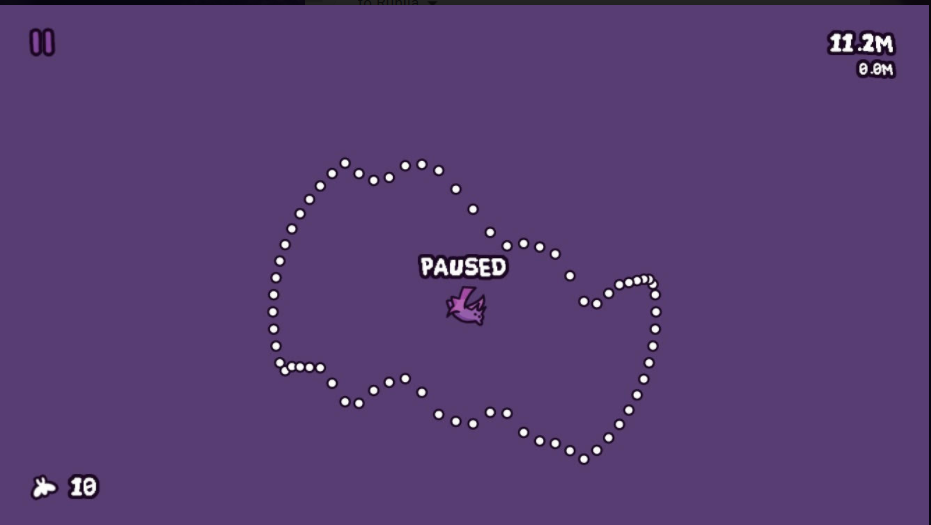
**Project Description**:

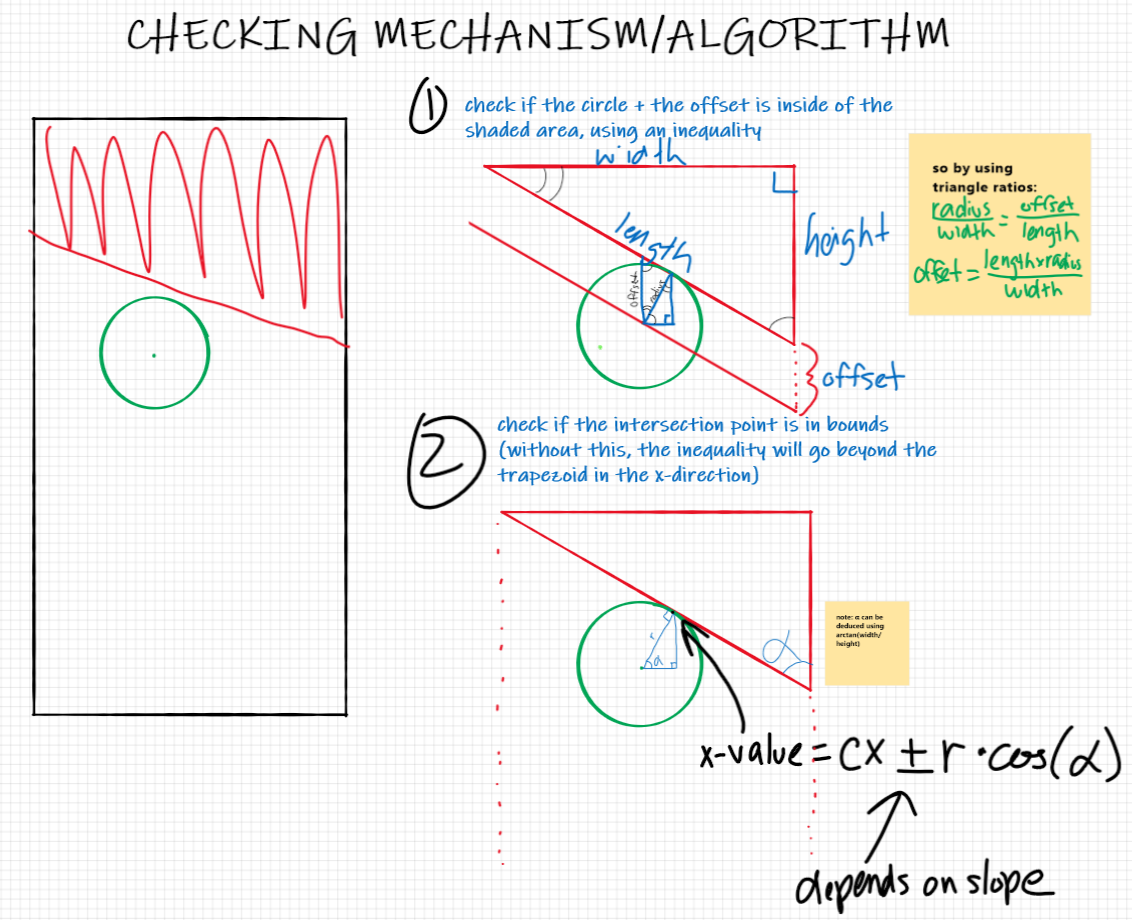
* Name: Batty Cave
* Description: Like Flappy Bird, but with a sidescrolling terrain instead of pipes. IN addition, the player cannot actually see the terrain. Instead, there is a circle around the player that reveals the terrain within a certain radius.
* Based off of a real game that can be played on a phone (also called Batty Cave)



**Structural Plan**:

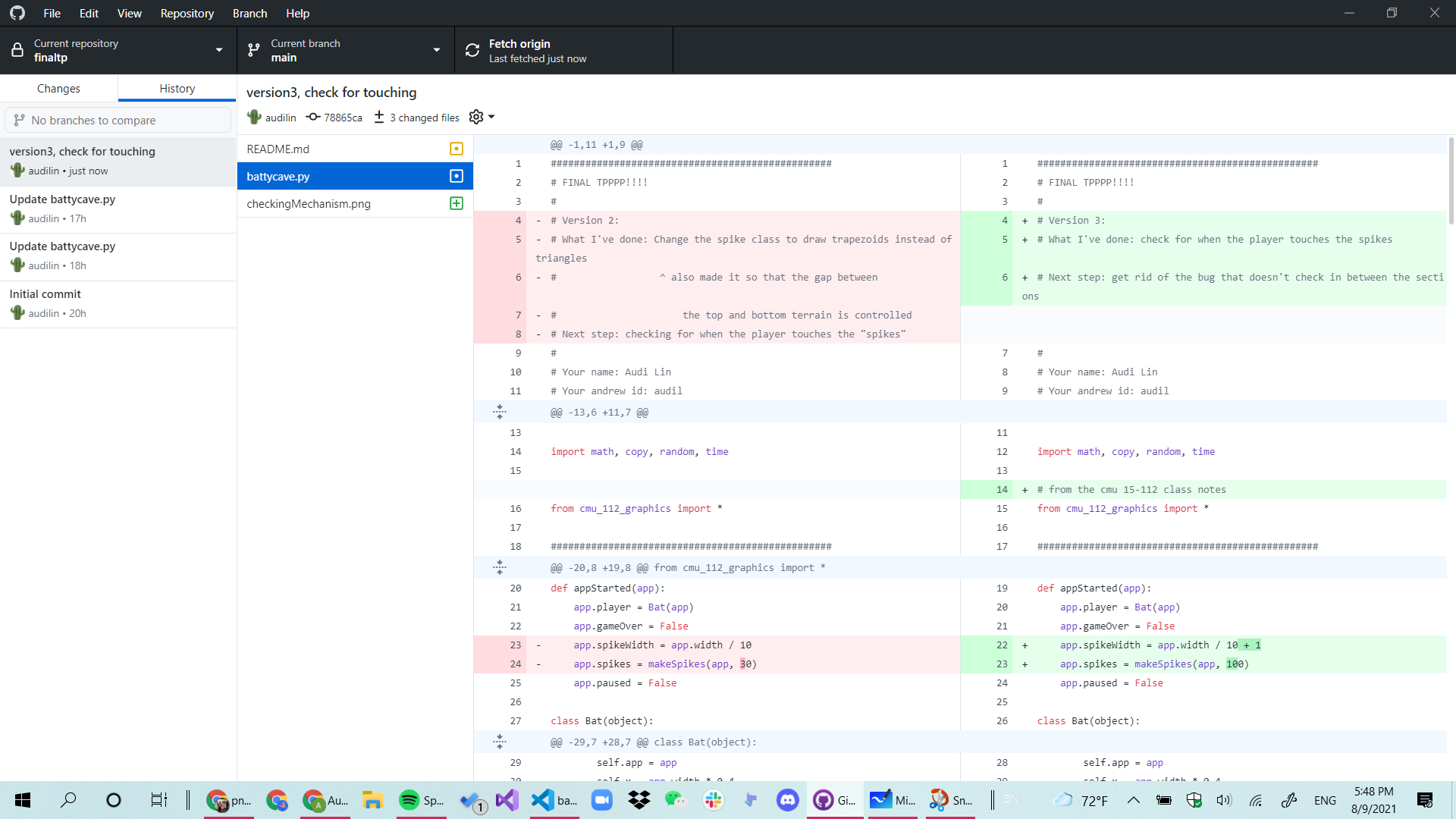
* Separate drawing functions - terrain, circle, and player
* Classes:
  + Player class with attributes: y-position, and things required for “jumping”
  + Spike class with attributes: height, width, etc.
  + Note: both of the classes have their own draw functions
* Key-presses:
  + debugging - pause, up, down
  + normal movement - SPACE for going up
* Functions:
  + checking - checking for game over and if “jumping”
  + terrain generation - terrain stored in list of Spike objects
* timerFired:
  + the player location is updated:
    - if they are jumping, go up
    - otherwise, go down
  + The spikes sidescroll

**Algorithmic Plan**:

* The most complex elements of this project will be:
  + Circle around the player that “fits” to the terrain
    - The circle is composed of individual dots
    - I will have to use algebra to calculate the location of each individual dot and where it potentially intersects with the spikes
  + Randomly generated terrain
    - I want it to be smooth, so I may have to make connected trapezoids instead of triangles(which I will start out with for simplicity)
    - Instead of generating the top and bottom separately, I will randomly generate the bottom and then randomly make generate the height of the gap
    - This will make it harder to check if the player touches the terrain
    - accurately checking if the player touches the terrain

**Version Control Plan**:

* I have chosen to use Github and also make safety uploads to Autolab
* In addition, I have plan on detailing the changes made in the header for clarity



**TP2 Update:**

Changes made:

* instead of doing individual dots, I will do a generic circle that will be covered by the terrain, as shown below:

