Below shows the development phase, where the purple marker was Justins writing, and the red was mine. I originally pitched the idea of a rhythm game to Justin because I remember hearing him say that he'd want to avoid a 2-d side scroller as much as possible. So, I came up with a solution of pitching a game that would be, for the most part, stationary: a rhythm game, one similar to guitar hero like the one on the DS, though I know Piano tiles was more renown. Justin separated all the parts into classes and how it could be done, while I added the occasional input since he was unfamiliar with rhythm games for the most part.

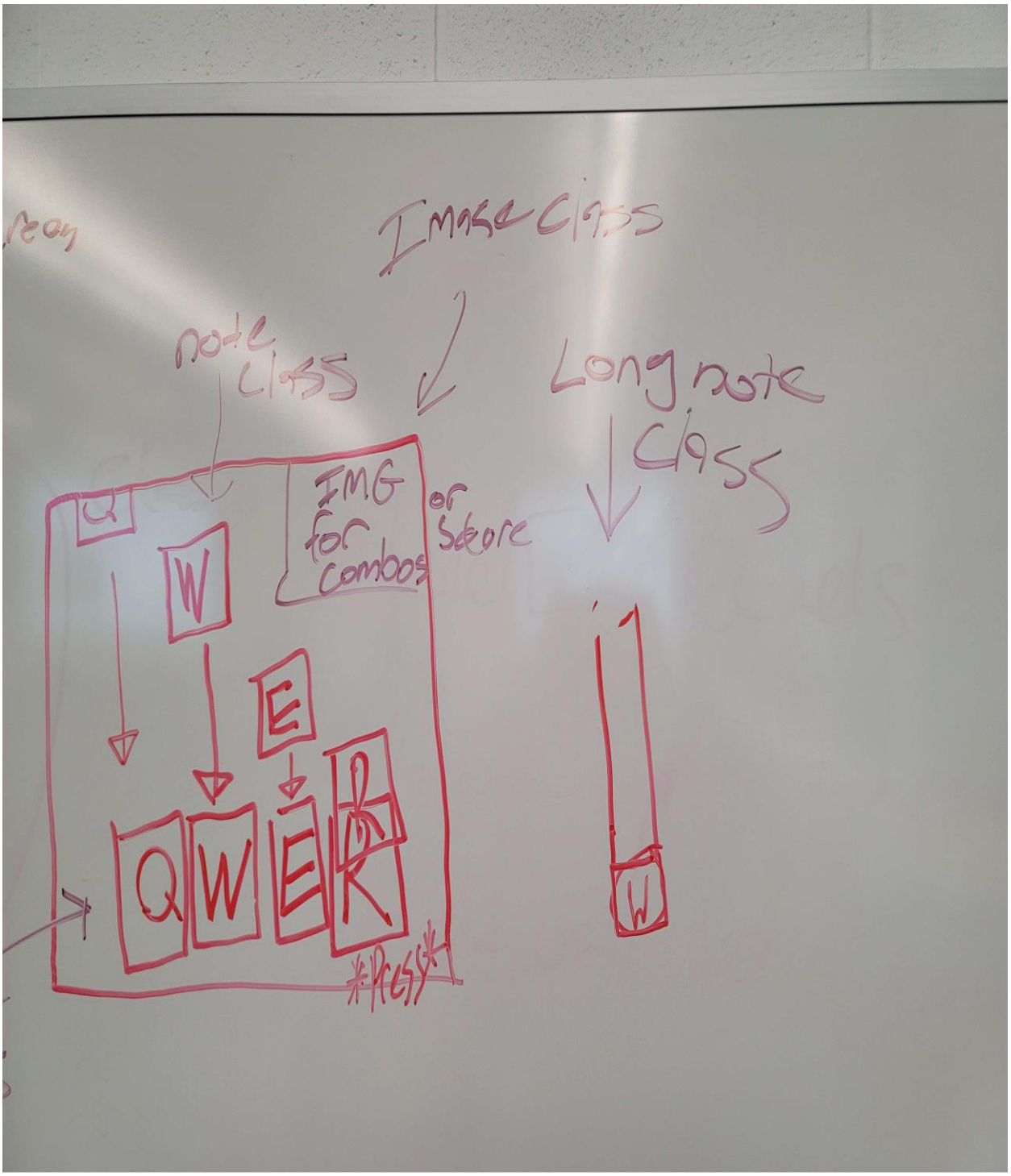


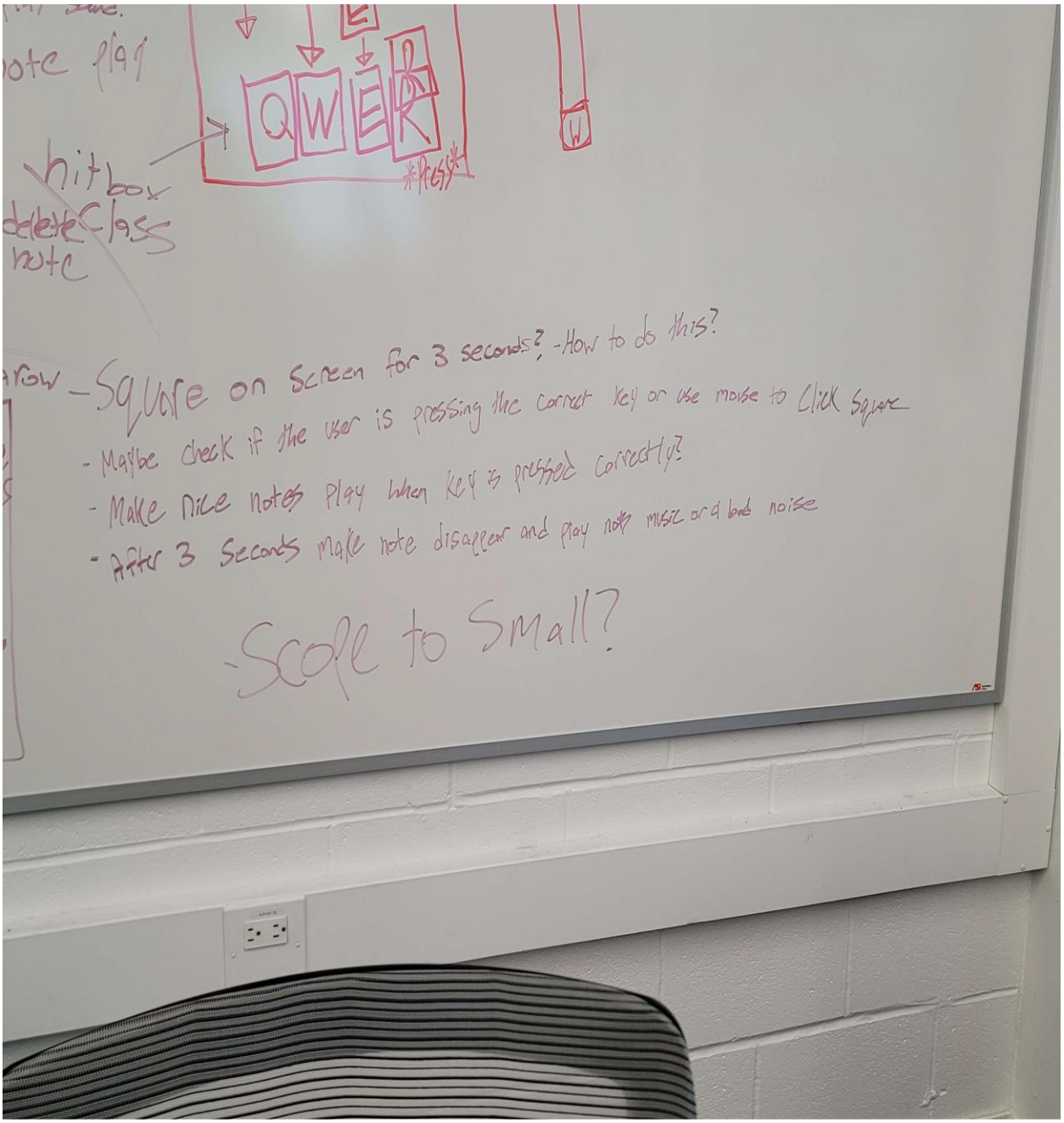
Image class
note class
Long note class
IMG for combos
or Score
Q
W
E
R
Q W E R
*Press*
W

Rhythm GAME #2

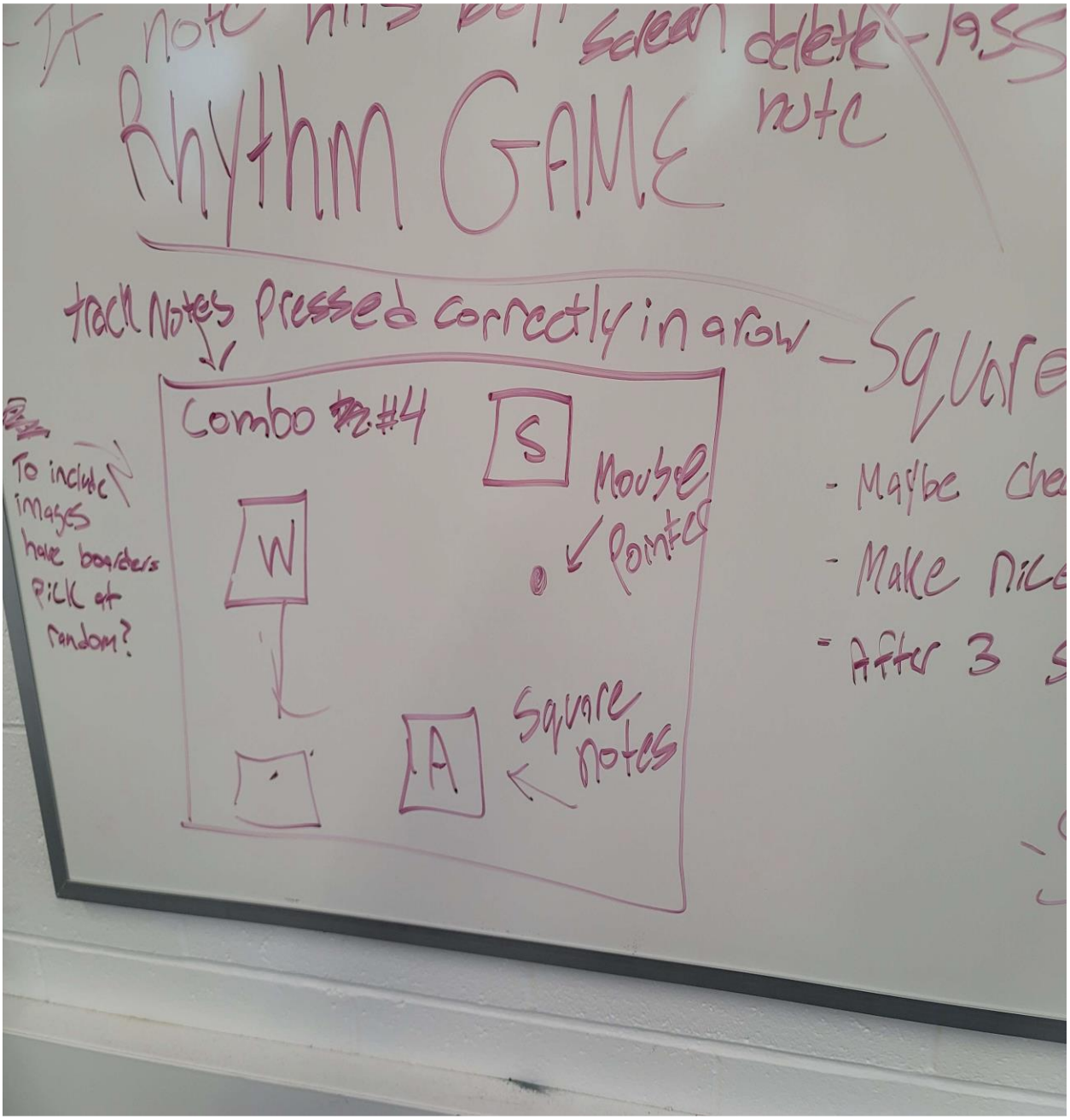
- Squares drop from the top of the screen
- Each key has text of "Q, W, E, R"
- At bottom of squares hitboxes/bottoms with text in boxes "Q, W, E, R"
- When the dropping square hits the hitbox at the bottom play at note.
- If player hits note correctly play sound.
- If player doesn't hit note play bad sound.
- If note hits bottom of screen delete class note


Rhythm GAME

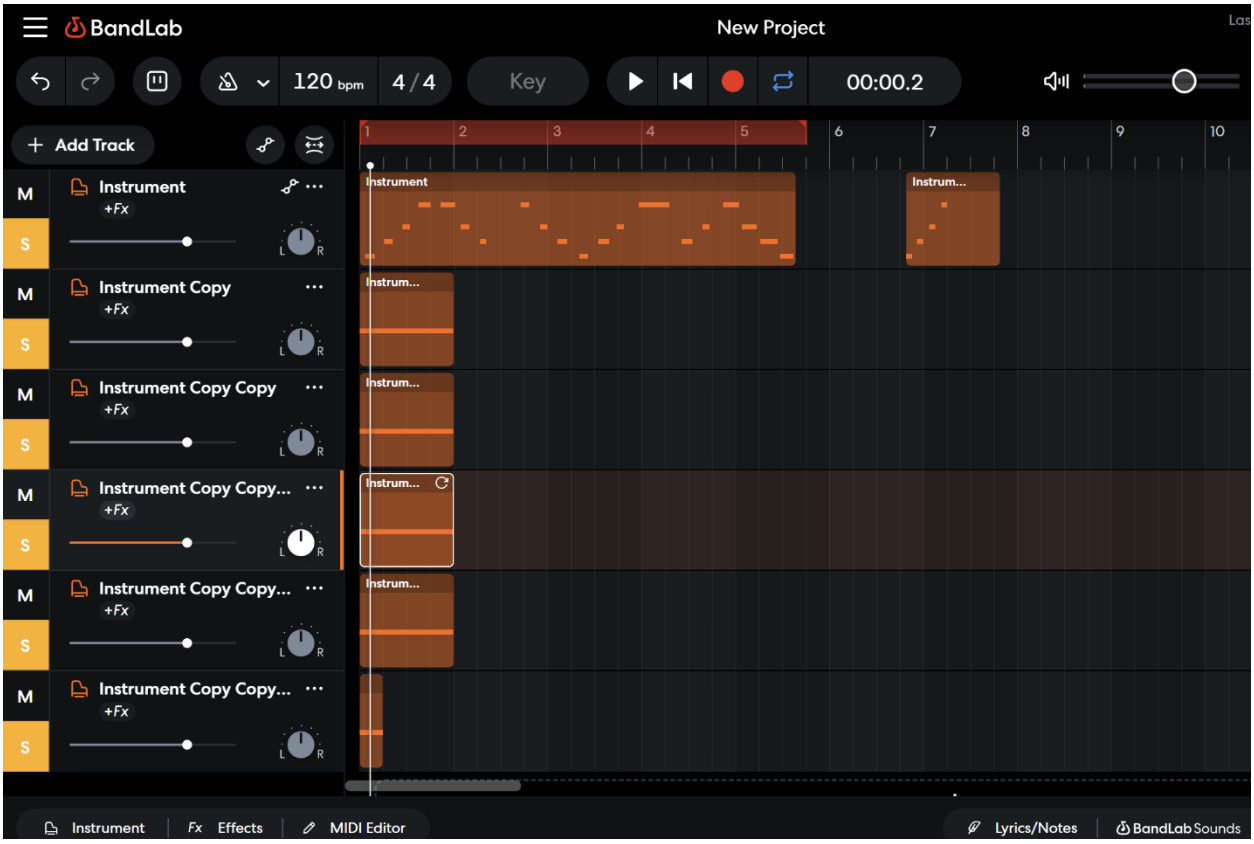
track Notes Pressed


ote play
hitbox
delete class
note
Q W E R
W
row
- Square on Screen for 3 seconds? - How to do this?
- Maybe check if the user is pressing the correct key or use mouse to click Square
- Make nice notes play when key is pressed correctly?
- After 3 seconds make note disappear and play note music or a bad noise
- Scale to Small?

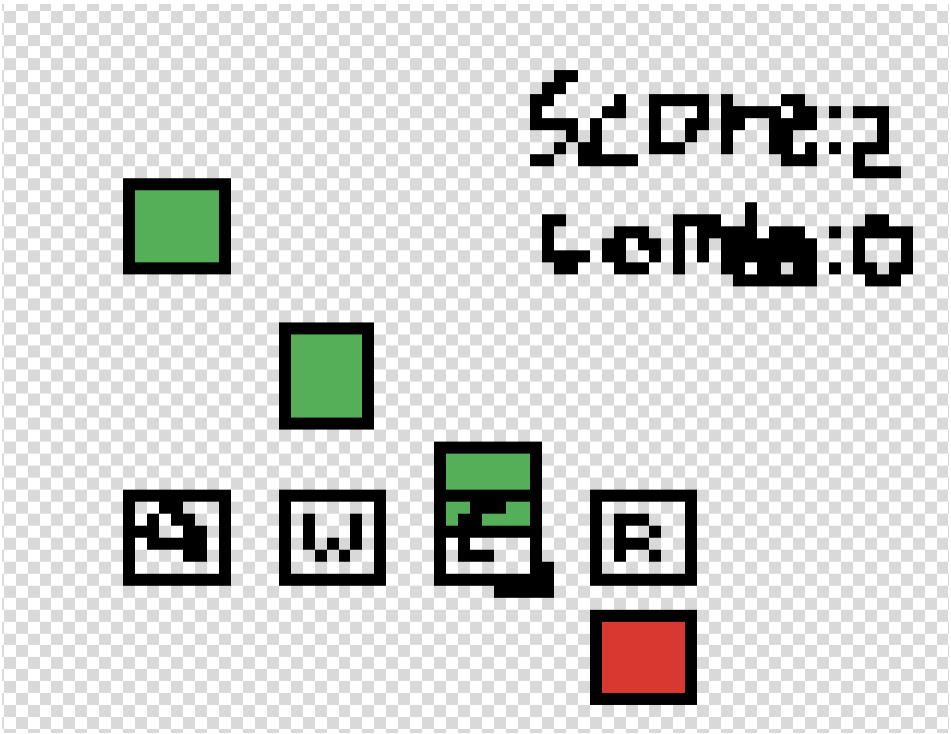
The picture below depicts Justins version of the game which was more of a mouse point and click game like Osu, though we ended up going with my idea of a traditional “block fall down and press key at right time” model.

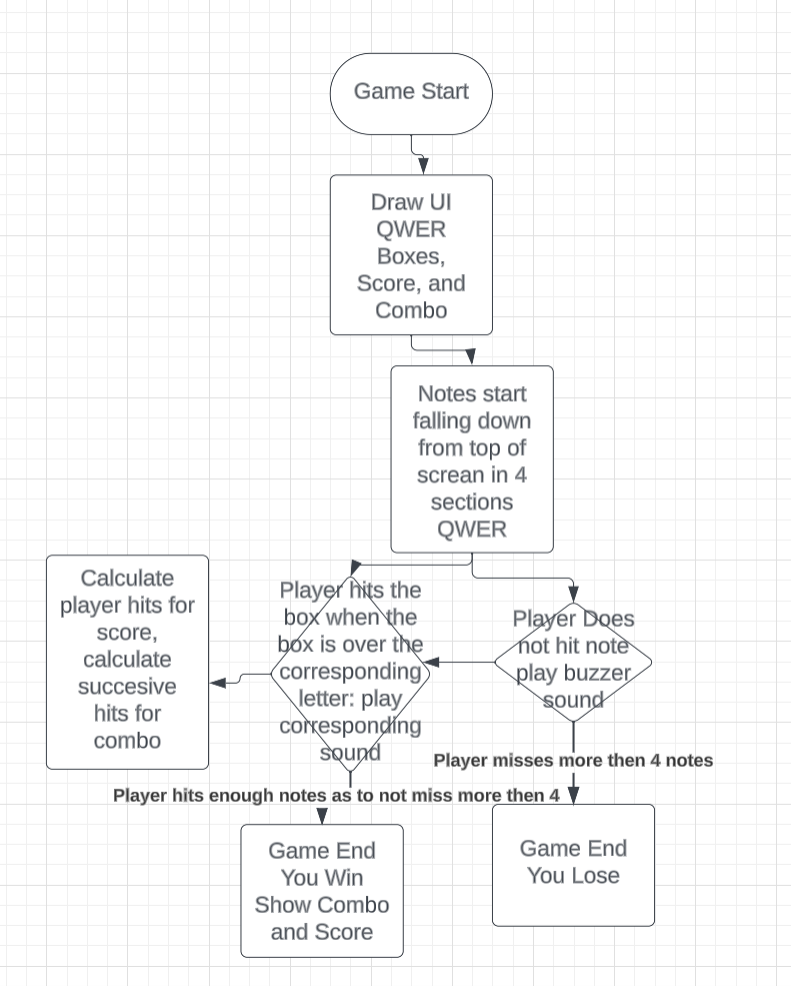


Below shows all my process work in terms of sound design for the game. I chose 4 black keys on the 4th octave because it sounds pleasant in a vacuum. The 4 blocks/bars you see below are the notes while underneath it is the buzzer a compound note that has a deep tone signifying the player did something wrong. The block up at the top is a design I came up with for a song the player can play, but I doubt it will




be used in the end due to its complexity. The block to the right of the song is a scale of the 4 notes. I chose a lo-fi sounding preset for the piano since it felt like it matched the arcade-y nature of the game.


Score:2
Combo:0
W
E
R




Coding: For the coding I made a new class that dealt with loading all the sounds, and created an array for the key sounds so that they can be called when needed super easily. I then added the buzzer as a separate sound and function to be called for whenever the player misses a note. Finally, I made it so that the program unloads all sounds on exit.

Challenges: Didn't know how to make classes work and how to upload to GitHub. So, Justin taught me how to make classes and how to upload commits to GitHub. Had some issues with the program not running when sounds were loaded, turns out references to ".wav" must be in lowercase to work. Couldn't find a free buzzer sound online without signing in so I just made my own using a music program. Had a tonne of work due from being sick with covid so I had to do a lot of work last second.