Individual Pitch:

Team Member: Trevor Siu

***One-liner*:**
I will be dealing with writing and reading data from our MongoDB database (backend), and non-essential quality-of-life game elements (power-ups and audio), as well as creating gameElement classes to allow us more flexibility in the future.

Outline:
In order to write and read data to the database, I will write asynchronous functions to save and read game save data from the JSON file produced by Nicolas' part using threads as we learned in class. I will also be using sound clips to enable feedback for certain actions in game (collecting power-ups, background music, etc.). Finally, after the game is functional, I will be adding the ability of power-ups to drop (based on random chance after a slab is destroyed) and be caught by the paddle to enable more features.

Technical Details:
I will be implementing DatabaseHandler as a singleton to work with Nicolas' GameSaveHandler class. GameSaveHandler.update() method will be able to call DatabaseHandler.save() to write to the database, and save() will use GameSaveHandler's produced json file. I will also have to use MongoDB's api to connect to it. SoundManager will also be a singleton that handles all the audio playback of the game. It will have background music constantly looping that plays upon the window opening, and play(String sound) to play sounds whenever an event on screen occurs. PowerUp.java will just be a class that holds the properties of the class. The implementation will deal with the visuals, collision, and movement of it, as well as different interactions based on the type of PowerUp that collided.

Impact:
The DatabaseHandler will allow for a "cloud save" feature. In other words, we can save data to and from the database so the user can continue their session from any device, so long as they're able to maintain connection to the database. SoundManager will allow for the game to produce audio which includes background music, feedback sounds, and ambient sounds. PowerUps will allow more expression in our design. For example, we could have power-ups that increase the amount of balls the player has, or makes the paddle wider.

Timeline:
I estimate that these features will take about a month to complete in total, given that the prerequisites are complete. For example, database saving requires Nicolas' code to be

functional before testing can begin. I also hope to be able to have a working implementation by the end of the week that the lab going over databases begin.

Risks and Mitigations:

A potential risk is that the features I implement in power-up will produce unintended interactions with the other game elements. For example, changing paddle length may cause errors in collision if we do not handle polymorphism well.

References:

I will be referencing the labs we do in class that go over how to read and write to databases (as per the course outline). For audio, I will consult geeksforgeeks as I found a page discussing usage of Clips to make audio playing methods.