Members: Jaina, Noor, Ebony

Pitch: Our group is developing an exciting one-player game that challenges the player's jumping skills with numerous obstacles throughout the game. The game involves a player avatar that must navigate through flying enemy obstacles while avoiding collisions. Each collision with an obstacle results in the end of the game, while hitting an enemy results in losing a single health point. If the player loses three health points by colliding with enemies, the game ends. It's similar to the popular game "Flappy Bird" but with the added twist of flying through enemy obstacles to make the gameplay even more engaging and challenging.

Requirements:

Requirement 1: The project must incorporate some visual interface using Processing.org libraries. All user interaction must be conducted via this interface.

We will be creating an interactive window with Sprites, arrow controls and possibly mouse click events, using the Papplet library given.

Requirement 2: The project must incorporate some kind of non-blocking concurrent/asynchronous processing that happens at regular intervals. For example, you might push or fetch data from in the background.

We will be using a timer that has a specific time value to continually update the game state, like an interval. It will be a separate thread that runs in the back to update the game state. This will ensure that the game runs smoothly while the user is interacting with the visuals.

Requirement 3: The project must incorporate some kind of non-trivial persistent data state that must be read, processed, and written at regular intervals. For example, you might save a game state in a JSON file. This may or may not be included with Requirement 2.

We will be saving and loading our game state with JSON objects(creating, writing and reading) to make it non-trivial. We can change it before saving the data and make some sort of serialized format to save the data.

Requirement 4: The project must incorporate some kind of self-managing custom iterable data structure. For example, you might have a collection of enemies that are added and deleted based on statistics maintained by the data structure.

To incorporate a self-managing custom iterable data structure for your Flappy Bird game with flying enemy obstacles, you can create a class that manages the collection of enemies and provides methods for adding and deleting enemies based on game statistics.

Requirement 5: The project must be well-documented, complete, and run without errors on final submission.

We will push every week to Github and set group meeting reminders every week on Discord. We'll also write meaningful commit messages. Also we will make weekly goals to ensure that we do not fall behind with the project.

1. Meeting time and format: when will you meet and how?

We will meet at least once a week depending on availability it will be in person or over discord. Tuesdays after school for at least 30 mins.

2. Communications expectations and format: what times of day are people available and by which medium?

Our rules are that we should ensure to respond to messages asap, trying to get back to group members within one day unless there are serious circumstances in which you cannot respond.

3. Roles and responsibilities: who will do what part of the group aspects of the project, including project management?

Roles:
Architect: Noor
Test maker: Noor
UI/UX lead: (Jaina, Ebony)
Backend: Jaina
Graphic design: Ebony

4. Abilities and expectations: how much effort do people want to put into the project and in what areas do they have expertise?

Everyone should do an equal amount, and people will be helping each other in complicated tasks. Ebony has talent in drawing/graphic design.