INFO1113 ASSIGNMENT REPORT

Name: Hoang Bach Nguyen **SID:** 530417293 **UNIKEY:** hngu0996

Object-oriented programming application

In this program, I define classes and create instances for each class to organize all the information. Each class encapsulates a set of data and functionalities represented through the attributes and methods, helping to reduce code duplication and promote code reusability. All classes in the program are listed below, along with their purpose and functionalities.

App.java

Purpose: The main entry point for the game, handling the game setup, user interaction, and visualization.

Functionality:

- Initialization: reads and extracts the game configuration information from a configuration file, initializes the game window, and loads any necessary images or fonts.
- User Input: capture the user interaction.
- Game Loop: Update the game state based on the user input and draw in-game elements.

Process.java

Purpose: Responsible for keeping the game state up-to-date and driving the core logic behind the gameplay.

Functionality:

- Extracting and organizing information: When creating a “Process” instance, all the information will be extracted and stored in other classes.
- Updating game state: Handling the logic of the game and updates all the state. This might involve switching the player’s turn, moving to the next round, calculating the score, etc.
- Interaction with other classes: The intermediate class connects all classes.

Setting.java

Purpose: Keeps all the data extracted from the configuration file. Each Setting object represents a setting of 1 round in the game.

Functionality:

- Extracting and storing information: Load the information stored in the given JSON object (storing the setting of each round) and store it.

Layout.java

Purpose: Stores the Terrain, Tree, and Tank elements and updates these objects throughout the game.

Functionality:

- Storing information: Creating and storing the Terrain, Tree, Tank objects
- Updating the objects: Keep all the objects up to date, like changing the terrain when something exploded, moving average the terrain, etc.

Wind.java

Purpose: Represents the wind data used in each round

Functionalities: Update the wind speed and generate the wind speed randomly.

Tank.java

Purpose: Represents the tank used in the game

Functionalities:

- Storing tank characteristics: The tank's color, health, and points… are kept here.
- Tank's in-game actions and abilities: Perform the tank's movements and their abilities such as firing projectiles, buying fuel, teleporting, etc.

Pointer.java

Purpose: Represents the pointer to show the location of the selected tank

Functionalities: Showing the pointer pointing towards the tank chosen

Projectile.java

Purpose: Represents all the projectile elements from every tank

Functionalities: Updating the projectile's X and Y coordinates are calculated throughout the active period.

Explosion.java

Purpose: Represents the explosion of both tanks and projectiles.

Functionalities: Illustrate the animation of the explosion

Terrain.java, Tree.java

Purpose: Represent the Terrain and Tree elements

Functionalities: Storing the X and Y coordinates of the elements throughout the game

Teleport.java

Purpose: Represents the teleport ability of the tank

Functionality: Display the pointer to choose the location to which the tank will move and move the tank to that location.