German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Nourhan Ehab
Dr. Ahmed Abdelfattah

CSEN 401 - Computer Programming Lab, *Spring 2024*
Attack on Titan: Utopia
Milestone 3
Deadline: 17.5.2024 @ 11:59 PM

In this milestone, you are required to implement the GUI to be able to play the game.

1 General Guidelines

- The effects of any action performed in the GUI should be reflected in the engine and vice versa.
- The player should be able to view all content at all times without the need to resize/minimize/maximize the window during runtime.
- The action that is currently happening in the game should always be clearly indicated in the GUI.
- Make sure to handle all exceptions and validations for any input or action performed. In case any exception implemented in the second milestone arises the player should be notified and the action should be prohibited and another action should be chosen by the player.
- The game should not be stopped/ terminated for any exception thrown. However, clicking the 'X' button on the window must be able to terminate the game at any instant.
- Try to adhere to the MVC architectural pattern to organize the codebase, enhancing maintainability and scalability:
  - **Model**: Manages the data, logic, and rules of the application independently of the user interface.
  - **View**: Represents the GUI which displays game information to the player. The view should be dynamic and reflect changes made in the model.

  - **Controller**: Accepts input, converting it into commands for the model or view. This ensures that the UI is separated from the data processing.

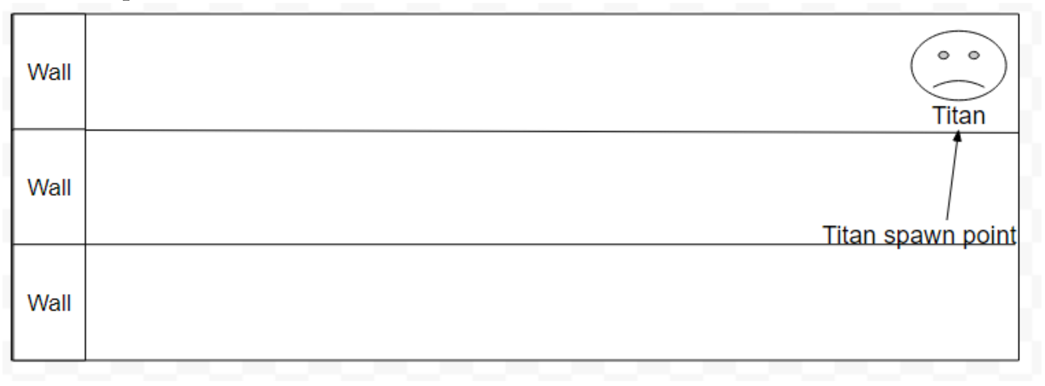
- You are **not allowed** to use Swing for the GUI, only JavaFX is allowed.
- You're free to use SceneBuilder when building the GUI.

2 Game Clarification and Rules

- This is a tower defence game, players' main goal is protect their lane walls from the approaching titans by buying weapons into the lanes to attack the titans present in it.
- The game must be initialized with an initial score of 0.
- The number of turns must start with 1.
- The game will have two modes: Easy and Hard.
- The modes will differ according to the following:

Specification	Easy	Hard
Initial Number of Lanes	3	5
Initial Resources per Lane	250	125

- Each Lane should have a wall at one end and a titan spawn point at the other. You should specify titan spawn distance according to your screen and chosen orientation of your game.
- For example:



Horizontal orientation

3 GUI Requirements

The requirements that should be covered in the GUI are explained below. You will be graded based on the requirements detailed in the following checklist:

Must be displayed whenever the player starts the game
A way to select a game mode (either Easy or Hard)
A way to start and play the game according to the chosen mode
Game instructions
Must be displayed and updated for the player throughout the game
Current score
Current turn
Current phase
Current Resources
Weapon shop
Available lanes
Must be displayed for each weapon in the weapon shop
Name
Type
Price
Damage points
Must be shown and updated to each lane
Wall
Wall current health
Danger level
Available weapons that were bought (if any)
Active titans
Must be shown and updated to each titan
Current health
Difference in height between types of titans
Difference in position according to the base wall
Difference in speed of titans (ie: distance moved each turn) (Hint: can be shown by using a grid for the lane)
Defeated titans must be removed
Player must be able to
Pass his turn
Select a lane to buy a weapon for
Select a weapon to buy from the shop into the selected lane
Buy the weapon selected from the shop
See the bought weapon in the selected lane

Distinguish between types of titans
Distinguish between types of weapons in both shop and lanes
Distinguish between lost and active lanes
Must be shown for any invalid action
Indicator to the player (whether insufficient resources or invalid lane), and reasons why the expected action could not be performed
The game should not be stopped/terminated for any invalid action
Closing the popup should NOT terminate the game
Must be shown whenever the player loses
Must be announced that the player has lost
Score must be displayed
Player must return to the start window
Any specific one of the following can be done additionally
Fantastic GUI + animations
AI (Automate player actions by choosing the best action to do)