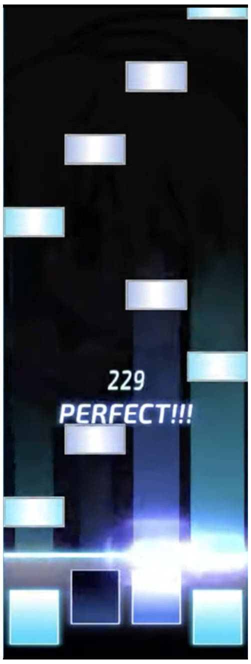
Lab 6: Event Handling and Thread

Instruction

1. Click the provided link on CourseVille to create your own repository.
2. Open Eclipse and then “File > new > Java Project” and set project name in this format **2110215_Lab6_2022_2_{ID}_{FIRSTNAME}**
    - Example: 2110215_Lab6_2022_2_6530000021_John.
3. Initialize git in your project directory
    - **Add .gitignore**
    - Commit and push initial codes to your GitHub repository.
4. Implement all the classes and methods following the details given in the problem statement file which you can download from CourseVille.
    - **The provided source files contain one folder:** src **make sure to add it into your project.**
    - You should create commits with meaningful messages when you finish each part of your program.
    - Don’t wait until you finish all features to create a commit.
5. Export your project into a jar file called Lab6_2022_2_{ID} and place it at the root directory of your project.
    - Example: **Lab6_2022_2_6530000021.jar**
6. Push all other commits to your GitHub repository

1. Problem Statement: Osu! Mania Lite

Osu! is a famous rhythm game with many modes, for example mania mode.

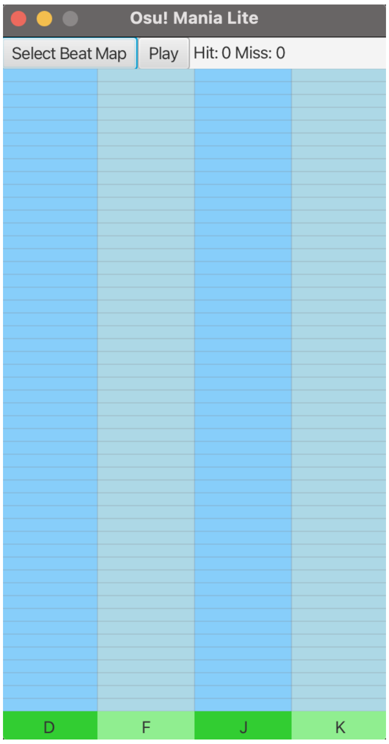


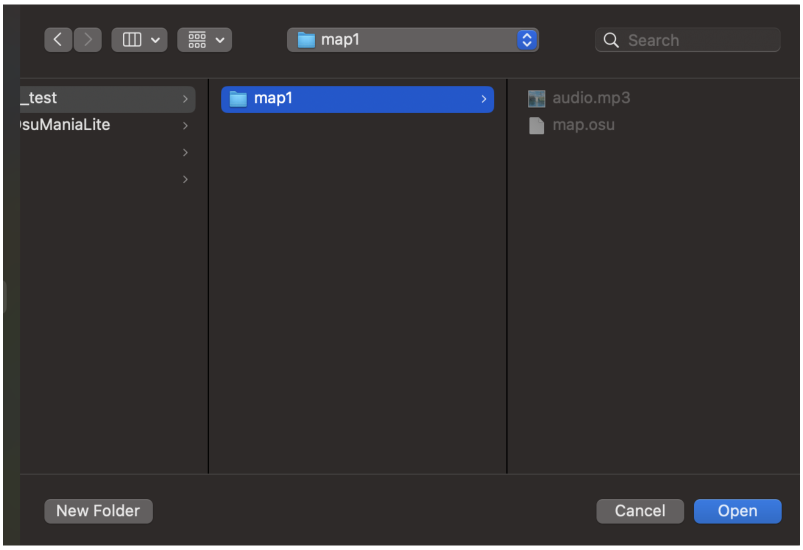
Johny, a CP student, wants to clone this game with Java. He also selected JavaFx to be a UI library in this map. However, he has made some bugs in the code so the program is freezed.

So Johny contacted you, the world famous Java Programmer, to fix the bug in this program.

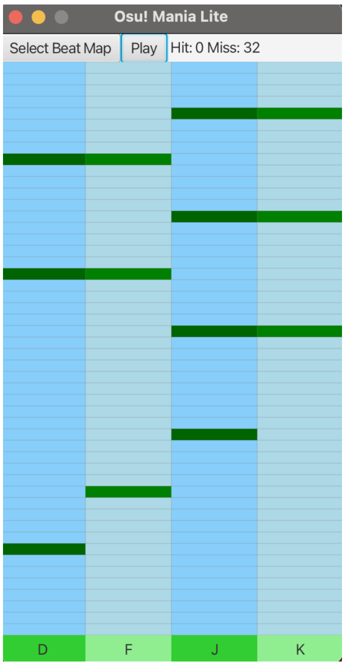
He will reward you as 2110215 Prog Meth Point for each task you implement successfully.

1.1 Overview





The program has UX/UI as the left image, when you click select beat map and select appropriate folder (the folder with 1 mp3 and 1 osu file inside) and click play the program should run an osu like gameplay.



However, due to some bug the program is not running and freezes instead.

Your task is to fix the program to make it run correctly.

2. Implementation Details

Most of the logic and game system has been provided. You only need to fix the bug of the program and implement some utility method.

Your target is:

- Make note flow in the program (example is in Overview section) without program freezing **with Thread**.
- Make hit, miss score counter working correctly **with Thread**.

Please note that:

- You only need to touch 3 files
    - GameCoreWrapper.java
    - ScoreProcessorWrapper.java
    - Utility.java
- Only important fields and relevance for your implementation is provided.
- Access Modifier Notations is listed below
    - + (public)
    - # (protected)
    - - (private)
    - underlined (static)
    - ALL_CAPS (final)
- Note that this is not the best practice to implement a program but to show students about how to use and benefit from thread only.
  To be best practice, the program needs to be refactor much more (which doesn't have enough time).
  However if you like this project and want to contribute and refactor it feel free to make pull requests at
  https://github.com/PongDev/Osu-Mania-Lite-Project
  (which will be made public after this lab score is published
  it may be little differ to the current provided code for making a lab).

2.1 Package application

2.1.1 Class Menu (This class is already provided)

This class contain the program menu with all utility

2.1.1.1 Field

- GameCoreWrapper gameCoreWrapper	GameCoreWrapper instance
- ScoreProcessorWrapper scoreProcessorWrapper	ScoreProcessorWrapper instance

2.1.1.2 Method

+ GameCoreWrapper getGameCoreWrapper()	getter for GameCoreWrapper
+ ScoreProcessorWrapper getScoreProcessorWrapper()	getter for ScoreProcessorWrapper

2.1.2 Class GameCoreWrapper (extends ???)

This is a wrapper class for GameCore class used to implement and execute GameCore

2.1.2.1 Field

- GameCore gameCore	GameCore class which contains the main game loop when executed.

2.1.2.2 Method

+ GameCoreWrapper(BeatMap beatMap, NoteTiles noteTiles)	Initiate GameCore instance with provided value
+ void ???()	This function executes the GameCore instance.

2.1.3 Class ScoreProcessorWrapper (extends ???)

This is a wrapper class for ScoreProcessor class used to implement and execute ScoreProcessor

2.1.3.1 Field

- ScoreProcessor scoreProcessor	ScoreProcessor class which contains the score processor loop when executed.

2.1.3.2 Method

+ ScoreProcessorWrapper(BeatMap beatMap, KeyPressTiles keyPressTiles, Label hitScoreLabel, Label missScoreLabel)	Initiate ScoreProcessor instance with provided value
+ void ???()	This function executes the ScoreProcessor instance.

2.1.4 Class Utility

This is a utility class contain utility function

2.1.4.1 Method

+ GameCoreWrapper setUpGameCore(GameCoreWrapper gameCoreWrapper, BeatMap beatMap, NoteTiles noteTiles)	(Currently, this function is not working correctly) Create and run GameCoreWrapper instance then return the created instance Note that if instance is not null it must be stop (with interrupt) before creating new instance
+ ScoreProcessorWrapper setUpScoreProcessor(ScoreProcessorWrapper scoreProcessorWrapper, BeatMap beatMap, KeyPressTiles keyPressTiles, Label hitScoreLabel, Label missScoreLabel)	(Currently, this function is not working correctly) Create and run ScoreProcessorWrapper instance then return the created instance Note that if instance is not null it must be stop (with interrupt) before creating new instance
+ void updateLabel(Label label, String newText)	Receive a label and set new text for that label Note that this method can be called from other threads
+ void cleanUp(Menu menu)	Clean up GameCoreWrapper and ScoreProcessorWrapper when exit with interrupt

3. Grading Criteria

Score will be given by below criteria 5 points (1.25 points for each) and will be scaled to 2.5 points

- GameCoreWrapper is Thread and can running correctly without freezing the program. (1.25 Points)
- ScoreProcessorWrapper is Thread and can running correctly without freezing the program. (1.25 Points)
- updateLabel can be working without exception. (1.25 Points)
- GameCoreWrapper and ScoreProcessorWrapper is interrupt when cleanup the program. (1.25 Points)

Appendix

- Image Reference From: https://www.youtube.com/watch?v=St24qYtcZvU
- Example Beat Map From:
  https://osu.ppy.sh/beatmapsets/1963261#mania/4069102
  https://osu.ppy.sh/beatmapsets/518660#mania/1462571
  https://osu.ppy.sh/beatmapsets/1904946#mania/3945230
- Osu File Structure:
  https://osu.ppy.sh/wiki/en/Client/File_formats/Osu_%28file_format%29

Note that this program currently support original osu file format (obtain from extract osz file download), so you can use any osu mania beat map with this application

Also note that the hold note is not implemented yet

You can also download more beatmap to try from
https://osu.ppy.sh/beatmapsets?m=3&s=any