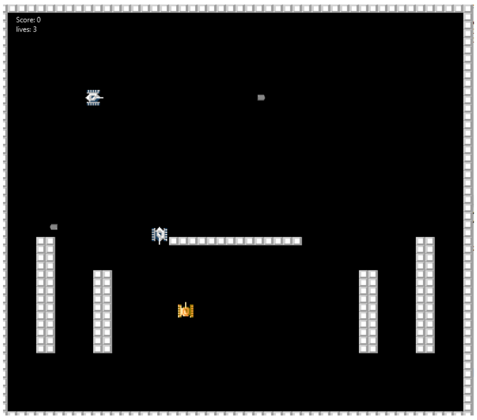
PROGRAMMING ASSIGNMENT 4

TAs : Emirhan SELİM
Due Date : 2.06.2024 (23:00:00)
Programming Language: Java 8 (Oracle)

Tank 2025



1 Introduction

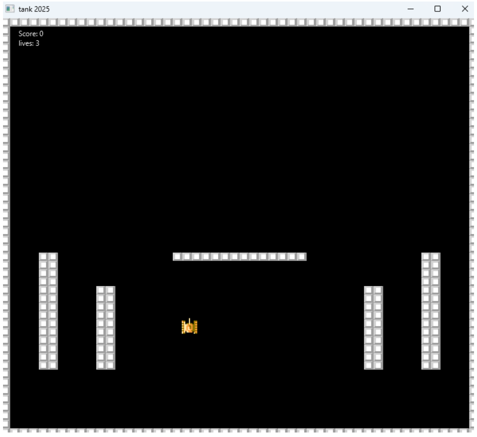
In this assignment you will design and implement a GUI(Graphical User Interface) using Javafx. As opposed to the command-line programs where the interaction between the user and the computer often relies on string of text, a GUI program offers a much richer type of interface where the user uses a mouse and keyboard to interact with GUI components such as windows, menus, buttons, check boxes, text input boxes, scroll bars, and so on. The fact that most people today interact with their computers exclusively through GUI, a developing a GUI-based application has become a must for the new developers.

There are lots of frameworks to develop a GUI application (such as Swing, SWT, AWT, and the like). In this assignment, you are to employ JavaFX framework to complete this assignment. JavaFX is a software platform for creating and delivering desktop applications, as well as Rich Internet Applications (RIAs) that can run across a wide variety of devices. JavaFX has support for desktop computers and web browsers on Microsoft Windows, Linux, and macOS. You are forbidden from using any framework other than Javafx

You will use the assets that will be shared alongside this assignment.

2 Game

The gameplay involves a player controlled tank and enemy tanks trying to destroy the player tank. tanks should move in 4 cardinal directions and each time a tank moves, there should be an animation showing thread movement. The game should take place in a map surrounded by indestrctible walls and there should also be indesctructible walls like in the image



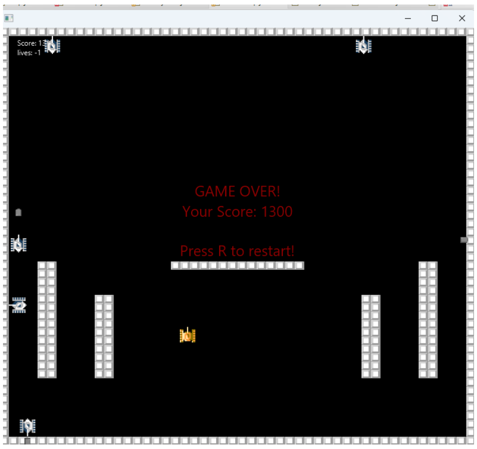
Score: 0
lives: 3

The player can move in 4 cardinal directions with arrow keys and fires a bullet with the "x" key. With some intervals, the enemy chooses a random direction. The enemy then moves in this direction until it chooses another direction. The enemy also constantly fires in a random interval.

The bullets should follow a constant speed after being shot until they hit a wall or tank. The bullets that hit indestructible walls should disappear with a small explosion effect that will last for a short time. It should not be possible to pass through the walls.

Enemy tanks spawn around the upper part of the map in random intervals, creating a dynamic challenge for the player If a bullet shot by the player hits an enemy, the enemy tank should be destroyed with an explosion effect that lasts for a short time before disappearing and the player's score written on the left upper corner should go up.

The player has three lives indicated below the score and each time the player is shot by an enemy's tank, the same exploison effect should be used and the player should lose a life. the player should then respawn some time later in the original starting position. If the player loses all their lives, a gameover screen with a score text should be shown, where the players can restart the game with "R" key and close the game with "Escape" key.



Score: 13
lives: -1

GAME OVER!
Your Score: 1300

Press R to restart!

Pressing "P" key during game should bring up a pause menu where players can similary restart the game or close the game. The pause menu shouldnt come up while gameover screen is active

The exact design of your maps and menus do not have to be the same as mine. If you think a different looking map would be better, you are free to do it like that. Just do notmake an empty map with no walls

3 Test and Execution

Your code must be compiled and executed under Java 8 (the one provided on Piazza) on Microsoft Windows Operating System (preferably Windows 10 or newer) or Apple MacOS (preferably MacOS 12 or newer) as you will not have any chance to compile and run your program on dev server since it does not contain any graphical user interface. If your code does not compile and execute under Java 8 by Oracle, then you will be graded as 0 for code part even if it works on your own machine under any other versions or providers. It is forbidden to use any external libraries such as JSON Library, and it is also forbidden to use any scene builders, FXML stuff etc. You are encouraged to build your own GUI just with Java Code as stated at JavaFX slides. It is also forbidden to use any extra CSS stuff. Moreover, your code must use JavaFX for GUI (Swing, AWT etc. is also forbidden). Sample run command is as follows:

```
    javac Tank2025.java
--
```

For this assignment, there is no need to test your code on the dev server

4 Extras

You can optionally include vertical scrolling and horizontal scrolling for bonus points. Make sure you first complete the requirements before attempting to do extra work.

5 Grading Policy

Task	Grade
Starting game succesfully	5
Player Movements and shooting	20
Enemy Movements and shooting	15
Bullet behaviour	15
Walls	5
Scoring and lives behaviour	10
Pause Menu and Game Over	10
Game Over Menu	10
Comments in JavaDoc Style	10*
Vertical Scrolling and side scrolling	30**
Total	100

*** The score of the clean code comment part will be multiplied by your overall score (excluding clean code comment part) and divided by the maximum score that can be taken from these parts. Say that you got 45 from all parts excluding clean code comment part and 10 from clean code comment part, your score for**

clean code comment part is going to be 10*(45/90) which is 5 and your overall score will be 45+5=50.

**** These are bonus points and not necessary to get 100 from this assignment**

6 CheckList

You must append a PDF file ("Checklist.pdf") that contains the requirements that you did not satisfy, please state your situation such as "This functionality does not work at all.", "There is a tiny issue such as . . . " so that your assignment will be graded by taking them into consideration, please note that if you did not stated a requirement, that is assumed that that functionality is fully working on your system and that will be graded as zero even if a tiny issue occurs that stated as has to be checked (not for the issues that they are not stated at requirements, but remember that there may be some extra conditions that you should check, but they are not going to be graded as zero if you do not satisfy them). Moreover, if you did not provide such a file, it is assumed that your code is working well without issue and that it will be graded as zero even if a tiny issue occurs that stated as must be checked.

7 Demo

You must provide a demo video that is at most 4 minutes in length and includes every feautere you implement

8 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported byz the system).

```
-b<studentID>.zip
  -Main.java, or *.java
```

Late Policy

You have two days for late submission. You will lose 10 points from maximum evaluation score for each day (your submitted study will be evaluated over 90 and 80 for each late submission day). You must submit your solution in at the most two days later than submission date, otherwise it will not be evaluated. Please do not e-mail to me even if you miss the deadline for a few seconds due to your own fault as it would be unfair for your friends, e-mail submissions will not be considered if you do not have a valid issue.

Notes and Restrictions

- You must use JavaDoc commenting style for this project, and you must give brief information about the challenging parts of your code, do not over comment as it is against clean code approach. Design your comments so that if someone wants to read your code they should be able to easily understand what is going on. You can check here to access Oracle's own guide about JavaDoc Sytle.

- You must obey given submit hierarchy and get score (1 point) from the submit system.
- Do not miss the submission deadline.
- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names for your variables, classes, and functions regardless of the length. The names of classes, attributes and methods must obey to the Java naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's Piazza group, and you are supposed to be aware of everything discussed in the Piazza group. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.