4CCSAPPA Programming Practice and Applications

Coursework 4: JavaFX Game Development

Your assignment is to build a game, from scratch, using vector graphics. You will have to use everything you have learned about GUI development and OOP to complete this assignment. The key to a successful game will depend on the amount of planning you have put into your game & software design. That is, before a single line of code is written, your team should know exactly which classes (and methods) are needed.

This assignment is a group project. You must work in groups of 3 or 4. **If you haven't done so already, register your group on KEATS as soon as possible.**

1 Introduction

Games have long served as a benchmark for evaluating artificial intelligence (AI) techniques. From classic board games like chess and Go to modern video games such as StarCraft II and Dota 2, researchers and developers have used these environments to test decision-making processes, cooperative behaviour, and learning models. By creating a structured environment with clear rules, challenges, and feedback mechanisms, games provide an excellent framework for implementing and assessing artificial intelligence.

For this assignment, you will develop a game of your choosing using vector graphics. Vector graphics are a form of computer graphics in which images are created directly from geometric shapes, e.g., points, lines, curves and polygons, drawn on a Cartesian plane. An example is shown in Figure 1. You will also have the option of using JavaFX Animation package. Thus, you will not need any game assets (sprites, art work etc.) Everything will be drawn either using the Animation package and/or the Canvas GraphicsContext object. While this assignment will not focus on implementing any AI, the game could serve as the foundation for research in AI.

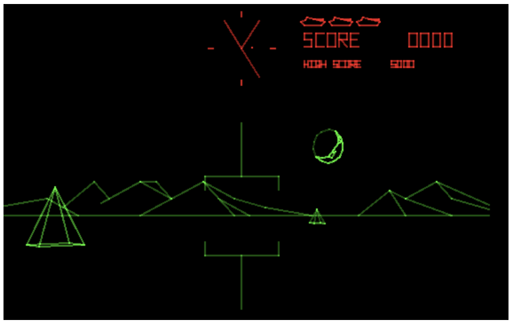

Figure 1: An example of a game that uses vector graphics. Notice the landscape is drawn using lines, even the planet in the sky is made of a series of lines.

Figure 2 shows some games you could use as inspiration given you are working with JavaFX and only have a few weeks. Of course you don't have to recreate the exact game; you can create a game with similar elements or plot.

2 Requirements (80 points)

2.1 General Requirements (40 points)

Your game must have the following

- **Menu Bar**. The menu bar should have 'File' and 'Help'. The 'File' menu item should have a 'Quit' option for ending the game. The 'Help' menu item should have an 'About' option which displays a pop-up showing the game's title and authors.
- **Score**. The game must track and display the player's score. The score should update dynamically based on game events (e.g., collecting items, defeating enemies, completing levels).
- **Animation**. The game must include at least one animated element. You may use: the JavaFX Animation package (e.g., FadeTransition, TranslateTransition) and/or custom animations drawn using the GraphicsContext object within a Canvas.
- **Interactivity**. The game should respond to user input (e.g., keyboard or mouse events). User input should influence the game state (e.g., movement, score changes, triggering animations).
- **Application Design**. The game should demonstrate an understanding of the fundamental of OOP: encapsulation, inheritance, polymorphism, and abstraction.

2.2 Game Quality (40 points)

- **Clear Objective**. The game should have a well-defined goal or purpose.
- **Intuitive Controls**. Players should easily understand how to interact with the game, e.g., players use the mouse, *wasd* keys, space-bar or arrow keys to play.
- **Stable Frame Rate**. The game should run smoothly with minimal lag, i.e., avoid resource hungry code/design that causes performance issues.
- **Professionalism**. The game should feel complete, not rushed or unfinished.

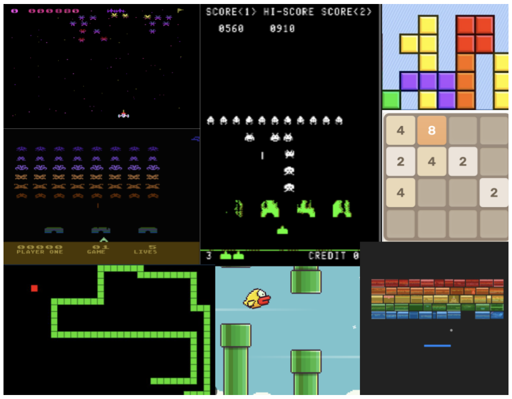

Figure 2: Here are a few games you could emulate using JavaFX: Space Invaders (Atari), Tetris, 2048, Snake, Flappy Bird, and Breakout (Atari)

2.3 Report (10 points)

Write a short report (no more than four pages long), containing the following:

- The names and student numbers of all students who worked on the submission
- A brief description of your game, game rules and mechanics
- At least two screenshots demonstrating gameplay and UI elements
- Description of any known issues or areas for future improvements

3 Submission (10 points)

- You'll submit a zip file containing the following

  1. A Jar file of your BlueJ project. —You can create a Jar from within BlueJ by going to Project, and then "Create Jar File...". You do not need to change any of the default options, and so you should just click the "Continue" button. **The Jar file must contain your source code, i.e., the *.java files, and it must run on BlueJ.**
  2. All of your Java files (*.java)
  3. Report (as a PDF)

- **Your assignment will be penalised if you are missing any files or included files that were not asked for in the task sheet.**

- Click the *submission link* to submit your work. Follow all instructions in the 'Student Submission Guide'. If you have any trouble submitting your work, email Jeffery Raphael as soon as possible. Do not wait until the last hour to attempt your first submission.

4 Deadline & Late Submissions

You must submit your assignment on Gradescope via KEATS by **Fri., Mar.** 28th **16:00 (4pm)**.

All coursework must be submitted on time. If you submit coursework late and have not applied for an extension or have not had a mitigating circumstances claim upheld, you will have an automatic penalty applied. If you submit late, but within 24 hours of the stated deadline, the work will be marked, and 10 raw marks will be deducted. If this deduction brings your mark below the pass mark (40%), your mark will be capped at the pass mark. **All work submitted more than 24 hours late will receive a zero.**