

Java Journey - Supplementary Assessment Project

Supplementary assessment project



Java Journey

Generative AI tools cannot be used in this assessment task: In this assessment, you must not use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.

Due time and date: 11:55pm, Friday 19 January 2024

Project overview

In order to pass FIT9131 you must gain at least a pass grade in this supplementary assessment. This is an individual assessment task and must be your own original work. You must attribute the source of any part of your code that you have not written yourself. Please note the section on plagiarism in this document.

This assessment task will require you to write a program to play the game '*Java Journey*'. You will be supplied with a series of specifications that your program will need to meet.

In preparing your program, please note the following:

-

You must use the workspace environment in the Ed platform to code all parts of your program.

-
-

You must not copy and paste large sections of code from somewhere else.

-

-

You must acknowledge all code in your project that you have taken from other sources.

-
-

The Java source code for this project must be implemented according to the FIT9131 Java Coding Standards.

-
-

Only a text interface is to be used for this program. More marks will be gained for a program that is easy to follow with clear information/error messages.

-
-

In this assessment, you must *not* use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.

-

Assessment details

You will complete this assessment in the workspace environment within Ed.

In this assessment, you will write a program to play the game "Java Journey". The objective of the game is for the player (human) to control a car on a highway and avoid obstacles in order to reach a destination safely and efficiently.

The player begins the game on a random lane on a **three-lane** highway. With each turn, the player has to make a choice between a few possible moves. Along the highway the player will encounter obstacles, some of which offer benefits and others that should be avoided. The vehicle the player chooses will have certain characteristics: *maximum fuel*, and *damage sustainability*. These characteristics help to determine the success or failure of the journey. The game ends when the player sustains damage greater than the maximum sustainable to their vehicle, runs out of fuel, or manages to get to the end of the highway.

With this assessment, you are free to add any additional aspects to the assessment to make it more engaging provided you do not miss out on any of the requirements outlined within this specification document. Any additional features that you add must be described in a document *features.txt* and included with your submission.

Gameplay

The following criteria outline the gameplay specifications that your program is required to meet:

•

When the game starts, the program reads the file 'vehicles.txt'. This file contains the different vehicles that can be selected by the player. Each line in the file has information for one vehicle, as follows: *vehicle type*, *maximum fuel allowed (max fuel)*, and *maximum sustainable damage (max damage)*.

•

•

The game begins by requesting the player enter their name. This name must be between 5 and 10 characters (inclusive) in length and contain only lower-case alphabetic characters. If these requirements are not met, the player should be asked to re-enter their name until the requirements are met.

•

•

The player selects from a difficulty level, with the corresponding specifications for each level as follows:

•

•

○

Easy

○

○

■

Highway length: randomly selected between 10 and 15 sections

■

■

Max fuel for vehicle of choice: 100%

■

■

Number of obstacles on highway: 12

■

○

Moderate

○ ○

■

Highway length: randomly selected between 16 and 30 sections

■

■

Max fuel for vehicle of choice: 80%

■

■

Number of obstacles on highway: 24

■

○
Hard

○ ○

■

Highway length: randomly selected between 31 and 50 sections

■

■

Max fuel for vehicle of choice: 50%

■

■

Number of obstacles on highway: 45

■

●

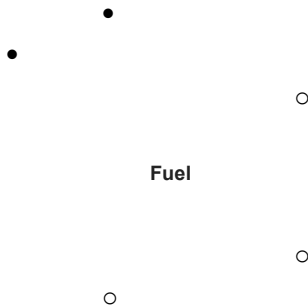
The player is then provided with an option to select their vehicle.

-
-

Once the player has chosen their vehicle they are informed of the length of the highway and the current fuel contained in their vehicle.

-
-

The game then randomly places each obstacle on the highway. No obstacles must be placed within the first 3 sections of the highway. Each obstacle can be one of the following types:



Provides the player with 10 fuel points

The probability of occurring is 30%

Represented in the game as an 'F'

Roadblock



■

The vehicle can't move past this obstacle and must change the lane to proceed

-
-

The probability of occurring is 40%

-
-

Represented in the game as a 'B'

-

○

Tyre Spikes

○

○

-

Cause 45 damage to the player's vehicle

-
-

Probability of occurring is 20%

-
-

Represented in the game as an 'S'

-

○

Deep pothole

○

○

-

Cause 60 damage to the player's vehicle

■
■

The probability of occurring is 10%

■
■

Represented in the game as an 'P'

■

●

The game will show at most 10 sections of the highway to the player at any point in time including all obstacles on each lane.

●
●

The player begins the game randomly placed on any lane of the highway in the first section. The player can be represented using the character '@'

●
●

The player is given the following options they can perform:

●
●

○

Move forward

○

○

■

This option allows the player to move forward one section on the highway. This move is always in the same lane and costs 1 fuel point.

■

○

Move up



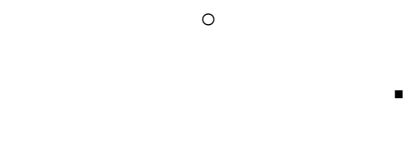
This option allows the player to change to the lane above their current lane. This move is only applicable if the player is not in the topmost lane of the highway. This move costs 2 fuel points.

Move down



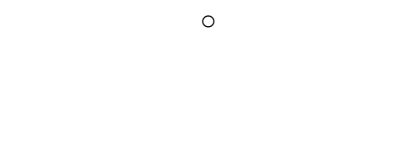
This option allows the player to change to the lane below their current lane. This move is only applicable if the player is not in the lowest lane on the highway. This move costs 2 fuel points.

Move diagonally up



This option allows the player to move forward one section on the highway and also change to the lane above their current lane. This move is only applicable if the player is not in the topmost lane of the highway. This move costs 4 fuel points.

Move diagonally down



■

This option allows the player to move forward one section on the highway and also change to the lane below their current lane. This move is only applicable if the player is not in the lowest lane on the highway. This move costs 4 fuel points.

■

•

During gameplay, if the player crosses an obstacle, the penalty or benefit is applied to the player, irrespective of whether the player ends up in a position in front of the obstacle once their move has been completed. If the fuel is picked up by the player, the total fuel cannot exceed the maximum capacity of the player's vehicle. Note that the roadblock obstacle cannot be crossed - the player must navigate around it.

•

•

The player must navigate the length of the highway without sustaining damage greater than their vehicle can sustain or running out of fuel before reaching the end of the highway.

•

•

On completion of the game, the game should write the distance covered, the number of moves made and the outcome of the journey, i.e. whether or not the player managed to successfully finish the journey), to a file called *'output.txt'*.

•

Game Requirements

•

You must use appropriate collections within your program and be able to justify your choice.

•

•

You must use appropriate good design within your program by correctly applying abstraction and modularization techniques.

•

•

Your program must only read and write to the file one time (at the start of the program to read and at the end of the program to write)

-
-

Your program must ensure to use exception handling correctly. Your program must not crash no matter what the user enters.

-

Assessment

Assessment for this project will be done via an **interview** with one of the FIT9131 Semester 2 teaching staff. The marks will be allocated as follows:

-

10% - Test strategy for a single class. This must not be the class used to generate random numbers.

-
-

10% - Class diagram, Java code quality and object-oriented design quality. This will be assessed on code quality (e.g. compliance with coding standards), appropriate design and implementation of classes, fields, constructors, methods, and validation of the object's state.

-
-

10% - Program functionality in accordance to the requirements.

-
-

70% - Oral assessment.

-

A reminder that you must use the workspace environment in the Ed platform (opposite this project specification) to code all parts of your program. You must not copy and paste large sections of code from other sources, and you must acknowledge *any* code in your project that has been taken from other sources.

Marks will be deducted for incomplete submissions and code that has not been prepared in the Ed workspace.

You must submit your work by the submission deadline on the due date (a late penalty of 10% per day, inclusive of weekends, of the possible marks will apply). There will be no extensions for this supplementary assessment.

All submitted source code must compile. Any submission that does not compile, as submitted, will receive a grade of 'N'.

Oral assessment

As part of the assessment, you will attend an interview following the submission date. At the interview, you will be asked questions about your code. You will be asked to explain your code/design, modify your code, and discuss your design decisions and alternatives. Marks will be awarded for your answers (the marker may also delete excessive in-code comments before you are asked to explain that code).

In other words, you will be assessed on your understanding of the code, and not on the actual code itself.

Interview times will be arranged after the submission date and will take place on campus or online. Please contact Judy Sheard: judy.sheard@monash.edu to arrange a time.

It is your responsibility to make yourself available for an interview time. **Any student who does not attend an interview will not receive a pass grade for this assessment.**

Submission Requirements

The project must be submitted by **11:55pm, Friday 19 January 2024**

The submission requirements for the project are as follows:

-

The main class in your program **MUST** be called **Journey.java** and it should contain the **main()** method to start the program.

-
-

Class diagram submitted as a pdf file.

-
-

Test strategy submitted as a pdf file.

-
-

Submit all your work (coding, class diagram and test strategy) via the Ed platform.

-
-

Re-submissions are allowed (and encouraged) before the submission deadline. Please ensure however that you do not click on the submit button *after* the due date. Your final submission will be used for grading purposes, and any submission made after the deadline will incur a late penalty.

-
-

A signed Assignment Cover Sheet. [Note: You are required to download the [Assignment Coversheet](#), sign the document and upload the pdf file in the Ed platform (you may drag and drop to the Toggle Pane)]

-

Marks will be deducted for any of these requirements that are not complied with.

Warning: there will be no extensions to the due date. Any late submission will incur a 10% per day penalty. It is strongly suggested that you submit the project well before the deadline, in case there are some unexpected complications on the day (e.g. interruptions to your home internet connection).