Assignment 1

Team number: 3
Team members:

Name	Student Nr.	Email
Quinn Rutherford	2670152	q.d.rutherford@student.vu.nl
J. Antonio Fortanet C.	2671921	j.a.fortanetcapetillo@student.vu.nl
Niatna Tesfaldet	2707692	n.h.tesfaldet@student.vu.nl
Nikita Sazhinov	2631755	n.sazhinov@student.vu.nl

This document has a maximum length of 5 pages (excluding the contents above).

Introduction

The system designed is a minigame taken from the 2020 highly acclaimed game “Cyberpunk 2077”. The game consists of a matrix of values from which the user must match a given sequence.

The game starts by the user selecting a value from the first row of the matrix. After that, the user has to pick a value from the column where the last selected element was selected, and so on alternating from row to column until one of the sequences is completed or the timer runs out. (Note: A correct sequence can include values in the buffer before the sequence that are not part of the sequence)

The game ends when the user completes one of the sequences by selecting values from the matrix in the correct order and therefore the user completes the puzzle. If the timer runs out the user loses the game. If the user wants at any point (even if the game is not done), they may press the Refresh button which restarts the game with a different puzzle.

If during the game the user thinks an error has been made, an undo button will be available so the user is able to remove one value from the buffer. This however, does not change the value of the timer.

The main type of user for this game could be both, somebody who has played the game and wants to practice on getting better at the minigame or a user who plays the minigame as a form of entertainment without necessarily being aware that it is taken from another game.

Video Explaining the Minigame:
[Cyberpunk 2077 Hacking Minigame Tutorial]

Features

Functional features

The features we chose for this are necessary for the game to work according to the characteristics from the game it is inspired from. The matrix, the buffer, the sequence and the timer are essential for the game to be functional. The puzzles will be premade and saved in .txt files so the game does not generate them each time a new game starts. The game will be presented in a GUI so that it is more accessible.

ID	Short name	Description	Champion
F1	Read puzzle	The system shall **read puzzle information** from .txt files and **store the information** in the matrix and buffer.	Antonio
F2	Display	The system shall **display the matrix, sequence(s), and the buffe**r using our GUI library.	Niatna
F3	Countdown Timer	The system shall have a **countdown timer** which stops the minigame when it runs out and **display this value** at all times.	Nico
F4	User Input	The system shall allow a user to **select a value** in the matrix with **mouse input** to add to the buffer when clicked on, this change could potentially be undone if the undo feature is implemented.	Antonio
F5	Code check	The system shall **check the buffer** to see if one of the sequences is completed. If the sequence is completed, then the game ends and the user wins.	Quinn
F6	Refresh Button	The system shall have a **refresh button** to generate a new puzzle, when the button is clicked the GUI refreshes with a newly generated puzzle taken from the puzzle pool, and an empty buffer and resetted timer.	Nico
F7	Row/Column Highlight	The system should **highlight the row or column** the user can pick from to add it to the buffer. Each turn the user picks a new element in the matrix, either the column of the element or the row of the element is selected for the next turn.	Niatna
F8	Move log/Undo	The system should allow the user to **undo a move** from the buffer. This could be done using game states.	Quinn

Quality requirements

The qualities we chose have the purpose of enhancing the user's experience and make it more seamless to keep solving more puzzles. They also work to make the game feel more complete and easy to use. The first requirement we chose is to make the game controls very easy to use and clear to the user. The second requirement we chose is so we can add more puzzles for the game without having to change the code of the project. The third and forth requirements were chosen to better the gaming experience.

ID	**Short name**	**Quality attribute**	**Description**
QR1	Input command checks	Reliability/Responsiveness	The player will only be allowed to select one value in the allowed row or column per turn.
QR2	Extensible puzzles	Maintainability	The minigame shall be easily extendable by adding new .txt files which can be displayed by the system.
QR3	Instantaneous results	Responsiveness	Once the player makes a move in the hacking game, the result of the move shall be available within 0.3 seconds.
QR4	Puzzle variety	Reliability/Usability	A specific puzzle will not be re-played by the player at least after 5 games of being played.

Java libraries

JFoenix

We decided to use JFoenix because it is a user friendly library which will allow us to learn it quickly which is great as we dont have much Java experience. We initially looked into this library because it was recommended to us and will provide us with a simple interface for the simple GUI functionalities we require. JFoenix's drag and drop components will help us quickly implement the basic capabilities of the minigame.

Cyclops

A Functional programming library will help us improve the cleanliness of the code and implement the game logic more easily. Cyclops include features such as Monad and stream utilities, comprehensions, pattern matching, functional extensions for all JDK collections, future streams, trampolines and much more. This library will help us with the implementation of the matrix and/or the checking of the values, and allow us to simplify our code.

Time logs

Team number	3		
Member	**Activity**	**Week number**	**Hours**
Quinn Rutherford	Define functional features	1	2
Antonio Fortanet	Search Java libraries	1	1
Antonio Fortanet	Add justification for feature	1	1
Nikita Sazhinov	Work on document intro	1	1
Niatna Tesfaldet	Java libraries	1	1
Nikita Sazhinov	Quality requirements	1	1
Antonio Fortanet	Revise intro	2	1
Antonio Fortanet	Add quality requirements	2	1
Quinn Rutherford	Java libraries explaination	2	1
		TOTAL	10