Assignment 1

Team number: 12
Team members

Name	Student Nr.	Email
Omer Faruk Cakici	2665502	omcakici99@gmail.com
Xiaoxuan Lu	2661045	junjunlujun72@gmail.com
Yasir Uçkun	2666346	ahmetyuckun@gmail.com
Ying Ying Ma	2686767	yingyingma@hotmail.nl
Emre Furkan Guduk		emregdk2853@gmail.com

Introduction

Author(s): All members

For our project, we have decided to implement our version of The Cyberpunk 2077 hacking minigame in Java. The system consists of five parts:

1. Code matrix (where players pick tiles)
2. Buffer size (the number of tiles players can pick)
3. Sequences required to upload (a list of possible sequences players can complete)
4. Time (the time remaining for players)
5. Score (the score shows how well the player solve the puzzle)

The data (code matrix, buffer size, sequence lists) for the game is implemented as follows:
The player can choose from 40 different puzzles (a set of puzzles from a Git repository). We reuse the set of puzzles instead of generating a code matrix and sequence lists randomly, as it might be possible that no solution exists in that case. The 40 different puzzles ensure that the system could have different values in the code matrix, buffer, and sequence lists. The time of different puzzles will be different based on a number of sequence lists. The puzzles with one sequence list give players 40 seconds to finish. One more sequence list will give players 20 seconds more to finish. For example, a puzzle with three sequence lists gives players 80 seconds to finish the game.

The basic dynamics of the system is shown below:
In the game, there is only one type of user, which is the player. The main goal of the minigame is to complete the sequence lists by using the code matrix in a limited buffer size and time.

Initially, the game starts the countdown when the player picks an index in the first row of the matrix. This index will be the first element of the buffer. Then, the orientation changes from horizontal to vertical. The next step is to pick another index in the same column with the first chosen index. As the game continues, the orientation is changed during every step until the buffer reaches its limit or the time runs out. The game detects the desired sequences as they appear in the buffer which will automatically give the player the reward. There are two ways to lose the game: if no sequences are found within the buffer space or within the time limit, the player loses. The system is able to check the success/failure of the game when the player finishes the game. Since there are multiple solutions to the puzzle, the system does not check every movement of the player. The player will be given scores as rewards. The player will get a score of 200 per sequence. The maximum score can be 600, which is in the case three sequences are shown. The score will be shown as Player Score/Total Score. The player can undo moves until the game finishes, but the time will continue, as extending the time would make the game too easy and boring. The player cannot undo moves when the game has finished, as they can restart the game.

The system will be implemented as a form of a basic player interface in a game window where the player will interact with the game by clicking only. The system will not be a command-line interface because it’s easier for players to click rather than typing for the game. Moreover, the graphic user interface is aesthetic and clear. The game works as follows:

1. The player runs the game and a window is opened.
2. The game description and the button to choose a puzzle will be displayed.
3. After the player clicks the puzzle button, 40 different puzzles will be displayed on the window.
4. The game will begin officially after the player chooses one puzzle.
5. The timer will start the countdown when the player picks the first tile.
6. Each tile picked by the player will be displayed in the buffer, and the player can keep picking tiles until the game ends.
7. The game ends when the buffer size is full or the time is up.
8. The player will get a score when the game ends.
9. The player is given three choices: restart the game, return to the puzzles page to pick another puzzle or exit the game.

External URLs

- Hacking minigame explained in a YouTube video:
  https://www.youtube.com/watch?v=ROr1BBqnGoo
- A complete description of the rules of the minigame:
  https://www.rockpapershotgun.com/cyberpunk-2077-hacking-minigame-breach-protocol-explained
- The GitHub repository of the set of puzzles
  https://github.com/kyle-rader/breach/tree/main/puzzles/txt

Features

Author(s): All members

Functional features

ID	Short name	Description	Champion
F1	Puzzle Choice	The system shall support different values in the tiles, sequences, and buffer in the game. The player is allowed to choose from 40 different puzzles to before the game begins. Format of file: ● The first line contains the buffer size (4-7) ● The matrix size (5x5 or 6x6) ● The sequence lists (1-4)	
F2	User Interface	There are 5 main parts showing when the game begins: time countdown, code matrix, buffer, Sequence list, and score the player gets. Players pick tiles by clicking on the tile. When the tile is picked, the code on the tile will change to ‘[ ]’.	
F3	Tile Picking	Starting in the first row, the player can pick a tile. Next, the player can pick a tile in the same column as the first tile. The game keeps alternating between rows and columns in a similar fashion until the game is over. When the player chooses a tile, the tile will be updated to the buffer.	
F4	Timer	The timer starts counting down when the player picks the first tile. When there are only 10 seconds remaining, the color of the timer’s colour will change from black to red.	

F5	End/ Outcome	**End** The game ends when the player either wins or loses. The score will be shown as Player score/Total score to let the player know how well they perform. **Win** There are two situations in which a player wins: 1. If the player finds all sequences within the time and does not exceed the buffer size, they win the game completely and they will get the corresponding score. For every sequence found, 200 points are rewarded. 2. If the player finds one sequence list within the time and does not exceed the buffer size, the player gets 200 points. In this case, players partially win the game. **Loss** When the buffer is completely filled or the time is up and the player has not found any sequence, they lose the game and get 0 points.	
F6	Undo/ Restart	**Undo** During the game, the player can undo his picked tiles. The timer will not stop or extend the time. **Restart/Puzzle page/Exit** After the game, the player has three options: 1. Restart the game with the same puzzle. 2. Return to the puzzle picking pages to pick another puzzle. 3. Exit the game.	

Quality requirements

Author(s): All members

The following table contains the quality requirements deemed the most important for a Cyberpunk 2077 hacking minigame. These requirements ensure that the player can enjoy the game without any difficulties unrelated to the gameplay itself.

ID	Short name	Quality attribute	Description
QR1	Pre-game Validation	Reliability	The text file that describes the buffer size, code matrix, and sequence lists shall be read

			and parsed when the player picks it. The system should display an error message when there is a null object found.
QR2	Extension	Maintainability	The game is not restricted by the predefined set of 40 configurations. In the future, the game can be easily extended to contain more than the original 40 configurations.
QR3	Instantaneous Results	Responsiveness	Once the player makes a move in the hacking game, the result of the move shall be available in the next round. When the player picks one tile, the tile will be on the buffer in 2 seconds.
QR4	Input Validation	Usability	Any action clicked by the player will be checked whether the player is within the rule. If the player clicked on a tile that breaks the rule, nothing will change on the system. Only if the player clicks a valid tile, the tile will appear on the buffer.

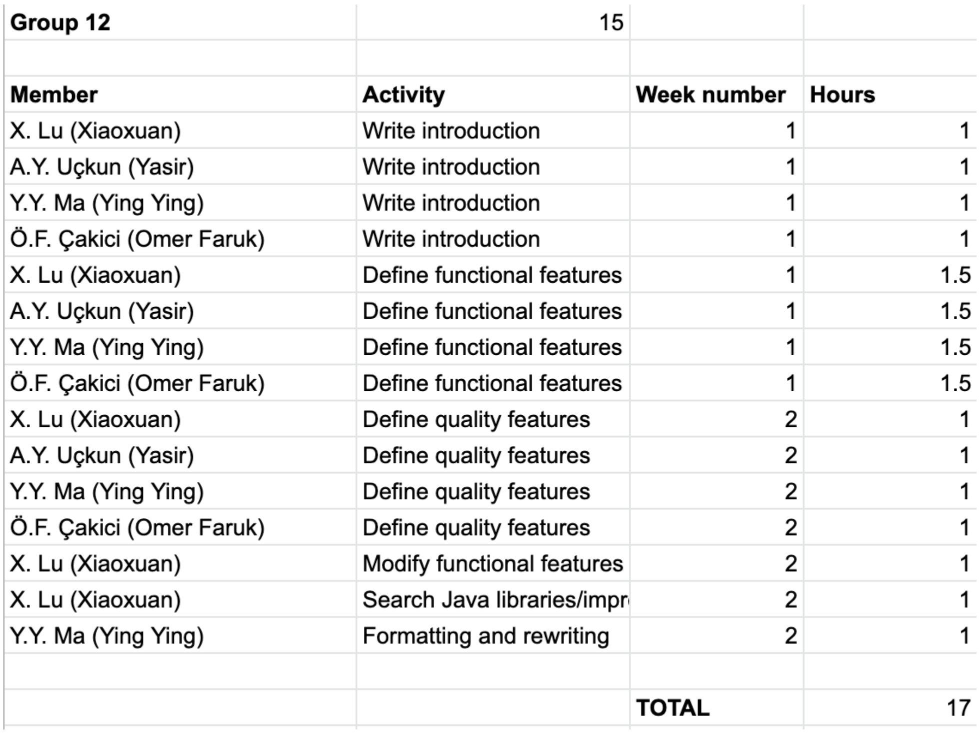
Java libraries

Author(s): Xiaoxuan Lu

JavaFX, JFoenix, Scene Builder for GUI
JavaFX and JFoenix are used to style the user interface of the system. We chose them among others because they can create a beautiful user interface. They can be used with Scene Builder which is simple and convenient to see the interface of the system. We can just drag and drop user interface design.

Apache Commons
Commons IO could be easier for inputting and outputting files. The library could be used as a helper for reading matrix, buffer size, and sequence lists in the text file. We will compare it with Java.io to see which one is easier and better to implement.

Time logs



Group 12	15		
Member	**Activity**	**Week number**	**Hours**
X. Lu (Xiaoxuan)	Write introduction	1	1
A.Y. Uçkun (Yasir)	Write introduction	1	1
Y.Y. Ma (Ying Ying)	Write introduction	1	1
Ö.F. Çakici (Omer Faruk)	Write introduction	1	1
X. Lu (Xiaoxuan)	Define functional features	1	1.5
A.Y. Uçkun (Yasir)	Define functional features	1	1.5
Y.Y. Ma (Ying Ying)	Define functional features	1	1.5
Ö.F. Çakici (Omer Faruk)	Define functional features	1	1.5
X. Lu (Xiaoxuan)	Define quality features	2	1
A.Y. Uçkun (Yasir)	Define quality features	2	1
Y.Y. Ma (Ying Ying)	Define quality features	2	1
Ö.F. Çakici (Omer Faruk)	Define quality features	2	1
X. Lu (Xiaoxuan)	Modify functional features	2	1
X. Lu (Xiaoxuan)	Search Java libraries/impr	2	1
Y.Y. Ma (Ying Ying)	Formatting and rewriting	2	1
		TOTAL	17