

Assignment 1

Team number: CyberpunkHackingMinigame - Team 1

Team members

Name	Student Nr.	Email
Irakliy Marsagishvili	2637597	i.marsagishvili@student.vu.nl
Ivan Ivanov	2672214	i.g.ivanov@student.vu.nl
Maxim Abramov	2673452	m.abramov@student.vu.nl
Man Chung Stephen Kwan	2631645	m.c.s.kwan@student.vu.nl

Introduction

Author(s): Irakliy Marsagishvili, Ivan Ivanov

Cyberpunk Hacking Minigame is a game that is inspired by CD Projekt's [Cyberpunk 2077](#) hacking minigame called Breach Protocol. In this game, we will allow the player to hack into the system by solving a puzzle which involves choosing a tile in a matrix and inserting it into the buffer according to the sequence. The game will also include some puzzles already, however the user would be able to customise their own puzzles and upload them to the game, where users can test their skills of creating complicated puzzles.

The main types of users will be:

- The Player: The user who plays the game according to the rules provided.
- The Creator: The user who has provided custom-made puzzles in the form of .txt files.

The main modules of the system will be:

- Puzzle: A choice of either a randomized or loadable puzzle will be presented to the player.
- UI: This will be the window where the game takes place. It will be a Graphical User Interface built with JavaFX and SceneBuilder. The inputs will be done by mouse only. The game will have a GUI version only as the team wanted to get as close to the looks of the original game.

The only entity in the game that is interactable is:

- The Code Matrix: This 6x6 matrix is made of tiles, which are hexadecimal symbols, that the player can select from a highlighted set of tiles in a row/column which then get inserted into the buffer.

The game will provide one type of customization to the Creator in the form of:

- Custom-made puzzles: The Creator will provide custom puzzles in the shape of standard Windows .txt files which will have two separate fields. The first one will be a set of hexadecimal symbols that will be parsed by the game and inserted in the Code Matrix's tiles. The second one will be a sequence of several other hexadecimal numbers which will be the sequence the player has to solve in order to win.

The game will be implemented in a Windows shell based window. Within the space set by that window the user(which we will refer to as *player* from now on) will interact with the game by the means of mouse inputs, the type of control used in the original game as well.

The game will generally work as specified below:

- The player will click on the .jar file and a window will open.
- The player will be presented with a choice of a randomized puzzle or a loadable one(from a .txt file).
- Dependent on the player's choice the game will load a type of puzzle.
- Instead of determining the difficulty by a sequence V1 until V3 like in the original game, the matrix has only one and only solution(I.E. V1 only), The emphasis lies on the matter of how fast the player is able to solve the puzzle. The faster the player is able to solve, the more points they obtain.
- The Code Matrix, our main playfield, will load on the left side of the screen, the buffer will show up on the top right and the timer will be below that.
- The game will wait for the user to provide an input in the form of a selected tile from the Code Matrix. The first input will be possible only in the rows of our Code Matrix.
- After choosing the first tile, the field of selectable tiles will be flipped from rows to columns, and the opposite will happen in the turn after that, etc.
- Each input provided will be added to the Buffer field, which in turn will decrease the Buffer's capacity by 1.
- While these actions are happening, the timer will count down from time set to 0.
- (Possible time-attack description)
- The game ends in Failure if the puzzle can't be solved, the timer reaches zero or the buffer gets filled up before solving a puzzle.
- The game ends in Success if the puzzle is solved(meaning that the given sequence has been solved correctly).
- Depending on the Success or Failure condition the player has reached the game will notify him by means of a Win/Lose screen showing up.

Author(s): Man Chung Stephen Kwan, Maxim Abramov

Functional features

ID	Short name	Description	Champion
F1	Timer	This game shall have a timer which limits the amount of time a player has to complete a stage.	<i>Irakliy Marsagishvili</i>

F2	Buffer	The player shall be able to copy selected tiles from the matrix into the buffer.	<i>Maxim Abramov</i>
F3	Code matrix	Code matrix is the central interactive object which shall contain tiles with values that the player should be able to collect and store inside the buffer.	<i>Irakliy Marsagishvili</i>
F4	Goal sequence	The main goal of the game is to match tiles in the buffer to the given sequence.	<i>Maxim Abramov</i>
F5	Tile selection	Each time the player selects a valid tile, the set of available tiles shall alternate between the vertical and horizontal lines containing the selected tile.	<i>Irakliy Marsagishvili</i>
F6	Sequence completion	If a selected value is not a continuation of the given sequence, the sequence shall not progress. If the player completes the sequence, the game shall end.	<i>Man Chung Stephen Kwan</i>
F7	User-game interaction	A player shall be able to use their mouse to interact with the game (left click to select a tile).	<i>Man Chung Stephen Kwan</i>
F8	Highlighting	Every currently available tile to click in the Code Matrix should be highlighted. In addition, if the player hovers over an available tile it should be highlighted with a different color.	<i>Man Chung Stephen Kwan</i>
F9	Hint indicator	The player should be able to hover over a tile and if its value occurs in the code sequence, the corresponding value should be highlighted.	<i>Ivan Ivanov</i>
F10	“Sequence unreachable” indicator	If a sequence cannot be constructed due to the position of the currently selected tile, the sequence should have an indicator reflecting the issue.	<i>Ivan Ivanov</i>
F11	“Out of memory” indicator	If a sequence cannot be constructed due to the player not having enough free space in the buffer, the sequence should have an indicator reflecting the issue.	<i>Maxim Abramov</i>
F12	Reward for speed	The player should receive an additional reward for completing the sequence faster.	<i>Ivan Ivanov</i>
F13	Quitting	The player shall be able to exit the game by clicking on the quit button.	<i>Ivan Ivanov</i>

F14	Loading a custom game	The user should have the option to load their own game in the form of a .txt file as the starting state.	<i>Irakliy Marsagishvili</i>
F15	Set of Puzzles	When the game has launched, the application shall generate a random puzzle from the set of pre-made puzzles.	<i>Irakliy Marsagishvili</i>

Quality requirements

Author(s): Man Chung Stephen Kwan, Ivan Ivanov

ID	Short name	Quality attribute	Description
QR1	Interface validation	Reliability	The interface will be rendered correctly when the game is initialized and will be interactive upon the events that have been taken by the player.
QR2	Pre-Game mechanics	Reliability	Once the game is initially run, the hexadecimal numbers(content of the matrix) will be loaded from a txt file along with the sequences that have to be done yet.
QR3	Developer friendly	Maintainability	The source code of the game will be written concise and comprehensible which results in a higher accessibility in case modification of the game has to be done by the initial original developer or another developer.
QR4	Instantaneous results	Responsiveness	Once the player makes a move in the hacking game, the result of the move shall be available within 0.2 seconds.
QR5	Input Reception	Usability	By clicking on the matrix by the player, one of the following game states occurs: -In progress -Game over -Completion
QR6	Interface	Usability	The game interface looks clean with adequate large buttons which enhances the feeling of navigation of the game.
QR7	Terminality	Reliability	The player is able to quit the game under any circumstances.

Java libraries

Author(s): Irakliy Marsagishvili, Man Chung Stephen Kwan, Maxim Abramov

[JavaFX](#)

A set of graphics and media packages that enables developers to design, create, test, debug and deploy rich client applications that operate consistently across diverse platforms. The graphical expressions will be realized along with the mouse click response (I.E. making it interactive) by using this library.

Tools

[Scene Builder](#)

Gluon's Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces. Users can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the layout that they are creating is automatically generated in the background. The result is an FXML file that can then be combined with a Java project by binding the UI to the application's logic.

We chose it among others because it allows us to make maintainable high-performance code.

Time logs

1	Team number:	CyberpunkHackingMinigame - Team 1	
2			
3	Member	Activity	Week number
4	Irakliy	Look into Java libraries	1
5	Ivan	Look into Java libraries	1
6	Maxim	Look into Java libraries	1
7	Stephen	Look into Java libraries	1
8	Irakliy	Meeting 1	1
9	Ivan	Meeting 1	1
10	Maxim	Meeting 1	1
11	Stephen	Meeting 1	1
12	Irakliy	Meeting 2	2
13	Ivan	Meeting 2	2
14	Maxim	Meeting 2	2
15	Stephen	Meeting 2	2
16	Irakliy	Quality attribute idea	2
17	Ivan	Quality attribute idea	2
18	Maxim	Quality attribute idea	2
19	Stephen	Quality attribute idea	2
20	Irakliy	Read example assignmen	2
21	Ivan	Read example assignmen	2
22	Maxim	Read example assignmen	2
23	Stephen	Read example assignmen	2
24	Irakliy	Write report session 1	2
25	Ivan	Write report session 1	2
26	Maxim	Write report session 1	2
27	Stephen	Write report session 1	2
28	Irakliy	Write report session 2	2
29	Ivan	Write report session 2	2
30	Maxim	Write report session 2	2
31	Stephen	Write report session 2	2
32			
33			
34		TOTAL	36