Assignment 1

Team number: Group 5
Team members:

Name	**Student Nr.**	**Email**
Irena Ristova	2642317	i2.ristova@student.vu.nl
Fouad Aljaber	2644243	f.aljaber@student.vu.nl
Emilija Baranauskaitė	2658783	e.b.baranauskaite@student.vu.nl
Iva Gunjača	2638025	i.gunjaca@student.vu.nl

Author(s): Emilija

Introduction

Cyberpunk Hacking minigame is not only a subtle feature of a new 2020 game but also a great way to test your logical abilities. Therefore, for the Software Design project, we decided to go on with this. Some game players hate it, and the others enjoy solving this logical puzzle. That is why we have decided to look more deeply into this game.

The original game consists of four primary components: six by six grid, sequence list, timer and buffer. At first, the player gets a list of possible sequences you can complete. The player has to pick tiles from the grid to form code sequences, that he or she at the beginning of the game. This process has a limited time. It starts ticking just when the player picks the first tile from the grid. Choosing tiles requires a specific logic to follow the player can only pick tiles in a particular column or row. It is possible to select a tile only from the same line as the previous tile you picked. If the last time you chose a tile from a row, the next move, you only can choose from a column. The player has to keep in mind that there exists a specific number of tiles, that he or she can choose. This number is called a “Buffer”. What is also important, the player cannot break the sequence up with an incorrect tile.

This game will be valuable for people who like playing the primary game Cyberpunk and want to master this minigame. It is also great for people who simply like this kind of brain teaser as their entertainment.

Our implementation will be designed in the graphical user interface. We thought that in CLI m, we will not be able to replicate this game that well because the original game indeed is created with GUI. Moreover, we wanted to get a practical touch of how this game works, not only from a logical perspective but also from a designer's point of view.

From this project, we expect not only to do our best replicating the original Cyberpunk hacking minigame but also to learn how working and creating such a game in a group feels like. We aspire to create an analogy which also has more than one level. Moreover, we want to make it possible for the player to choose the size of the buffer, save the game, undo/redo their moves.

Features

Author(s): Iva

Functional features

Cyberpunk hacking minigame is a puzzle game that tests users ability to make a sequence from an offered matrix, with a set of rules, limited time and limited amount of moves. Therefore, most crucial functional features in cyberpunk hacking minigame are grid, buffer, set of sequences, timer and choosing the tiles.

Help button can be also considered a crucial feature, for a user that has never played a game it is the only way to understand how a game works, but it is not a necessity for creating a functional game.

Furthermore, to make this game enjoyable for a player some kind of reward needs to be given after successfully finding a sequence, by implementing awards, the game offers stimulation to the user and feeling of achievement. It pushes the user to do repetitive puzzles and lowers the possibility that the user will get bored of the game, because even if the game becomes repetitive and boring the user will try to play more to get more rewards such as coins, higher score or maybe power-ups. This reward based system is what most games today are based on and its main reason why they are so addictive, even the most repetitive mobile games such as Subway Surfers see huge success as you receive a reward as soon as you do a certain task.

Levels are one more feature that makes a game more enjoyable. Making sequences harder to make, longer or having multiple creates a feeling of progress and makes the game more interesting. Leveling also offers a user ability to slowly gain skill from most

simple puzzles to harder ones. Giving a user randomly generated puzzles without considering how easy it is to solve them results in anger and frustration or boredom, which can result in bad reviews of the game.
Option to undo moves offers a player a chance to fix a mistake that they made accidentally or on purpose, however it should be limited to once or twice per level as undoing a move removes challenge from the game.

ID	Short name	Description	Champion
F1	Grid	A 5x5 tiles grid that contains a numeric puzzle	Irena
F2	Buffer	The game should have a buffer of limited length in which the player's choice of sequences will be stored.	Irena
F3	Set of sequences	A list of possible sequences the user can complete, and the reward one will achieve.	Irena
F4	Awards	The user should receive a set of rewards corresponding to the completed set of sequences.	Fouad
F4	Choosing a tile	The user should be able to click on tiles and make a sequence to fill up the buffer.	Fouad
F5	Timer	The game should display a real-time timer that has a countdown.	Iva
F6	Help button	Players can press the help button that shows rules of the game.	Iva
F7	Undo	The player shall be able to undo one or more moves.	Emilija
F8	Levels	Puzzles get more complex as player plays	Emilija

Author(s):Irena

Quality requirements

To create an enjoyable puzzle game main concern is users enjoyment, therefore we are focusing on reliability and usability. While creating a game, or any program in general, developers have a picture of how a program should be used, however users will often not follow that path. By explaining the rules, highlighting current state and limiting users interaction, such as setting a minimum size buffer, we are guiding the user how to use the program and not allowing breaking the game or making the levels impossible.
Furthermore, performance and maintainability are also highly important parts for user experience, lag and unexpected bugs are something that will most certainly drive a user away from playing our game.
To provide availability on a variety of different platforms, the game should be able to be run from a BASH terminal.
We don't focus too much on security as we are not making this game for profits but for entertainment.

ID	Short name	Quality attribute	Description
QR1	Success/fail Check	Reliability	When the player picks a sequence the system shall be able to check the success/fail of the moves and assign the corresponding reward.
QR2	Extensible buffer	Maintainability	The buffer shall be extendable in terms of size of the number of tiles that can be picked.
QR3	Instantaneous results	Performance	Once the player makes a move in the hacking game, the result of the move shall be visible immediately.
QR4	Explanation of rules	Usability	The user shall be presented with a description to follow, making it easy to play.
QR5	Minimum size of buffer	Reliability	The buffer size will have to be a minimum of the size of the sequence.
QR6	Highlight current state	Usability	The game should highlight the column/row the player is to choose a tile from.
QR7	Cross Platform	Availability	The game shall run from a BASH terminal

Author(s): Fouad

Java Libraries

JavaFX

Used for implementing the graphical user interface of the game.
It is our library of choice given its popularity and ease of use.

Time4J

This library gives useful time tricks that can be used for calculating and measuring the game duration and can also be used for the timer.

FastJSON

Simple and fast json processor library. Will be used to keep the data of the game for future use. Such as the level of the player.