**Project Template**

**Project Report: Group Number**

Friday, October 20, 2023

## Table of contents

[**1 Introduction** **2**](#_Toc3329)

[**2 JavaCraft’s Workflow** **2**](#_Toc3330)

[**3 Functionality Exploration** **2**](#_Toc3331)

[**4 Finite State Automata (FSA) Design** **3**](#_Toc3332)

[**5 Git Collaboration & Version Control** **3**](#_Toc3333)

[**6 Extending the Game Code (For Final Submission)** **3**](#_Toc3334)

[**7 Interacting with Flags API (For Final Submission)** **3**](#_Toc3335)

[**8 Conclusion (For Final Submission)** **3**](#_Toc3336)

[**9 Appendix** **3**](#_Toc3337)

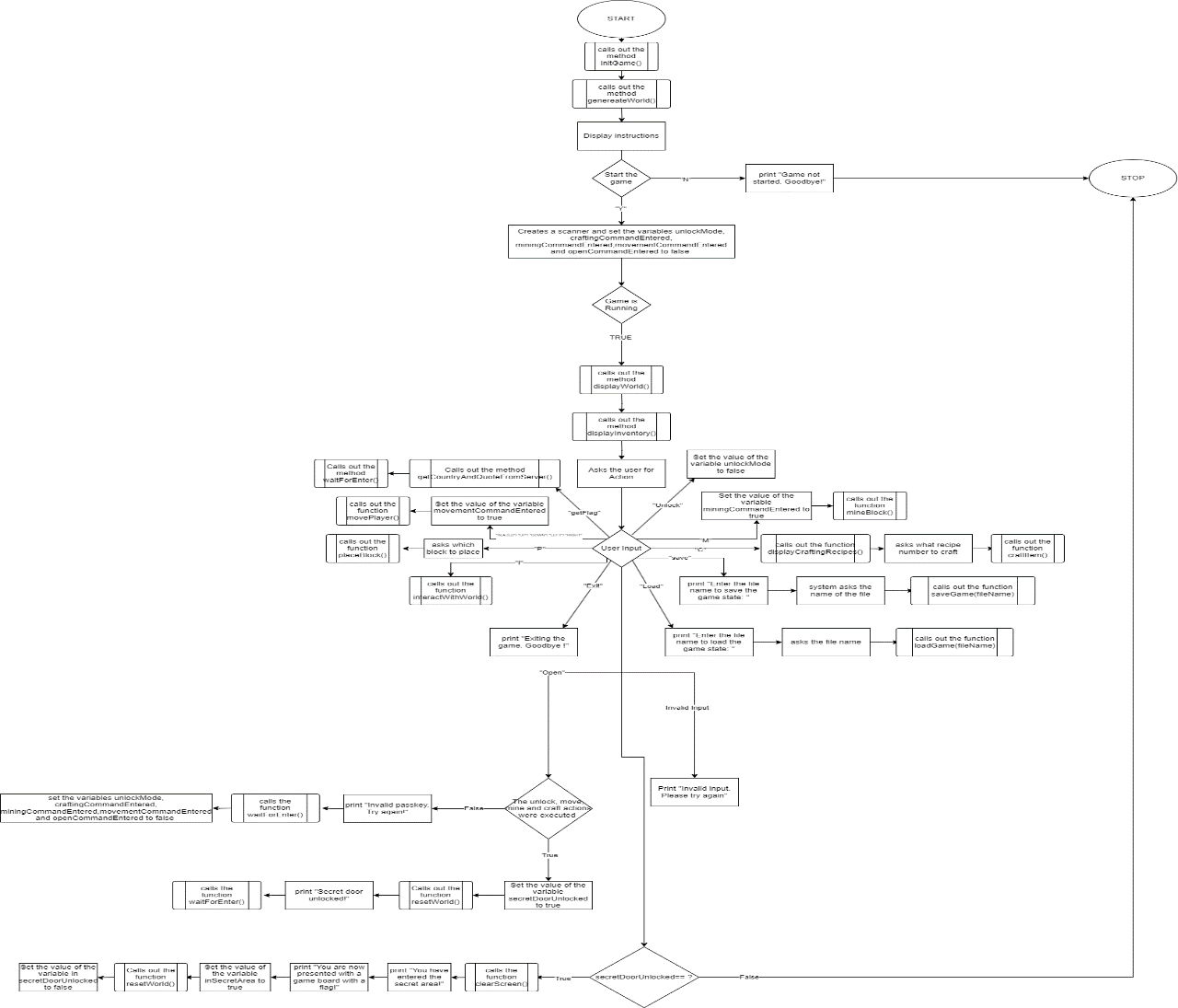
[**10 References** **3**](#_Toc3338)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | | Details | | |
| Group Name | | Group79 | | |
| Group Number | | 79 | | |
| TA | | [Name] | | |
| Student Name | | Student ID |
| Hagenbeek Mika | | [Student ID] |
| Fernandes Kennedy | | [Student ID] |
| Vieversys Gustas | | [Student ID] |
| Zerbi Liam | | I6368807 |

# Introduction

[Provide a brief introduction to the report here.]

# JavaCraft’s Workflow

* Flowchart For Game: 
* Pseudocode For Game: [Provide pseudocode here]

# Functionality Exploration

List of key functionalities explored:

|  |  |  |
| --- | --- | --- |
| No. | Function Name | Description |
| 1 | Main | This is the entry point of the program. It initializes the game, generates the game world, and handles user input to start the game or exit. |
| 2 | initGame | this function initializes the game loading up any initial information such as the welcome page as well as basic instructions on how to start up the program |
| 3 | generateWorld | this function generates a world, filling up the 2D array with different blocks based on a random value generator. Based on the value, each ‘block’ is filled with a certain element such as leaves |
| 4 | displayWorld | this method goes through the array and displays a certain character based on what value is stored in that part of the array. This repeats for each value until the boundaries ‘worldHeight’ and ‘worldWidth’ are reached |
| 5 | getBlockSymbol | this method ‘gets’ the block symbol depending on what value is stored within that space in the array. |
| 6 | getBlockChar | Returns the ASCII character for a given block type used in crafting recipes. |
| 7 | startGame | this function starts the game, this means that after the user inputs ‘y’, the game begins and the function startGame calls upon functions such as displayWorld and others in order to show and display the game |
| 8 | movePlayer | Moves the player's position based on the input direction (WASD or arrow keys). |
| 9 | mineBlock | Makes the player mine and obtain the block placed in the position of the player, if there is indeed a block this function will notify that the action has been successful and the type of block mined. In the same way, if there is no block that can be mined the system will tell it to the player |
| 10 | displayInventory | The system will display a section which is the player’s inventory, containing every item and block the player has obtained, specifying the quantity of each |
| 11 | placeBlock | Allows the player to place a block from their inventory at their current position. |
| 12 | displayCraftingRecipes | The system will display a section which is the player’s inventory, containing every item and block the player has obtained, specifying the quantity of each |
| 13 | craftItem | Crafts an item based on the selected recipe. |
| 14 | interactWithWorld | Allows the player to interact with different block types in the game world, adding them to the inventory. |
| 15 | saveGame | Saves the current game state, including world data, player position, and inventory, to a file. |
| 16 | loadGame | Loads a saved game state from a file and restores it to continue the game. |
| 17 | lookAround | Displays a limited view of the nearby blocks from the player's current position. |
| 18 | getCountryAndQuoteFromServer | Makes an HTTP request to a server to retrieve and display a country name and a quote. |
| 19 | waitForEnter | Pauses the game and waits for the player to press Enter. |
| 20 | resetWorld | Resets the game world to its initial state for the secret door unlock sequence. |
| 21 | clearScreen | Clears the console screen, providing a visual update for the player. |
| 22 | fillInventory | this method first clears the player’s inventory, then it loops and ‘fills up’ the inventory of the player with the certain block type. This loop ends once inventory is full. |
| 23 | generateEmptyWorld | generates a new map with the a custom width and height ,as well as the default  red block, white block and blue block amount, it also divides the height into 3 equal parts |
| 24 | getBlockName | this function gets the block name depending on the block type, it returns wood if the block type is wood etc etc |
| 25 | getCraftedItemFromBlockType | Converts a block type to its corresponding crafted item |
| 26 | getCraftedItemName | Returns the name of a crafted item based on its type. |
| 27 | craftWoodenPlanks | this method creates the item WoodenPlanks and adds it to your inventory. To do this you need to have the blocks needed to build it, that being 2 wood blocks. It also removes 2 wood blocks from your inventory. |
| 28 | craftStick | this method creates the item stick and adds it to your inventory. To do this you need to have the blocks needed to build it, that being 1 wood block. It also removes 1 wood block from your inventory. |
| 29 | craftIronIngot | this method creates the item ironIngot and adds it to your inventory. To do this you need to have the blocks needed to build it, that being 3 iron ore. It also removes 3 iron ore from your inventory. |
| 30 | getBlockTypeFromCraftedItem | depending on what type of block the user has the code lets the user print either wooden\_planks, crafted\_sticks or crafted\_iron\_ingot |
| 31 | getCraftedItemFromBlockType | Converts a block type to its corresponding crafted item. |
| 32 | displayLegend | this function allows the player to understand which block is represented by what symbol on the map. It shows colored text and says what type of block it shows |
| 33 | removeItemsFromInventory | this method removes a certain item as well as the amount you remove from a player's inventory |
| 34 | inventoryContains | using the name of the item as a parameter, this function checks if the inventory contains that item. |
| 35 | getBlockColor | this method is used in order to show the color of each block. For example, air is blank, wood is red, etc etc. |
| 36 | addCraftedItem | Adds a crafted item to the player's list of crafted items. |

Note: Provide flowchart and pseudocode for at least 15 functions in the Appendix.

# Finite State Automata (FSA) Design

* Secret Door Logic Analysis: [Describe the secret door’s functionality]
* FSA Illustration & Description: [Attach FSA diagram]

# Git Collaboration & Version Control

* Repository Link: <https://gitlab.maastrichtuniversity.nl/bcs1110/javacraft/-/tree/group79?ref_type=heads>
* Branch Details: List branch names and corresponding members • Changes & Conflicts: Discuss how changes and conflicts were handled.

# Extending the Game Code (For Final Submission)

[Provide details on the new block types, craft recipes, and their integration into the game. Include code snippets where appropriate]

[Provide Java code here]

# Interacting with Flags API (For Final Submission)

[Details on Flags API exploration and flag rendering on the grid.]

# Conclusion (For Final Submission)

[Provide a summary of achievements, challenges, and learnings.]

# Appendix

Include any additional pseudocode, flowcharts, or supplementary material.

# References

1. Source Name - Description
2. …