JavaCraft Project

BCS1110, Introduction to Computer Science

Group 11

Full Name	Student ID
Long Luong	I6359380
Élisa Donéa	I6356213
Chris Munteanu	I6344912
Alexia Raportaru	I6355814

Professors: Dr. Ashish Sai, Dr. Thomas Bitterman

Maastricht, October 3, 2023

Contents

1 Introduction **1**

2 JavaCraft's Workflow **2**

3 Functionality Exploration **3**

4 Finite State Automata (FSA) Design **4**

5 Git Collaboration & Version Control **5**

6 References **6**

A Appendix: pseudocode and flowcharts **7**

1 | Introduction

JavaCraft is a a multifaceted text-based Java game inspired by Minecraft. The game is a relatively complex Java program that brings over 35 functions to create a diverse gameplay experience. This project is an academic exercise in computer science, logical thinking, and collaboration. Working in teams of four, we used our creativity, analytical skills, and technical skills to analyse and expand upon the existing JavaCraft game.

2 | JavaCraft's Workflow

In order to fully understand the mechanism of the game, a flowchart of the entire game is provided below. The flowchart is accompanied by pseudocode.
Insert flowchart of game Insert pseudocode for game

3 | Functionality Exploration

Table 3.1: A table that describes functions used in javacraft

No.	Function Name	Description
1	`void generateWorld`	assigns integer to every tile of the world
2	`void initGame`	creates world with width `worldWidth` and height `worldHeight`
3	`void main`	main function
4	`void startGame`	starts the game
5	`void movePlayer`	moves player horizontally or vertically
6	`void mineBlock`	mines block player is on if block is not air
7	`String getBlockSymbol`	returns symbol of `blockType`
8	`void resetWorld`	clears the world and sets player position in middle
9	`void generateEmptyWorld`	generates an empty world
10	`void clearScreen`	clears terminal
11	`void lookAround`	prints out adjacent squares to player
12	`void fillInventory`	completely fills up inventory of player
13	`void displayLegend`	displays a legend of what each tile represents
14	`void displayWorld`	prints out all tiles of the world
15	`void displayInventory`	prints out obtained items & crafted items
16	`void loadGame`	loads the game from file `fileName`
17	`void saveGame`	saves the game in file `fileName`
18	`void interactWithWorld`	interacts with item player is standing on
19	`void addCraftedItem`	adds item `craftedItem` to array `craftedItems`
20	`void removeItemsFromInventory`	removes item `item` `count` times from inventory
21	`boolean inventoryContains`	returns boolean of whether `item` is in inventory
22	`void placeBlock`	places block `blockType` at player position
23	`void displayCraftingRecipes`	prints out available crafting recipes
24	`void craftItem`	crafts an item based on argument `recipe`
25	`void craftIronIngot`	crafts an iron ingot
26	`void craftStick`	crafts a stick
27	`void waitForEnter`	waits for operator to press Enter
28	`String getBlockTypeFromCraftedItem`	returns integer of `craftedItem`
29	`String getCraftedItemFromBlockType`	returns integer of `blockType`
30	`String getBlockName`	returns name of `blockType`
31	`String getBlockColor`	returns color of `blockType`
32	`String getCraftedItemName`	returns name of `craftedItem`
33	`String getCraftedItemColor`	returns color of `craftedItem`
34	`char getBlockChar`	returns char of `blockType`
35	`void craftWoodenPlanks`	crafts wooden planks
36	`void getCountryAndQuoteFromServer`	makes HTTP request and writes data to server and prints country and quote

4 | Finite State Automata (FSA) Design

Secret Door Logic Analysis: Describe the secret doors functionality • FSA Illustration and Description: Attach FSA diagram

5 | Git Collaboration & Version Control

The link to the JavaCraft branch can be found here: https://gitlab.maastrichtuniversity.nl/bcs1110/javacraft/-/tree/group11

All the files of the project can be found in the repository.
Everything was done on one branch called group11. There were no conflicts during the process.

6 | References

A | Appendix: pseudocode and flowcharts

This appendix provides the full blocks of pseudocode and its flowcharts.