# JavaCraft Project Report

**Project Report: 36**

Sunday, October 22, 2023

## Table of contents

| Attribute | Details |
|---|---|
| Group Name | Group 36 |
| Group Number | 36 |
| TA | Sam Goldie |

| Student Name | Student ID |
|---|---|
| R. Y. Avarzaman | i6356849 |
| D.H.F. Wicker | i6350383 |
| E. Silva Chagas | i6352037 |
| D. Iacovone | i6363887 |

## 0. Overview of Who Did What

| Task | Students |
|---|---|
| Introduction | E. Silva Chagas |
| JavaCraft Workflow Flowchart | R. Y. Avarzaman, E. Silva Chagas |
| JavaCraft Workflow Pseudocode | D.H.F. Wicker |
| Functions Description | R. Y. Avarzaman, E. Silva Chagas, D. Iacovone, D.H.F. Wicker |
| Functions Flowcharts | R. Y. Avarzaman, E. Silva Chagas, D. Iacovone, D.H.F. Wicker |
| Functions Pseudocode | R. Y. Avarzaman, E. Silva Chagas, D. Iacovone, D.H.F. Wicker |
| Secret Door Logic Analysis | R. Y. Avarzaman, D. Iacovone |
| FSA Illustration and Description | R. Y. Avarzaman, E. Silva Chagas, D. Iacovone, D.H.F. Wicker |
| Interacting with the Flags API | R. Y. Avarzaman, D.H.F. Wicker |
| Extending the game's code | R. Y. Avarzaman, E. Silva Chagas, D. Iacovone, D.H.F. Wicker |
| Writing the project report | R. Y. Avarzaman, E. Silva Chagas, D.H.F. Wicker |

## 1 Introduction

In the following report, our team delved into a detailed overview of our exploration of JavaCraft, a terminal based adventure game written in Java.

We began by sharing our understanding of the general game's workflow through a straightforward, visual-friendly flowchart diagram. We then worked on writing pseudocode for all the game's core functionality, which can be found in section 2 of the report.

After a general understanding of the game's workflow had been achieved, we were free to dive into JavaCraft's functions. Hence, we collaborated in making a succinct description of each of the game's functions. Furthermore, we then explored eighteen functions in more depth, developing flowcharts and pseudocode for each of them, which can be found in the Appendix.

An important component of the game was its secret door functionality. We analyzed the logic behind it before developing a Deterministic Finite Automata.

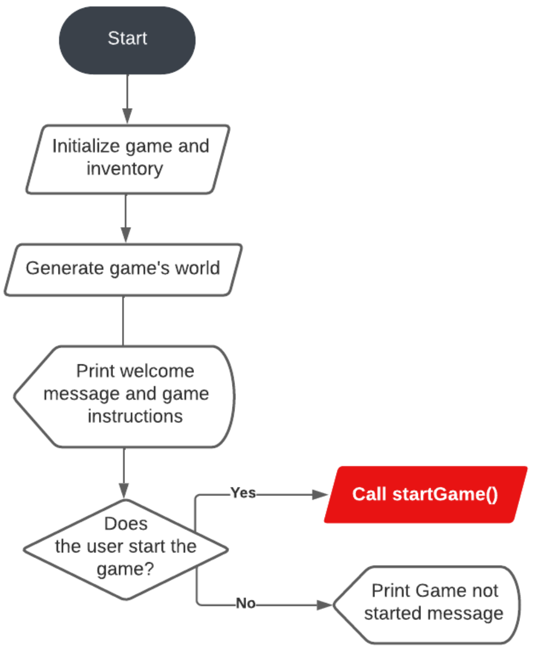
After making sure everyone was comfortable working with the version control software "git" and that everyone was able to make commits without error, we began extending the game's code, implementing 3 new blocks as well as a new crafting recipe.

Lastly, we completed the implementation of the `getCountryAndQuoteFromServer` method to interact with the Flags API and to retrieve the flag of the Philippines (categorized as hard), which we then drew by changing the implementation of the `generateEmptyWorld` method.

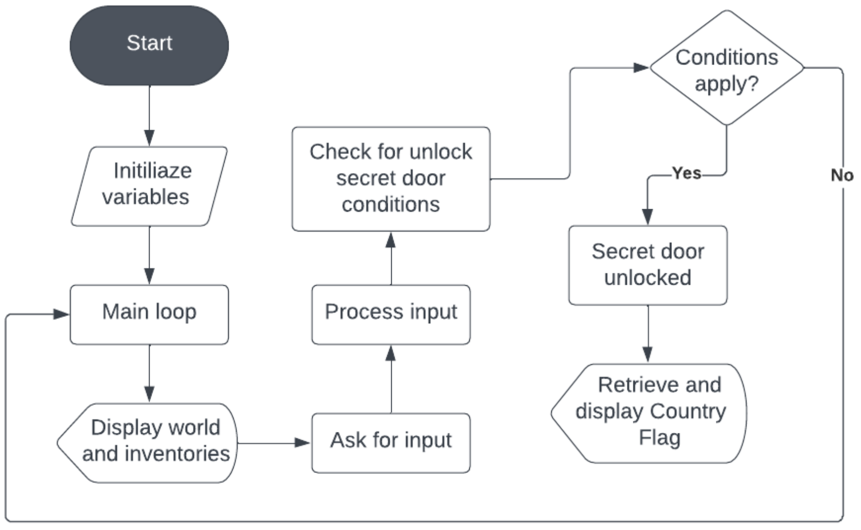
## 2 JavaCraft's Workflow

### Game's Flowchart

**main()**


Start
Initialize game and inventory
Generate game's world
Print welcome message and game instructions
Does the user start the game?
Yes
Call startGame()
No
Print Game not started message

**startGame()**



## Game’s Pseudocode

```
FUNCTION main()
    CALL initGame()
    CALL generateWorld()
    PRINT Welcome message and instructions legend
    AWAIT User input and save to startGameChoice
    IF startGameChoice is "Y" THEN
        CALL startGame()
    ELSE
        PRINT Game not started message
    ENDIF
ENDFUNCTION
```

```
FUNCTION startGame()
    SET unlockMode to false
    SET craftingCommandEntered to false
    SET miningCommandEntered to false
    SET openCommandEntered to false

    WHILE true
        CALL clearScreen()
        CALL displayLegend()
        CALL displayWorld()
        CALL displayInventory()
        PRINT Possible commands legend
        AWAIT user input and save to input

        IF input is w, up, s, down, a, left, d or right THEN
            IF unlockMode is true THEN
                SET movementCommandEntered to True
            ENDIF
            CALL movePlayer() with input

        ELSE IF input is m THEN
            IF unlockMode is true THEN
                SET miningCommandEntered to true
            ENDIF
            CALL mineBlock()

        ELSE IF input is p THEN
            CALL displayInventory()
            PRINT message to ask block selection
            AWAIT user input and save to recipe
            CALL craftItem() with recipe

        ELSE IF input is i THEN
            CALL interactWithWorld()

        ELSE IF input is save THEN
            PRINT message to ask for file name
            AWAIT user input and save to fileName
            CALL saveGame() with fileName

        ELSE IF input is load THEN
            PRINT message to ask for file name
            AWAIT user input and save to fileName
            CALL loadGame() with fileName

        ELSE IF input is exit THEN
            PRINT goodbye message
            BREAK WHILE LOOP

        ELSE IF input is unlock THEN
            SET unlockMode to true

        ELSE IF input is getflag THEN
            CALL getCountryAndQuoteFromServer()
            CALL waitForEnter()

        ELSE IF input is open THEN
            IF unlockMode, craftingCommandEntered, miningCommandEntered
            AND movementCommandEntered are all true THEN
                SET secretDoorUnlocked to true
                CALL resetWorld()
                PRINT secret door unlocked!
                CALL waitForEnter()
            ELSE
                PRINT invalid passkey
                CALL waitForEnter()
                SET unlockMode to false
                SET craftingCommandEntered to false
                SET miningCommandEntered to false
                SET openCommandEntered to false
        ELSE
            PRINT invalid input
        ENDIF

        IF unlockMode is true THEN
            IF input is c THEN
                SET craftingCommandEntered to true
            ELSE IF input is m THEN
                SET miningCommandEntered to true
            ELSE IF input is open THEN
                SET openCommandEntered to true
            ENDIF

        ENDIF

        IF secretDoorUnlocked is true THEN
            CALL clearScreen()
            PRINT message to welcome user to secret area
            SET inSecretArea to true
            CALL resetWorld()
            SET secretDoorUnlocked to false
            CALL fillInventory()
            CALL waitForEnter()
        ENDIF
    ENDWHILE
ENDFUNCTION
```

## 3 Functionality Exploration

List of key functionalities explored:

| No. | Function Name | Description |
|---|---|---|
| 1 | **initGame** | Initializes the game and its inventory, setting width and height parameters and placing the players in the middle of the world. |
| 2 | **generateWorld** | Generates the game's world, placing random blocks in it. |
| 3 | **displayWorld** | Prints the layout of the game's world. |
| 4 | **getBlockSymbol** | Assigns colors to the blocks. |
| 5 | **getBlockChar** | Assigns the symbol of the blocks. |
| 6 | **startGame** | Initializes the game and resets everything using all the functions to make that possible. |
| 7 | **fillInventory** | Fills your inventory with 100 blocks of 4 possible block types (wood, iron_ore, leaves, and stone). |
| 8 | **resetWorld** | The function is called when the secret door is opened. It clears the world of the initial blocks. |
| 9 | **generateEmptyWorld** | Updates the world with blocks to display the Dutch flag. |
| 10 | **clearScreen** | It clears the terminal based on the OS. |
| 11 | **lookAround** | When the user inputs the command "look", this function prints surrounding blocks based on the position of the player. |
| 12 | **movePlayer** | Moves the player depending on the input ("W","A","S","D" and the arrows). |
| 13 | **mineBlock** | Checks if the block the player is on is not AIR. If that condition is met, the block type is added to inventory and replaced in the world with AIR. |
| 14 | **placeBlock** | Place a chosen block type at the user's position, if the user's inventory has that block type. |
| 15 | **getBlockTypeFromCraftedItem** | Returns block type of the crafted item. |
| 16 | **getCraftedItemFromBlockType** | Returns crafted item of the block type. |
| 17 | **displayCraftingRecipes** | Displays the different options of crafting recipes on the terminal. |
| 18 | **craftItem** | Calls the chosen crafting recipe's method |

| | | |
|---|---|---|
| 19 | **craftWoodenPlanks** | Crafts wooden planks with 2 woods. |
| 20 | **craftStick** | Craft a stick with one wood. |
| 21 | **craftIronIngot** | Crafts an iron ingot from 3 iron ore. |
| 22 | **inventoryContains** | Returns true if the inventory has the inputted amount of the inputted item, otherwise false. |
| 23 | **removeItemsfromInventory** | Removes items used for crafting from the inventory. |
| 24 | **addCraftedItem** | Adds the crafted item to the inventory. |
| 25 | **interactWithWorld** | Lets users interact with the world by gathering different items. |
| 26 | **saveGame** | Saves the current game state data to a selected file. |
| 27 | **loadGame** | Loads game state data from a file into the game's program. |
| 28 | **getBlockName** | Returns block types names. |
| 29 | **displayLegend** | Displays a list of all the types of blocks that you can have. |
| 30 | **displayInventory** | Displays user inventory. |
| 31 | **getBlockColor** | Returns the color of the different block types. |
| 32 | **waitForEnter** | This function is called usually after another function is finished (e.g. after mineBlock). This function waits for the player to press the Enter key. |
| 33 | **getCraftedItemName** | It gets the name of the crafted item depending on its type. It's used in the function displayInventory to print the names of all the crafted items' types. |
| 34 | **getCraftedItemColor** | It gets the color of the crafted item. It's used in the displayInventory to print the color of the crafted item. |
| 35 | **getCountryAndQuoteFromServer()** | Uses a try-catch to connect to an API Server to get a Country and a Quote. If the connection doesn't go well it will show an Error. |

Flowcharts and pseudocode for functions **1**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **12**, **14**, **15**, **21**, **23**, **25**, **29**, **30**, **31**, **32** are provided in the Appendix.

# 4 Finite State Automata (FSA) Design

## Secret Door Logic Analysis

In order to open the secret door, the user needs to type "unlock" first. Then, use the move command (W,A,S,D), mine command, craft command in any combination. All the commands must appear at least one, with exception to the unlock command. Lastly, the user needs to type "open" to open the secret door.

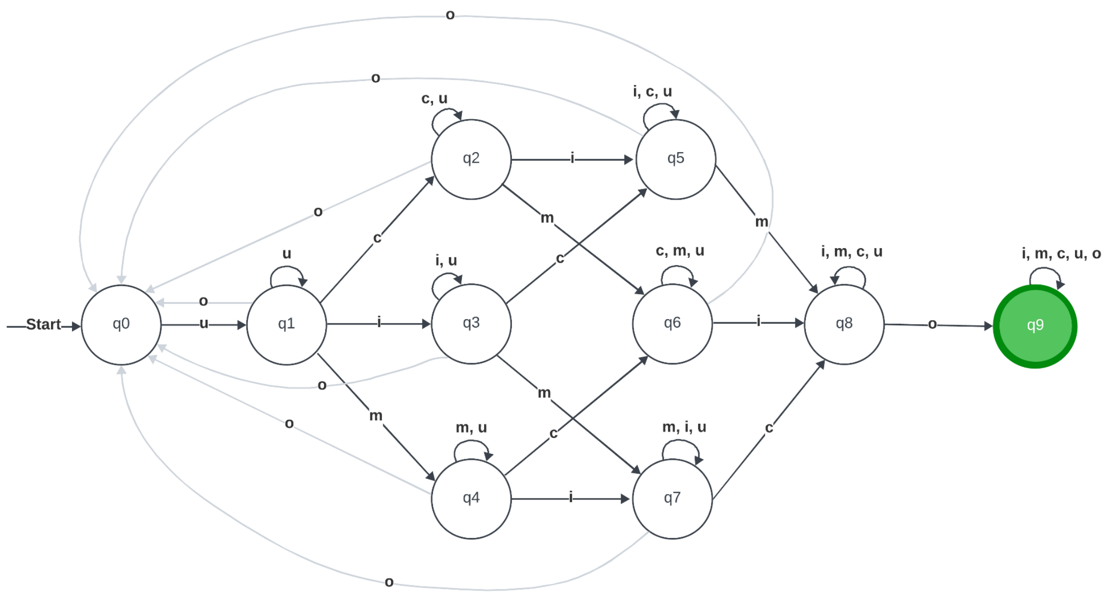
## FSA Illustration & Description

**`D = { Q, Σ, δ, q0, F }`**

- Q is the set of states: **`Q = { q0, q1, q2, q3, q4, q5, q6, q7, q8, q9 }`**
- Σ is the alphabet: **`Σ = { u, m, c, i, o }`**, where
    - u: unlock command
    - m: move commands (W, A, S, D)
    - c: craft command
    - i: mine command
    - o: open command
- L is the language: **`L = { w in Σ* | first character is u, then m, c, i, u in any combination, where m, c, i must appear at least once, and lastly o }`**
- **δ** is the transition function (defined on page 9)
- **q0** is the starting state
- F is the set of accepting states: **`F = { q9 }`**

An example of an accepted string would be "**`umcio`**", while an example of a string **not** accepted by the FSA would be "**`moccio`**".

FSA Design:

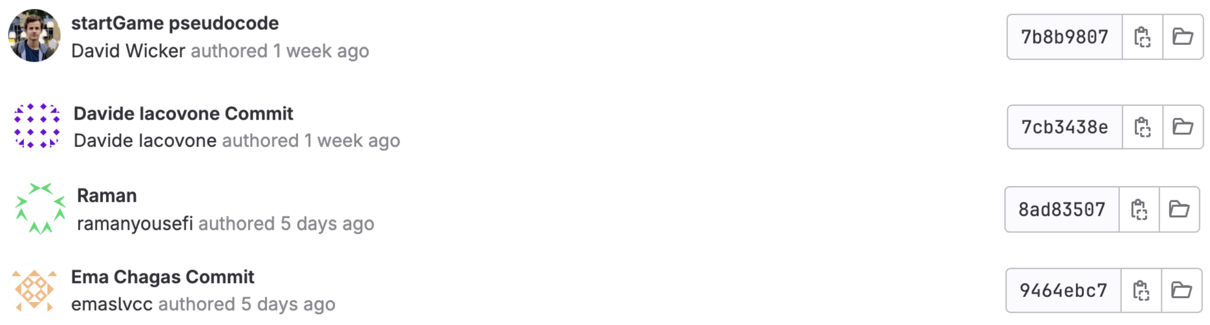


FSA Transition function:

| | u | m | c | i | o |
|---|---|---|---|---|---|
| q0 | q1 | q0 | q0 | q0 | q0 |
| q1 | q1 | q4 | q2 | q3 | q0 |
| q2 | q2 | q6 | q2 | q5 | q0 |
| q3 | q3 | q7 | q5 | q3 | q0 |
| q4 | q4 | q4 | q6 | q7 | q0 |
| q5 | q5 | q8 | q5 | q5 | q0 |
| q6 | q6 | q6 | q6 | q8 | q0 |
| q7 | q7 | q7 | q8 | q7 | q0 |
| q8 | q8 | q8 | q8 | q8 | q9 |
| q9 | q9 | q9 | q9 | q9 | q9 |

## 5 Git Collaboration & Version Control

A list of all the team members' first git commit to the `Group36` branch of the javacraft repository.



**startGame pseudocode**
David Wicker authored 1 week ago
7b8b9807

**Davide Iacovone Commit**
Davide Iacovone authored 1 week ago
7cb3438e

**Raman**
ramanyousefi authored 5 days ago
8ad83507

**Ema Chagas Commit**
emaslvcc authored 5 days ago
9464ebc7

## 6 Extending the Game Code

When extending the game code, we implemented three new blocks:

- Diamond Ore
- Gold Ore
- TNT → This block cannot be crafted or mined by the player, but when the player steps on it, their inventory is cleared and an area of 3x3 around the player's position is emptied to represent an explosion.

We also implemented one crafting recipe, the Iron Sword, which can be crafted with 1 Wood block and 2 Iron Ore blocks. To implement such features, we had to push changes to different functions, such as `generateWorld`, `getBlockSymbol`, `getBlockChar`, `interactWithWorld`, `getBlockName`, `placeBlock` and others. We also had to implement a new function, `craftIronSword`.

In particular, to implement the TNT feature, we had to push changes to the `startGame` function: whenever the player moves, the function had to check whether the new player position matched with the position of a TNT block, in which case the TNT's functionality would be applied.

## 7 Interacting with Flags API

**Retrieving the flag from API**

In order to communicate with the Flags API, we had to complete the implementation of the `getCountryAndQuoteFromServer` function, which uses the Java.Net package to send an HTTP request to API endpoint https://flag.ashish.nl/get_flag with method POST and including in the payload the group number and **difficulty Hard**.

When running the function, we retrieved the flag of the Philippines, which we then started plotting by altering the function `generateEmptyWorld` and the function `displayWorld`.

**Plotting the flag**

- We divided the flag in 2 halves, which we approached in separate for loops.
- For the upper half of the flag, we looped over every row and increased the number of white blocks per row at every iteration, while the remaining of the row blocks were filled with blue.
- For the lower half we used the same approach, but at every iteration the number of white blocks would decrease.
- Finally, to implement the golden sun and stars, we simply replaced some of the white blocks with newly implemented golden blocks.

The function which plots the flag of the Philippines can be found on our GitLab branch `Group36`.
A picture of the flag can be found in the appendix (9.19).

## 8 Conclusion

Throughout our project we encountered various challenges that we were able to overcome as a team. These, along with the achievements we accomplished and the learnings we acquired, are listed below.

**Achievements**

- Collaborating as a group through git version control
- Completing the FSA after a number of attempts
- Coding the algorithm to plot the flag of the Philippines
- Writing a concise but extensive project report

**Challenges**

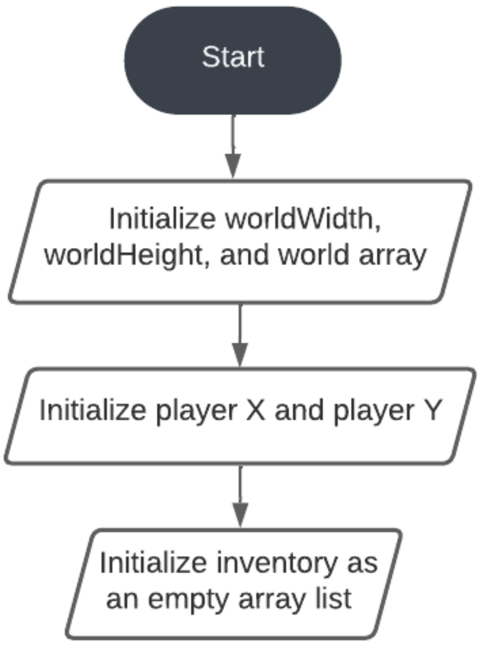
- Managing git merge conflicts
- Distributing work amongst the team members

**Learnings**

- Coordinating git pushes in order to avoid conflicts
- Contributing to a team project
- Writing clean and consistent pseudocode

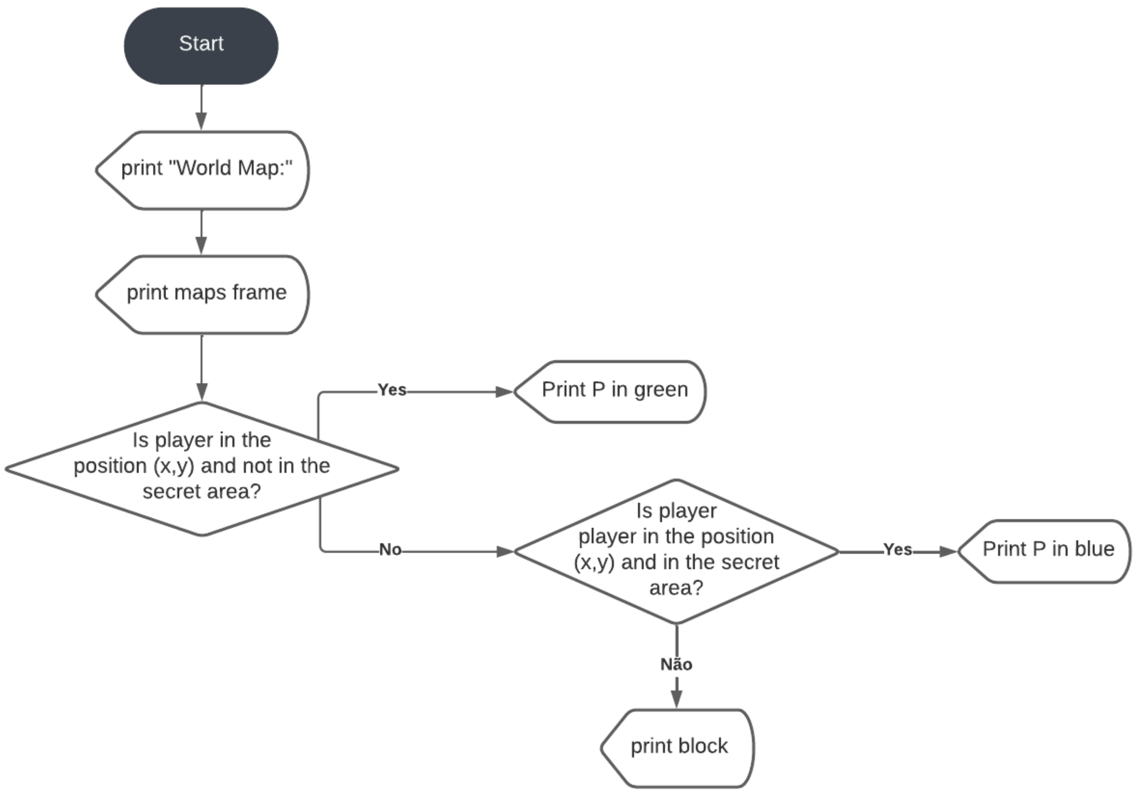
# 9 Appendix

### 9.1. Flowchart and Pseudocode for function initGame()


Start
Initialize worldWidth, worldHeight, and world array
Initialize player X and player Y
Initialize inventory as an empty array list

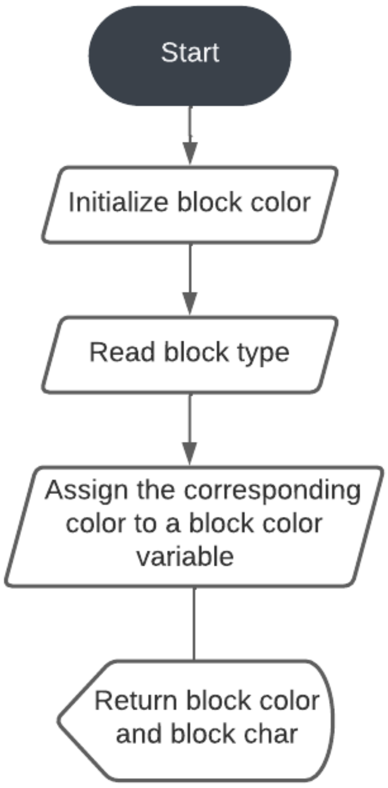
```
FUNCTION initGame() with worldWidth and worldHeight
      SET world width to worldWidth
      SET world height to worldHeight

      SET world to new array with dimensions worldWidth and worldHeight
      SET playerX to worldWidth/2
      SET playerY to worldHeight/2

      SET inventory to an empty ArrayList
ENDFUNCTION
```

### 9.2. Flowchart and Pseudocode for function displayWorld()


Start
print "World Map:"
print maps frame
Is player in the position (x,y) and not in the secret area?
Yes
Print P in green
No
Is player player in the position (x,y) and in the secret area?
Yes
Print P in blue
Não
print block

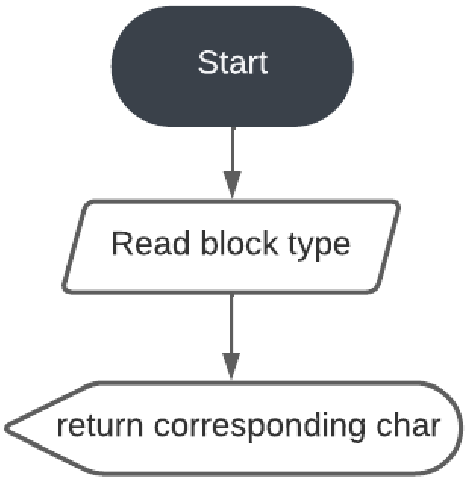
```
FUNCTION displayWorld()
   PRINT "World Map" header
   PRINT world top border characters

   FOR each row in the world
   PRINT world left border character

      FOR each column in the world
         IF player is in world position AND inSecretArea is false THEN
            PRINT player icon in green
         ELSE IF player is in world position AND inSecretArea is true THEN
       PRINT player icon in blue
         ELSE
            CALL getBlockSymbol() with world item at position (column, row)
            RETURN the block character and color
            PRINT the block character
         ENDIF

      ENDFOR
   ENDFOR
ENDFUNCTION
```

**9.3. Flowchart and Pseudocode for function getBlockSymbol()**


Start
Initialize block color
Read block type
Assign the corresponding color to a block color variable
Return block color and block char

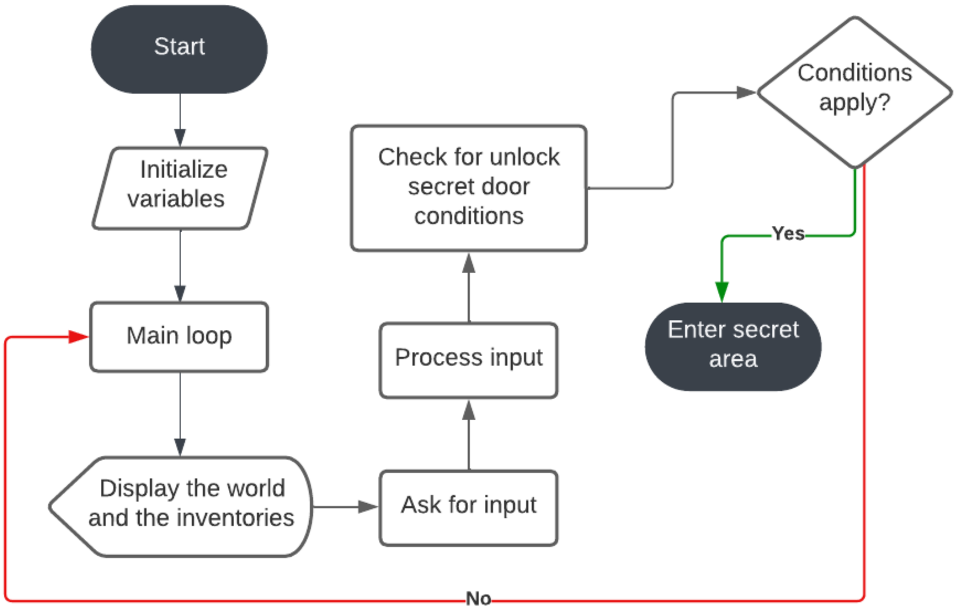
```
FUNCTION getBlockSymbol() with blockType index RETURNING colored symbols
      CASE blockType OF
            AIR(0): RETURN ANSI reset character followed by "-"
            WOOD(1): SET blockColor to red
            LEAVES(2): SET blockColor to green
            STONE(3): SET blockColor to blue
            IRON_ORE(4): SET blockColor to white
            OTHERS: RETURN ANSI reset character
      ENDCASE

      RETURN block symbol with blockColor and a space
ENDFUNCTION
```

**9.4. Flowchart and Pseudocode for function getBlockChar()**



```
FUNCTION getBlockChar() with blockType index RETURNING block character
      CASE blockType OF
            WOOD(1): RETURN wood character
            LEAVES(2): RETURN leaves character
            STONE(3): RETURN stone character
            IRON_ORE(4): RETURN iron ore character
            OTHERS: RETURN '-'
      ENDCASE
ENDFUNCTION
```

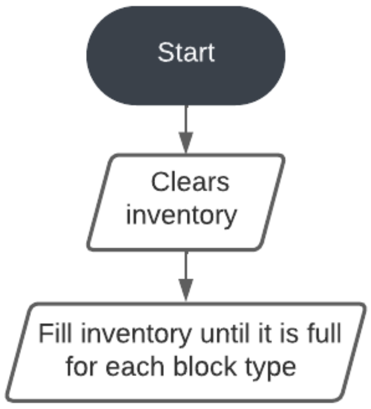
**9.5. Flowchart and Pseudocode for function startGame()**



```
FUNCTION startGame()
      SET scanner to new Scanner object
      SET unlockMode to false
      SET craftingCommandEntered to false
      SET miningCommandEntered to false
      SET openCommandEntered to false

      WHILE true

            CALL clearScreen()
            CALL displayLegend()
            CALL displayWorld()
            CALL displayInventory()

            PRINT possible commands legend
            AWAIT user input and save to input

            IF input is w, up, s, down, a, left, d or right THEN
                  IF unlockMode is true THEN
                        SET movementCommandEntered to True
                  ENDIF
                  CALL movePlayer() with input

            ELSE IF input is m THEN
                  IF unlockMode is true THEN
                        SET miningCommandEntered to true
                  ENDIF
                  CALL mineBlock()
            ELSE IF input is p THEN
                  CALL displayInventory()
                  PRINT message to ask block selection
                  AWAIT user input and save to recipe
                  CALL craftItem() with recipe

            ELSE IF input is i THEN
                  CALL interactWithWorld()

            ELSE IF input is save THEN
                  PRINT message to ask for file name
                  AWAIT user input and save to fileName
                  CALL saveGame() with fileName

            ELSE IF input is load THEN
                  PRINT message to ask for file name
                  AWAIT user input and save to fileName
                  CALL loadGame() with fileName

            ELSE IF input is exit THEN
                  PRINT goodbye message
                  BREAK WHILE LOOP

            ELSE IF input is unlock THEN
                  SET unlockMode to true

            ELSE IF input is getflag THEN
                  CALL getCountryAndQuoteFromServer()
                  CALL waitForEnter()

            ELSE IF input is open THEN
                  IF unlockMode, craftingCommandEntered,
miningCommandEntered and movementCommandEntered are all true THEN
                        SET secretDoorUnlocked to true
                        CALL resetWorld()
                        PRINT secret door unlocked!
                        CALL waitForEnter()

                  ELSE
                        PRINT invalid passkey
                        CALL waitForEnter()
                        SET scanner to new Scanner object
                        SET unlockMode to false
                        SET craftingCommandEntered to false
                        SET miningCommandEntered to false
                        SET openCommandEntered to false
            ELSE
                  PRINT invalid input

            ENDIF
```

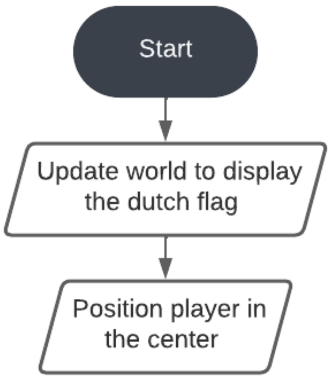
```
        IF unlockMode is true THEN
            IF input is c THEN
                SET craftingCommandEntered to true
            ELSE IF input is m THEN
                SET miningCommandEntered to true
            ELSE IF input is open THEN
                SET openCommandEntered to true
            ENDIF
        ENDIF

        IF secretDoorUnlocked is true THEN
            CALL clearScreen()
            PRINT message to welcome user to secret area
            SET inSecretArea to true
            CALL resetWorld()
            SET secretDoorUnlocked to false
            CALL fillInventory()
            CALL waitForEnter()
        ENDIF
    ENDWHILE
ENDFUNCTION
```

## 9.6. Flowchart and Pseudocode for function fillInventory()



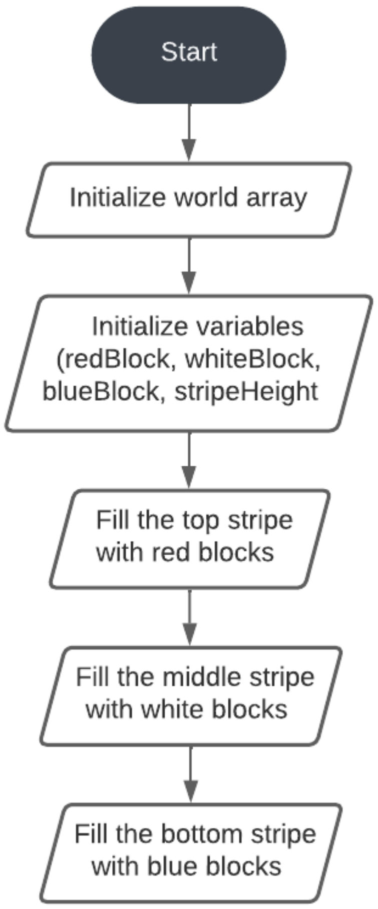
```
FUNCTION fillInventory()
      CALL clear() on inventory array

      FOR every blockType
            FOR size of the inventory
                  ADD blockType to inventory
            ENDLOOP
      ENDFOR
ENDFUNCTION
```

## 9.7. Flowchart and Pseudocode for function resetWorld()


Start
Update world to display the dutch flag
Position player in the center

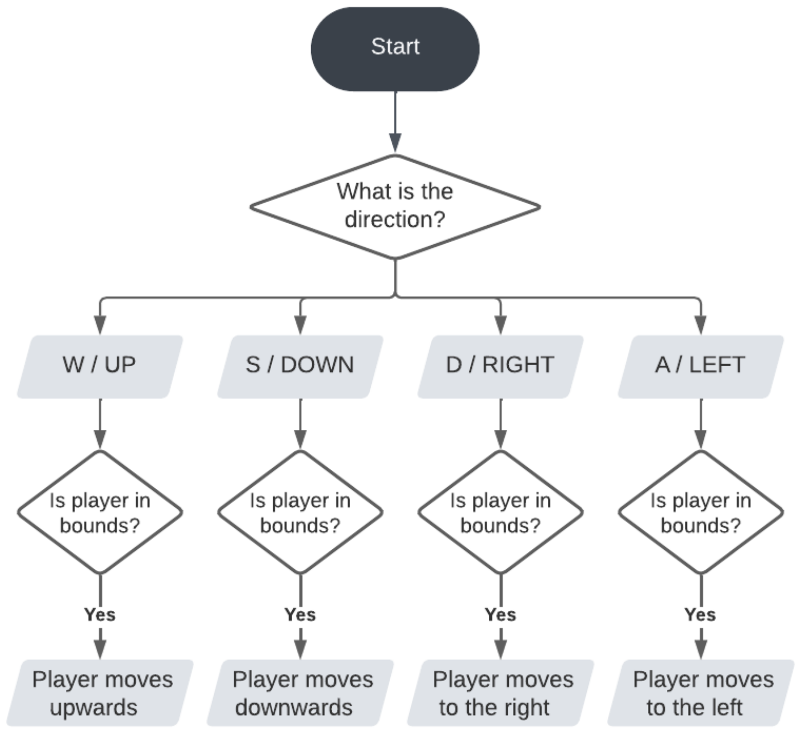
```
FUNCTION resetWorld()
      CALL generateEmptyWorld()

      SET playerX to worldWidth / 2
      SET playerY to worldHeight / 2

ENDFUNCTION
```

## 9.8. Flowchart and Pseudocode for function generateEmptyWorld()


Start
Initialize world array
Initialize variables (redBlock, whiteBlock, blueBlock, stripeHeight
Fill the top stripe with red blocks
Fill the middle stripe with white blocks
Fill the bottom stripe with blue blocks

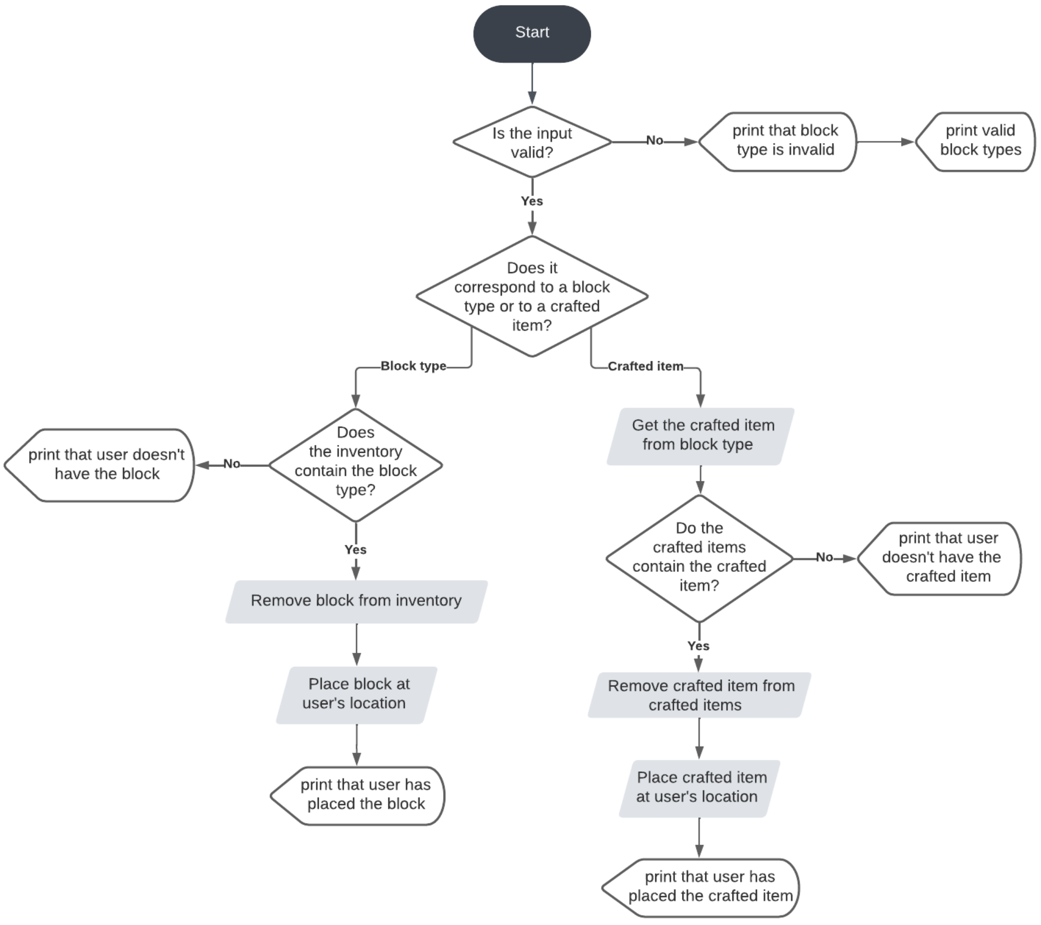
```
FUNCTION generateEmptyWorld()
      SET world to empty array (NEW_WORLD_WIDTH(25),NEW_WORLD_HEIGHT(15))
      SET redBlock to 1
      SET whiteBlock to 4
      SET blueBlock to 3
      SET stripeHeight to a third of the NEW_WORLD_HEIGHT

      FOR each y in range 0 to stripeHeight
            FOR each x in range 0 to NEW_WORLD_WIDTH
                  INSERT redBlock at coordinates (x,y)
      ENDFOR

ENDFUNCTION
```

### 9.9. Flowchart and Pseudocode for function movePlayer()



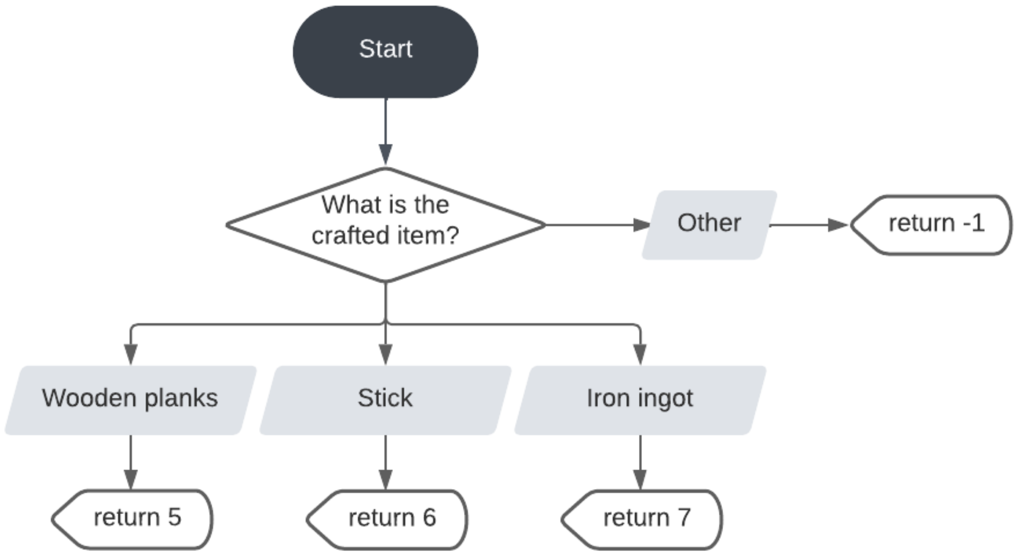
```
FUNCTION movePlayer()
      IF direction is W OR UP THEN
            IF player's Y coordinate is in bounds THEN
                  MOVE player upwards
            ENDIF
      ELSE IF direction is S OR DOWN THEN
            IF player's Y coordinate is in bounds THEN
                  MOVE player downwards
            ENDIF
      ELSE IF direction is D OR RIGHT THEN
            IF player's X coordinate is in bounds THEN
                  MOVE player to the right
            ENDIF
      ELSE IF direction is A OR LEFT THEN
            IF player's X coordinate is in bounds THEN
                  MOVE player to the left
            ENDIF
      ENDIF
ENDFUNCTION
```

## 9.10. Flowchart and Pseudocode for function placeBlock()


Start
Is the input valid?
No
print that block type is invalid
print valid block types
Yes
Does it correspond to a block type or to a crafted item?
Block type
Crafted item
Does the inventory contain the block type?
No
print that user doesn't have the block
Yes
Remove block from inventory
Place block at user's location
print that user has placed the block
Get the crafted item from block type
Do the crafted items contain the crafted item?
No
print that user doesn't have the crafted item
Yes
Remove crafted item from crafted items
Place crafted item at user's location
print that user has placed the crafted item

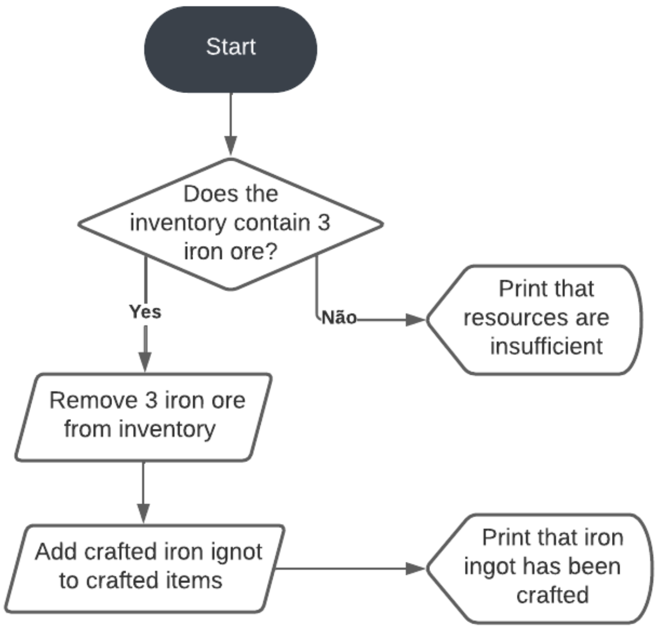
```
FUNCTION placeBlock()
        IF block number between 0 and 7 THEN
                IF block number is smaller or equal to 4 THEN
                        IF blockType is in inventory THEN
                                REMOVE blockType from inventory
                                PLACE blockType at players coordinates
                                PRINT that player has placed the block
                        ELSE
                                PRINT that user doesn’t have the block
                        ENDIF
                ELSE
                        GET craftedItem from blockType
                        IF craftedItem is in craftedItems THEN
                                REMOVE craftedItem from CraftedItems
                                PLACE craftedItem at player’s coordinates
                                PRINT that user has placed the crafted item
                        ELSE
                                PRINT that user doesn’t have the crafted item
                        ENDIF
                ENDIF
        ELSE
                PRINT that block number is invalid and which ones are valid
        ENDIF
ENDFUNCTION
```

## 9.11. Flowchart and Pseudocode for function getBlockTypeFromCraftedItem()



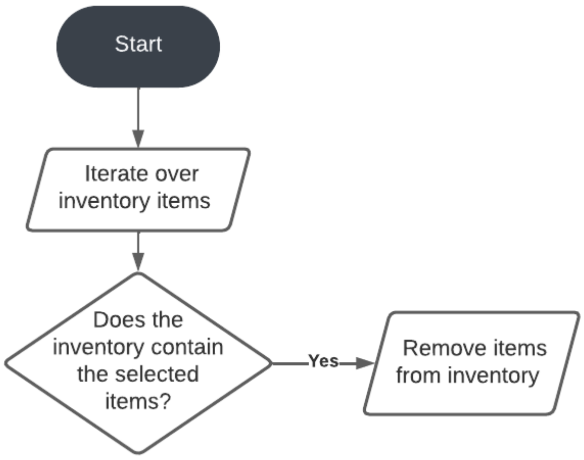
```
FUNCTION getBlockTypeFromCraftedItem()
      IF craftedItem is CRAFTED_WOODEN_PLANKS THEN
            RETURN 5
      ELSE IF craftedItem is CRAFTED_STICK THEN
            RETURN 6
      ELSE IF craftedItem is IRON_INGOT THEN
            RETURN 7
      ELSE
            RETURN -1
      ENDIF
ENDFUNCTION
```

## 9.12. Flowchart and Pseudocode for function craftIronIngot()


Start
Does the inventory contain 3 iron ore?
Yes
Não
Print that resources are insufficient
Remove 3 iron ore from inventory
Add crafted iron ignot to crafted items
Print that iron ingot has been crafted

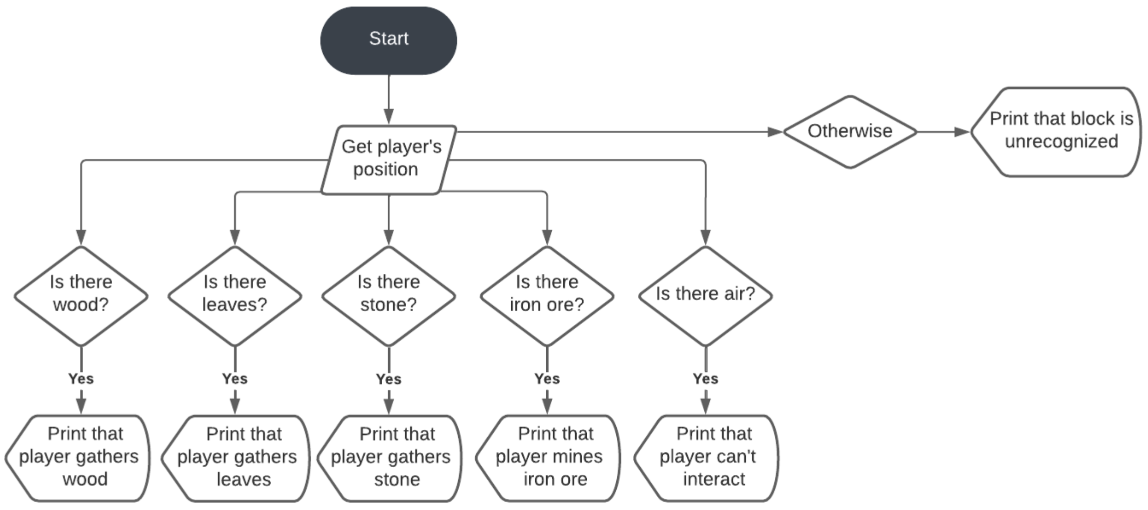
```
FUNCTION craftIronIngot()
      IF inventory contains 3 IRON_ORE items THEN
            REMOVE 3 IRON_ORE items from the inventory
            CALL addCraftedItem() WITH CRAFTED_IRON_INGOT (200)
      ELSE
            PRINT that the user has insufficient resources
      ENDIF
ENDFUNCTION
```

### 9.13. Flowchart and Pseudocode for function removeItemsFromInventory()


Start
Iterate over inventory items
Does the inventory contain the selected items?
Yes
Remove items from inventory

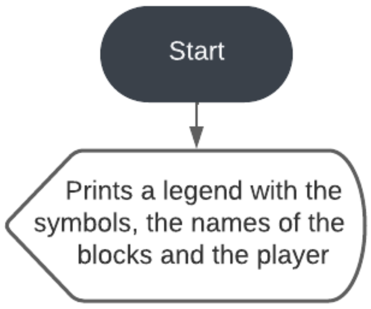
```
FUNCTION removeItemsFromInventory() WITH item id AND count
      SET removeCount to 0

      LOOP over the inventory
            SET nextItem to next inventory item in the loop
            IF the nextItem is the same as item id THEN
                  REMOVE item from inventory
                  IF removedCount is the same as count THEN
                        BREAK the loop
                  ENDIF
            ENDIF
      ENDLOOP
ENDFUNCTION
```

## 9.14. Flowchart and Pseudocode for function interactWithWorld()


Start
Get player's position
Otherwise
Print that block is unrecognized
Is there wood?
Is there leaves?
Is there stone?
Is there iron ore?
Is there air?
Yes
Yes
Yes
Yes
Yes
Print that player gathers wood
Print that player gathers leaves
Print that player gathers stone
Print that player mines iron ore
Print that player can't interact

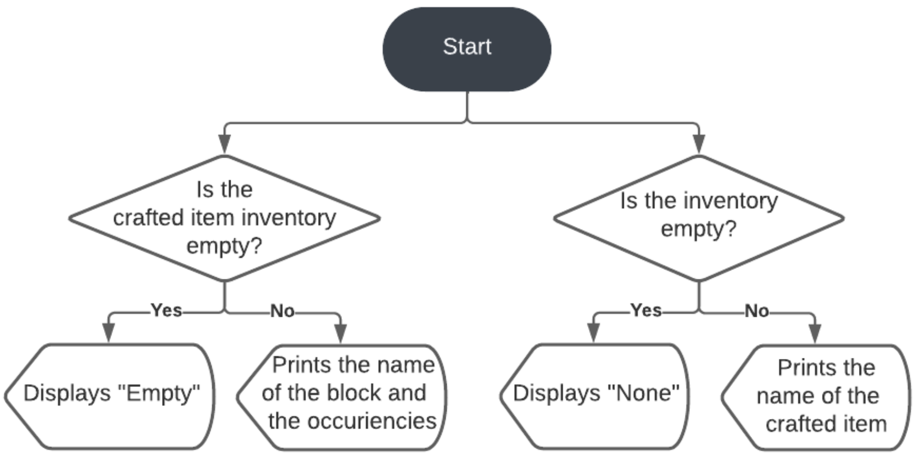
```
FUNCTION interactWithWorld()
      SET blockType to the blockType id on which the player is located

      CASE blockType OF
            WOOD(1):
                  PRINT message informing user they gathered wood
                  ADD WOOD to the inventory
            LEAVES(2):
                  PRINT message informing user they gathered leaves
                  ADD LEAVES to the inventory
            STONE(3):
                  PRINT message informing user they gathered stone
                  ADD STONE to the inventory
            IRON_ORE(4):
                  PRINT message informing user they gathered iron ore
                  ADD IRON_ORE to the inventory
            AIR(0):
                  PRINT message informing user they cannot interact with
                  air
            DEFAULT:
                  PRINT to the user that the block is unrecognized
      ENDCASE

      CALL waitForEnter()
ENDFUNCTION
```

## 9.15. Flowchart and Pseudocode for function displayLegend()


Start
Prints a legend with the symbols, the names of the blocks and the player

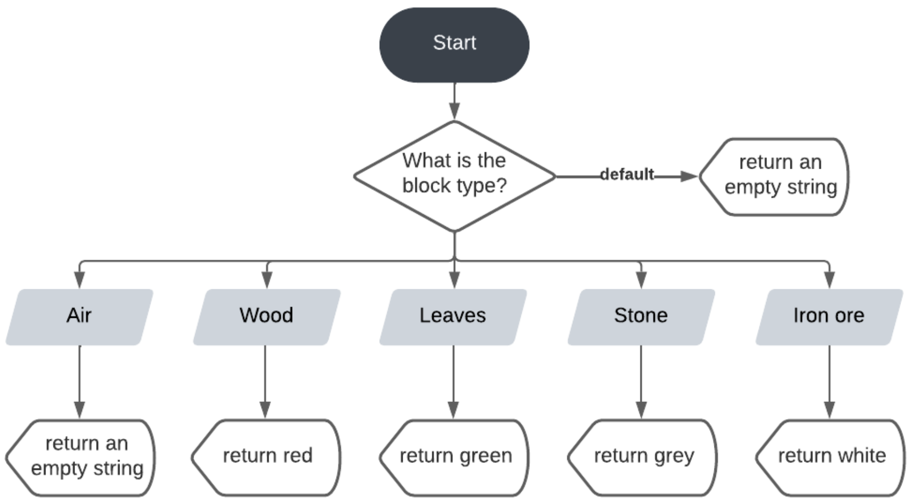
```
FUNCTION displayLegend()
      PRINT "Legend"
      PRINT "Empty Block" in white
      PRINT "Wood Block" in red
      PRINT "Leaves Block" in greed
      PRINT "Stone Block" in blue
      PRINT "Iron ore Block" in white
      PRINT "Player" in blue

ENDFUNCTION
```

### 9.16. Flowchart and Pseudocode for function displayInventory()


Start
Is the crafted item inventory empty?
Yes
No
Displays "Empty"
Prints the name of the block and the occuriencies
Is the inventory empty?
Yes
No
Displays "None"
Prints the name of the crafted item

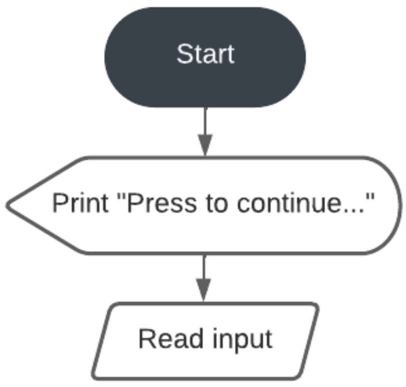
```
FUNCTION displayInventory()
        PRINT "Inventory"
        IF inventory is empty THEN
                PRINT message displaying empty inventory
        ELSE
                SET blockCounts to empty array of size 5

                FOR each item in the inventory
                        SET block to the inventory item
                        INCREMENT blockCount at index item
                ENFOR

                FOR blockType in blockCounts
                        SET occurrences to blockCounts
                        IF occurrences is greater than 0
                              PRINT the number of occurrences for the blockType
                        ENDIF
                ENDFOR
        ENDIF

        PRINT "Crafted items"
        IF craftedItems is null OR is empty THEN
                PRINT "None" in yellow
        ELSE
                FOR each item in craftedItems
                        CALL getCraftedItemColor() WITH item
                        CALL getCraftedItemName() WITH item
                        PRINT crafted item with appropriate color and name
                ENDFOR
        ENDIF
ENDFUNCTION
```

## 9.17. Flowchart and Pseudocode for function getBlockColor()



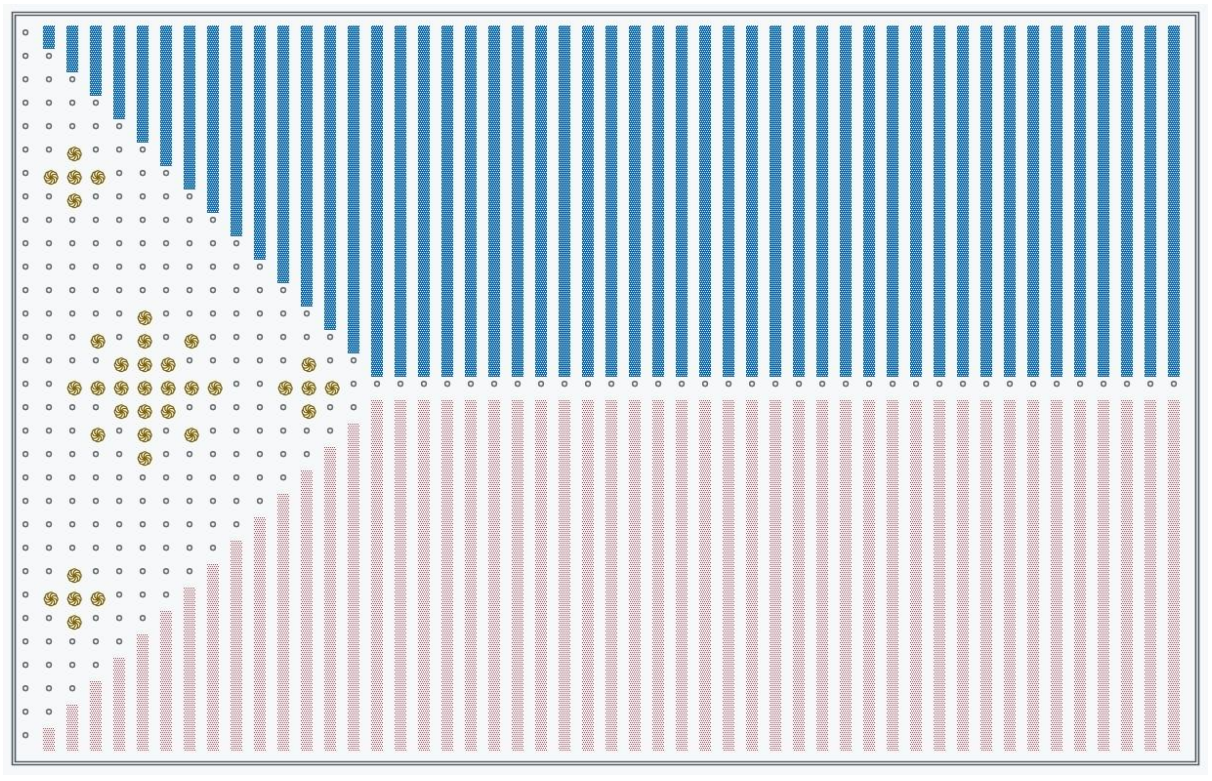
```
FUNCTION getBlockColor() WITH blockType RETURNING ANSI color character

      CASE blockType OF
            AIR:
                  RETURN ""
            WOOD:
                  RETURN ANSI character for red
            LEAVES:
                  RETURN ANSI character for red
            STONE:
                  RETURN ANSI character for red
            IRON_ORE:
                  RETURN ANSI character for red
            OTHERS:
                  RETURN ""
      ENDCASE
ENDFUNCTION
```

## 9.18. Flowchart and Pseudocode for function waitForEnter()



```
FUNCTION waitForEnter()
      PRINT "Press enter to continue"
      AWAIT user input
ENDFUNCTION
```

## 9.19. Printed Philippines Flag



## 10 References

1. Metwalli, Sara. “Pseudocode: What It Is and How to Write It | Built In.” *Builtin.com*, 16 May 2022, builtin.com/data-science/pseudocode.