# 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 5, 2023

# Contents

# 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

This section describes several functions found in the JavaCraft.java file. Out of the functions, there are fifteen who have a flowchart and pseudocode.

**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

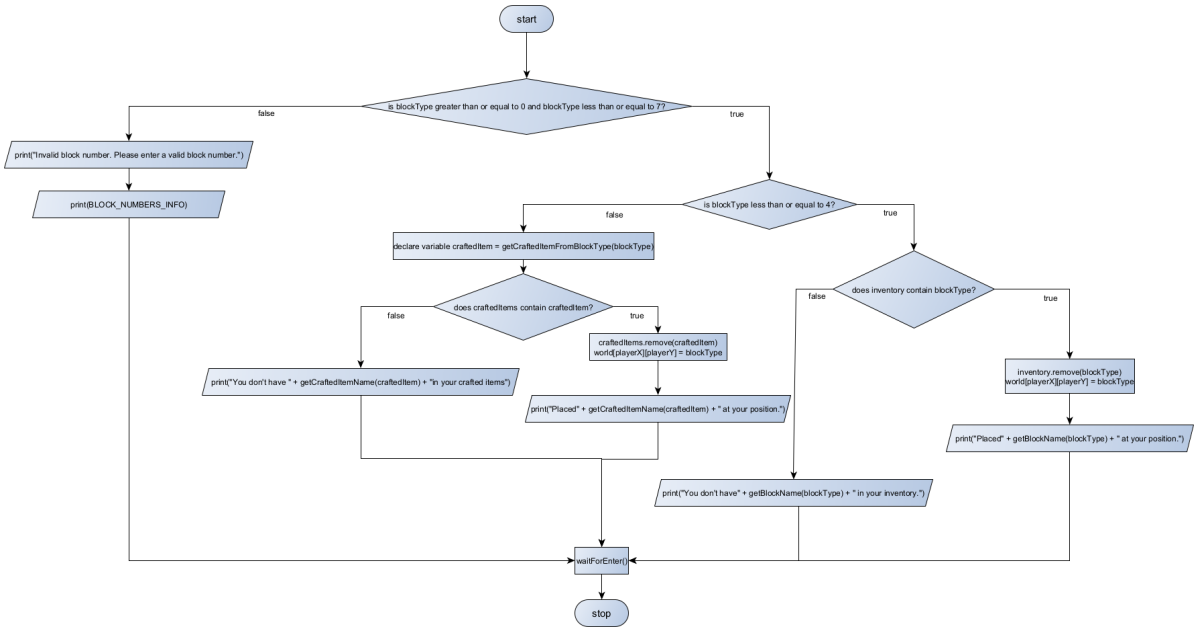
The link to the JavaCraft branch can be found here: https://gitlab.maastrichtuniversity.nl/bcs1110/javacraft/-/tree/group11

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.

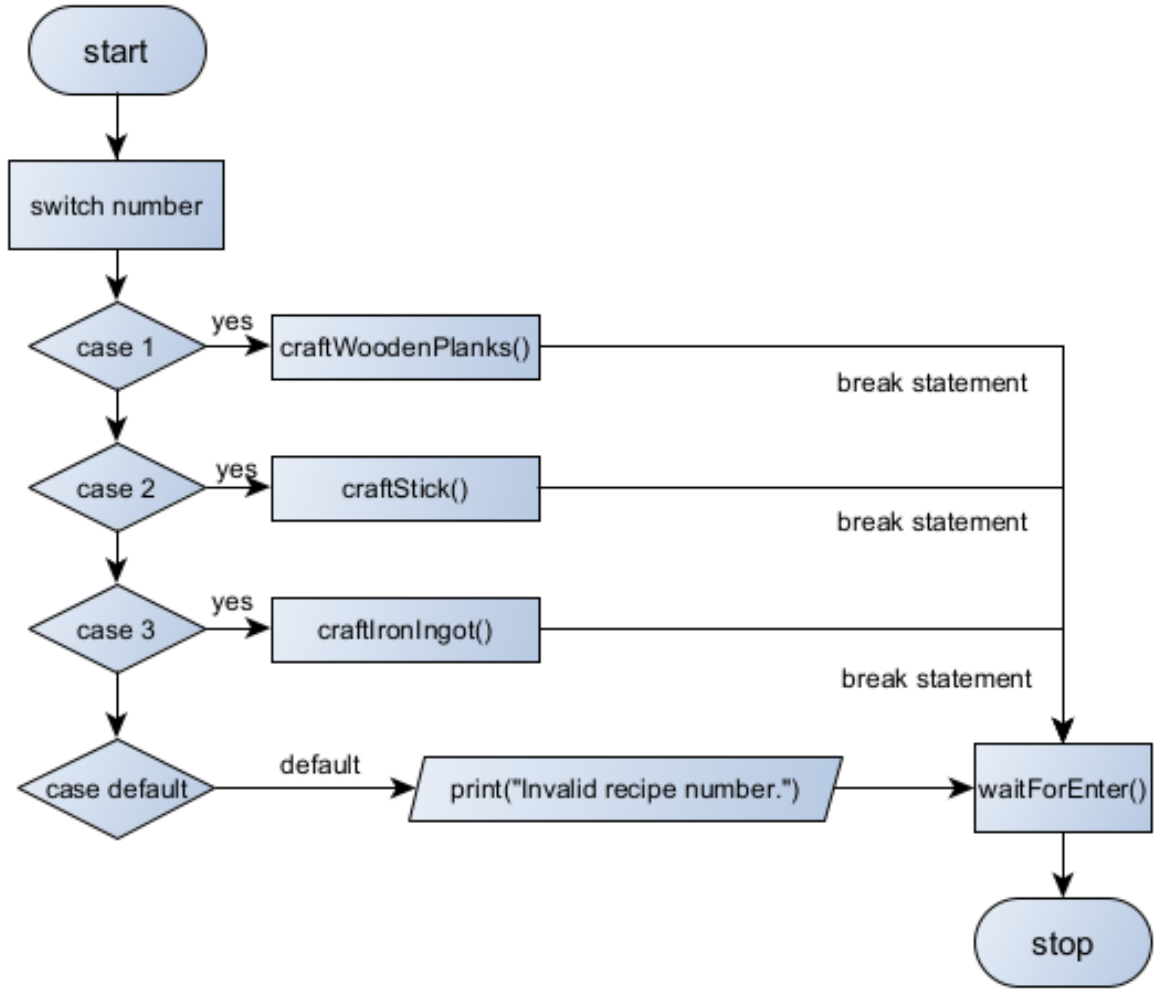
# 6 | References

# A | Appendix: pseudocode and flowcharts

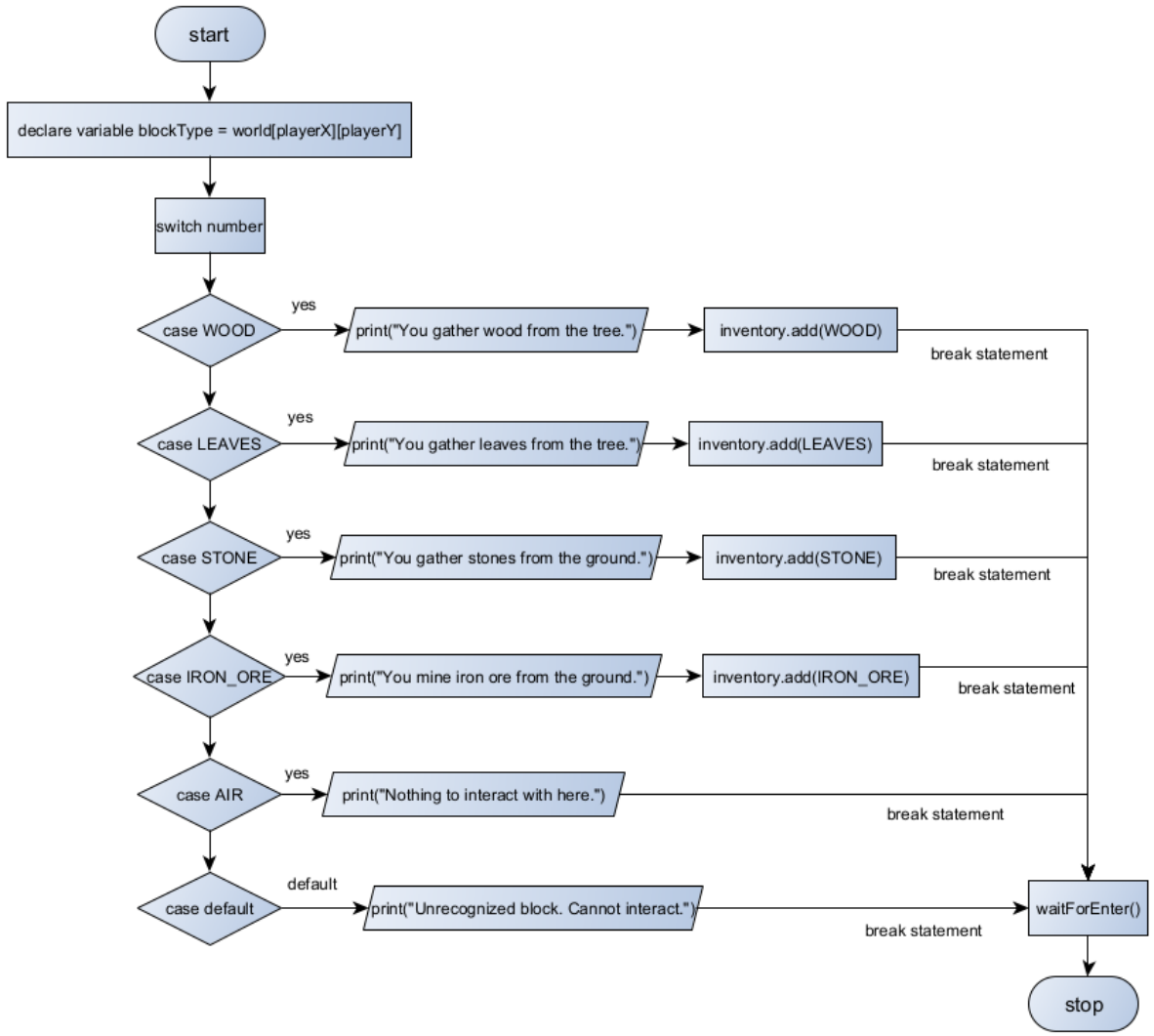
This appendix provides the full blocks of pseudocode and its flowcharts.




```
function placeBlock(blockType)

if blockType is greater than or equal to 0 and blockType is less than or equal to 7 then
   if blockType less than or equal to 4 then
      if inventory contains blockType then
         inventory.remove(blockType)
         world[playerX][playerY] = blockType
         print("Placed" + getBlockName(blockType) + " at your position.")

      else print("You don't have " + getBlockName(blockType) + " in your inventory")
      end if
   else
      craftedItem = getCraftedItemFromBlockType(blockType)
      if craftedItems contains craftedItem then
         craftedItems.remove(craftedItem)
         world[playerX][playerY] = blockType
         print("Placed" + getCraftedItemName(craftedItem)+ " at your position.")
      else print("You don't have " + getCraftedItemName() + " in your inventory.")
      end if
   end if
else print("Invalid block number. Please enter a valid block number.")
print(BLOCK_NUMBERS_INFO)
end if
waitForEnter()
end function
```

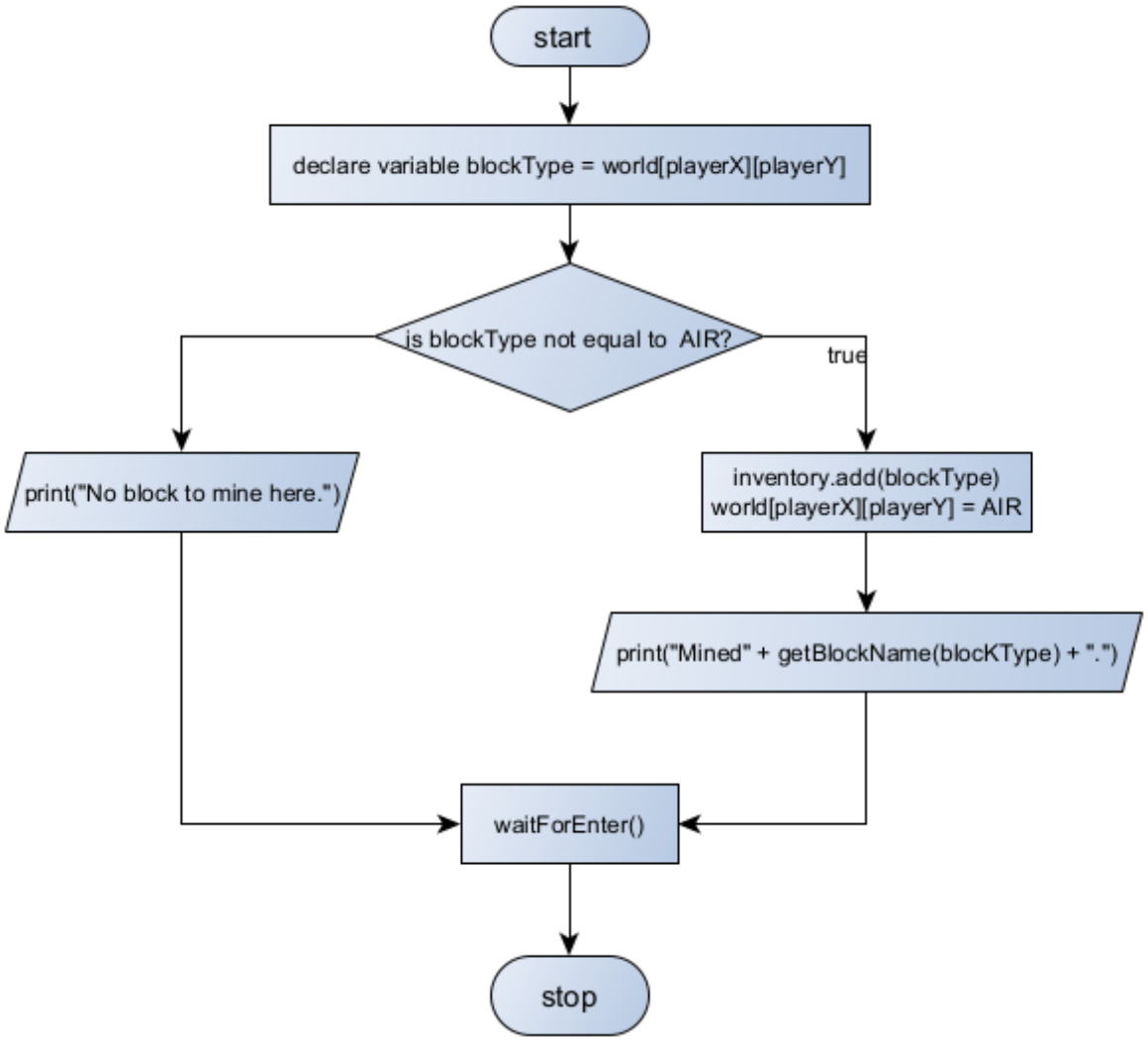



```
function craftItem(recipe)

switch(recipe)
case 1:
   craftWoodenPlanks()
end if
case2:
   craftStick()
end if
case3:
   craftIronIngot()
end if
default:
   print("Invalid recipe number.")
end switch
waitForEnter()
end function
```

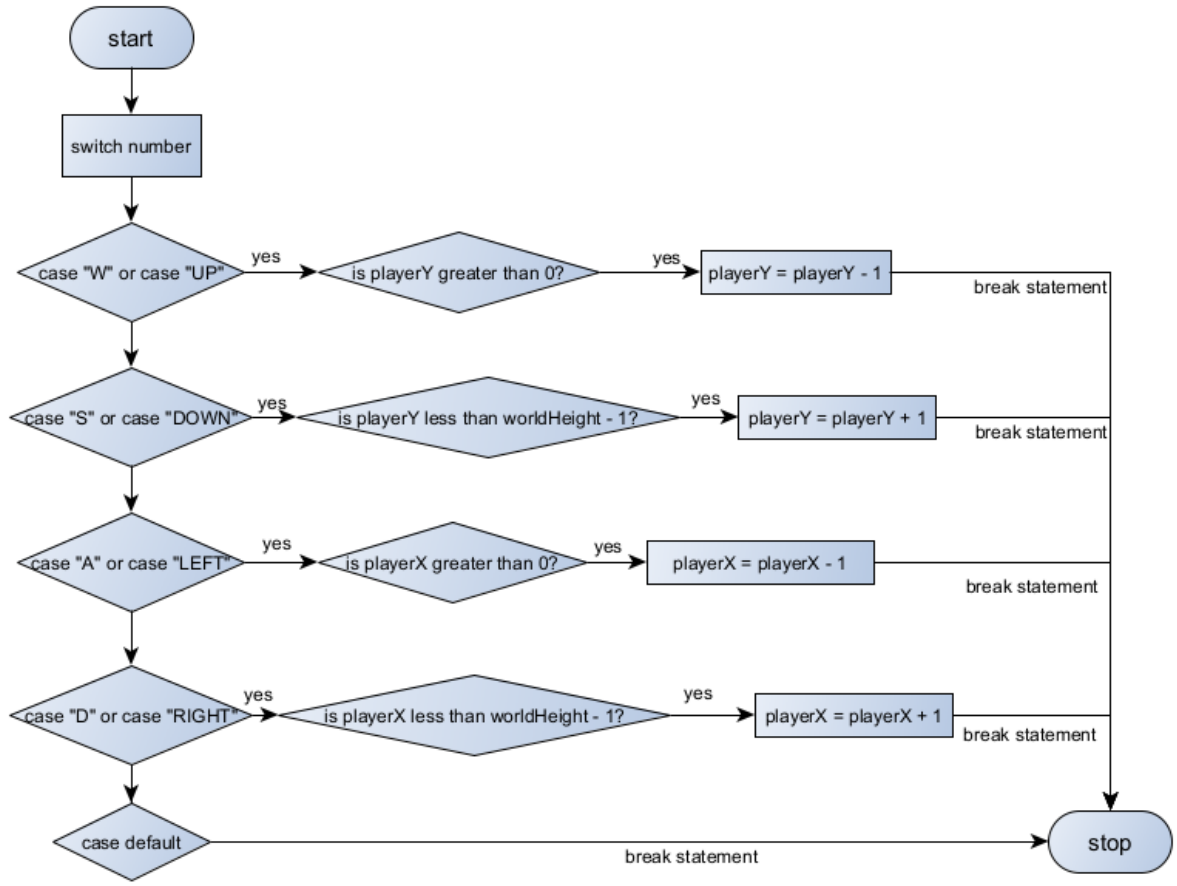



```
function interactWithWorld()

blockType = world[playerX][playerY]
switch(blockType):
case WOOD:
   print("You gather wood from the tree.")
   inventory.add(WOOD)
end if
case LEAVES:
   print("You gather leaves from the tree.")
   inventory.add(LEAVES)
end if
case STONE:
   print("You gather stones from the ground.")
   inventory.add(STONE)
end if
case IRON_ORE:
   print("You mine iron ore from the ground.")
   inventory.add(IRON_ORE)
end if
case AIR:
   print("Nothing to interact with here.")
end if
default: print("Unrecognized block. Cannot interact.")
end switch
```

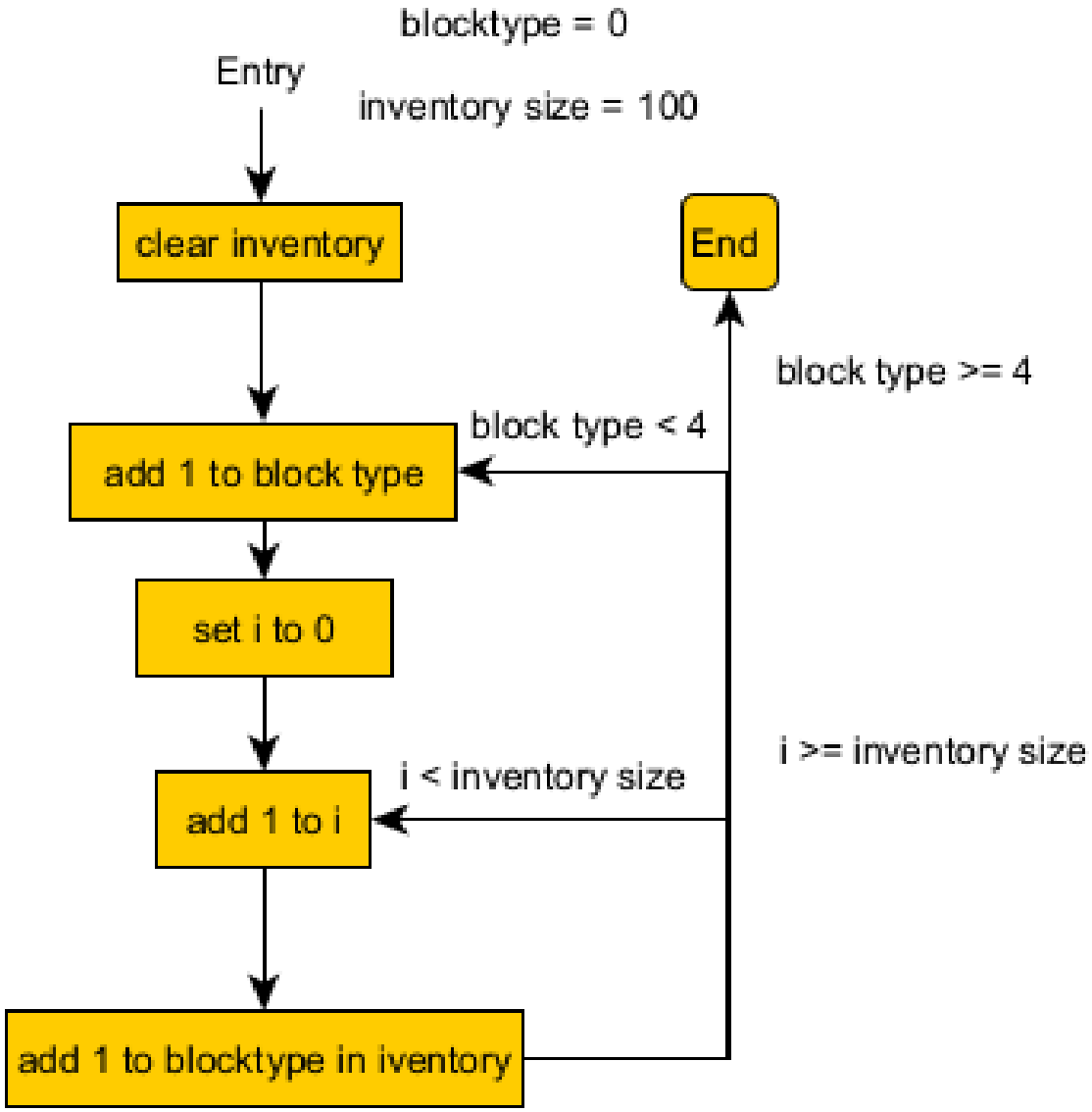
```
waitForEnter()
end function
```



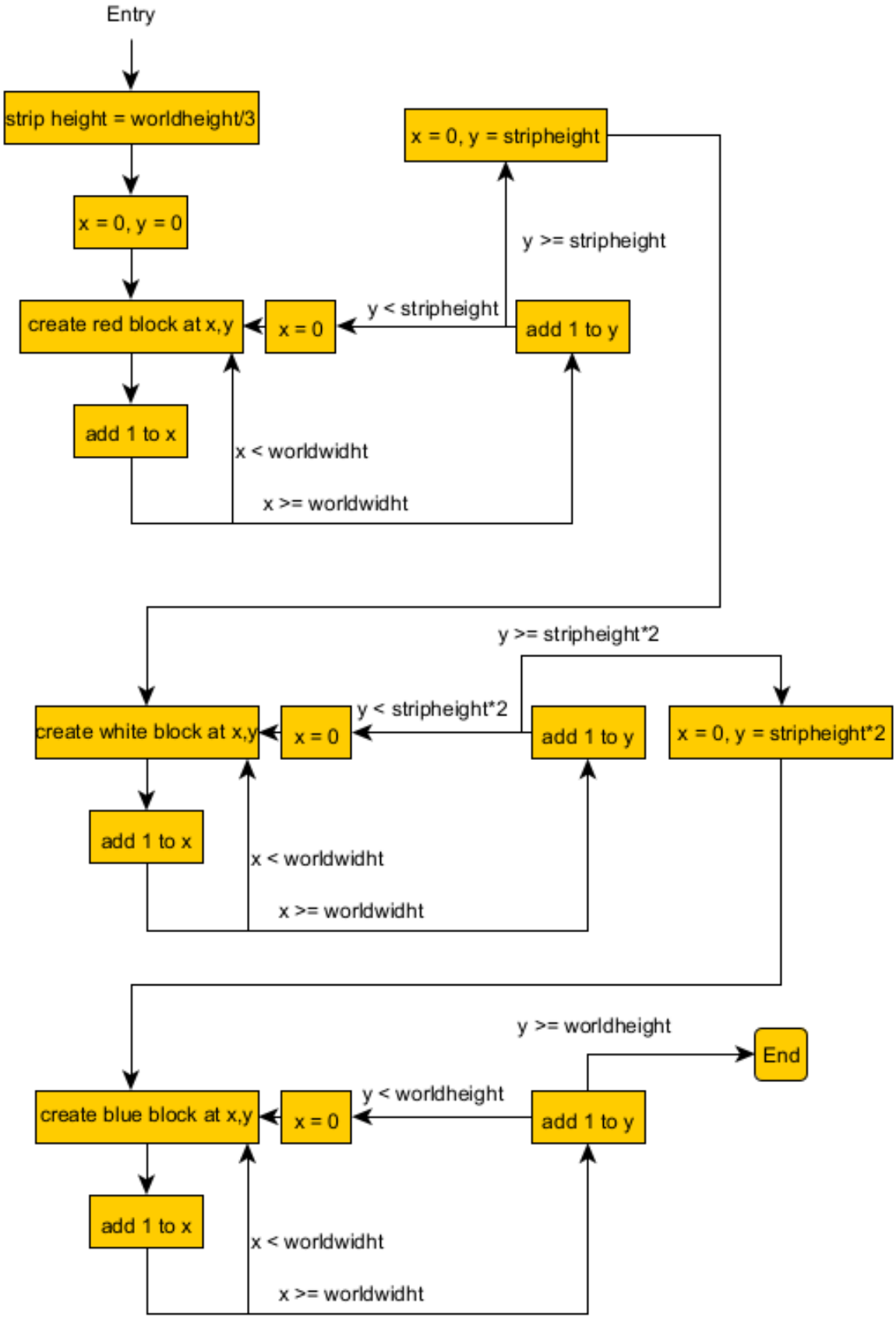

```
function mineBlock()

blockType = world[playerX][playerY]
if blockType is not equal to AIR then
   inventory.add(blockType)
   world[playerX][playerY] = AIR
   print("Mined " + getBlockName(blockType) + ".")
else print("No block to mine here.")
end if
waitForEnter()
end function
```




```
function movePlayer(direction):

direction = uppercase(direction) //converts direction to uppercase for consistency
switch(direction):
case "W" or "UP":
   if playerY > 0 then playerY = playerY - 1
   end if
case "S" or "DOWN":
   if playerY < worldHeight - 1 then playerY = playerY + 1
   end if
case "A" or "LEFT":
   if playerX > 0 then playerX = playerX - 1
   end if
case "D" or "RIGHT":
   if playerX < worldWidth - 1 then playerX = playerX + 1
   end if
default: //do nothing
end switch
end function
```

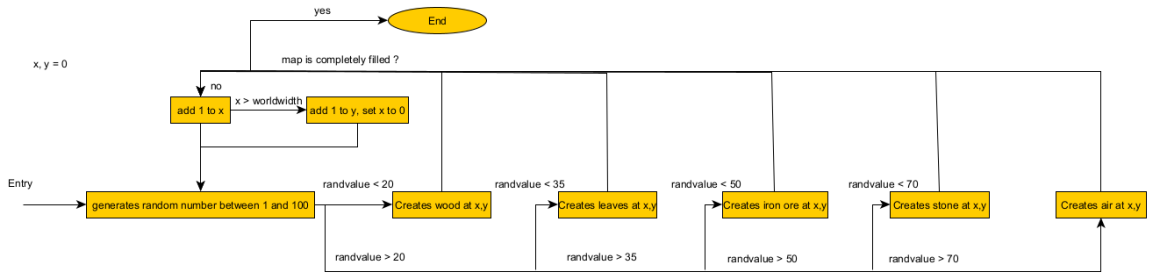



```
function fillInventory()

blocktype = 0
inventory size = 100
clear inventory
if block type < 4
add 1 to block type
set i to 0
if i < inventory size
   add 1 to i
   add 1 blocktype in inventory
```



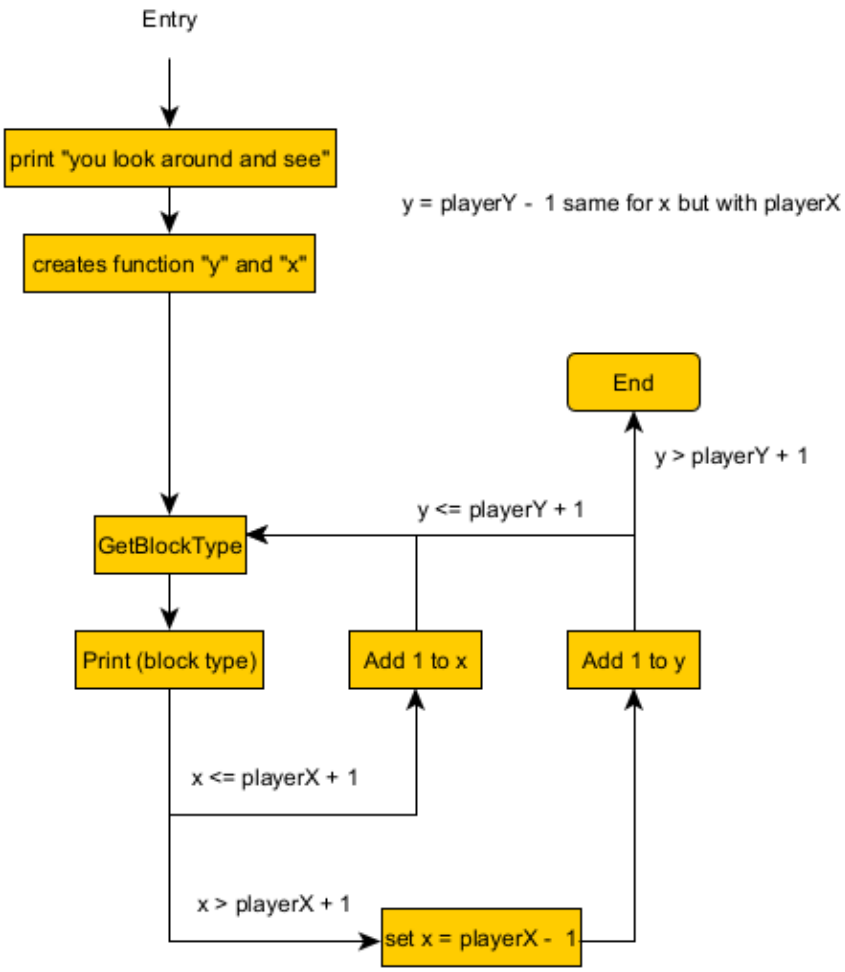

```
function generateEmptyWorld()
```

```
redBlock = 1
whiteBlock = 4
blueBlock = 3

stripheight = world height / 3

y = 0
if y < stripheight(
   if x < worldwidth(

   create redblock at x, y
   add 1 to x
     )
    add 1 to y)

y = stripheight
if y < stripheight*2(
   if x < worldwidth(

   create redblock at x, y
   add 1 to x
     )
    add 1 to y)

y = stripheight*2
if y < worldheight(
   if x < worldwidth(

   create redblock at x, y
   add 1 to x
     )
    add 1 to y)
```

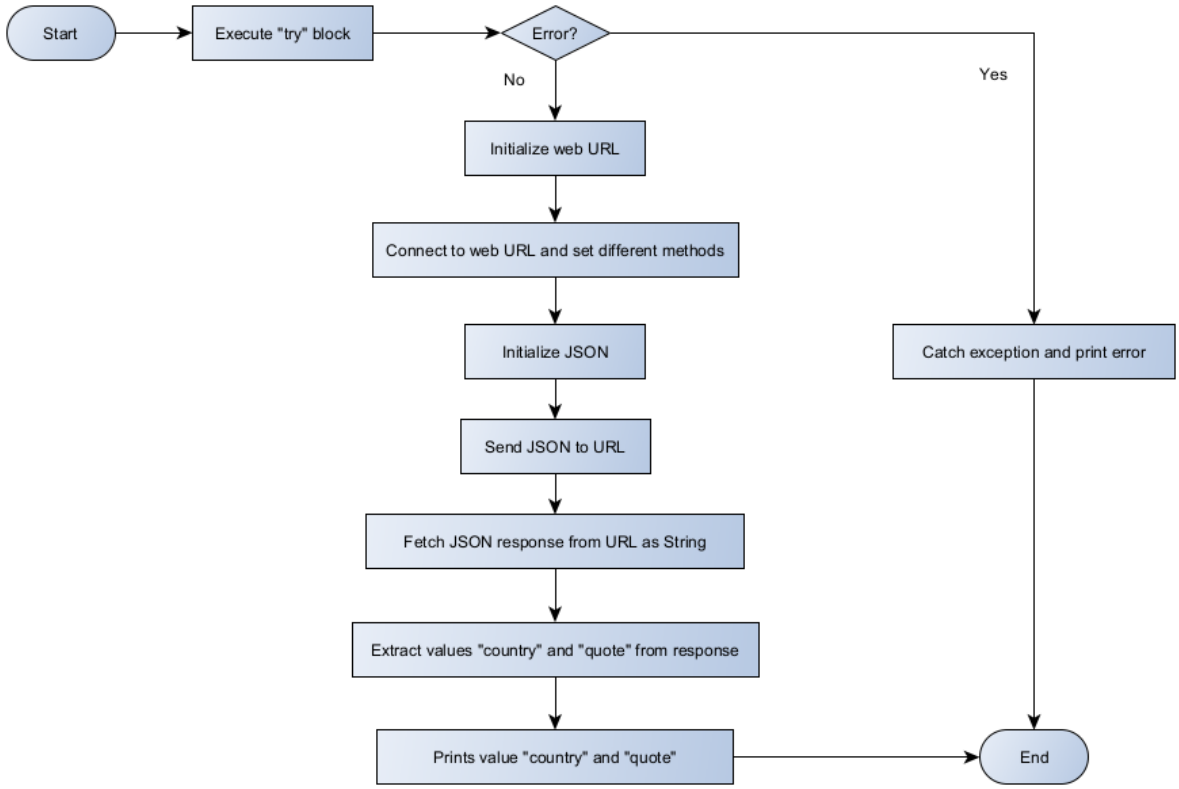



```
function generateWorld()

x = 0
y = 0
if y < worldheight
    if x < worldwidth
        creates random number between 1 and 100
        if random number < 20
            creates wood at x, y
        else if random number < 35
            creates leaves at x, y
        else if random number < 50
            creates stone at x, y
        else if random number < 20
            creates iron ore at x, y
        else create air at x, y
    add 1 to x
add 1 to y
```


```
function initgame()

set worldwidth
set worldheight
set world = [worldwidht][worldheight]
set playerx = worldwidght / 2
set playery = worldheight / 2
creates arraylist inventory
```




```
function lookAround()

print "You look around and see:"

set x = player x -1 , y = player y - 1
if y <= player Y + 1
if x <= player X + 1
  get block type (x, y)
  print (block type)
else set x to player x - 1
     add 1 to y
```




```
function getCountryAndQuoteFromServer():
    TRY:
        let link = "https://flag.ashish.nl/get_flag"
        Setup a connection to link
        Set request method of connection to "POST"
        Set request property of connection to "Content-Type" as json
        Enable output of connection

        let payload be stringified json
        let writer be OutputStreamWriter of connection
        Write payload to writer
        Flush writer
        Close writer

        let reader be BufferedReader of connection
        let sb be StringBuilder
        let line be empty string
        WHILE (line is not null):
            let line read next line of reader
            Append line line to sb
        json = ConvertToString(sb)

        let countryStart = FindSubstringIndex(json, " ") + 11
        let countryEnd = FindSubstringIndex(json, " ", countryStart)
        let country = Substring(json, countryStart, countryEnd)

        let quoteStart = FindSubstringIndex(json, " ") + 9
        let quoteEnd = FindSubstringIndex(json, " ", quoteStart)
        let quote = Substring(json, quoteStart, quoteEnd)

        quote = ReplaceSpaces(quote)
        Print("Country: " + country)
        Print("Quote: " + quote)
    CATCH Exception AS e:
```

```
        stackTrace = GetStackTrace(e)
        Print("Error connecting to the server")
        Print(stackTrace)
end function
```