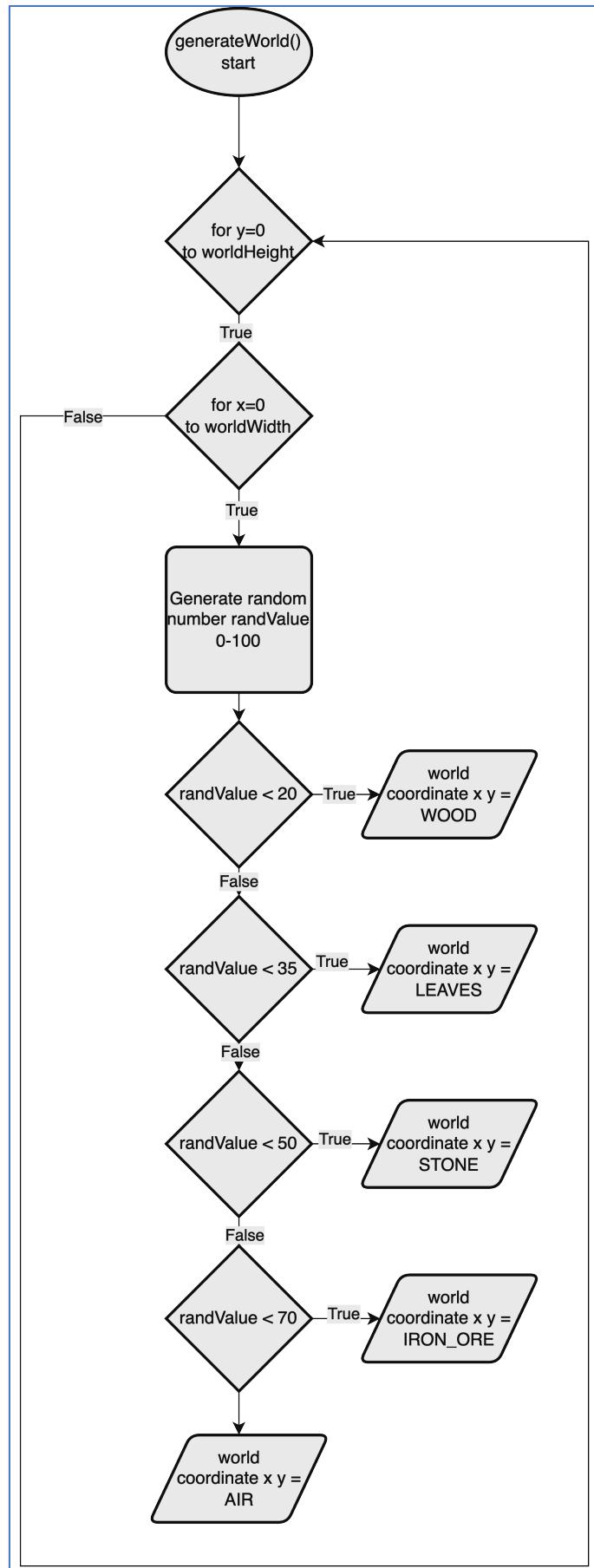


# *APPENDIX*

# *Flowcharts & Pseudocode*

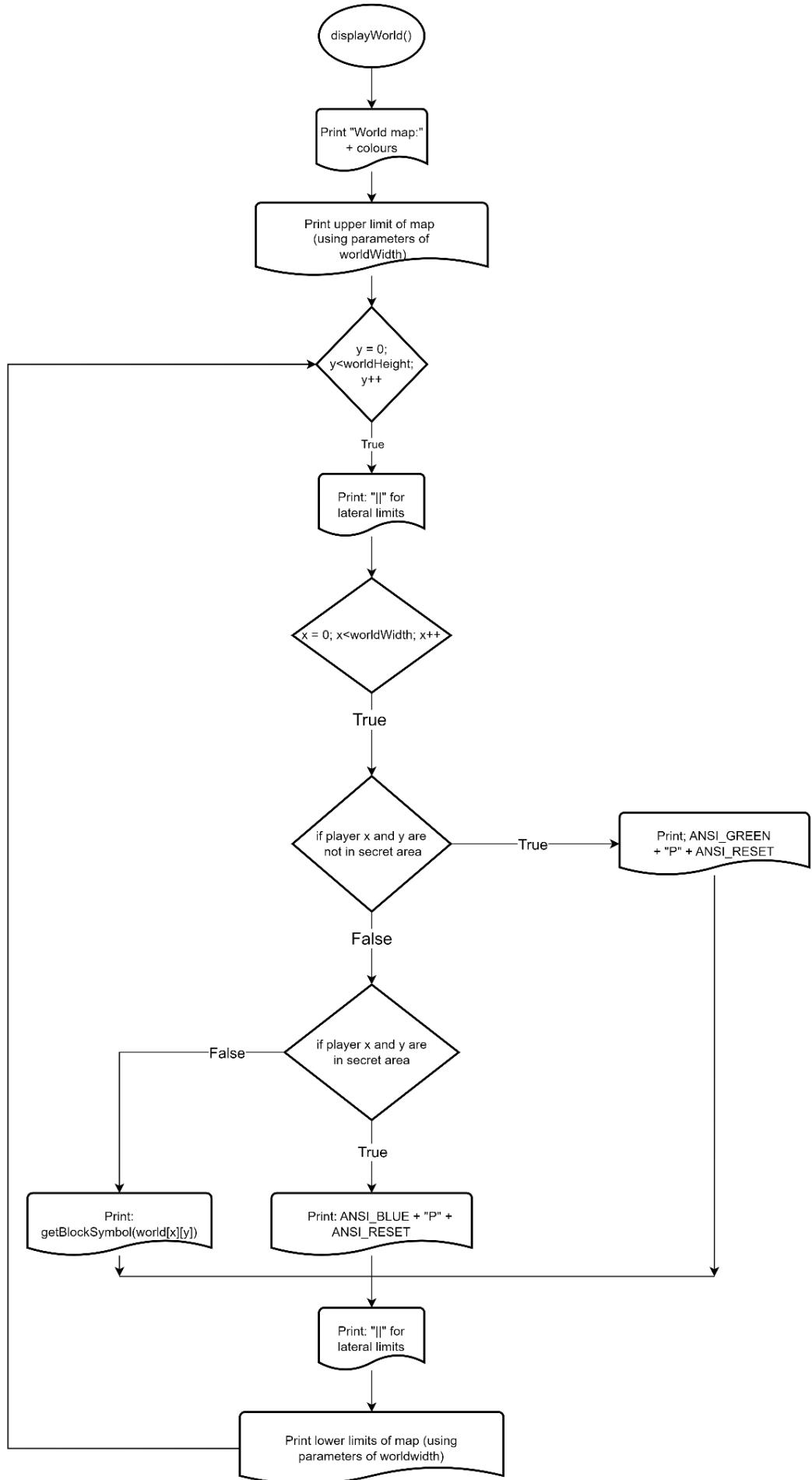
## 03 – generateWorld



Pseudocodes > ≡ 03-generateWorld

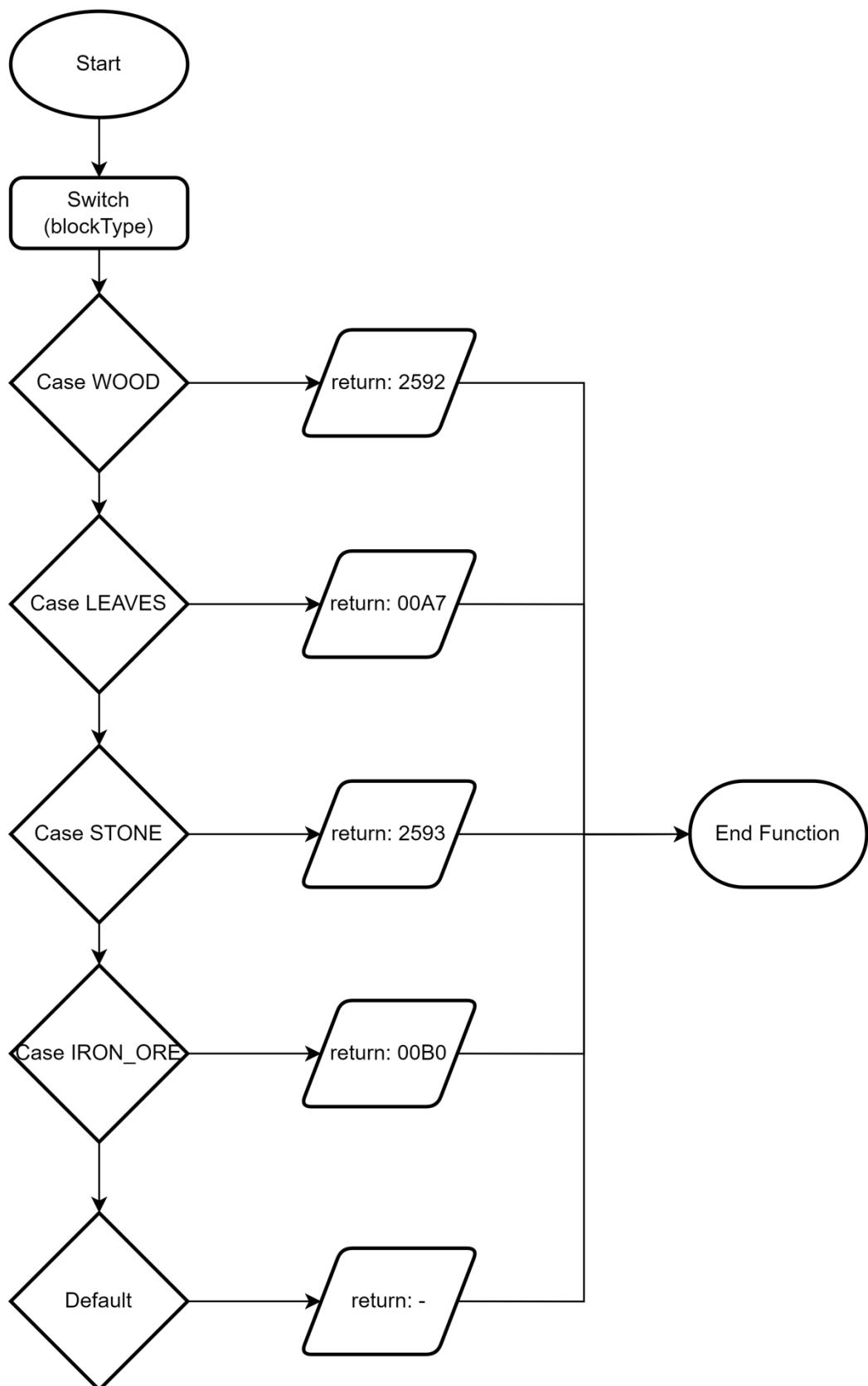
```
1   FUNCTION generateWorld
2       OBJECT Random new rand
3       FOR each y coordinate in wordHeight
4           FOR each x coordinate in worldWidth
5               SET randValue to new random value between 0-100
6               IF randValue is smaller than 20 THEN
7                   SET world xy coordinate to WOOD
8               ELSE IF randValue is smaller than 35 THEN
9                   SET world xy coordinate to LEAVES
10              ELSE IF randValue is smaller than 50 THEN
11                  SET world xy coordinate to STONE
12              ELSE IF randValue is smaller than 70 THEN
13                  SET world xy coordinate to IRON_ORE
14              ELSE
15                  SET world xy coordinate to AIR
```

## 04 – displayWorld



## 06 – getBlockChar

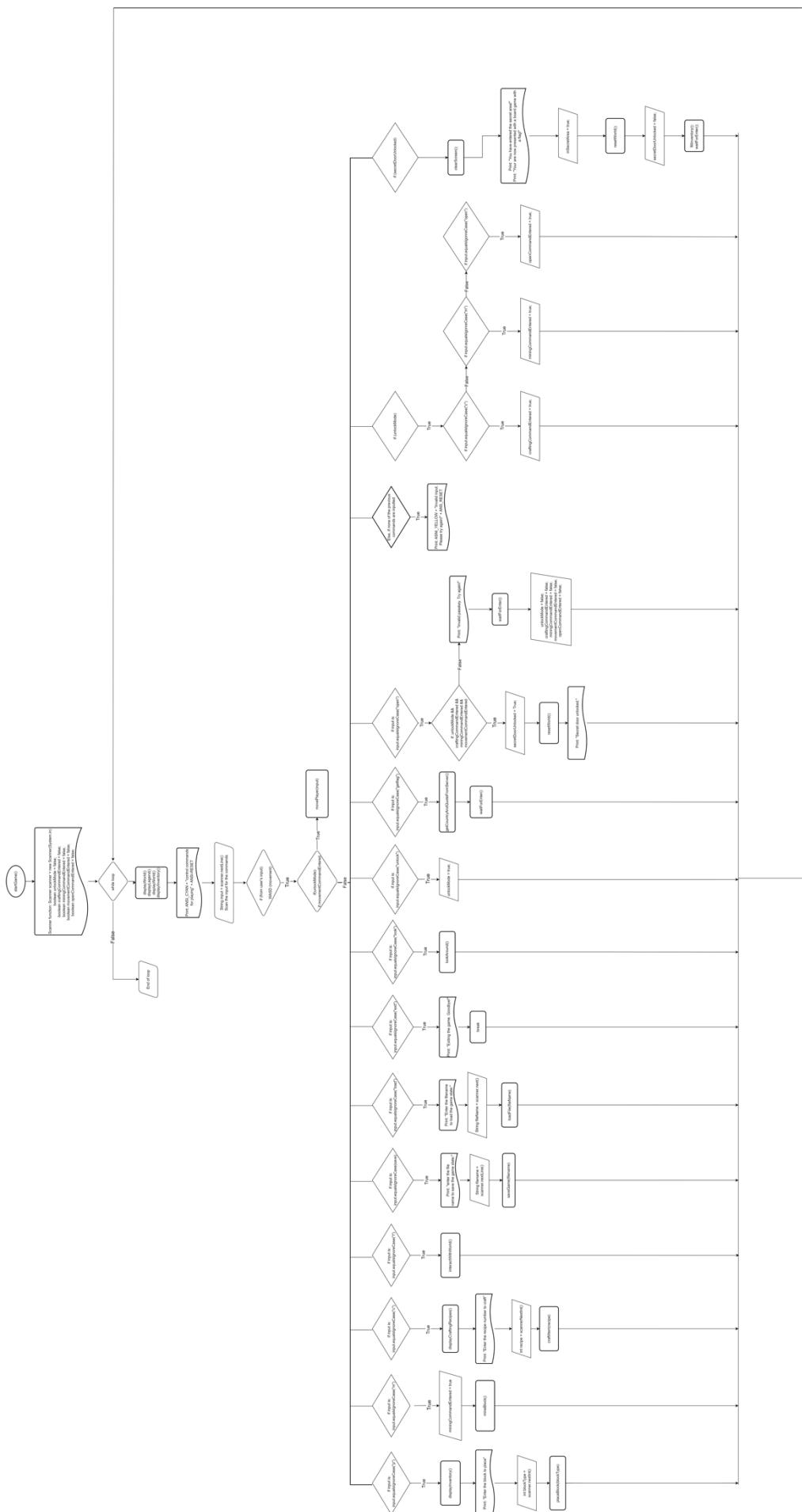
getBlockChar



Pseudocodes > ! 06-getBlockChar

```
1  FUNCTION getBlockChar
2      CASE block type
3          "WOOD":
4              RETURN: 2592
5          "LEAVES":
6              RETURN: 00A7
7          "STONE":
8              RETURN: 2593
9          "IRON_ORE":
10         RETURN: 00B0
11         "DEFAULT":
12         RETURN: nothing
13     ENDCASE
```

## 07 – startGame



## Pseudocodes &gt; 07-startGame

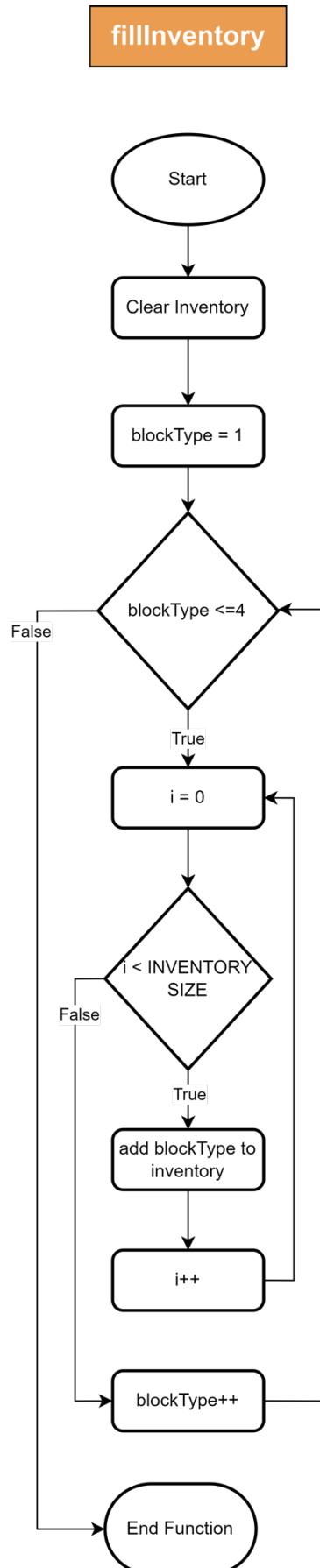
```
1 START GAME
2     scanner is equal to new Scanner(System.in)
3     unlockMode is false
4     craftingCommandEntered is false
5     miningCommandEntered is false
6     movementCommandEntered is false
7     openCommandEntered is false
8 WHILE true
9     clearScreen
10    displayLegend
11    displayWorld
12    displayInventory
13    PRINT "Enter your action: 'WASD': Move, 'M': Mine, 'P': Place, 'C': Craft,
14    'I': Interact, 'Save': Save, 'Load': Load, 'Exit': Quit, 'Unlock': Unlock Secret Door"
15    scanner scans the value for unserinput
16    IF any of these commands are entered (ignoring case) ("w" || "up", "s" || "down", "a" || "left", "d" || "right")
17        IF it is in unlockMode THEN
18            movementCommandEntered is true
19            movePLAYER(Input)
20        ELSE IF INPUT is "m"
21            miningCommandEntered is true
22            mineBlock
23        ELSE IF INPUT is "p"
24            displayInventory
25            PRINT "Enter the block type to place: "
26            blockType is input(integer)
27            placeBlock(blockType)
28        ELSE IF INPUT is "c"
29            displayCraftingRecipes
30            PRINT "Enter the recipe number to craft: "
31            recipe is input(integer)
32            craftItem(recipe)
33        ELSE IF INPUT is "i"
34            interactWithWorld
35        ELSE IF INPUT is "save"
36            PRINT "Enter the file name to save the game state: "
37            fileName is input
38            saveGame(fileName)
39        ELSE IF INPUT is "load"
40            PRINT "Enter the file name to load the game state: "
41            fileName is input
42            loadGame(fileName)
```

```

43     ELSE IF INPUT is "exit"
44         PRINT "Exiting the game. Goodbye!"
45         break
46     ELSE IF INPUT is "look"
47         lookAround
48     ELSE IF INPUT is "unlock"
49         unlockMode is true
50     ELSE IF INPUT is "getFlag"
51         getCountryAndQuoteFromServer
52         waitForEnter
53     ELSE IF INPUT is "open"
54         IF unlockMode && craftingCommandEntered && miningCommandEntered && movementCommandEntered THEN
55             secretDoorUnlocked is true
56             resetWorld
57             PRINT "Secret door unlocked!"
58             waitForEnter
59         ELSE
60             PRINT "Passkey invalid. Try again!"
61             unlockMode is false
62             craftingCommandEntered is false
63             miningCommandEntered is false
64             movementCommandEntered is false
65             openCommandEntered is false
66     ELSE
67         PRINT "Invalid input. Please try again" in yellow
68 IF unlockMode
69     IF INPUT is "c" THEN
70         craftingCommandEntered = true
71     ELSE IF INPUT is "m"
72         miningCommandEntered is true
73     ELSE IF INPUT is "open"
74         openCommandEntered is true
75 IF secretDoorUnlocked
76     clearScreen
77     PRINT "You have entered the secret area!"
78     PRINT "You are not presented with a board game with a flag!"
79     inSecretArea is true
80     resetWorld
81     secretDoorUnlocked is false
82     fillInventory
83     waitForEnter

```

## o8 – fillInventory

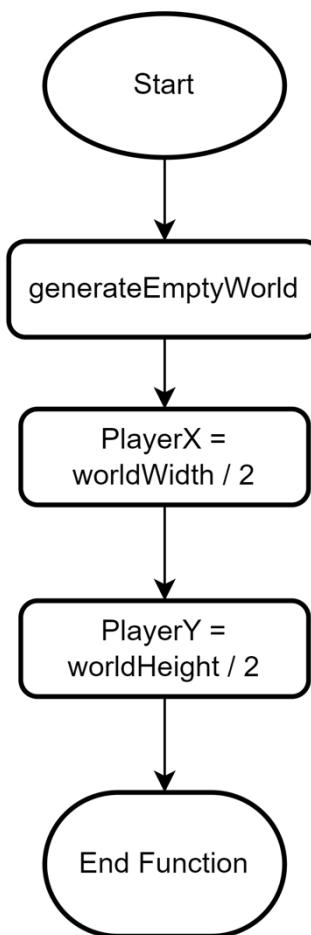


Pseudocodes > Ξ 08-fillInventory

```
1  FUNCTION fillInventory
2      CLEAR inventory
3      FOR each blockType from 1 to 4
4          FOR each in the inventory size
5              ADD block type to the inventory
```

## 09 – resetWorld

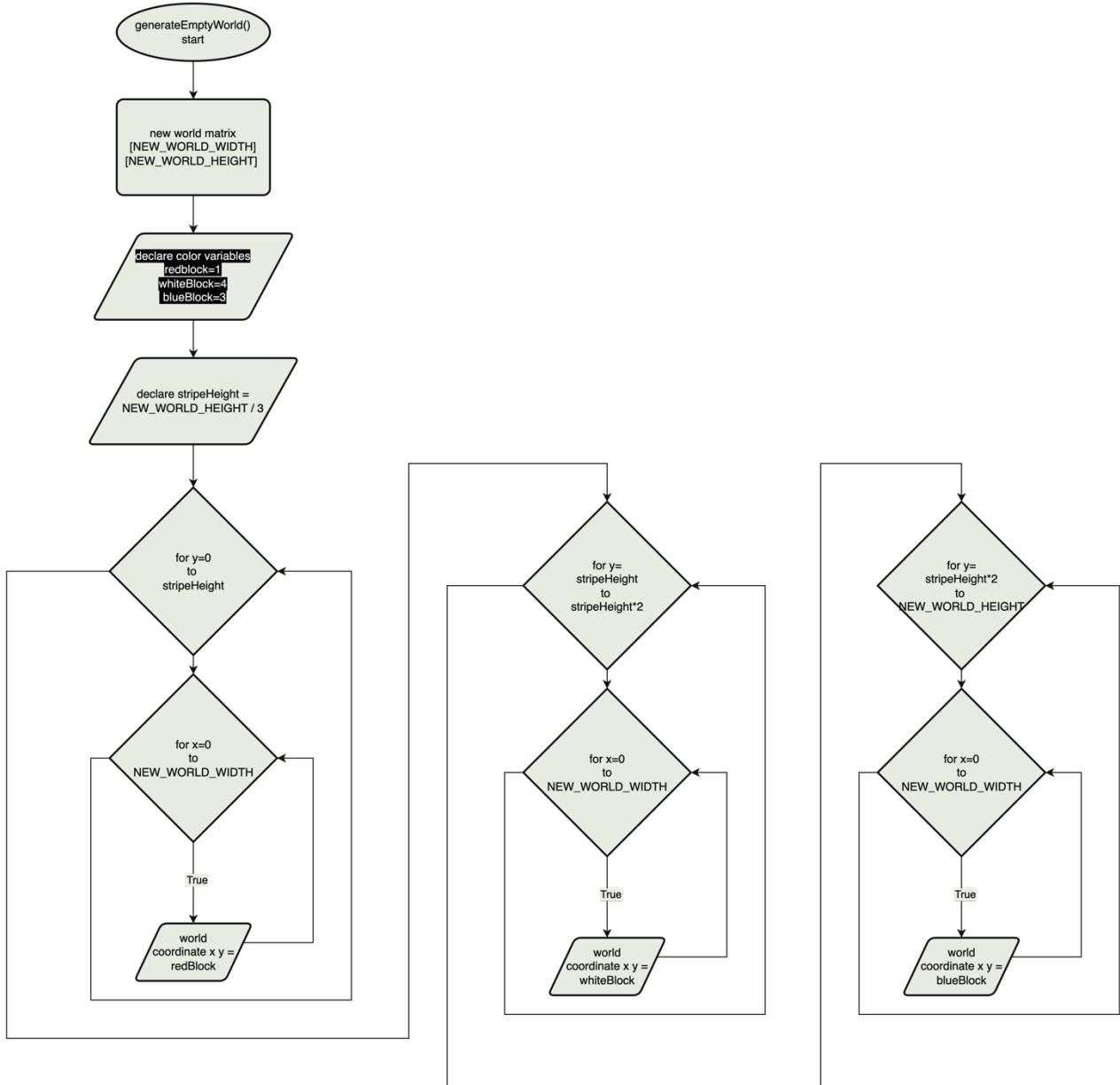
resetWorld



Pseudocodes > 09-resetWorld

```
1  FUNCTION resetWorld
2    GENERATE empty world
3    SET variable playerX to the worldwidth divided by 2
4    SET variable playerY to the worldheight divided by 2
```

## 10 – generateEmptyWorld

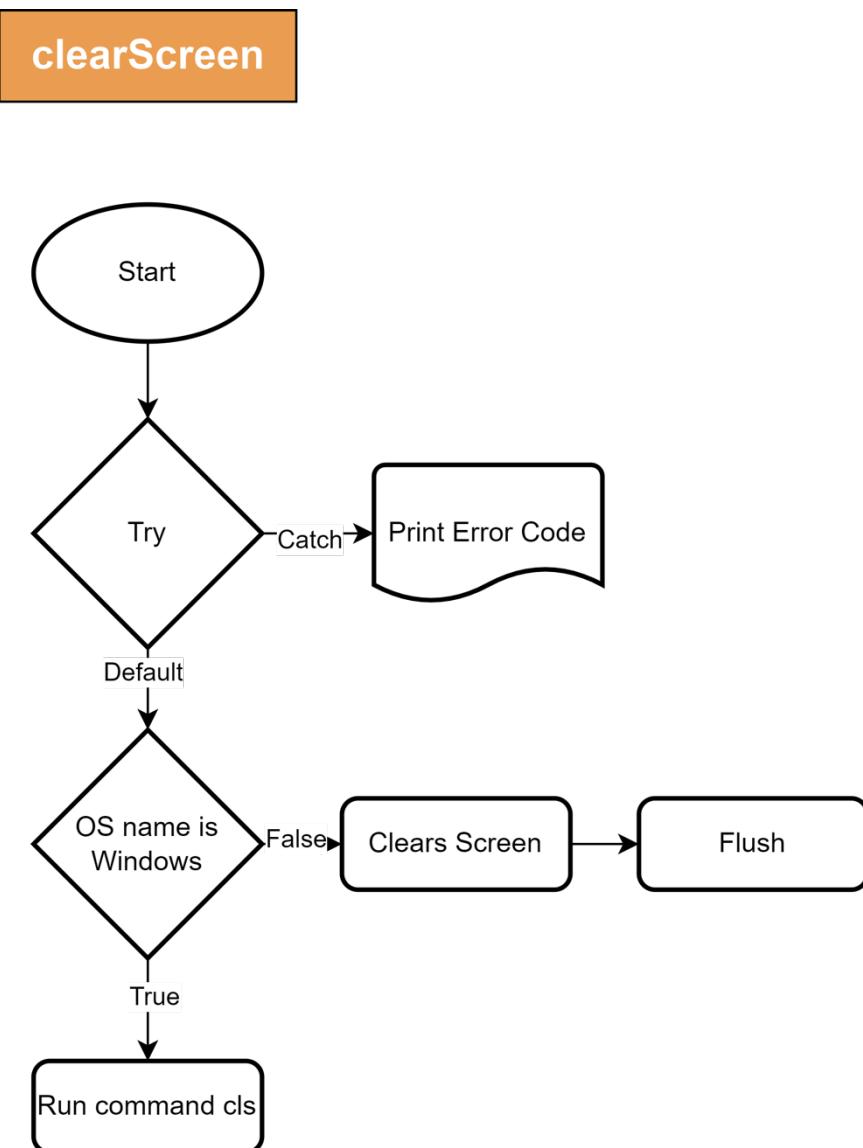


Pseudocodes > 10-generateEmptyWorld

```

1  FUNCTION generateEmptyWorld
2      SET world to integer array (NEW_WORLD_WIDTH, NEW_WORLD_HEIGHT)
3      SET redBlock to 1
4      SET whiteBlock to 4
5      SET blueBlock to 3
6      SET stripeHeight to one third of the NEW_WORLD_HEIGHT
7
8      FOR each y coordinate in stripeHeight
9          FOR each x coordinate in NEW_WORLD_WIDTH
10             SET world xy coordinate to redBlock
11
12     FOR each y coordinate between stripeHeight and 2 times stripeHeight
13         FOR each x coordinate in NEW_WORLD_WIDTH
14             SET world xy coordinate to whiteBlock
15
16     FOR each y coordinate between 2 times stripeHeight and NEW_WORLD_HEIGHT
17         FOR each x coordinate in NEW_WORLD_WIDTH
18             SET world xy coordinate to blueBlock
  
```

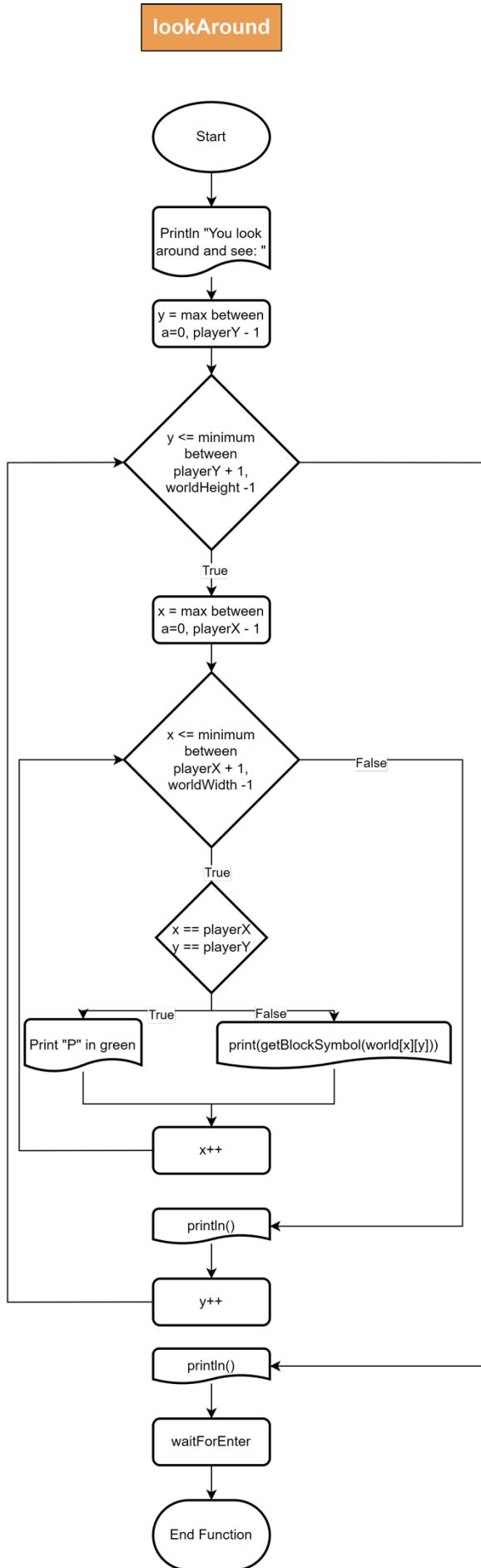
## 11 – clearScreen



Pseudocodes > ≡ 11-clearScreen

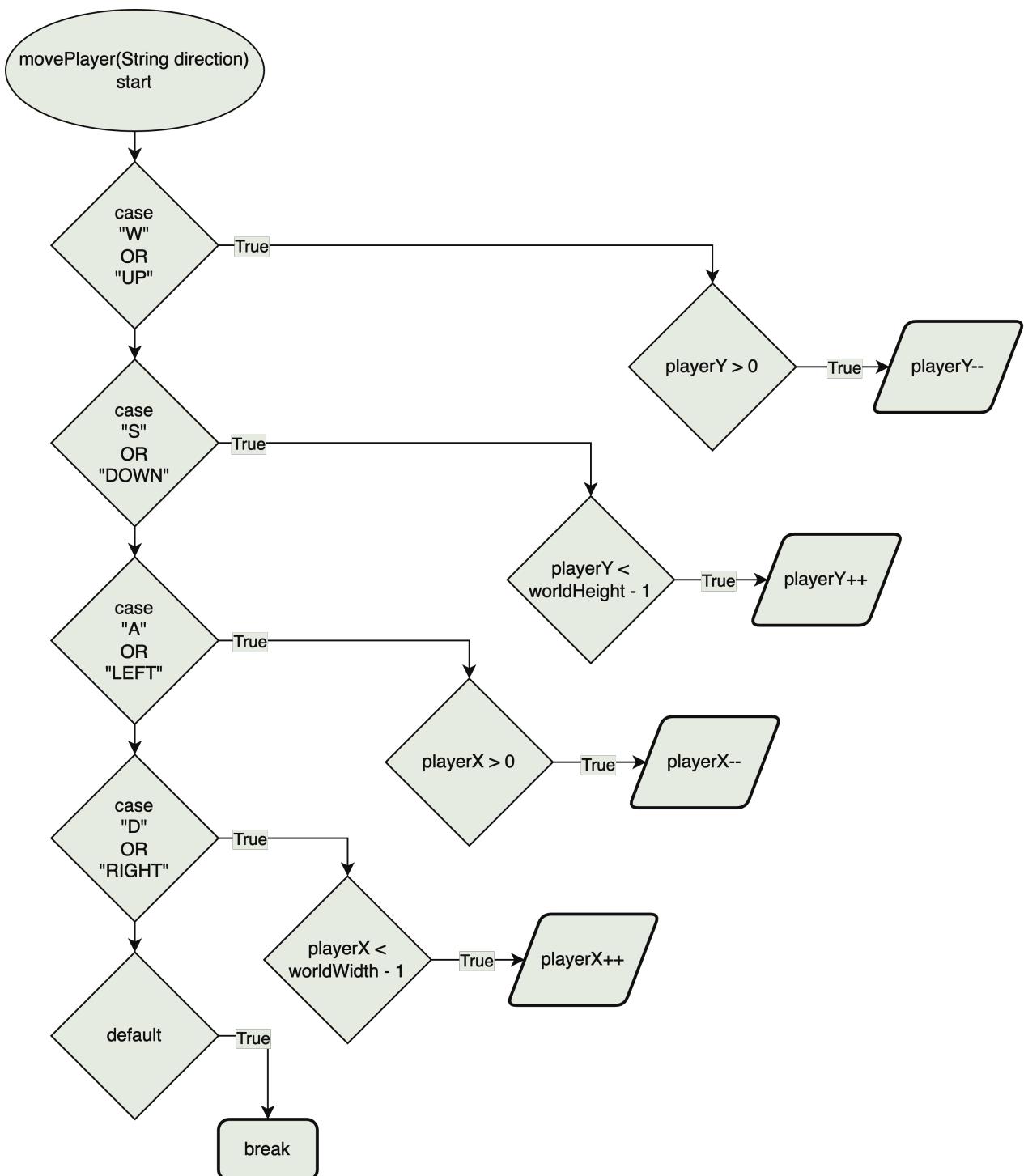
```
1  FUNCTION clearScreen
2      TRY unless an error happens
3          IF the OS name is Windows
4              RUN command cls
5          ELSE
6              CLEAR the screen
7              FLUSH
8      CATCH when an error happens
9          PRINT error code
```

## 12 – lookAround



```
Pseudocodes > ≡ 12-lookAround
1  FUNCTION lookAround
2    PRINT "You look around and see"
3    FOR each y coordinate that is lower or equal to the minimum between playerY +1, and the worldheight -1
4      FOR each x coordinate that is lower or equal to the minimum between playerX +1, and the worldwidth -1
5        IF the x coordinate equals playerX, and the y coordinate equals playerY
6          PRINT "P" in green
7        ELSE
8          PRINT getBlockSymbol(world[x][y])
9        PRINT a new line
10       PRINT a new line
11       WAIT for enter
```

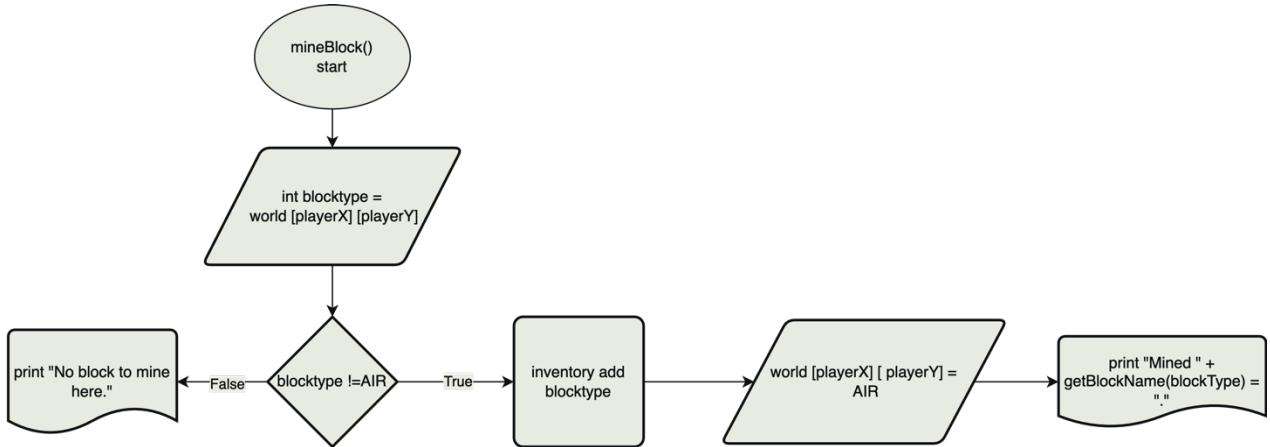
## 13 – movePlayer



Pseudocodes > ≡ 13-movePlayer

```
1  FUNCTION movePlayer
2      CASE uppercase direction
3          "W" OR "UP":
4              IF playerY is greater than 0 THEN
5                  DECREMENT playerY
6          "S" OR "DOWN":
7              IF playerY is smaller than wordHeight-1 THEN
8                  INCREMENT playerY
9          "A" OR "LEFT":
10             IF playerX is greater than 0 THEN
11                 DECREMENT playerX
12             "D" OR "RIGHT":
13                 IF playerX is smaller than worldWidth-1 THEN
14                     INCREMENT playerX
15             DEFAULT:
16         ENDCASE
```

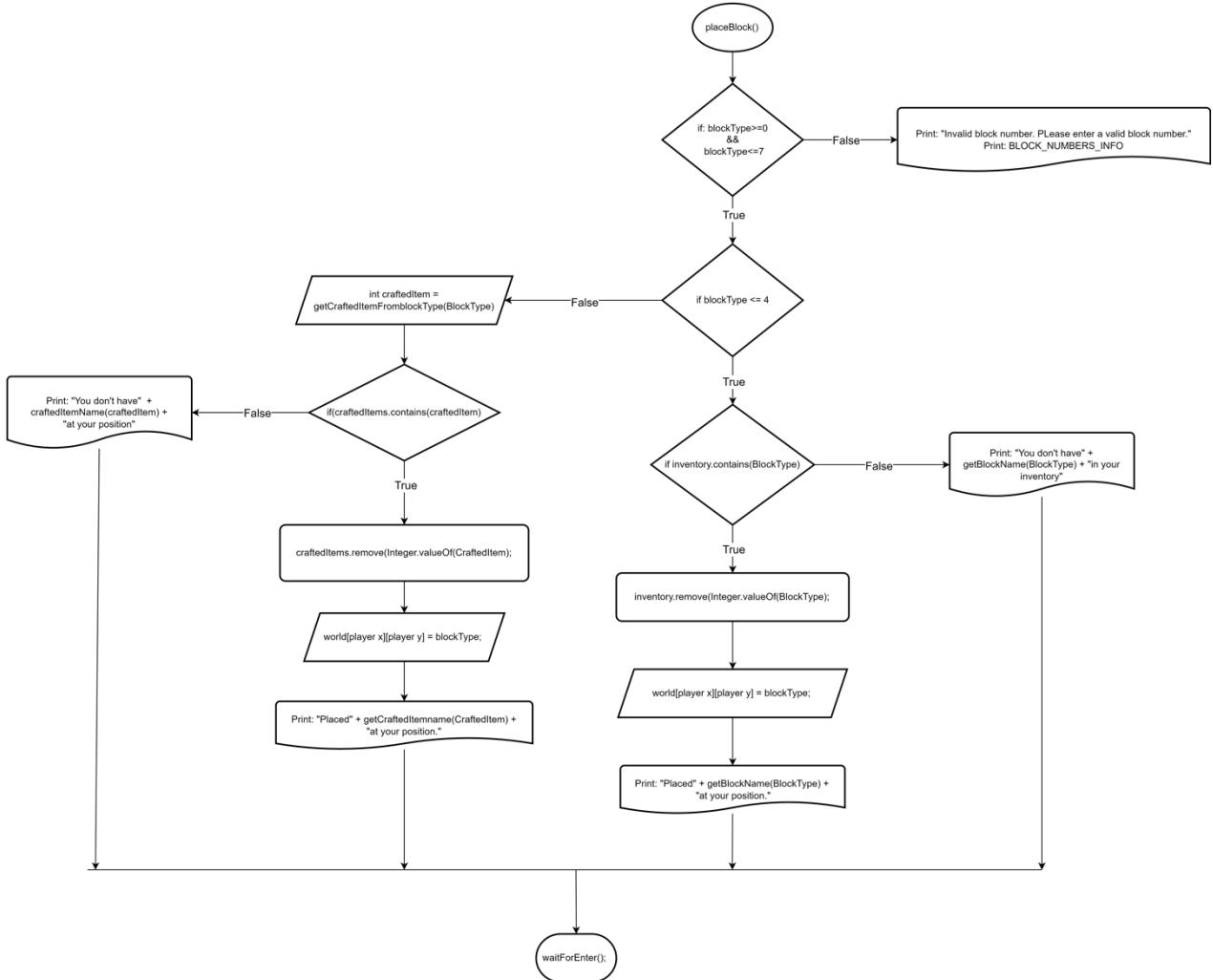
## 14 – mineBlock



### Pseudocodes > ≡ 14-mineBlock

```
1  ↘ FUNCTION mineBlock
2          SET blockType to world(playerX, playerY)
3  ↘      IF blockType is not AIR THEN
4          ADD inventory blockType
5          SET world(playerX,playerY) to AIR
6          PRINT "Mined" GET blockname "."
7  ↘      ELSE
8          PRINT "No block to mine here."
```

# 15 – placeBlock

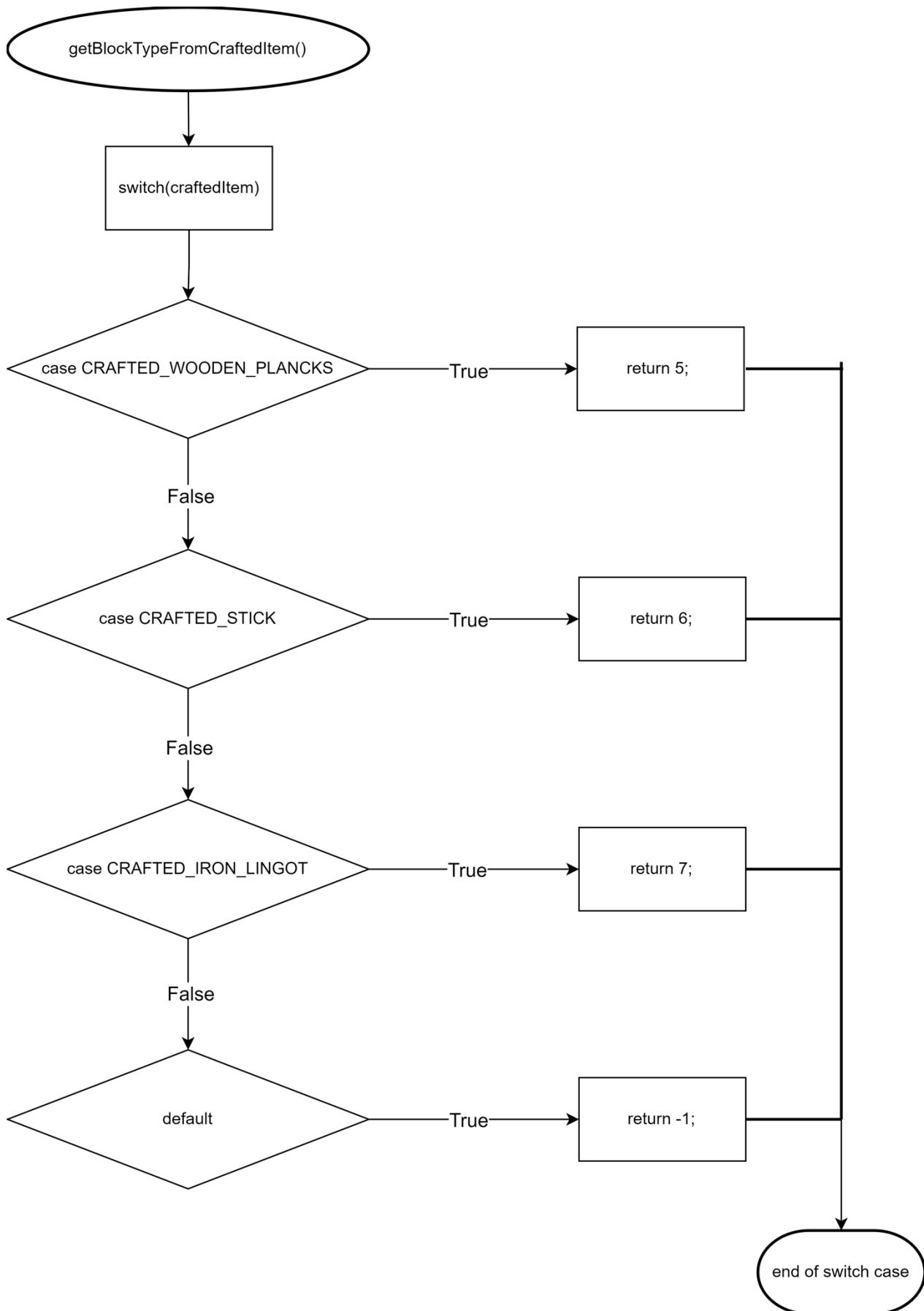


Pseudocodes > 15-placeBlock

```

1  PLACE BLOCK
2  IF blockType is greater or equal to 0 and blockType is also smaller or equal to 7
3    IF blockType is smaller or equal to 4
4      IF inventory contains blockType
5        Remove an integer value of blockType in inventory
6        world[player X][player Y] is equal to blockType
7        PRINT "Placed " + getBlockName(blockType) + " at your position."
8      ELSE
9        PRINT "You don't have " + getBlockName(blockType) + " in your inventory."
10     ELSE
11       craftedItem equals to getCraftedItemFromBlockType(blockType)
12       IF craftedItems contains craftedItem THEN
13         Remove an integer value in craftedItem
14         world[playerX][playerY] is equal to blockType
15         PRINT "Placed " + getCraftedItemName(craftedItem) + " at your position."
16       ELSE
17         PRINT "You don't have " + getCraftedItemName(craftedItem) + " in your crafted items."
18     ELSE
19       PRINT "Invalid block number. Please enter a valid block number."
20       PRINT BLOCK_NUMBERS_INFO
21   waitForEnter
  
```

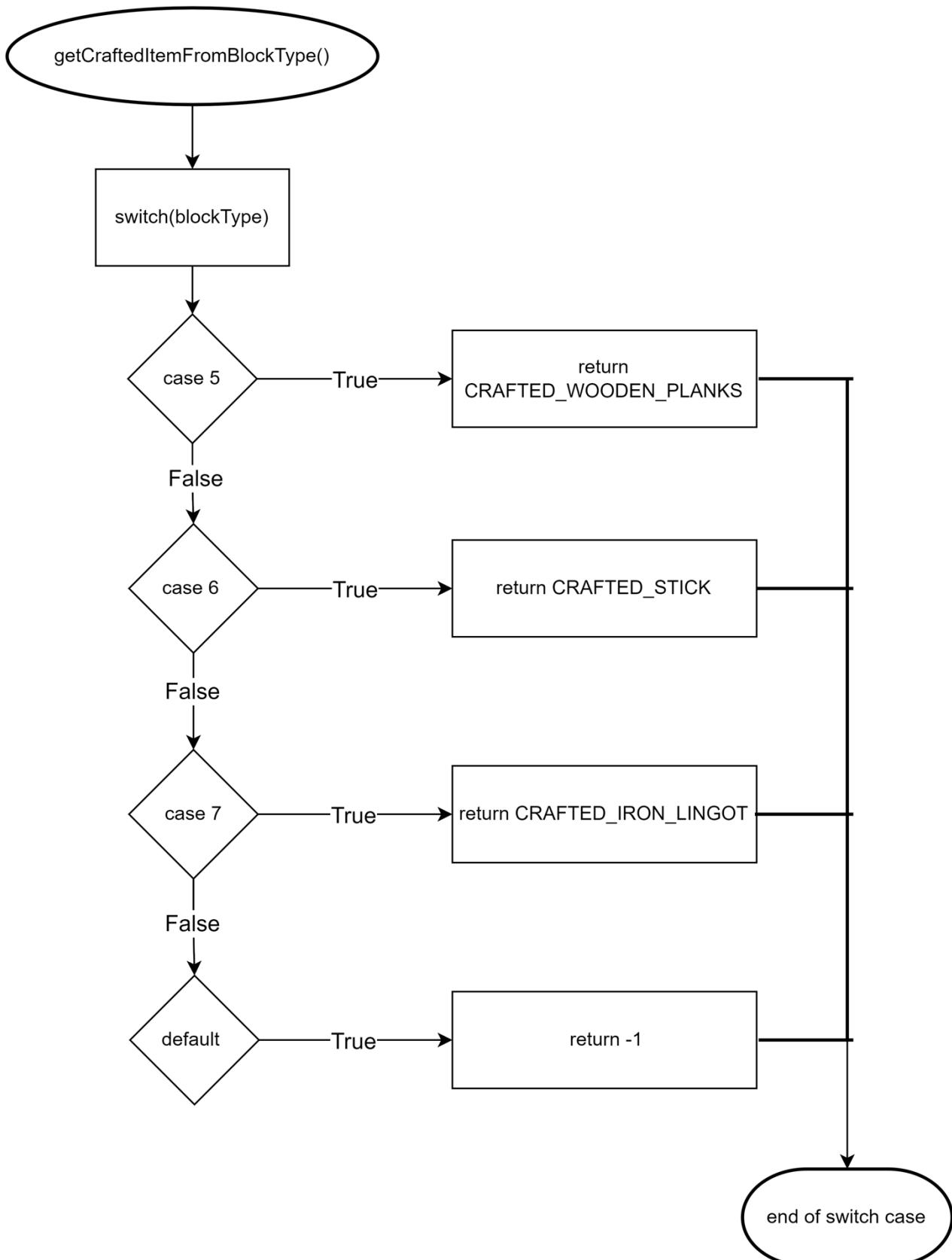
## 16 – getBlockTypeFromCraftedItem



Pseudocodes > ≡ 16-getBlockTypeFromCraftedItem

```
1   GET BLOCK TYPE FROM CRAFTED ITEM
2       SWITCH craftedItem
3           CASE CRAFTED_WOODEN_PLANKS:
4               return 5
5           CASE CRAFTED_STICK:
6               return 6
7           CASE CRAFTED_IRON_INGOT:
8               return 7
9           DEFAULT:
10              return -1
```

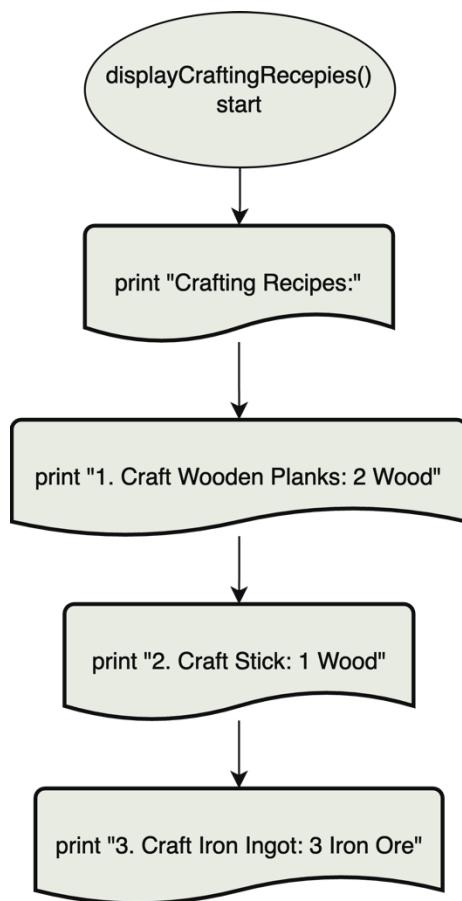
## 17 – getCraftedItemFromBlockType



Pseudocodes > ≡ 17-getCraftedItemFromBlockType

```
1   GET CRAFTED ITEM FROM BLOCK TYPE
2       SWITCH blockType
3           CASE 5
4               return CRAFTED_WOODEN_PLANKS
5           CASE 6
6               return CRAFTED_STICK
7           CASE 7
8               return CRAFTED_IRON_INGOT
9       DEFAULT
10      return -1
```

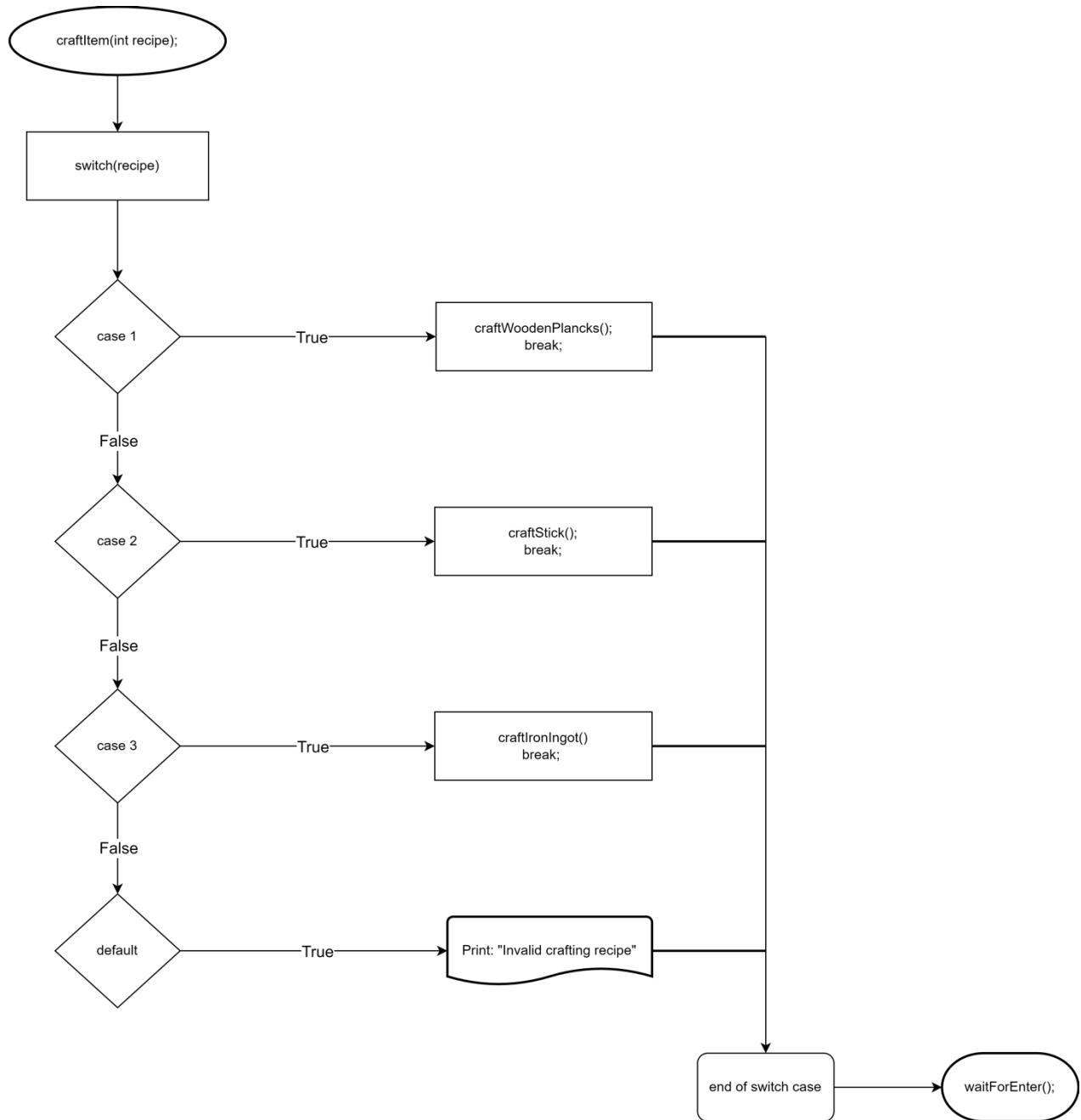
## 18 – displayCraftingRecepies



Pseudocodes > 18-displayCraftingRecipes

```
1  FUNCTION displayCraftingRecipes
2      PRINT "Crafting Recipes:"
3      PRINT "1. Craft Wooden Planks: 2 Wood"
4      PRINT "2. Craft Stick: 1 Wood"
5      PRINT "3. Craft Iron Ingot: 3 Iron Ore"
```

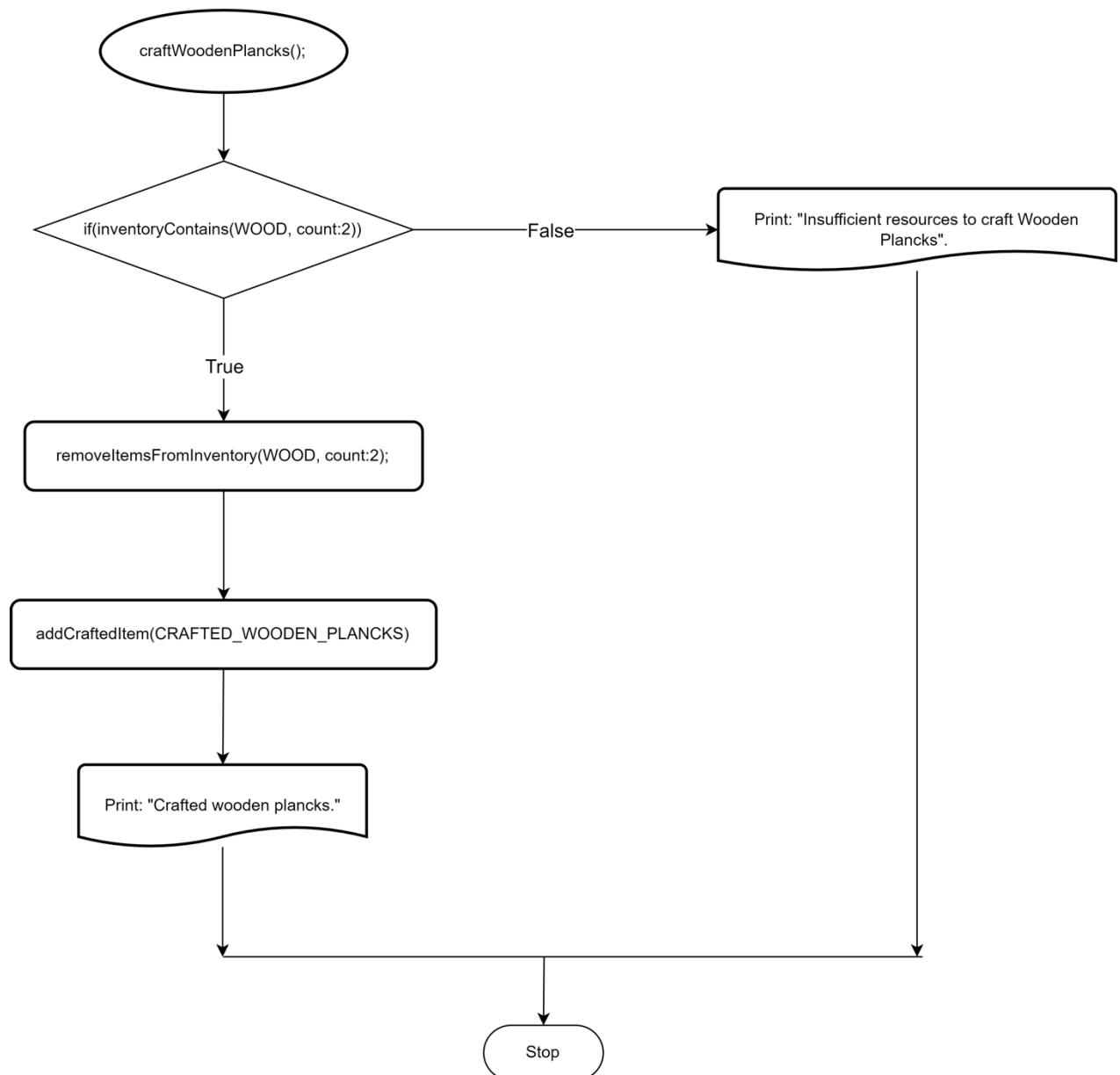
## 19 – craftItem



Pseudocodes > 19-craftItem

```
1 CRAFT ITEM
2     SWITCH recipe
3         CASE 1
4             craftWoodenPlanks
5             break
6         CASE 2
7             craftStick
8             break
9         CASE 3
10            craftIronIngot
11            break
12        DEFAULT
13            PRINT "Invalid recipe number."
14    waitForEnter
```

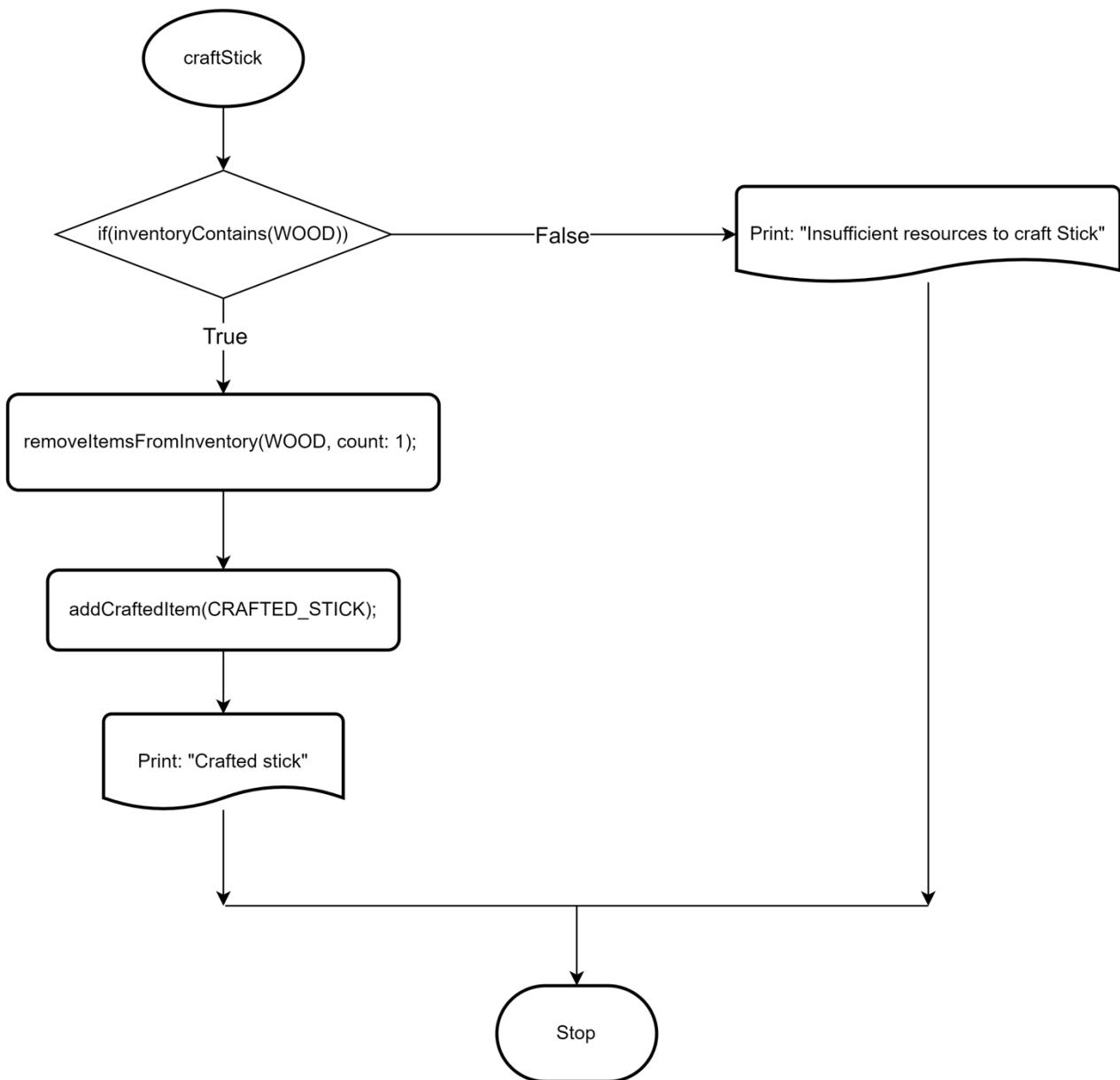
## 20 – craftWoodenPlanks



Pseudocodes > 20-craftWoodenPlanks

```
1 CRAFT WOODEN PLANKS
2 IF inventory contains wood (2 units min) THEN
3     Remove 2 units of wood from inventory
4     addCraftedItem(CRAFTED_WOODEN_PLANKS)
5     PRINT "Crafted wooden planks"
6 ELSE
7     PRINT "Insufficient resources to craft Wooden Plancks. "
```

## 21 – craftStick



Pseudocodes > ≡ 21-craftStick

```
1 CRAFT STICK
2     IF inventory contains wood THEN
3         Remove 1 unit of wood from inventory
4         addCraftedItem(CRAFTED_STICK)
5         PRINT "Crafted Stick."
6     ELSE
7         PRINT "Insufficient resources to craft Stick."
```

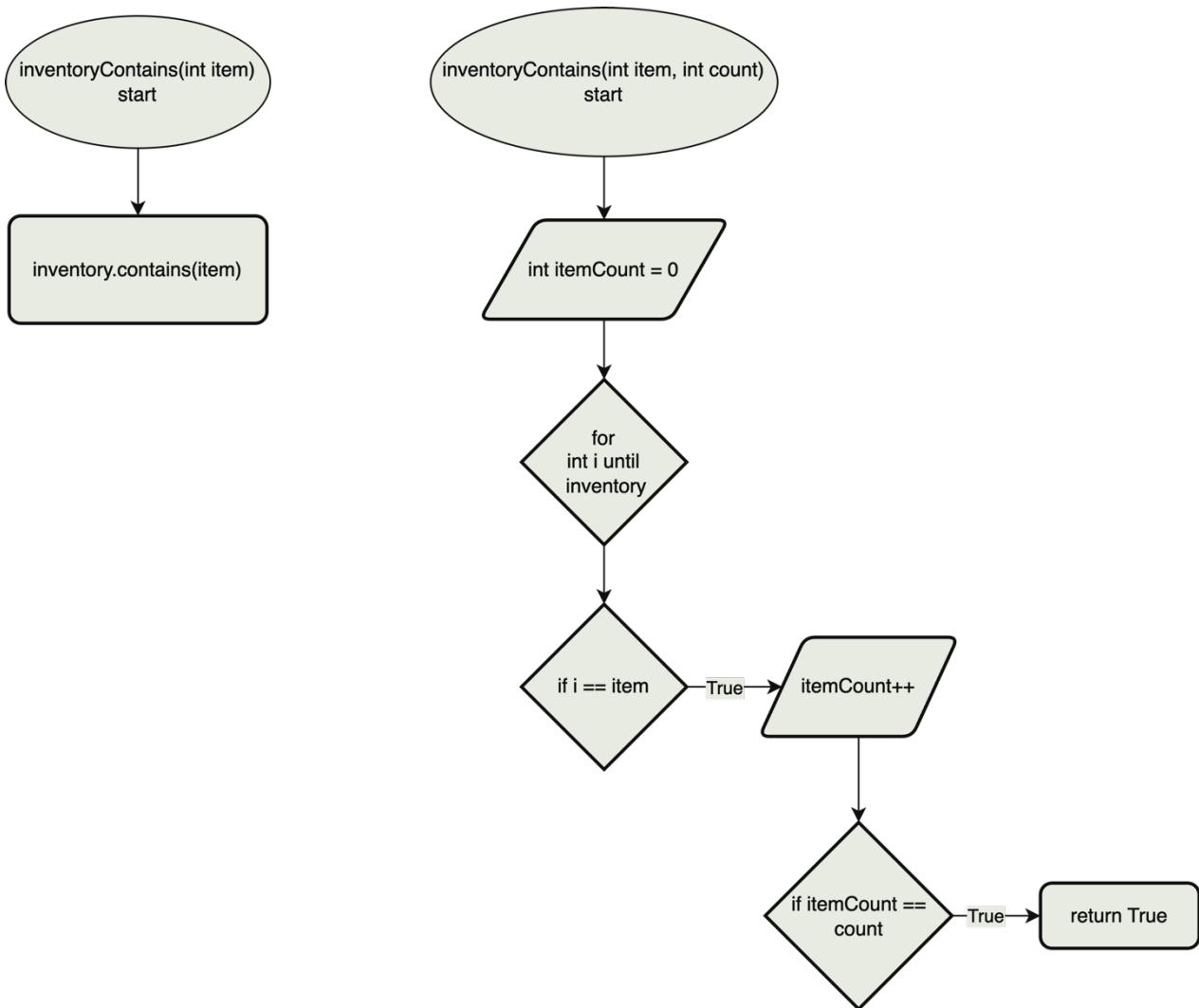
## 22 – craftIronIngot



Pseudocodes > 22-craftIronIngot

```
1 CRAFT_IRON_INGOT
2     IF inventory contains iron ore (3 units min) THEN
3         Remove 3 units of iron ore from inventory
4         addCraftedItem(CRAFTED_IRON_INGOT)
5         PRINT "Crafted Iron Ingot."
6     ELSE
7         PRINT "Insufficient resources to craft Iron Ingot"
```

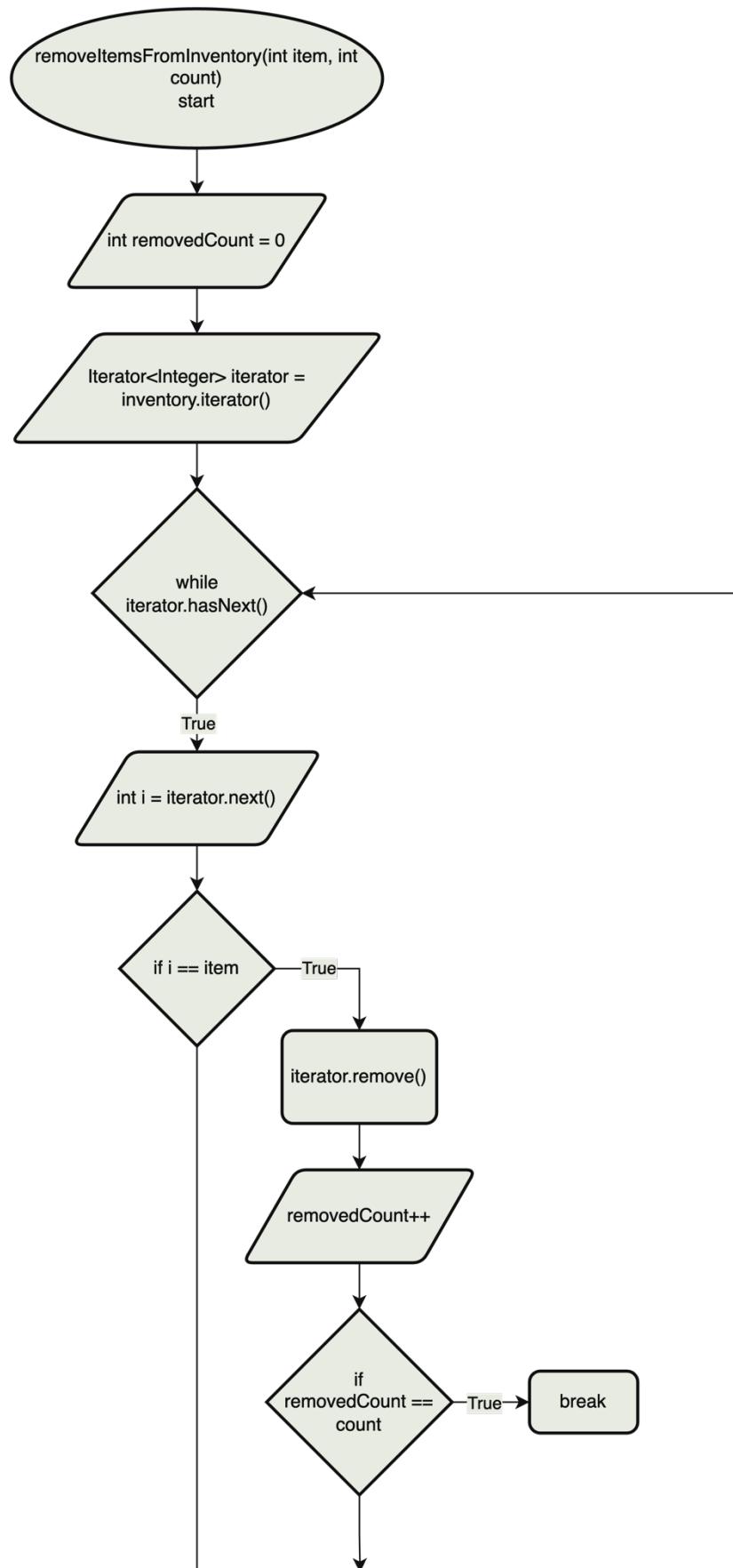
## 23-24 – inventoryContains



Pseudocodes > ≡ 23-24-inventoryContains

```
1  FUNCTION inventoryContains(integer item)
2      RETURN wheter inventory contains(item)
3
4  FUNCTION inventoryContains(integer item, integer count)
5      SET itemCount to 0
6      FOR each i in inventory
7          IF i equals to item THEN
8              INCREMENT itemCount
9              IF itemCount equals count THEN
10                 RETURN true
11
12             RETURN false
```

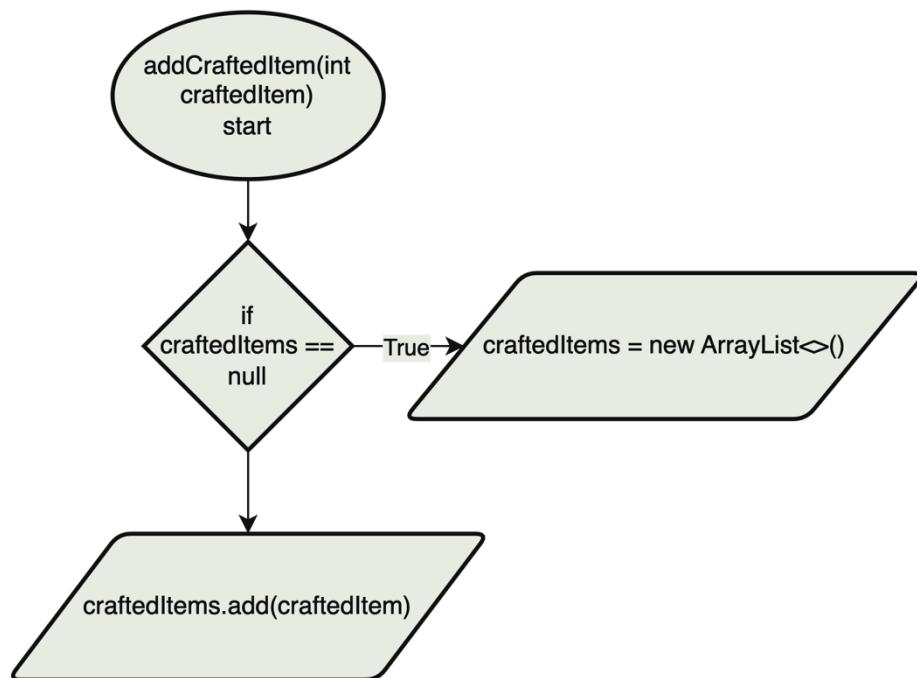
## 25 – removeItemsFromInventory



Pseudocodes > ≡ 25-removeItemsFromInventory

```
1   FUNCTION removeItemsFromInventory
2     SET removedCount to 0
3     NEW iterator inventory.iterator
4     WHILE iterator has next item
5       SET i to next iterator item
6       IF i equals to item
7         REMOVE iterator item
8         INCREMENT removedCount
9         IF removedCount equals to count THEN
10           BREAK
```

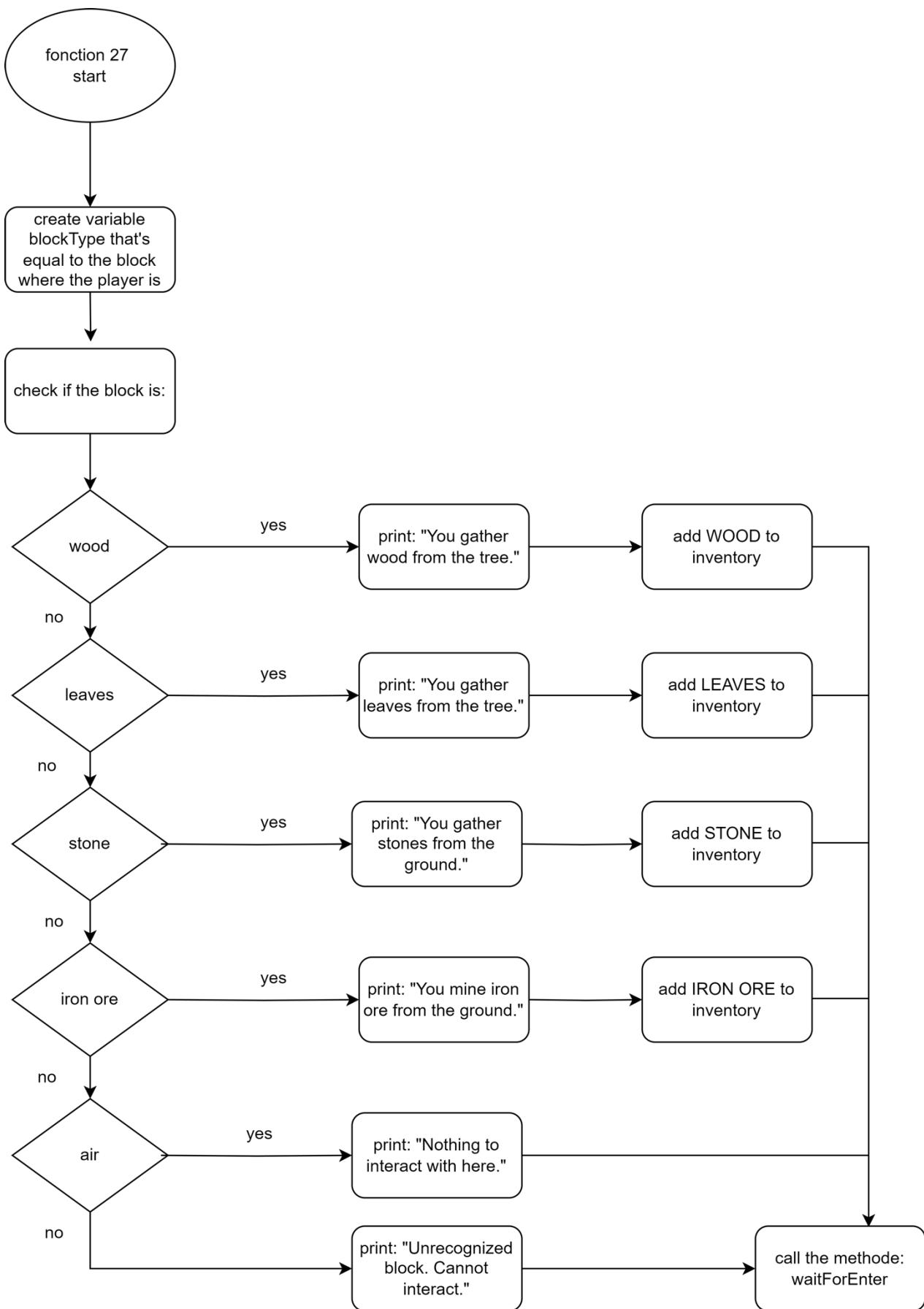
## 26 – addCraftedItem



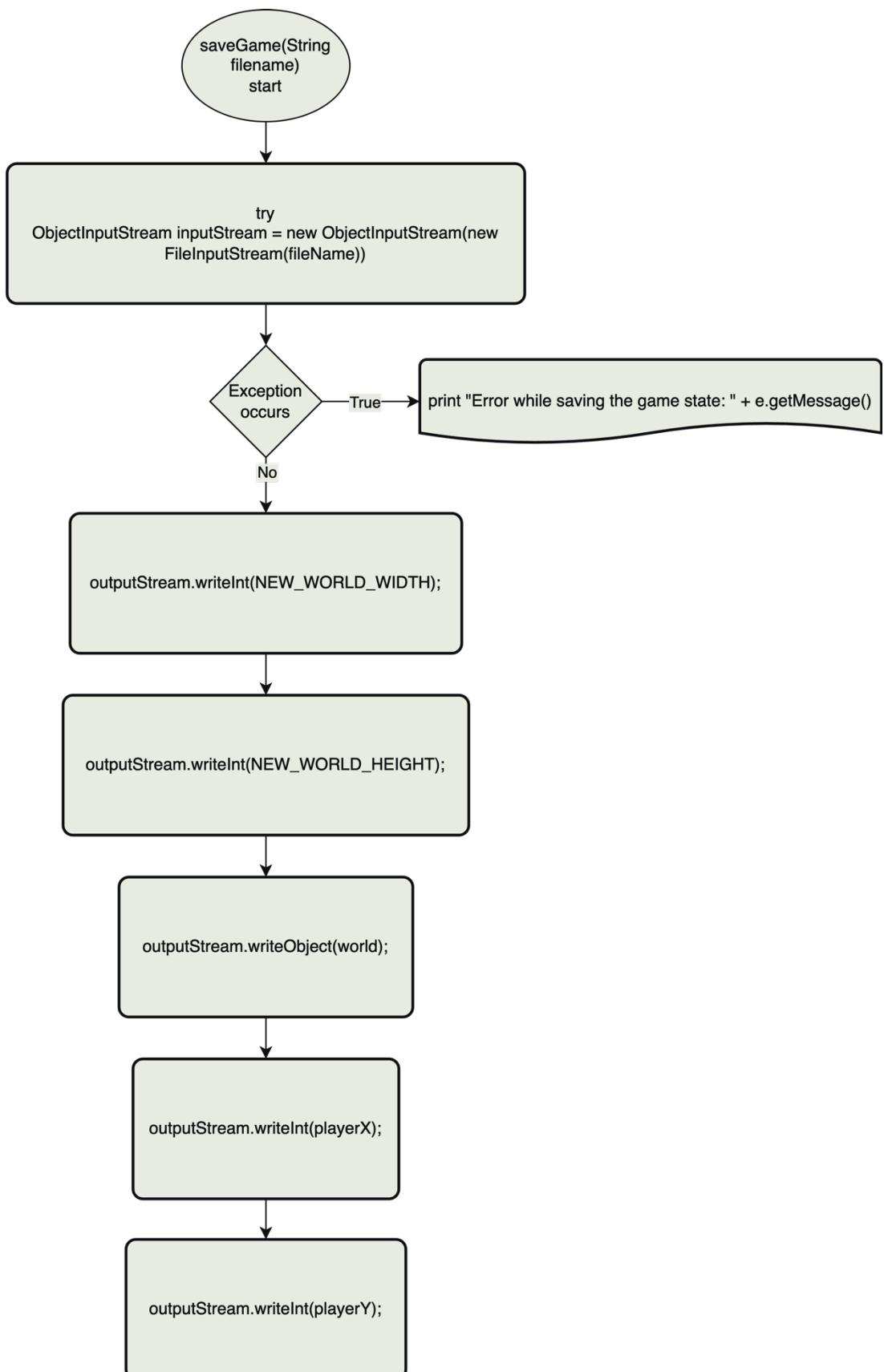
Pseudocodes > 26-addCraftedItem

```
1  FUNCTION addCraftedItem
2    IF craftedItems equals to null THEN
3      NEW ArrayList craftedItems
4      ADD craftedItem to craftedItems
```

## 27 – interactWithWorld

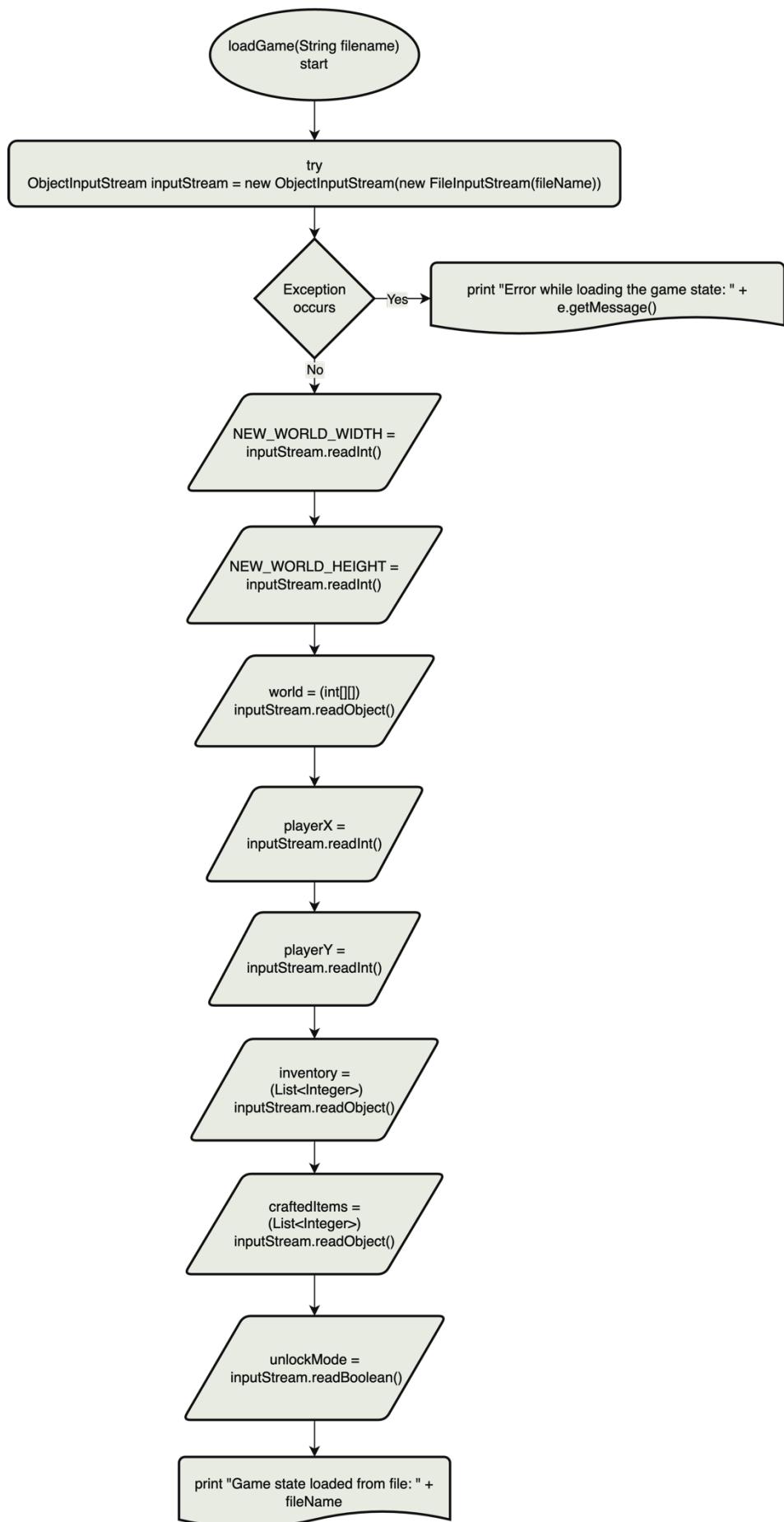


## 28 – saveGame



```
FUNCTION saveGame(string fileName)
    TRY NEW ObjectOutputStream(NEW FileOutputStream(fileName)) outputStream
        WRITE NEW_WORLD_WIDTH to outputStream
        WRITE NEW_WORLD_HEIGHT to outputStream
        WRITE world to outputStream
        WRITE playerX to outputStream
        WRITE playerY to outputStream
        WRITE inventory to outputStream
        WRITE craftedItems to outputStream
        WRITE unlockMode to outputStream
        PRINT "Game state saved to file: " + fileName
    CATCH
        PRINT "Error while saving the game state: " + error message
    waitForEnter()
```

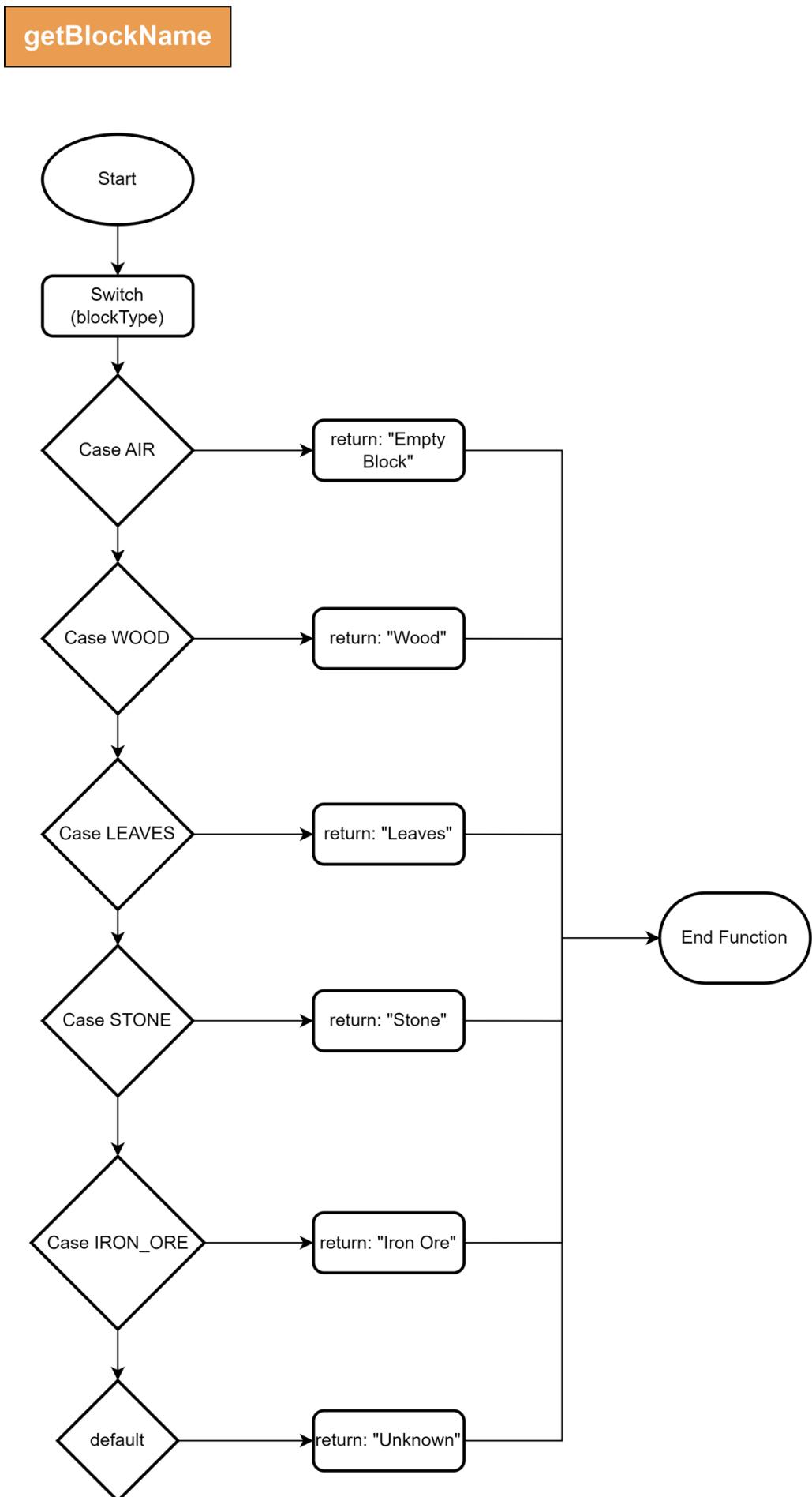
## 29 – loadGame



Pseudocodes > 29-loadGame

```
1  FUNCTION loadGame(string fileName)
2      TRY NEW ObjectInputStream(NEW FileInputStream(fileName)) inputStream
3          SET NEW_WORLD_WIDTH to READ inputStream
4          SET NEW_WORLD_HEIGHT to READ inputStream
5          SET world to READ inputStream
6          SET playerX to READ inputStream
7          SET playerY to READ inputStream
8          SET inventory to READ inputStream
9          SET craftedItems to READ inputStream
10         SET unlockMode to REAF inputStream
11         PRINT "Game state loaded from file: " + fileName
12     CATCH
13         PRINT "Error while loading the game state: " + error message
14     waitForEnter()
```

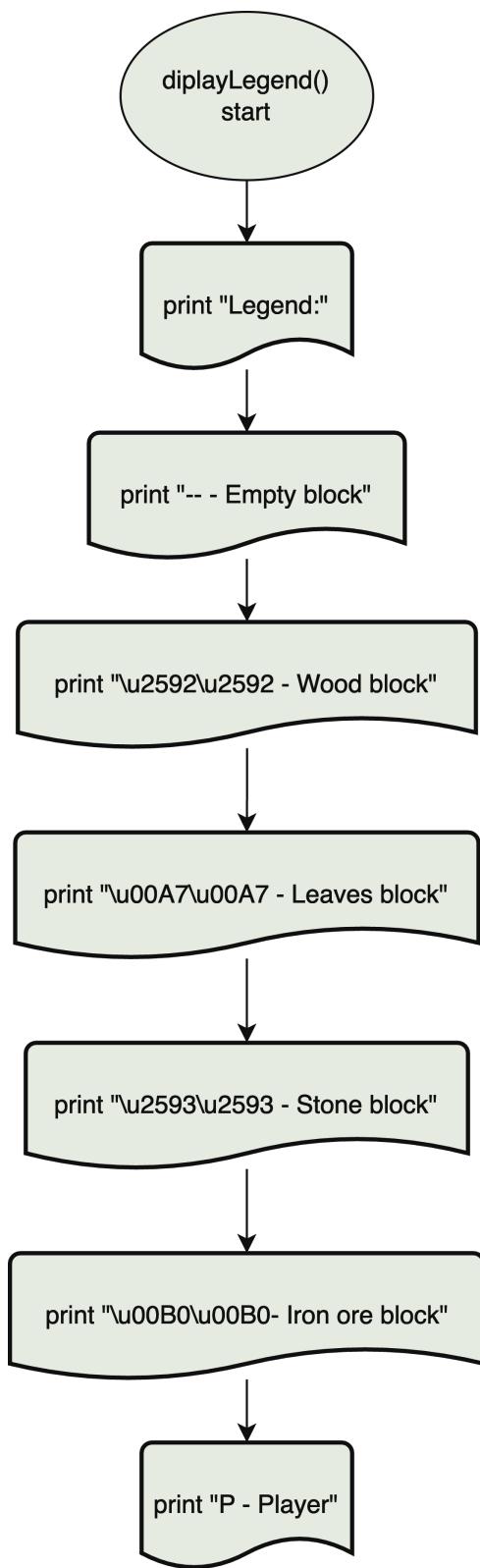
## 30 – getBlockName



## Pseudocodes > 30-getBlockName

```
~/javacraft/Pseudocodes lockName
2      CASE block type
3          "AIR":
4              RETURN: "Empty block"
5          "WOOD":
6              RETURN: "Wood"
7          "LEAVES":
8              RETURN: "Leaves"
9          "STONE":
10         RETURN: "Stone"
11         "IRON_ORE":
12         RETURN: "Iron Ore"
13         "DEFAULT":
14         RETURN: "Unknown"
15     ENDCASE
```

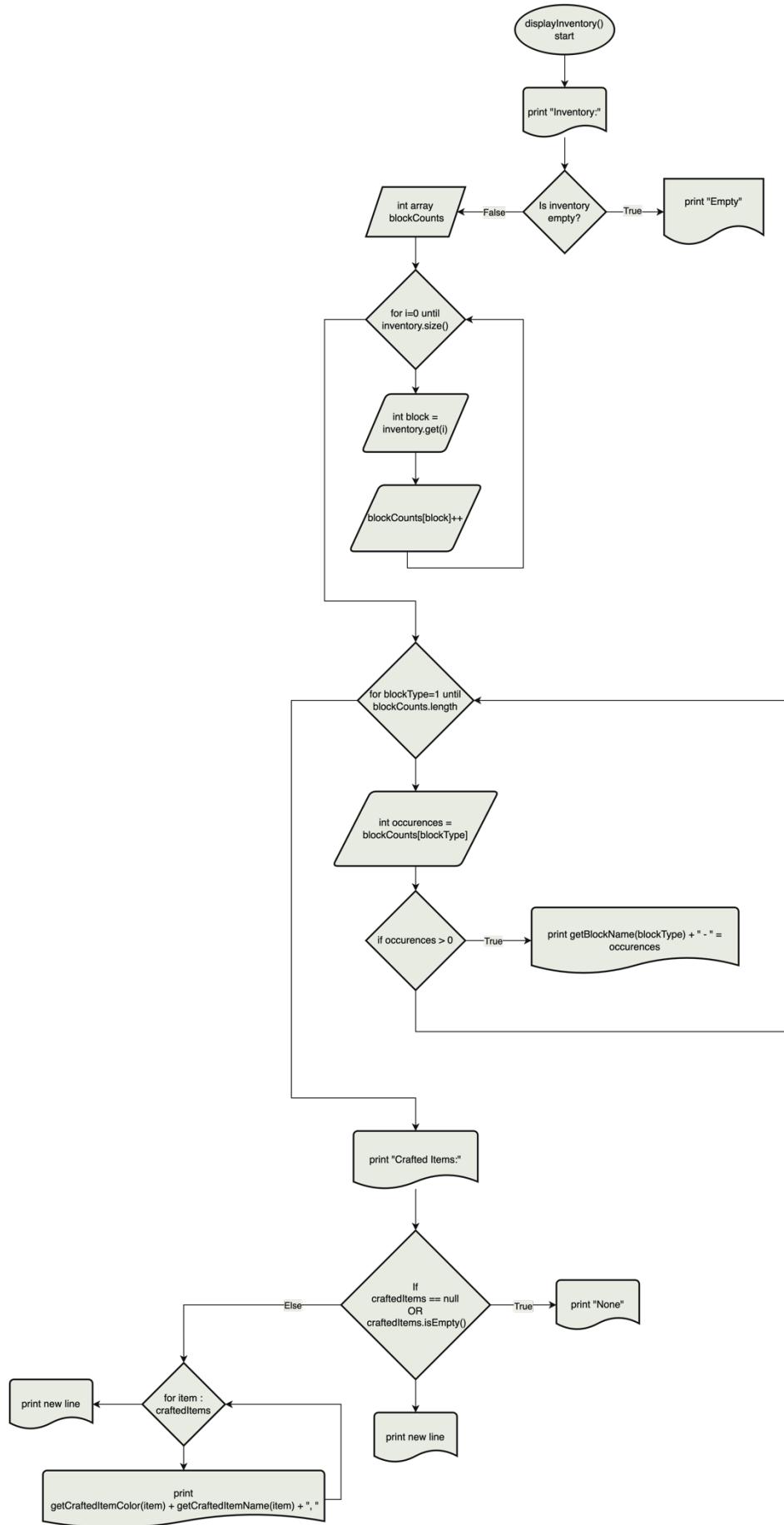
## 31 – displayLegend



## Pseudocodes > 31-displayLegend

```
1   FUNCTION displayLegend
2       PRINT "Legend:" in blue
3       PRINT "-- - Empty block" in white
4       PRINT "█ - Wood block" in red
5       PRINT "§§ - Leaves block" in green
6       PRINT "██ - Stone block" in blue
7       PRINT "○○- Iron ore block" in white
8       PRINT "P - Player" in blue
```

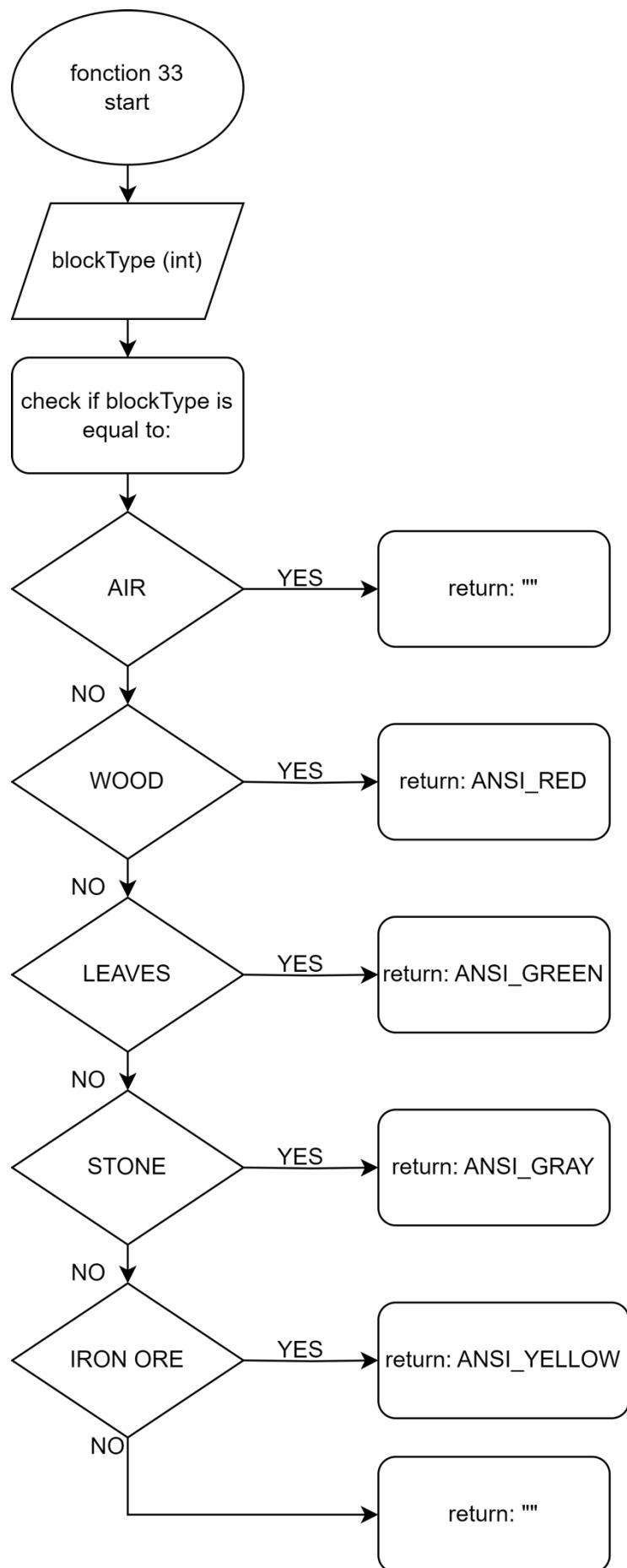
## 32 – displayInventory



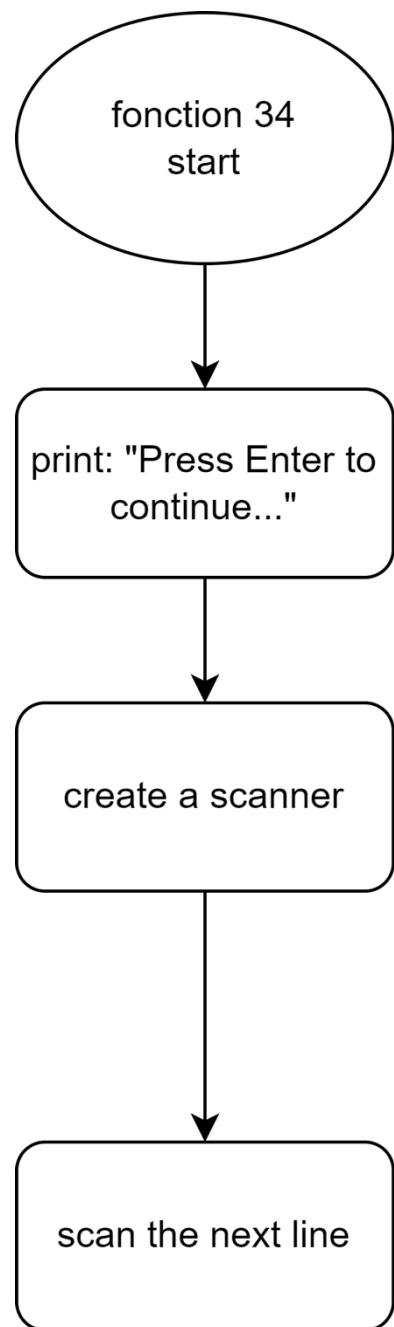
Pseudocodes > 32-displayInventory

```
1  FUNCTION displayInventory
2      PRINT "Inventory:"
3      IF inventory is empty THEN
4          PRINT "Empty" in yellow
5      ELSE
6          SET blockCounts to NEW integer array[5]
7          FOR each i in inventory
8              SET block to GET inventory(i)
9              INCREMENT blockCounts[block]
10         FOR each blockType in blockCounts
11             SET occurrences to blockCounts[blockType]
12             IF occurrences bigger than 0 THEN
13                 PRINT getBlockName(blockType) + " - " + occurrences
14             PRINT "Crafted Items:"
15             IF craftedItems equals to null THEN
16                 PRINT "None" in yellow
17             ELSE
18                 FOR each item in craftedItems
19                     PRINT getCraftedItemColor(item) + getCraftedItemName(item) + ", "
20             PRINT NEW LINE
21             PRINT NEW LINE
```

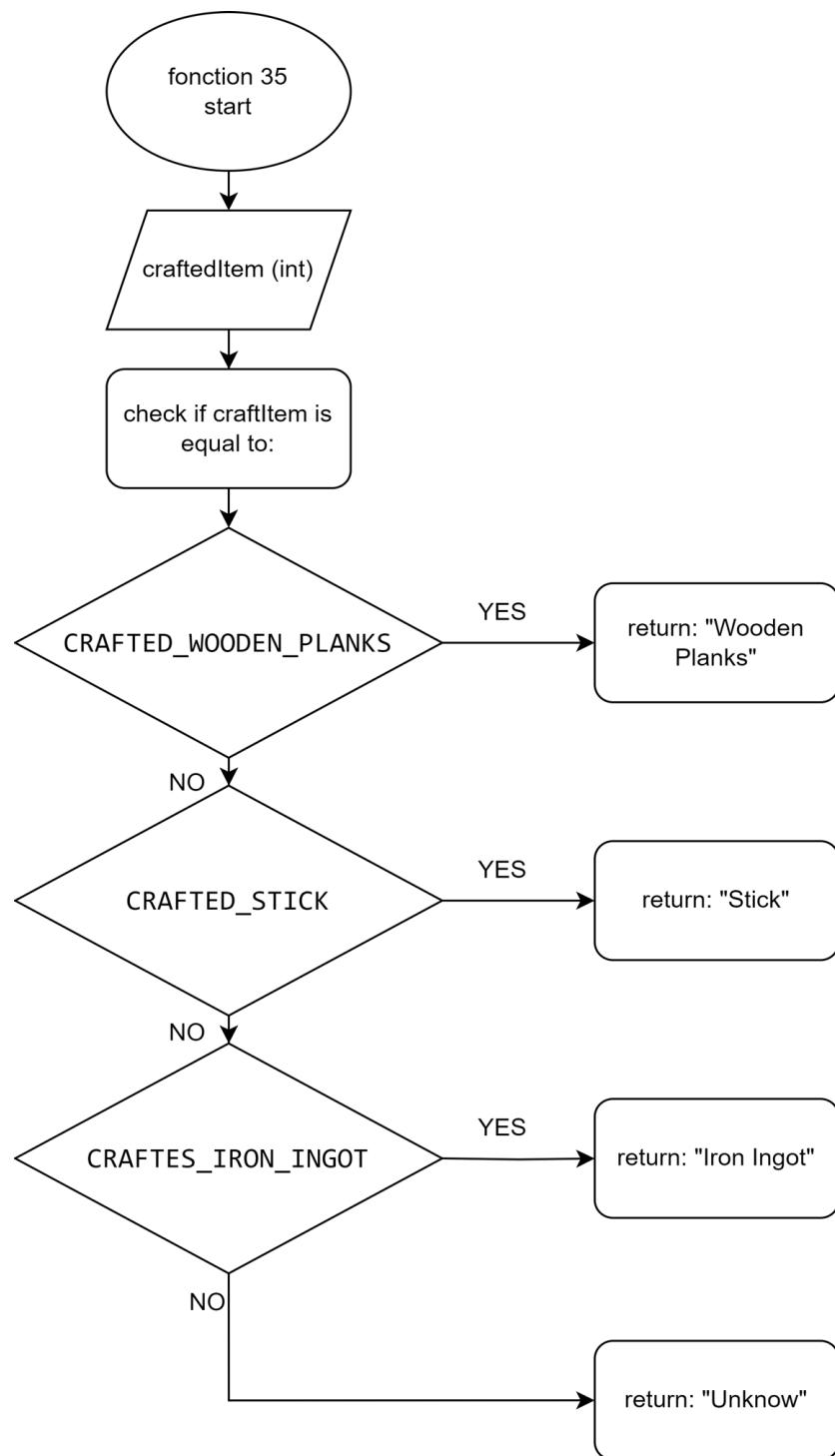
## 33 – getBlockColor



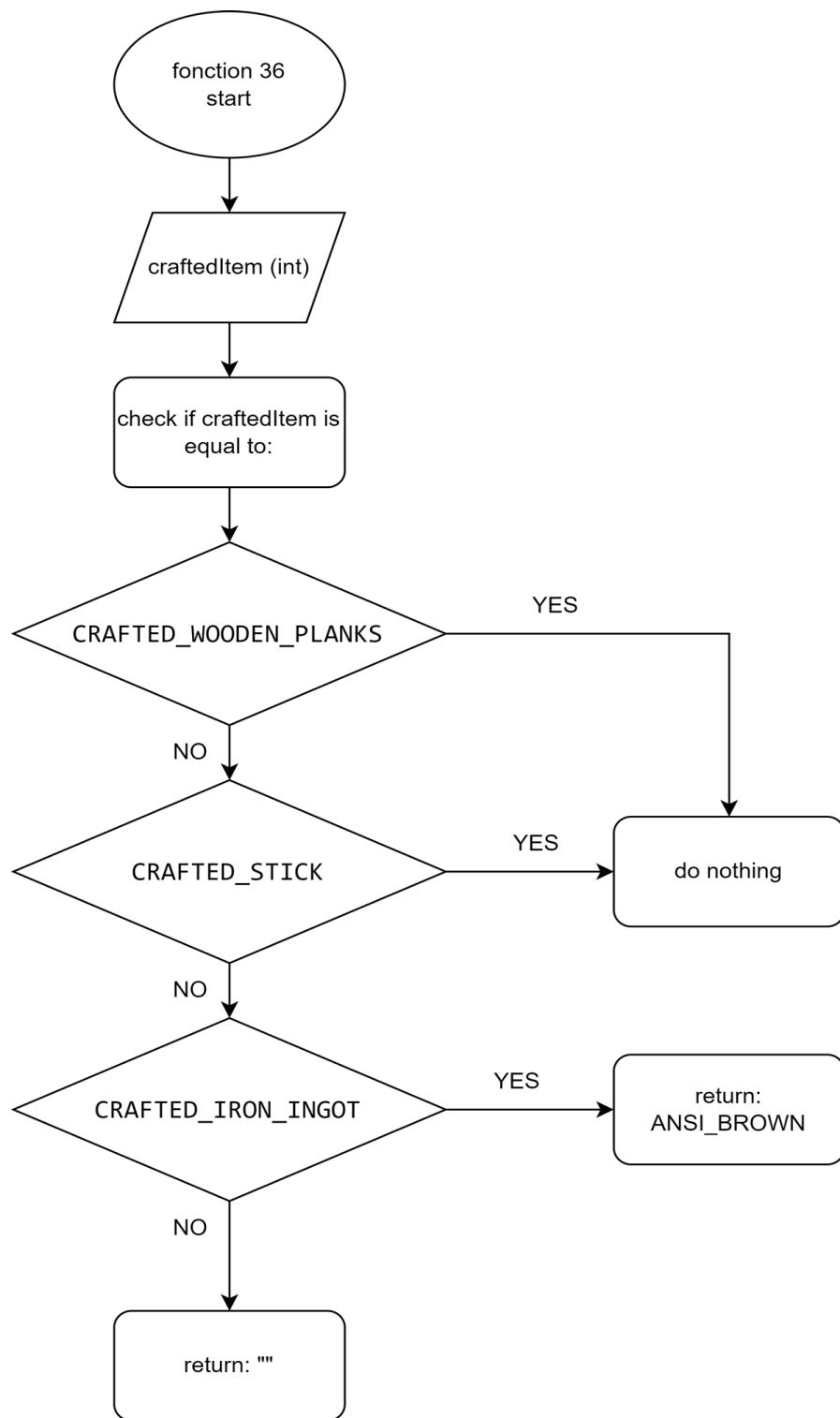
## 34 – waitForEnter



## 35 – getCraftedItemName



## 36 – getCraftedItemColor



## Flowchart for the game:

