

```

public static void mineBlock() {
    int blockType = world[playerX][playerY];
    if (blockType != AIR) {
        inventory.add(blockType);
        world[playerX][playerY] = AIR;
        System.out.println("Mined " + getBlockName(blockType) + ".");
    } else {
        System.out.println("No block to mine here.");
    }
    waitForEnter();
}

```

1. Function mineBlock()

Initialize variable blocktype to player's location

If the blocktype isn't equal to air

 Add the block to the inventory

 Place an AIR block in the player's location

 Print (Mined " + getBlockName(blockType) + ".")

Else

 Print ("No block to mine here.")

Call waitForEnter() method

End if

End function

```

public static void craftWoodenPlanks() {
    if (inventoryContains(WOOD, 2)) {
        removeItemsFromInventory(WOOD, 2);
        addCraftedItem(CRAFTED_WOODEN_PLANKS);
        System.out.println("Crafted Wooden Planks.");
    } else {
        System.out.println("Insufficient resources to craft Wooden
Planks.");
    }
}

```

2.Function craftWoodenPlanks()

If inventory contains 2 wood pieces

 Remove 2 wood pieces from inventory

 Add Wooden Planks to the inventory

 print("Crafted Wooden Planks")

Else

 print("Insufficient resources to craft Wooden Planks.")

End if

End function

```

public static boolean inventoryContains(int item, int count) {
    int itemCount = 0;
    for (int i : inventory) {
        if (i == item) {
            itemCount++;
            if (itemCount == count) {
                return true;
            }
        }
    }
    return false;
}

```

3.Function inventoryContains(int item, int count)

Initialize int variable itemCount to 0

For each int i in inventory

 If i is equal to value of item

 itemCount++

 If itemCount is equal to count

 Return true

End if

End if

End for

Return false

End function

```

public static void addCraftedItem(int craftedItem) {
    if (craftedItems == null) {
        craftedItems = new ArrayList<>();
    }
    craftedItems.add(craftedItem);
}

```

4.Function addCraftedItem(int craftedItem)

If there are no craftedItems

 Create new ArrayList called craftedItems

End if

Add craftedItem to the craftedItems arraylist

End function

```
public static void interactWithWorld() {  
    int blockType = world[playerX][playerY];  
    switch (blockType) {  
        case WOOD:  
            System.out.println("You gather wood from the tree.");  
            inventory.add(WOOD);  
            break;  
        case LEAVES:  
            System.out.println("You gather leaves from the tree.");  
            inventory.add(LEAVES);  
            break;  
        case STONE:  
            System.out.println("You gather stones from the ground.");  
            inventory.add(STONE);  
            break;  
        case IRON_ORE:  
            System.out.println("You mine iron ore from the ground.");  
            inventory.add(IRON_ORE);  
            break;  
        case AIR:  
            System.out.println("Nothing to interact with here.");  
            break;  
        default:  
            System.out.println("Unrecognized block. Cannot interact.");  
    }  
    waitForEnter();  
}
```

5.Function interactWithWorld()

Initialize variable blockType to player's location

Switch (blockType)

case wood

print("You gather wood from the tree.")

Add wood to inventory

break

case leaves

print("You gather leaves from the tree.")

Add leaves to inventory

break

case stone

print("You gather stones from the ground.")

Add stone to inventory

break

case iron

print("You mine iron ore from the ground.")

Add iron to inventory

break

case air

print("Nothing to interact with here.")

break

default:

print("Unrecognized block. Cannot interact.")

Call waitForEnter()

End function

```
private static void fillInventory() {
    inventory.clear();
    for (int blockType = 1; blockType <= 6; blockType++) {
        for (int i = 0; i < INVENTORY_SIZE; i++) {
            inventory.add(blockType);
        }
    }
}
```

6.Function fillInventory()

Clear inventory

For blockType from 1 to 6, increment by 1

 For i from 0 to INVENTORY_SIZE - 1, increment by 1

 Add blockType to inventory

End for

End for

End function

```
private static void waitForEnter() {
    System.out.println("Press Enter to continue...");
    Scanner scanner = new Scanner(System.in);
    scanner.nextLine();
}
```

7.Function waitForEnter()

print("Press Enter to continue...")

Create new scanner

Scan the user's input

End function

```

public static void displayInventory() {
    System.out.println("Inventory:");
    if (inventory.isEmpty()) {
        System.out.println(ANSI_YELLOW + "Empty" + ANSI_RESET);
    } else {
        int[] blockCounts = new int[5];
        for (int i = 0; i < inventory.size(); i++) {
            int block = inventory.get(i);
            blockCounts[block]++;
        }
        for (int blockType = 1; blockType < blockCounts.length; blockType++)
        {
            int occurrences = blockCounts[blockType];
            if (occurrences > 0) {
                System.out.println(getBlockName(blockType) + " - " +
occurrences);
            }
        }
    }
    System.out.println("Crafted Items:");
    if (craftedItems == null || craftedItems.isEmpty()) {
        System.out.println(ANSI_YELLOW + "None" + ANSI_RESET);
    } else {
        for (int item : craftedItems) {
            System.out.print(getCraftedItemColor(item) +
getCraftedItemName(item) + ", " + ANSI_RESET);
        }
        System.out.println();
    }
    System.out.println();
}

```

```

8.Function displayInventory()
print("Inventory:")
If inventory is empty
    print("Empty")
Else
    Create new int array blockCounts[5]
    For int i from 0 to inventory size - 1, increment by 1
        Initialize int variable block to inventory.get(i)
        Increment blockCounts[block] by 1
    End for
    For int blockType from 1 to blockCounts.length -1, increment by 1
        Initialize int variable occurrences to blockCounts[blockType]
        If occurrences > 0
            print(getBlockName(blockType) + " - " + occurrences)
        End if
    End for
End if
print("Crafted items:")
If craftedItems is null or empty
    print("None")
Else
    For each int item in craftedItems
        print(getCraftedItemColor(item) + getCraftedItemName(item) + ", " +
ANSI_RESET)
    End for
    Print empty line
End if
Print empty line
End function

```

```
public static void placeBlock(int blockType) {
    if (blockType >= 0 && blockType <= 7) {
        if (blockType <= 4) {
            if (inventory.contains(blockType)) {
                inventory.remove(Integer.valueOf(blockType));
                world[playerX][playerY] = blockType;
                System.out.println("Placed " + getBlockName(blockType) + " at
your position.");
            } else {
                System.out.println("You don't have " + getBlockName(blockType) +
" in your inventory.");
            }
        } else {
            int craftedItem = getCraftedItemFromBlockType(blockType);
            if (craftedItems.contains(craftedItem)) {
                craftedItems.remove(Integer.valueOf(craftedItem));
                world[playerX][playerY] = blockType;
                System.out.println("Placed " + getCraftedItemName(craftedItem) +
" at your position.");
            } else {
                System.out.println("You don't have " +
getCraftedItemName(craftedItem) + " in your crafted items.");
            }
        }
    } else {
        System.out.println("Invalid block number. Please enter a valid block
number.");
        System.out.println(BLOCK_NUMBERS_INFO);
    }
    waitForEnter();
}
```


9.Function placeBlock(int blockType)

If blockType is from 0 to 7

 If blockType is less than or equal to 4

 If inventory contains blockType

 Remove block from inventory

 Place the block on player's location

 print("Placed " + getBlockName(blockType) + " at your position.")

 Else

 print("You don't have " + getBlockName(blockType) + " in your inventory.");

 End if

 Else

 Initialize int variable craftedItem to blockType

 If array craftedItems contains craftedItem

 Remove craftedItem from craftedItems

 Place the crafted item on player's location

 print("Placed " + getCraftedItemName(craftedItem) + " at your position.")

 Else

 print("You don't have " + getCraftedItemName(craftedItem) + " in your

crafted items.")

 End if

 End if

Else

 print("Invalid block number. Please enter a valid block number.")

 Print block numbers info

End if

Call waitForEnter() function

End function

```
private static void generateEmptyWorld() {
    world = new int[NEW_WORLD_WIDTH][NEW_WORLD_HEIGHT];
    int redBlock = 1;
    int whiteBlock = 4;
    int blueBlock = 3;
    int stripeHeight = NEW_WORLD_HEIGHT / 3; // Divide the height into
three equal parts

    // Fill the top stripe with red blocks
    for (int y = 0; y < stripeHeight; y++) {
        for (int x = 0; x < NEW_WORLD_WIDTH; x++) {
            world[x][y] = redBlock;
        }
    }

    // Fill the middle stripe with white blocks
    for (int y = stripeHeight; y < stripeHeight * 2; y++) {
        for (int x = 0; x < NEW_WORLD_WIDTH; x++) {
            world[x][y] = whiteBlock;
        }
    }

    // Fill the bottom stripe with blue blocks
    for (int y = stripeHeight * 2; y < NEW_WORLD_HEIGHT; y++) {
        for (int x = 0; x < NEW_WORLD_WIDTH; x++) {
            world[x][y] = blueBlock;
        }
    }
}
```

10. Function generateEmptyWorld()

Create a new 2d array called world with the world width and world height

Initialize int redBlock to 1

Initialize int whiteBlock to 4

Initialize int blueBlock to 3

Initialize int stripeHeight to the world height / 3

For int y from 0 to stripeHeight -1, increment by 1

 For int x from 0 to NEW_WORLD_WIDTH - 1, increment by 1

 Assign value of redBlock to coordinates of world[x][y]

 End for

End for

For int y from stripeHeight to stripeHeight * 2 - 1, increment by 1

 For int x from 0 to NEW_WORLD_WIDTH - 1, increment by 1

 Assign value of whiteBlock to coordinates of world[x][y]

 End for

End for

For int y from stripeHeight * 2 to NEW_WORLD_HEIGHT - 1, increment by 1

 For int x from 0 to NEW_WORLD_WIDTH - 1 , increment by 1

 Assign value of blueBlock to coordinates of world[x][y]

 End for

End for

End function

```

public static void displayWorld() {
    System.out.println(ANSI_CYAN + "World Map:" + ANSI_RESET);
    System.out.println("═" + "=".repeat(worldWidth * 2 - 2) + "╗");
    for (int y = 0; y < worldHeight; y++) {
        System.out.print("║");
        for (int x = 0; x < worldWidth; x++) {
            if (x == playerX && y == playerY && !inSecretArea) {
                System.out.print(ANSI_GREEN + "P " + ANSI_RESET);
            } else if (x == playerX && y == playerY && inSecretArea) {
                System.out.print(ANSI_BLUE + "P " + ANSI_RESET);
            } else {
                System.out.print(getBlockSymbol(world[x][y]));
            }
        }
        System.out.println("║");
    }
    System.out.println("═" + "=".repeat(worldWidth * 2 - 2) + "╝");
}

```

11. Function displayWorld()

print("World Map:")

print("═" + "=".repeat(worldWidth * 2 - 2) + "╗")

For int y from 0 to worldHeight - 1, increment by 1

 print("║")

 For int x from 0 to worldWidth - 1, increment by 1

 if(x equals playerX and y equals playerY and not inSecretArea

 print(ANSI_GREEN + "P " + ANSI_RESET)

 Else if x equals playerX and y equals playerY and inSecretArea

 print(ANSI_BLUE + "P " + ANSI_RESET)

 Else

 print(getBlockSymbol(world[x][y]))

 End if

 End for

print("║")

End for

print("═" + "=".repeat(worldWidth * 2 - 2) + "╝")

End function

```

public static void initGame(int worldWidth, int worldHeight) {
    JavaCraft.worldWidth = worldWidth;
    JavaCraft.worldHeight = worldHeight;
    JavaCraft.world = new int[worldWidth][worldHeight];
    playerX = worldWidth / 2;
    playerY = worldHeight / 2;
    inventory = new ArrayList<>();
}

```

12. Function initGame(int worldWidth, int worldHeight)

Initialize variable worldWidth to the value passed as the worldWidth parameter

Initialize variable worldHeight to the value passed as the worldHeight parameter

Create a new 2d array named world and initialize it using the worldWidth and worldHeight

Assign worldWidth / 2 to the variable playerX

Assign worldHeight / 2 to the variable playerY

Create a new arraylist called inventory

End function

```

public static void generateWorld() {
    Random rand = new Random();
    for (int y = 0; y < worldHeight; y++) {
        for (int x = 0; x < worldWidth; x++) {
            int randValue = rand.nextInt(100);
            if (randValue < 20) {
                world[x][y] = WOOD;
            } else if (randValue < 35) {
                world[x][y] = LEAVES;
            } else if (randValue < 50) {
                world[x][y] = STONE;
            } else if (randValue < 70) {
                world[x][y] = IRON_ORE;
            } else {
                world[x][y] = AIR;
            }
        }
    }
}

```

13. Function generateWorld()

Create a new random variable rand

For y from 0 to worldHeight -1, increment by 1

For int x from 0 to worldWidth -1, increment by 1

Create a randValue variable that generates a random number from 0 to 100

If randValue is less than 20

Set the block at position [x][y] to wood

Else if randValue is less than 35

Set the block at position [x][y] to leaves

Else if randValue is less than 50

Set the block at position [x][y] to stone

Else if randValue is less than 70

Set the block at position [x][y] to iron ore

Else

Set the block at position [x][y] to air

End if

End for

End for

End function

```
public static void main(String[] args) {
    initGame(25, 15);
    generateWorld();
    System.out.println(ANSI_GREEN + "Welcome to Simple Minecraft!" +
ANSI_RESET);
    System.out.println("Instructions:");
    System.out.println(" - Use 'W', 'A', 'S', 'D', or arrow keys to move
the player.");
    System.out.println(" - Press 'M' to mine the block at your position
and add it to your inventory.");
    System.out.println(" - Press 'P' to place a block from your inventory
at your position.");
    System.out.println(" - Press 'C' to view crafting recipes and 'I' to
interact with elements in the world.");
    System.out.println(" - Press 'Save' to save the game state and 'Load'
to load a saved game state.");
    System.out.println(" - Press 'Exit' to quit the game.");
    System.out.println(" - Type 'Help' to display these instructions
again.");
    System.out.println();
    Scanner scanner = new Scanner(System.in);
    System.out.print("Start the game? (Y/N): ");
    String startGameChoice = scanner.next().toUpperCase();
    if (startGameChoice.equals("Y")) {
        startGame();
    } else {
        System.out.println("Game not started. Goodbye!");
    }
}
```

```
14. Function main(String[] args)
Call initGame method with parameters 25,15
Call generateWorld() method
Print game instructions
Create a new scanner
print("Start the game? (Y/N): ")
Scan next line in upper case and assign to variable startGameChoice
If startGameChoice equals "Y"
    Call startGame() function
Else
    print("Game not started. Goodbye!")
End if
End function
```



```

public static void craftItem(int recipe) {
    switch (recipe) {
        case 1:
            craftWoodenPlanks();
            break;
        case 2:
            craftStick();
            break;
        case 3:
            craftIronIngot();
            break;
        default:
            System.out.println("Invalid recipe number.");
    }
    waitForEnter();
}

```

15. Function craftItem(int recipe)

Switch (recipe)

Case 1

Call craftWoodenPlanks() method

Break

Case 2

Call craftStick() method

Break

Case 3

Call craftIronIngot() method

Break

Default

print("Invalid recipe number.")

End switch

Call waitForEnter() function

End function