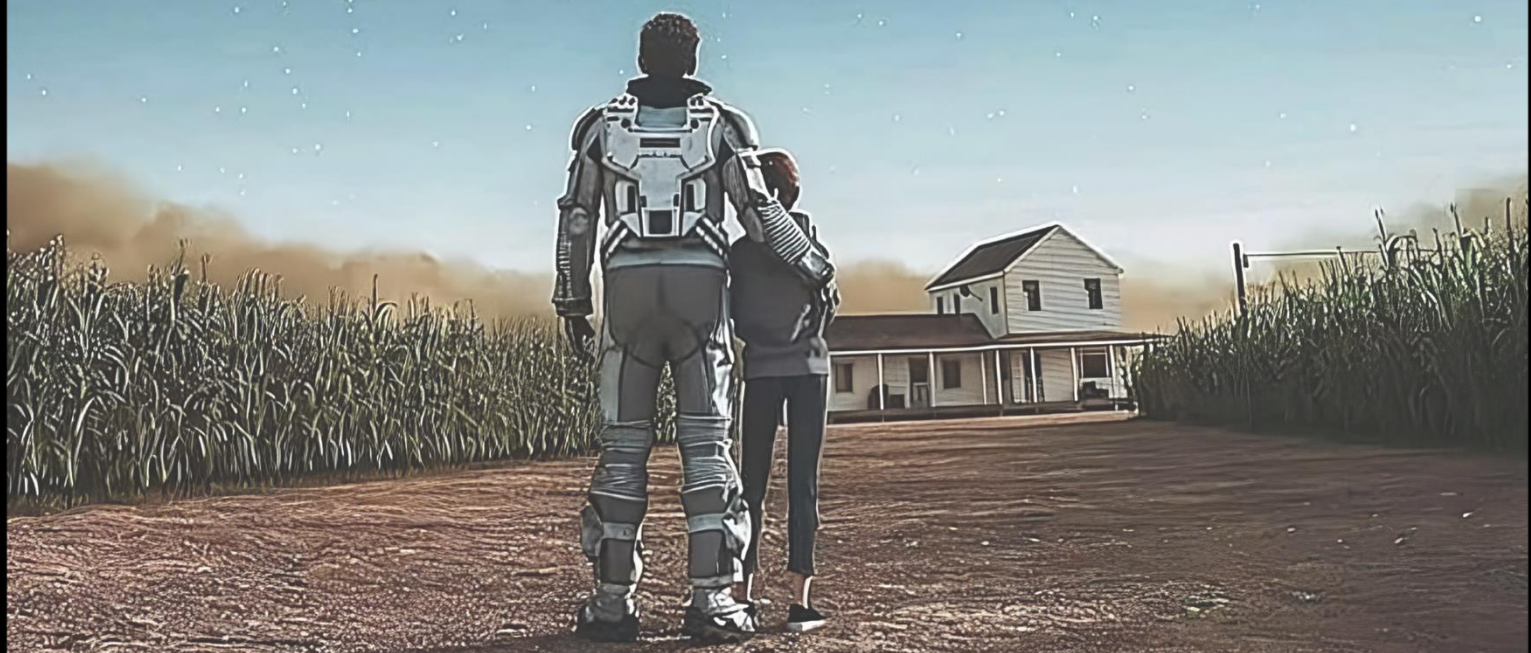
SPACE RACER



Java Programming Coursework 2023

Peter Paul Farrugia

10 Diamonds

Table of Contents

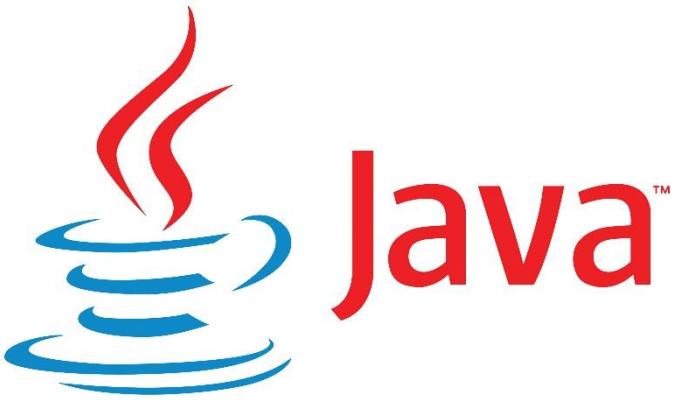
Introduction

What is Programming?

Programming is the process of performing a particular computation, usually by designing and building an executable computer program. Programming involves tasks such as analysis, generating algorithms, algorithms' accuracy, etc.

Programming can be written using varies programming languages. Some popular programming languages are:

- Java
- Javascript
- Python
- PHP
- C#
- C
- C++
- SQL
- HTML
- CSS



Program used in this Project

The program used to code this project was Visual Studio Code (VS Code). VS Code is an integrated development environment from Microsoft. It is used to develop computer programs including websites, web apps, web services and mobile apps. It can support all the popular programming languages including Python and Java.

Problem

Problem Definition

You are required to choose Problem 1 or Problem 2:

Problem 1:

Class Marks - You need to program an application to handle basic class mark management procedures. These could possibly include all or some of the following:

- Entry of names and marks in an array
- Basic data processing:
    - Generation of statistics: average mark, lowest mark, highest mark, marks above average, marks below average.
    - Generation and display of grades.
    - Generation of simple histogram showing grade distribution.
- Output of data as requested by the user.

OR

Problem 2:

Game – Example: Space Racer

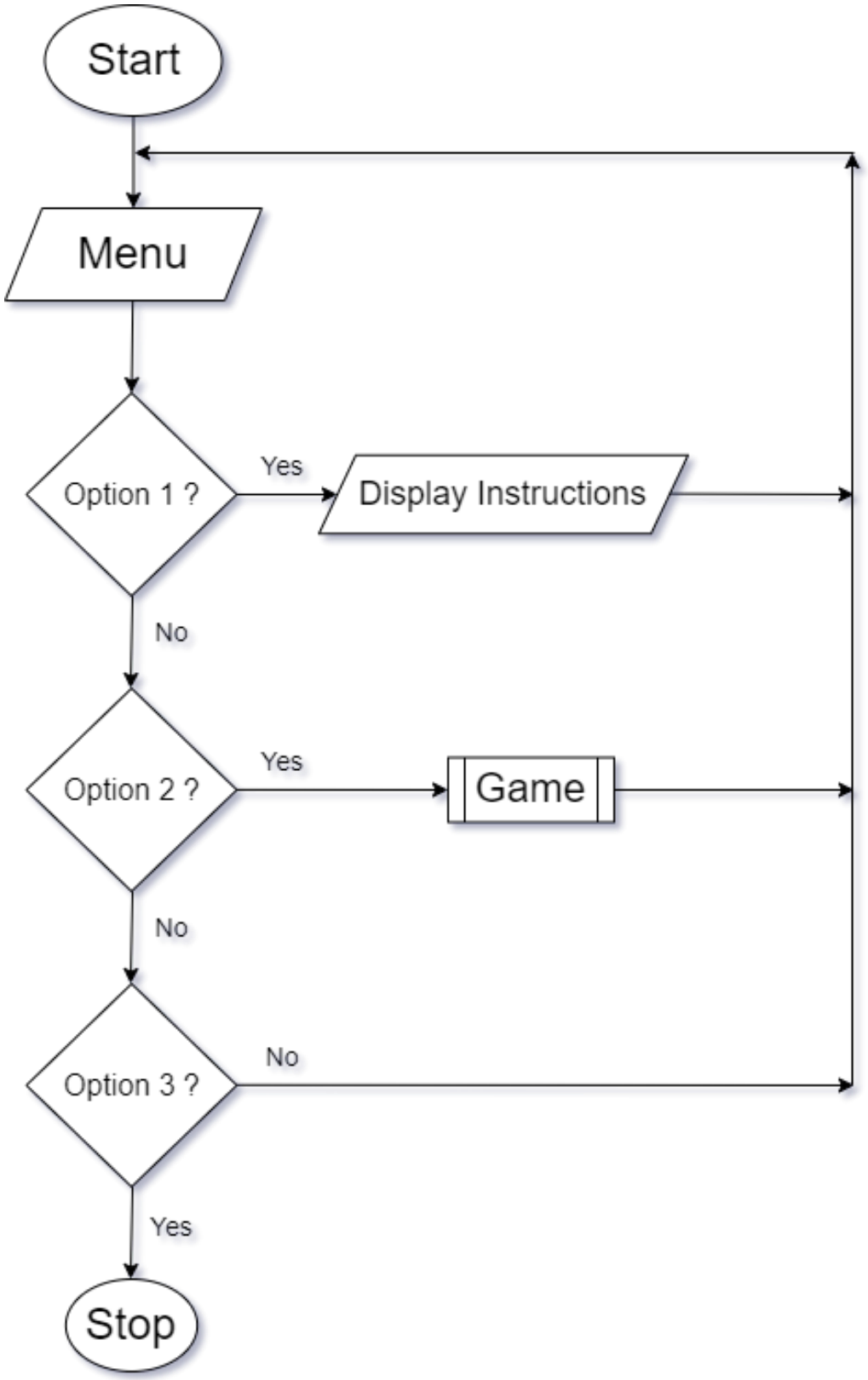
A spaceship captain decides to fly to Saturn and goes all the way to Neptune, with 40 asteroids in-between. The captain must roll a dice and move forward several asteroids depending on the result of the die. The captain should keep doing this until the spaceship reaches the 40^{th} asteroid which is the one next to Neptune. However, during the journey there are some surprises – certain asteroids contain gold deposits and others have nearby wormholes.

Chosen Problem: Problem 2

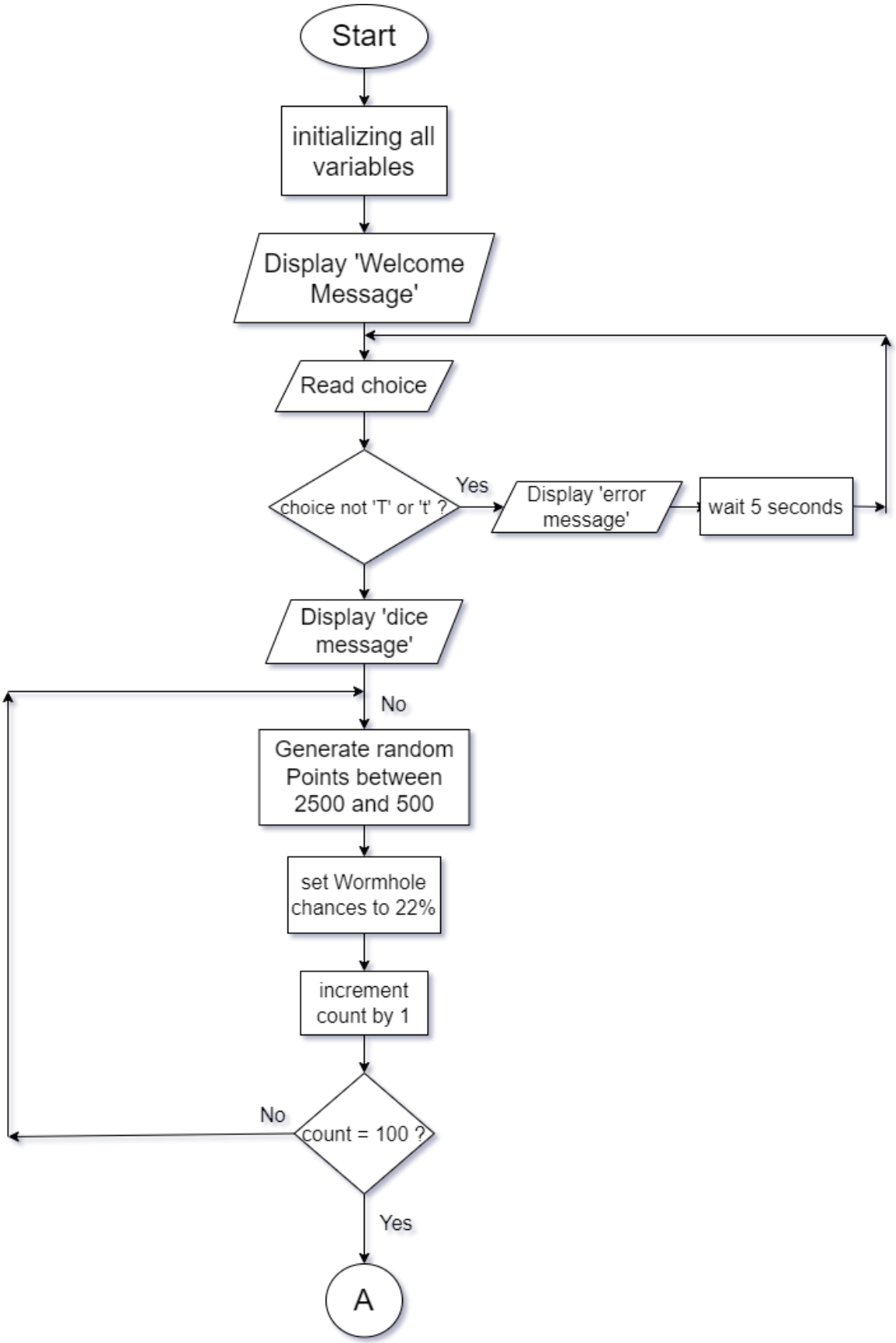
Problem Solution – Game Description

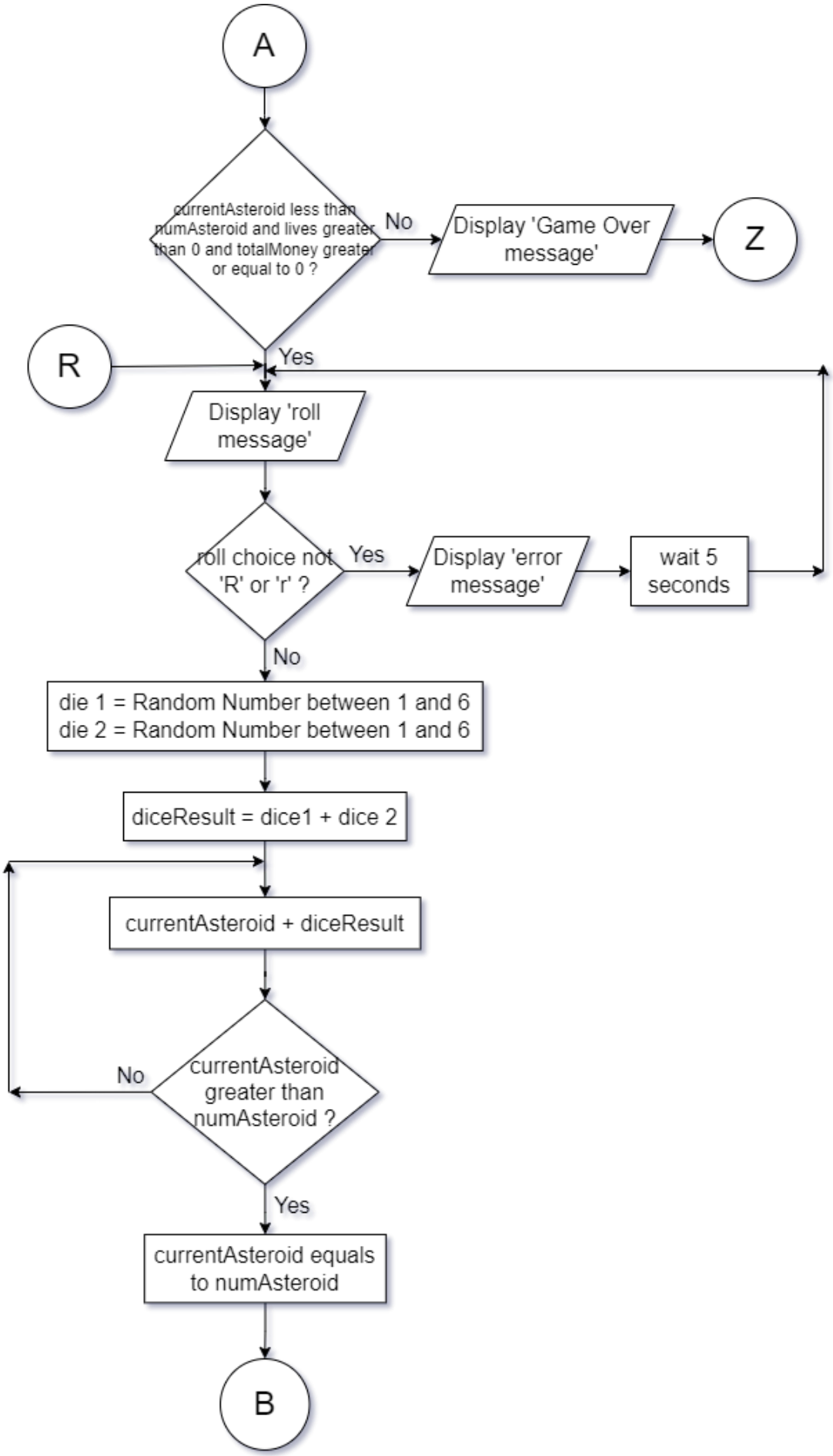
The program (game) that I developed is a space game about a space traveler who has a mission. His mission is to make it and land on Europa (Jupiter's Moon). But he first must travel from Earth to Mars and from there he has to travel across 100 asteroids, meanwhile he has several random chances of being teleported by wormholes to the Sun, an unknown area of the universe, or get hit by a meteoroid. If so, he'll have to get back to Earth to repair his ship. At the end of the day his mission is to make it to Europa in one piece!

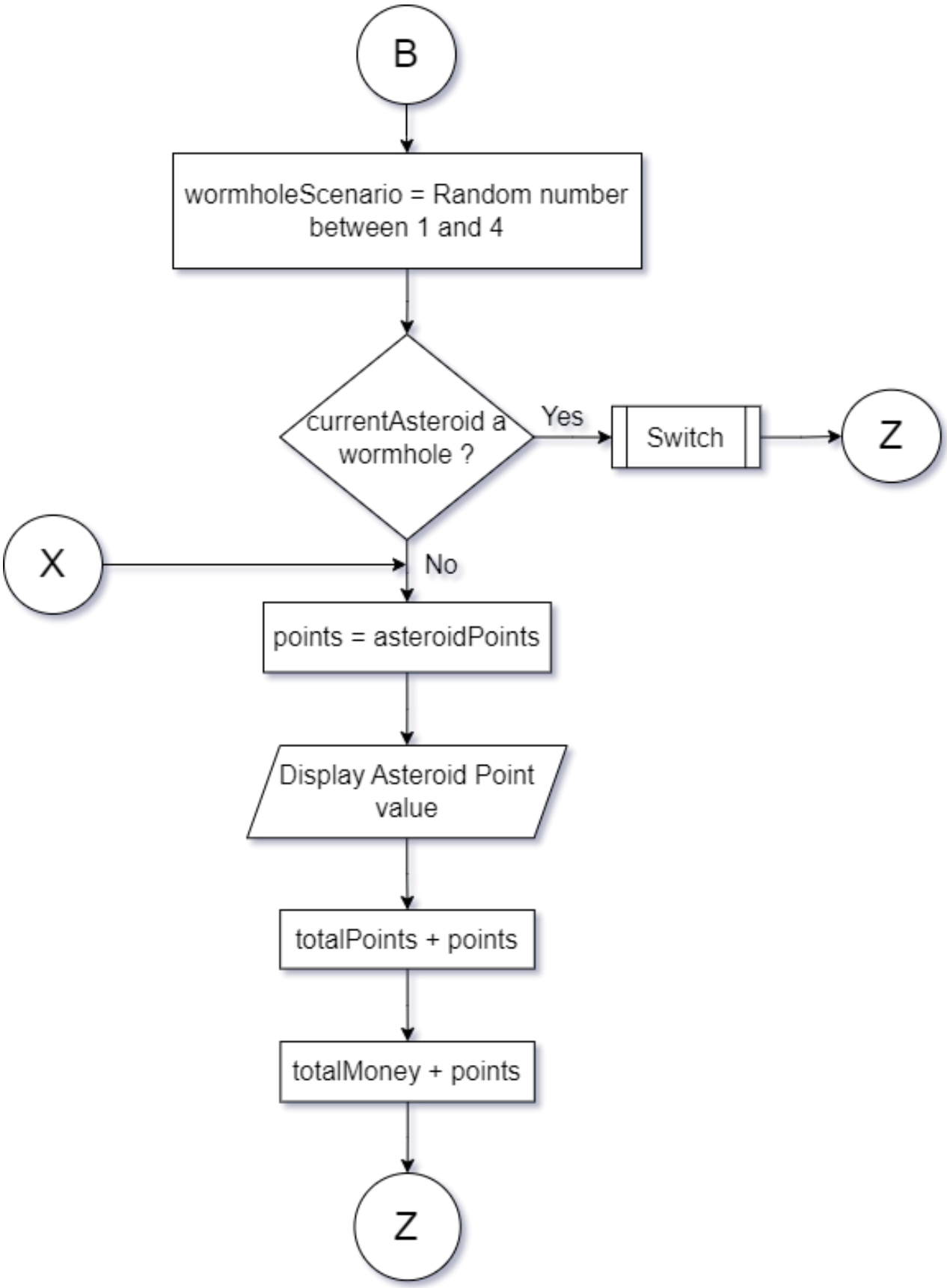
Flowchart – Main Menu



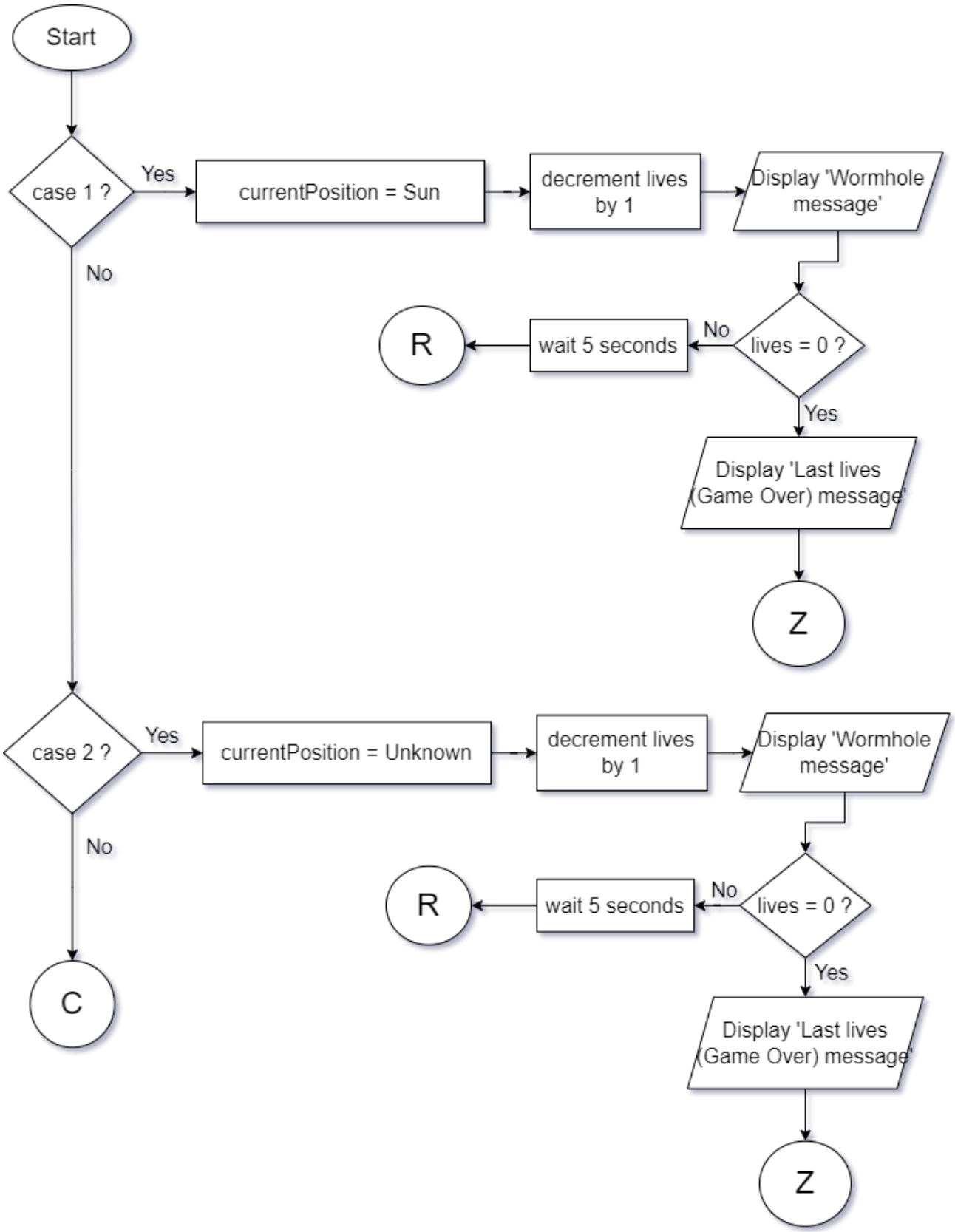
Flowchart – Game Method

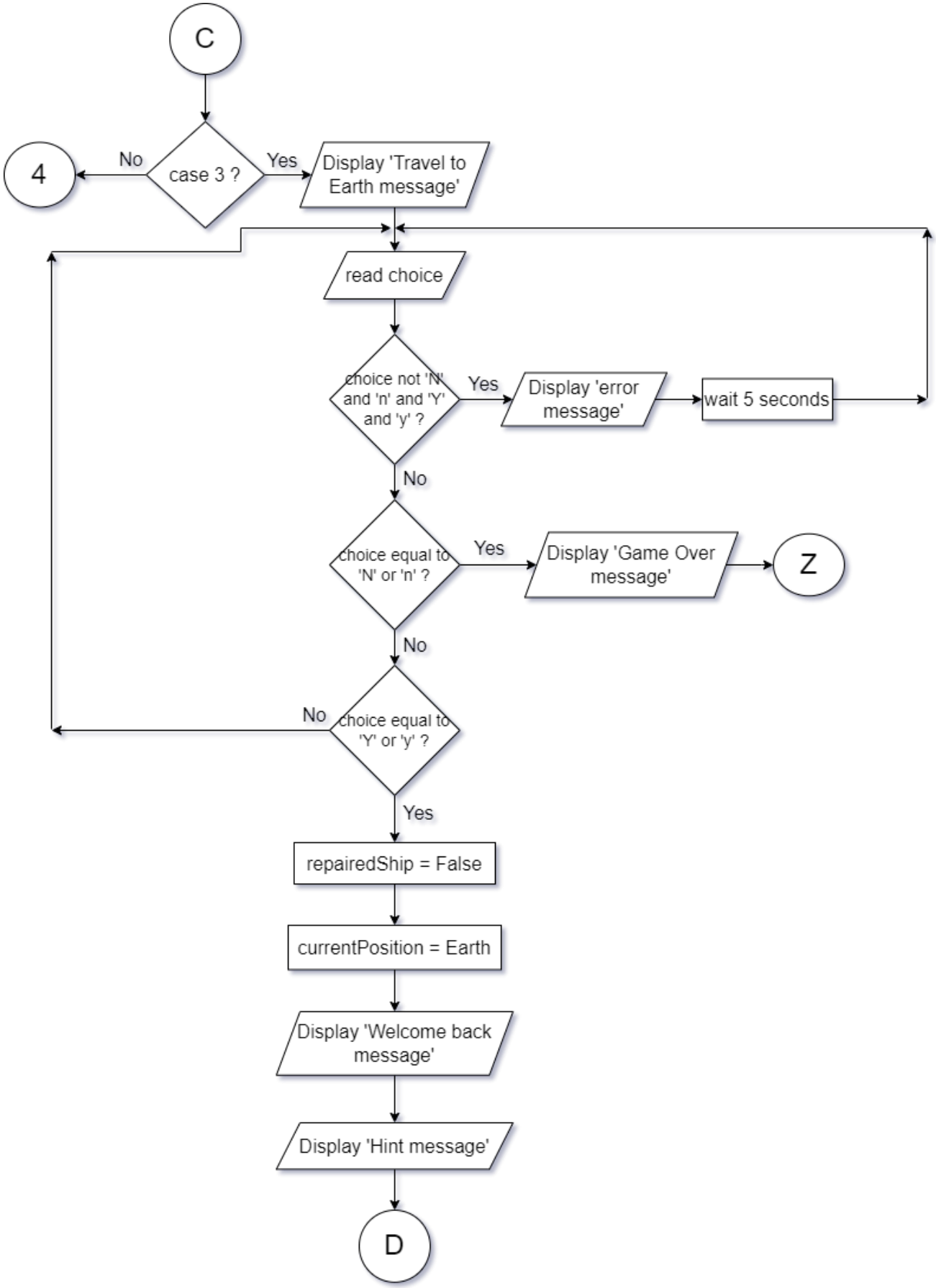


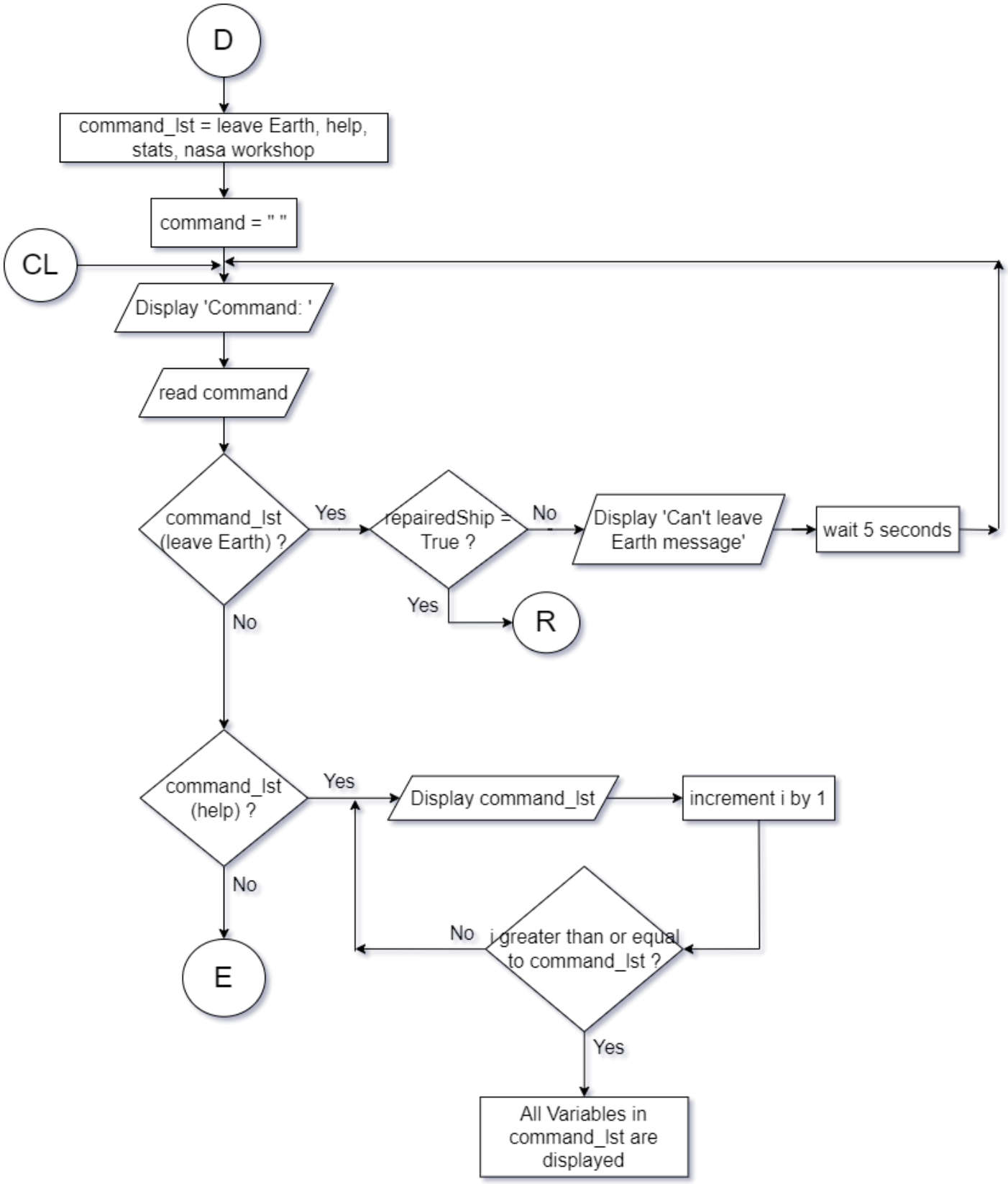

A
currentAsteroid less than numAsteroid and lives greater than 0 and totalMoney greater or equal to 0 ?
No
Display 'Game Over message'
Z
R
Yes
Display 'roll message'
roll choice not 'R' or 'r' ?
Yes
Display 'error message'
wait 5 seconds
No
die 1 = Random Number between 1 and 6
die 2 = Random Number between 1 and 6
diceResult = dice1 + dice 2
currentAsteroid + diceResult
currentAsteroid greater than numAsteroid ?
No
Yes
currentAsteroid equals to numAsteroid
B

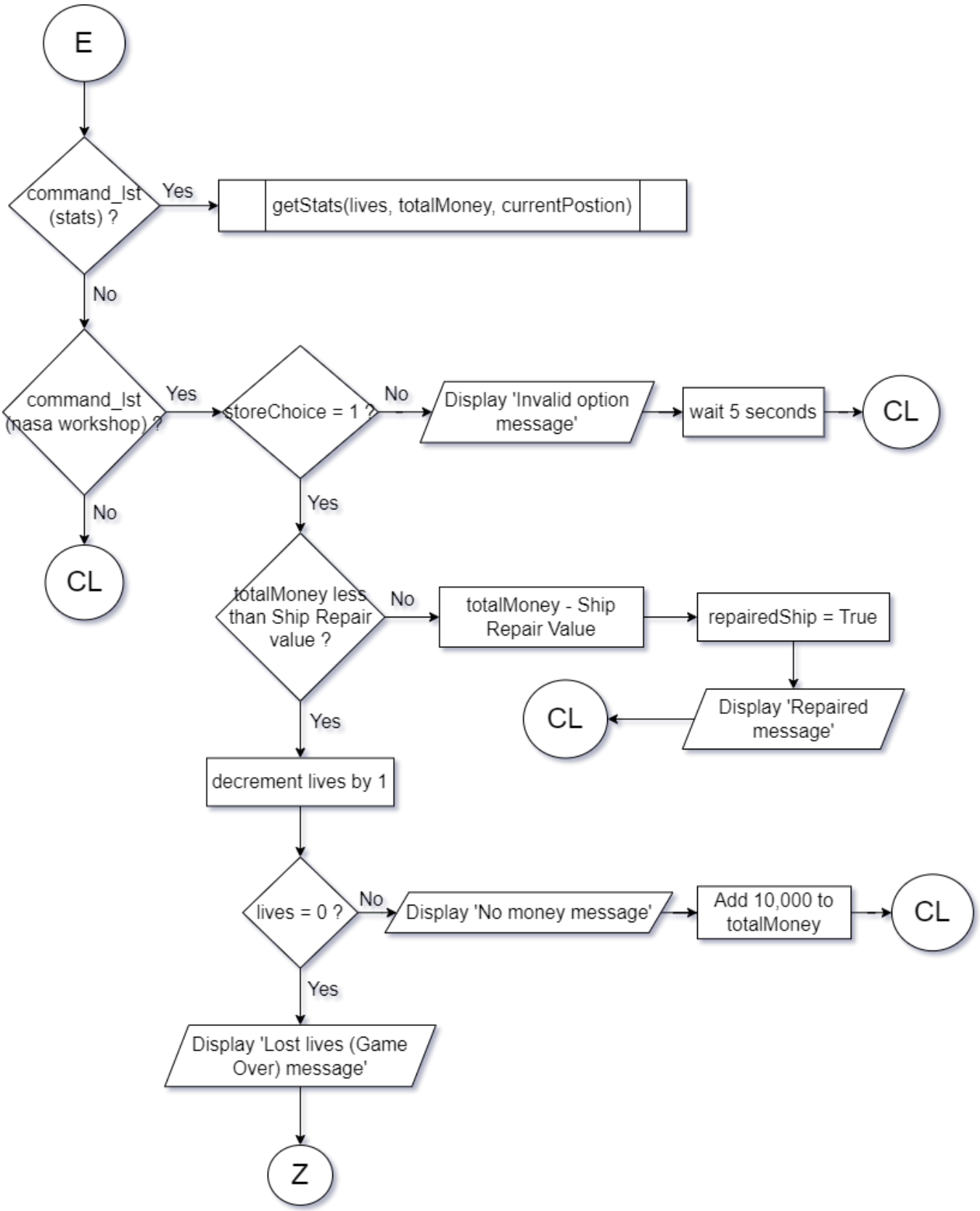

B
wormholeScenario = Random number between 1 and 4
currentAsteroid a wormhole ?
Yes
Switch
Z
X
No
points = asteroidPoints
Display Asteroid Point value
totalPoints + points
totalMoney + points
Z

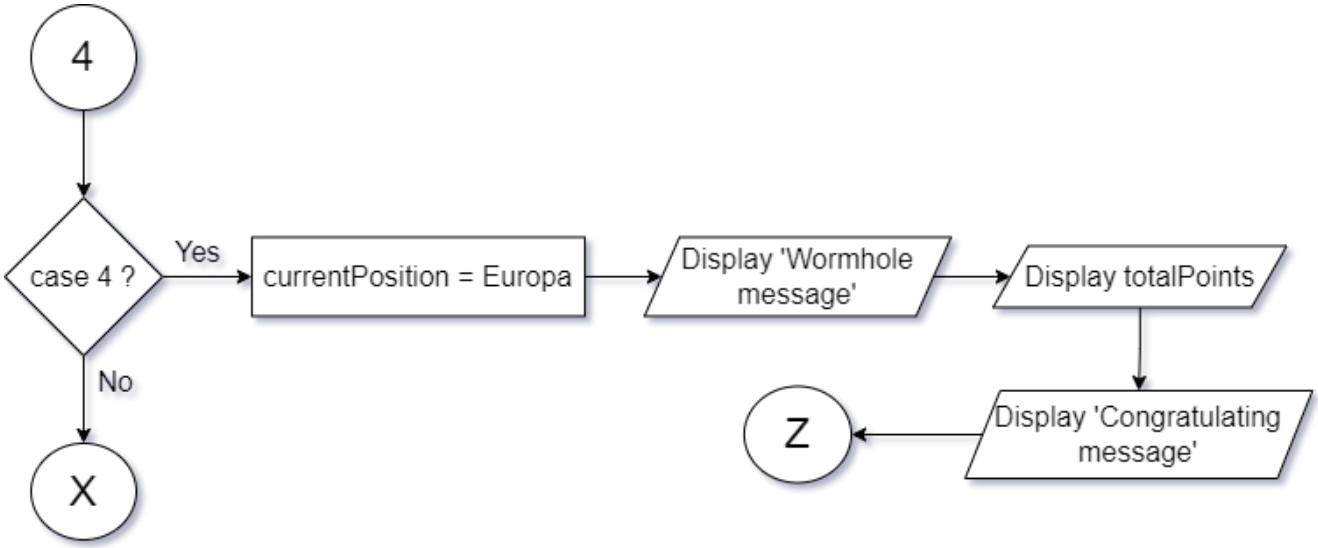
Flowchart – Switch Method




C
case 3 ?
No
Yes
4
Display 'Travel to Earth message'
read choice
choice not 'N' and 'n' and 'Y' and 'y' ?
Yes
Display 'error message'
wait 5 seconds
No
choice equal to 'N' or 'n' ?
Yes
Display 'Game Over message'
Z
No
choice equal to 'Y' or 'y' ?
No
Yes
repairedShip = False
currentPosition = Earth
Display 'Welcome back message'
Display 'Hint message'
D


D
command_lst = leave Earth, help, stats, nasa workshop
command = " "
CL
Display 'Command: '
read command
command_lst (leave Earth) ?
Yes
repairedShip = True ?
No
Display 'Can't leave Earth message'
wait 5 seconds
Yes
R
No
command_lst (help) ?
Yes
Display command_lst
increment i by 1
No
E
No
i greater than or equal to command_lst ?
Yes
All Variables in command_lst are displayed


E
command_lst (stats) ?
Yes
getStats(lives, totalMoney, currentPostion)
No
command_lst (nasa workshop) ?
Yes
storeChoice = 1 ?
No
Display 'Invalid option message'
wait 5 seconds
CL
No
CL
Yes
totalMoney less than Ship Repair value ?
No
totalMoney - Ship Repair Value
repairedShip = True
CL
Display 'Repaired message'
Yes
decrement lives by 1
lives = 0 ?
No
Display 'No money message'
Add 10,000 to totalMoney
CL
Yes
Display 'Lost lives (Game Over) message'
Z


4
case 4 ?
Yes
currentPosition = Europa
Display 'Wormhole message'
Display totalPoints
No
X
Z
Display 'Congratulating message'

Listing of Program

```java
//Importing extra tools for game
import java.io.IOException;
import java.util.Map;
import java.util.HashMap;

class SpaceRacer {
    /*****************Welcome Screen*****************/
    public static int mainMenu()throws IOException, InterruptedException {
        new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

        System.out.println(x:" _____________________________________");
        System.out.println(x:"|      WELCOME TO SPACE RACER       |");
        System.out.println(x:"|                                   |");
        System.out.println(x:"| 1. Game Instructions              |");
        System.out.println(x:"| 2. Play Game                      |");
        System.out.println(x:"| 3. Exit                           |");
        System.out.println(x:"|___________________________________|");
        System.out.println();

        System.out.print(s:"Enter a choice: ");
        int choice = Keyboard.readInt();
        System.out.println();

        return choice;
    }
    /*********************************************************/

    /****************Game Methods & Functions*****************/
    //This method is used to display the users' stats
    public static void getStats(int lives, int totalMoney, String currentPosition) {
        if (totalMoney == 0) {  //This if statement checks how much money the sure has and outputs the table required
            System.out.println(x:"  ___________________________________");
            System.out.println(x:" |   Lives   |   Money   |   Location   |");
            System.out.printf(format:" |     %d     |   S-$%d    |    %s     |\n", lives, totalMoney, currentPosition);
            System.out.println(x:" |___________________________________|");
        }
        else if (totalMoney >= 1000 && totalMoney <= 9999) {
            System.out.println(x:"  ___________________________________");
            System.out.println(x:" |   Lives   |   Money   |   Location   |");
            System.out.printf(format:" |     %d     | S-$%d   |    %s     |\n", lives, totalMoney, currentPosition);
            System.out.println(x:" |___________________________________|");
        }
        else if (totalMoney >= 10000 && totalMoney <= 99999) {
            System.out.println(x:"  ___________________________________");
            System.out.println(x:" |   Lives   |   Money   |   Location   |");
            System.out.printf(format:" |     %d     | S-$%d  |    %s     |\n", lives, totalMoney, currentPosition);
            System.out.println(x:" |___________________________________|");
        }
        else if (totalMoney >= 100000 && totalMoney <= 999999) {
            System.out.println(x:"  ___________________________________");
            System.out.println(x:" |   Lives   |   Money   |   Location   |");
            System.out.printf(format:" |     %d     | S-$%d |    %s    |\n", lives, totalMoney, currentPosition);
            System.out.println(x:" |___________________________________|");
        }
    }

```

```java
//This method is used to assign the different values of different stuff available in the NASA Workshop
public static char getWorkshop(Map<String, Integer> inventory) {
    System.out.println(x:" _____________________________________________________________");
    System.out.println(x:"|                          NASA Workshop                      |");
    System.out.println(x:"|                                                             |");
    System.out.printf(format:"|              1. Repair Ship:             S-$%d            |\n", inventory.get(key:"Ship Repairs"));
    System.out.println(x:"|                                                             |");
    System.out.println(x:"|_____________________________________________________________|");

    System.out.println();
    System.out.println(x:"Pick what you need Lieutenant.");
    System.out.print(s:"Choice: ");
    char storeChoice = Keyboard.readChar();

    return storeChoice;
}

//This method is used to generate a random number
public static int getRandom(int max, int min) {
    int random = (int)(Math.random() * (max - min) + min);  //Generates a random number depending on the number assigned to the variable
    return random;
}

//This method will print out a message depending on the Wormhole Sceneario generated
public static void getWormholeScenario(boolean scenario3, int diceResult, String currentPosition, int lives, String message, String customMessage) {
    if (!scenario3) {  //This if statement will check whether the generated scenario was 3, if so the below code will be executed
        if (customMessage.equals(anObject:"")) {  //If customMessage was used then the below code will execute
            System.out.println();
            System.out.printf(format:"You rolled a %d and arrived at a wormhole!\n", diceResult);

            System.out.printf(format:"Oh no! The wormhole led to %s%s\n", currentPosition, message);
            System.out.printf(format:"Lives left: %d\n", lives);
        }
        else  //If the above if statement does not execute then the below code will
            System.out.printf(format:"\n%s", customMessage);
    }
    else {  //If the above if statement does not execute then the below code will instead
        System.out.println();
        System.out.println("You rolled a " + diceResult + " and you got hit by a meteoroid!");
        System.out.println(x:"You have to go back to Earth to repair your ship or else you will die and LOSE!!");
    }
}

//This method will generate a message depending on what it was assigned to during coding
public static void generateMessage(String message, int miliSec) {
    System.out.println(message);
    try {
        Thread.sleep(miliSec);
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
    }
}
```

```java
//This method will execute whenever the user doesn't win, wins or to show them their points
public static void gameResult(boolean won, int totalPoints, String message, int miliSec) throws IOException, InterruptedException {
    new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

    if (!won) {  //This if statement will check whether the user has not won, if so the below code will be executed
        System.out.printf(format:"GAME OVER!\n%s", message);

        try {
            Thread.sleep(miliSec);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
    else {  //If the above if statement does not execute then the below code will
        if (message.equals(anObject:"")) {  //If Message was used then the below code will execute
            System.out.println(x:"CONGRATULATIONS!");
            System.out.println(x:"You have reached Europa!");
            System.out.println("Your total score is: " + totalPoints + " points.");
        }
        else //If the above if statement does not execute then the below code will
            System.out.print(message);

        try {
            Thread.sleep(miliSec);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
}

//This method will have a for loop running until all variables in command_lst[i] will be displayed
public static void getCommandLst(String[] command_lst) {
    System.out.print(s:"\n| ");
    for (int i = 0; i < command_lst.length; i++) {
        System.out.print(command_lst[i] + " : ");
    }
    System.out.print(s:"|\n");
}
/*********************************************************/
```

```java
/**********************Main Game Method**********************/
public static void game() throws IOException, InterruptedException {
    new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

    /***************Variable Initialization***************/
    int currentAsteroid = 1; //Declaring a variable named currentAsteroid to 1
    int numAsteroids = 100;  //Declaring a variable named numAsteroids to 100

    int[] asteroidPoints = new int[numAsteroids];        //Declaring an array named asteroidPoints to the number of asteroids
    boolean[] isWormhole = new boolean[numAsteroids];    //Declaring an array named isWormhole to the number of asteroids

    int totalPoints = 0;  //Declaring a variable named totalPoints to 0
    int totalMoney = 10000;   //Declaring a variable named totalMoney to 10000
    int lives = 3;        //Declaring a variable named lives to 3

    Map<String, Integer> inventory = new HashMap<>(); //This map type String, String is connected to the method getStore()
    inventory.put(key:"Ship Repairs", value:15000); //Declaring a map type String, String to "Repair Ship", "S-$15,000"

    String[] position = {"Sun", "Mercury", "Venus", "Earth", "Mars", "Europa",
    "Jupiter", "Saturn", "Uranus", "Neptune", "Pluto", "UNKNOWN"}; //Declaring an array named position and it stores all of the positions used in this game
    String currentPosition = position[3]; //Starting position is set to position 3 (Earth)
    /*****************************************************/

    char choice = ' ';
    while (choice != 'T' || choice != 't') { //This while loop will work only if the user enters the letter 'T' or 't', otherwise it will print an error
        new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

        System.out.println(x:"Welcome Lieutenant!");
        System.out.println("You are currently on " + currentPosition + ".");

        System.out.print(s:"Press T to travel to Mars: ");
        choice = Keyboard.readChar();

        if (choice != 'T' && choice != 't')  //If user entered another letter than 'T' or 't', the game will print an error message
            generateMessage(message:"\nError. Press T to travel to Mars.", miliSec:2000);
        else { //if user entered 'T' or 't' in the variable choice
            generateMessage(new String("\nYou have arrived on " + position[4] + "!\nUse the dice to travel between asteroids and reach Europa!"), miliSec:0);

            //Randomizing Asteroid & Wormhole Chances
            for (int i = 0; i < numAsteroids; i++) {
                asteroidPoints[i] = getRandom(max:2500,min:500);
                isWormhole[i] = Math.random() < 0.22;
            }

            while (currentAsteroid < numAsteroids && lives > 0 && totalMoney >= 0) {  //This while loop will keep running until currentAsteroids value is < numAsteroids, lives value is > 0 and totalMoney >= 0
                System.out.println();
                System.out.print(s:"Press R to roll the dice and travel to a new asteroid: ");
                char roll = Keyboard.readChar();

                if (roll != 'R' && roll != 'r') //If user entered another letter than 'R' or 'r', the game will print an error message
                    generateMessage(message:"Error. Press R to roll the dice and travel to a new asteroid.\n", miliSec:2000);
                else { //Checks if user entered 'R' or 'r' in the variable choice
                    int die1 = getRandom(max:6,min:1);  //This variable uses the method getRandom() and the range in it is 6 which equals to (max 6 - min 0) and starts from min == 1
                    int die2 = getRandom(max:6,min:1);  //This variable uses the method getRandom() and the range in it is 6 which equals to (max 6 - min 0) and starts from min == 1
                    int diceResult = die1 + die2;
                    currentAsteroid += diceResult;
```

```
207                 //This if will run everytime until currentAsteroid is equal to numAsteroids
208                 if (currentAsteroid > numAsteroids) {
209                     currentAsteroid = numAsteroids;
210                 }
211 
212                 String wormhole_scenario = Integer.toString(((int)(Math.random() * 4) + 1)); //Variable challengeType is created to randomize 4 chances of different types of challenges
213                 if (isWormhole[currentAsteroid-1]) {  //This if statement checks if the currentAsteroid is a Wormhole, if so it will execute
214                     switch (wormhole_scenario) {
215                         //Case 1 will send the user next to the Sun, therefore they'll die and lose a life. If lives == 0 then it's Game Over
216                         case "1":
217                                 currentPosition = position[0];
218                                 lives--;
219 
220                                 //The below code uses the getWormholeScenario() method
221                                 getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:". Oops you burned up..", customMessage:"");
222 
223                                 if (lives != 0) {
224                                     try {
225                                         Thread.sleep(millis:5000);
226                                     } catch (InterruptedException e) {
227                                         Thread.currentThread().interrupt();
228                                     }
229                                 }
230                                 else {
231                                     //The below code uses the getResults() method
232                                     gameResult(won:false, totalPoints, message:"You lost all of your lives!", miliSec:5000);
233                                     return;
234                                 }
235                         break;
236 
237                         //Case 2 will send the user in an Unknown location in the Universe, therefore they'll die and lose a life. If lives == 0 then it's Game Over
238                         case "2":
239                                 currentPosition = position[11];
240                                 lives--;
241 
242                                 //The below code uses the getWormholeScenario() method
243                                 getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:" location in the universe and you're lost!", customMessage:"");
244 
245                                 if (lives != 0) {
246                                     try {
247                                         Thread.sleep(millis:5000);
248                                     } catch (InterruptedException e) {
249                                         Thread.currentThread().interrupt();
250                                     }
251                                 }
252                                 else {
253                                     //The below code uses the getResult() method
254                                     gameResult(won:false, totalPoints, message:"You lost all of your lives!", miliSec:8000);
255                                     return;
256                                 }
257                         break;
258 
```

```
//Case 3 will hit the user's ship with a meteoroid, therefore they'll have to go back to Earth to fix the ship.
case "3":
    //The below code uses the getWormholeScenario() method
    getWormholeScenario(scenario3:true, diceResult, currentPosition, lives, currentPosition, customMessage:"");

    choice = ' ';
    while (choice != 'n' && choice != 'N' && choice != 'y' && choice != 'Y') {  //This while loop will run again if the user enters any letter other than 'n', 'N', 'y', 'Y'
        System.out.println();
        System.out.print(s:"Do you want to travel back to Earth? (Y/N): ");
        choice = Keyboard.readChar();

        if (choice != 'n' && choice != 'N' && choice != 'y' && choice != 'Y') {  //Checks if user entered 'n' or 'N' or 'y' or 'Y' in the variable choice, if not error message will print
            System.out.print(s:"Error. Press (Y/N) to continue.");

            try {
                Thread.sleep(millis:2000);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }

        else if (choice == 'n' || choice == 'N') {  //Checks if user entered 'n' or 'N', if so the below code will execute
            //The below code uses the getResult() method
            gameResult(won:false, totalPoints, message:"\nYou did not repair your ship, therefore you couldn't continue the mission!", miliSec:8000);
            return;
        }

        else if (choice == 'y' || choice == 'Y') {  //Checks if user entered 'y' or 'Y', if so the below code will execute
            boolean repairedShip = false;    //Resets the value of repairedShip to false everytime it runs again
            currentPosition = position[3];   //Updates user's current position to position[3] (Earth)

            new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

            //The below code uses the generateMessage() method
            generateMessage(new String("Welcome back to " + currentPosition + " Lieutenant!"), miliSec:0);
            generateMessage(message:"We noticed your ship needs a few repairs.", miliSec:0);
            generateMessage(message:"\nHINT: Type 'help' for a list of commands.", miliSec:0);

            String command_lst[] = {"leave Earth", "help", "stats", "nasa workshop"};

            String command = "";
            while(!command.equals(command_lst[0]) || !repairedShip) { //This while loop will run until user enters "Leave Earth" and ship must be repaired
                System.out.print(s:"\nCommand: ");
                command = Keyboard.readString();

                if (command.equals(command_lst[0])) {
                    if (repairedShip)  //Checks whether the user repaired their ship, if yes it will clear screen
                        new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

                    else   //Checks whether the user repaired their ship, if no below message will execute
                        generateMessage(message:"\nCan't Leave Earth, your ship is still broken ;)", miliSec:5000);
                }
                else if (command.equals(command_lst[1])) {
                    getCommandLst(command_lst);  //This code uses the getCommadLst() method
                }
                else if (command.equals(command_lst[2])) {
                    getStats(lives, totalMoney, currentPosition);  //This code uses the getStats() method
                }
```

```
317                                 else if (command.equals(command_lst[3])) {
318                                     char storeChoice = getWorkshop(inventory);  //This code uses the getWorkshop() method
319
320                                     if (storeChoice == '1') {  //Checks if user entered '1', if so the below code will execute
321                                         if (totalMoney < inventory.get(key:"Ship Repairs")) {
322                                             lives--;
323                                             if (lives == 0) {
324                                                 System.out.println();
325                                                 System.out.println(x:"It looks like you lost all of your lives!!");
326                                                 System.out.println("Lives left: " + lives);
327                                                 System.out.println(x:"GAME OVER, Lieutenant!");
328
329                                                 try {
330                                                     Thread.sleep(millis:5000);
331                                                 } catch (InterruptedException e) {
332                                                     Thread.currentThread().interrupt();
333                                                 }
334                                                 return;
335                                             }
336                                             else {
337                                                 //The below code uses the generateMessage() method
338                                                 generateMessage(message:"\nYou don't have enough money to buy this!", miliSec:0);
339                                                 generateMessage(message:"Therefore you lost a life.", miliSec:0);
340                                                 System.out.println("Lives left: " + lives);
341                                                 generateMessage(message:"", miliSec:3000);
342                                                 generateMessage(message:"In return..", miliSec:2500);
343                                                 generateMessage(message:"We have given you S-$10000 since you used up all of your money.", miliSec:0);
344                                                 totalMoney += 10000;
345                                             }
346                                         }
347                                         else {
348                                             //Subtracts the cost of ship repairs from totalMoney(current money value)
349                                             totalMoney -= inventory.get(key:"Ship Repairs");
350                                             repairedShip = true;
351
352                                             //The below code uses the generateMessage() method
353                                             generateMessage(message:"\nLieutenant, we have just repaired your ship.", miliSec:0);
354                                             generateMessage(new String("That costed you " + inventory.get(key:"Ship Repairs") + " Starfleet Dollars!"), miliSec:2500);
355                                             generateMessage(new String("\nYou have " + totalMoney + " Starfleet Dollars left!"), miliSec:0);
356                                             generateMessage(message:"Lieutenant, now you can keep going on your journey!\nGood Luck and don't DIE!", miliSec:5000);
357                                         }
358                                     }
359                                     else {  //If user entered another number than '1', the game will print an error message
360                                         System.out.println();
361                                         System.out.println(x:"Invalid Option. Type 'Store' again to retry...");
362                                         System.out.println();
363                                     }
364                                 }
365                             }
366                         }
367                     }
368                 break;
369
```

```
                        //Case 4 will send the user directly to Europa. Catch is will the user have collected a lot of points or not.
                        case "4":
                            currentPosition = position[5];

                            //The below code uses the getWormholeScenario() method
                            getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:"", new String("Congratulations you won!\nThis wormhole led straight to " + currentPosition + "!"));

                            try {
                                Thread.sleep(millis:3000);
                            } catch (InterruptedException e) {
                                Thread.currentThread().interrupt();
                            }

                            //The below code uses the generateMessage() method
                            generateMessage(message:"\nLet's hope you managed to get as many points as you could then, Lieutenant!", miliSec:2000);
                            generateMessage(message:"Because...", miliSec:2000);
                            generateMessage(new String("Your total score is: " + totalPoints + " points."), miliSec:3000);

                            //The below code uses the gameResult() method
                            gameResult(won:true, totalPoints, message:"CONGRATULATIONS Again Lieutenant, Excellent Job!", miliSec:7000);
                        return;
                    }
                }
                else { //If challengeType isn't executed then user is told the dice result and that they moved to the next asteroid
                    int points = asteroidPoints[currentAsteroid-1];

                    //The below code uses the generateMessage() method
                    generateMessage(new String("\nYou rolled a " + diceResult + " and arrived at asteroid " + currentAsteroid + "."), miliSec:0);
                    generateMessage(new String("This asteroid is worth " + points + " points!"), miliSec:1000);

                    totalPoints += points;  //Adds all points and stores them in the variable totalPoints
                    totalMoney += points;   //Adds all points and stores them in the variable totalMoney
                }
            }
        }

        //Win Game method
        if (currentAsteroid >= numAsteroids) { //If the currentAsteroid the user is on is greater or equal to the numAsteroids(100) then the user is done from game and congratulating message is outputted
            //The below code uses the gameResult() method
            gameResult(won:true, totalPoints, message:"", miliSec:7000);
            return;
        }
    }
  }
}
/*********************************************************/
```

```java
/*****************Instructions Menu*****************/
public static void instructions() throws IOException, InterruptedException {
    char choice = ' ';

    while (choice != 'B' && choice != 'b') { //The instructions will be shown to user until they press 'B' or 'b', if they do it will return to main(String[] args)
        new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();

        System.out.println(x:" _____________________________________________________________________________________________________");
        System.out.println(x:"|                                                Instructions:                                        |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|You are a new Lieutenant in Starfleet.                                                               |");
        System.out.println(x:"|You're assigned to Company Alpha under Captain James T. Kirk. Assigned to the Orbiter IV!            |");
        System.out.println(x:"|You're heading to Europa, one of Jupiter's Moons to explore any Alien life you may find.             |");
        System.out.println(x:"|Unfortunately on the way you have to cross a set of asteroids.                                       |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|Mission:                                                                                             |");
        System.out.println(x:"|Help Captain Kirk make it to Europa safely.                                                          |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|Objective:                                                                                           |");
        System.out.println(x:"|You must roll two dice (6 & 6) and move forward several asteroids depending on the dice's result.     |");
        System.out.println(x:"|You must keep doing this until you reach Europa.                                                     |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|CAUTION:                                                                                             |");
        System.out.println(x:"|Space junk is dangerous!                                                                             |");
        System.out.println(x:"|Some asteroids will give you a set of points and some will be wormholes.                             |");
        System.out.println(x:"|If you pass through a wormhole you can end up:                                                       |");
        System.out.println(x:"|                                              - near the Sun                                         |");
        System.out.println(x:"|                                              - in an Unknown Location in the universe               |");
        System.out.println(x:"|                                              - gettig hit by meteoroids                             |");
        System.out.println(x:"|                                              - or making it to Europa!                              |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|EXPLANATION:                                                                                         |");
        System.out.println(x:"|If you end up near the Sun you'll burn up and lose a life.                                           |");
        System.out.println(x:"|If you en up in an Unknown location in the universe you'll die and lose a life.                      |");
        System.out.println(x:"|If you end up getting hit by a meteoroid you'll have to go back to Earth,                            |");
        System.out.println(x:"|to repair your ship from the NASA Workshop otherwise you'll die and it's GAME OVER!!                 |");
        System.out.println(x:"|If you repair your ship you'll be brought back to the last asteroid you were at before.              |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|Starting Stats:                                                                                      |");
        System.out.println(x:"|                  - 3 Lives                                                                          |");
        System.out.println(x:"|                  - S-$10,000 (Starfleet Dollars)                                                    |");
        System.out.println(x:"|                  - Starting Location: Earth                                                         |");
        System.out.println(x:"|                                                                                                     |");
        System.out.println(x:"|Tips:                                                                                                |");
        System.out.println(x:"|      Try: 'Help' for a list of commands!                                                            |");
        System.out.println(x:"|           This will give you the options for you to see your:                                       |");
        System.out.println(x:"|           Stats and the NASA Workshop                                                               |");
        System.out.println(x:"|_____________________________________________________________________________________________________|");

        System.out.println();
        System.out.print(s:"Press B to continue: ");
        choice = Keyboard.readChar();

        if (choice != 'B' && choice != 'b') {
            System.out.print(s:"\nError. Press B to continue.");

```

```
474                 try {
475                     Thread.sleep(millis:2000);
476                 } catch (InterruptedException e) {
477                     Thread.currentThread().interrupt();
478                 }
479             }
480         }
481     }
482     /*************************************************************/
483 
484 
485     /***********************main Method***********************/
        Run | Debug
486     public static void main(String[] args) throws IOException, InterruptedException {
487         int choice = 0;
488 
489         while (choice != 3) { //This while loop will keep running until 3 is pressed, if so program will exit
490             choice = mainMenu();
491 
492             switch (choice) {
493                 case 1:
494                     instructions();
495                 break;
496 
497                 case 2:
498                     game();
499                 break;
500 
501                 case 3:
502                 break;
503 
504                 default:
505                     System.out.println(x:"Error. Please enter a number from 1 to 3.");
506 
507                     try {
508                         Thread.sleep(millis:2000);
509                     } catch (InterruptedException e) {
510                         Thread.currentThread().interrupt();
511                     }
512                 break;
513             }
514         }
515     }
516     /*************************************************************/
517 }
```

Coursework Papers

FORM 4 ANNUAL – JAVA PROGRAMMING COURSEWORK

This Coursework carries 15% of the annual examination mark. This exercise is to be fully word processed and to include the items outlined below.

Problem Definition

- A brief description of the application developed and a possible context for its use.

Solution of the problem

- A structured algorithm described by means of a suitable diagrammatic methodology or pseudo-code.

- A hardcopy listing of the program. The program should include the following constructs:
  - inline documentation
  - input and output statements
  - assignment expressions
  - decision construct
  - looping construct
  - implementation of array
  - simple methods
  - any special design features.

Coursework Example 1: Class marks

An application designed to handle basic class mark management procedures. These could possibly include all or some of the following:

- Entry of names and marks in an array.
- Basic data processing:
  - Generation of statistics: average mark, lowest mark, highest mark, marks, above average, marks below average.
  - Generation and display of grades.
  - Generation of simple histogram showing grade distribution.

- Output of data as requested by the user

Coursework Example 2: Game

Example: Space Racer

A spaceship captain decides to fly to Saturn and goes all the way to Neptune, with 40 asteroids in-between. The captain must roll a dice and move forward several asteroids depending on the result of the die. The captain should keep doing this until the spaceship reaches the 40th asteroid which is the one next to Neptune. However, during the journey there are some surprises - certain asteroids contain gold deposits and others have nearby wormholes.

Year 10 – Programming Coursework – Marking Sheet

Criteria for Assessment	Maximum Mark	Actual Mark
Problem Definition:		
Description of the scope of the problem to be tackled	2	
Solution of the problem		
Algorithm (flowchart or pseudocode)	4	
Computer listing of program involving		
Declaration of user-defined classes	2	
Simple methods	3	
Input and output statements	2	
Assignment expressions	2	
Decision construct	3	
Looping construct	4	
Implementation of an array	4	
Inline documentation	2	
Special design features	2	
TOTAL	30	
TOTAL/15 (rounded to nearest integer)	15	

Student's Name: _____

Class: _____

Teacher's Name: _____

Signature: _____

Date: _____