

Table of Contents

INTRODUCTION	
What is Programming?	1
Program used in this Project	2
PROBLEM.....	
Problem Definition	3
Problem Solution – Game Description	4
Flowchart - Main Menu.....	5
Flowchart - Game Method	6
Flowchart - Switch Method	9
LISTING OF PROGRAM.....	
Hardcopy of the Program	14
Coursework Papers	
Coursework – Requirements	24
Coursework – Marking Sheet	25

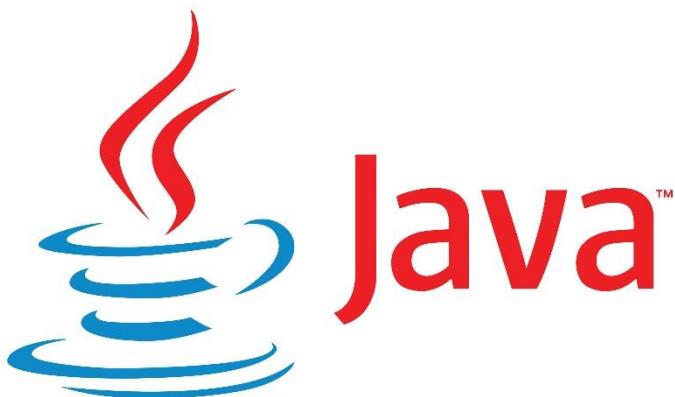
Introduction

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.

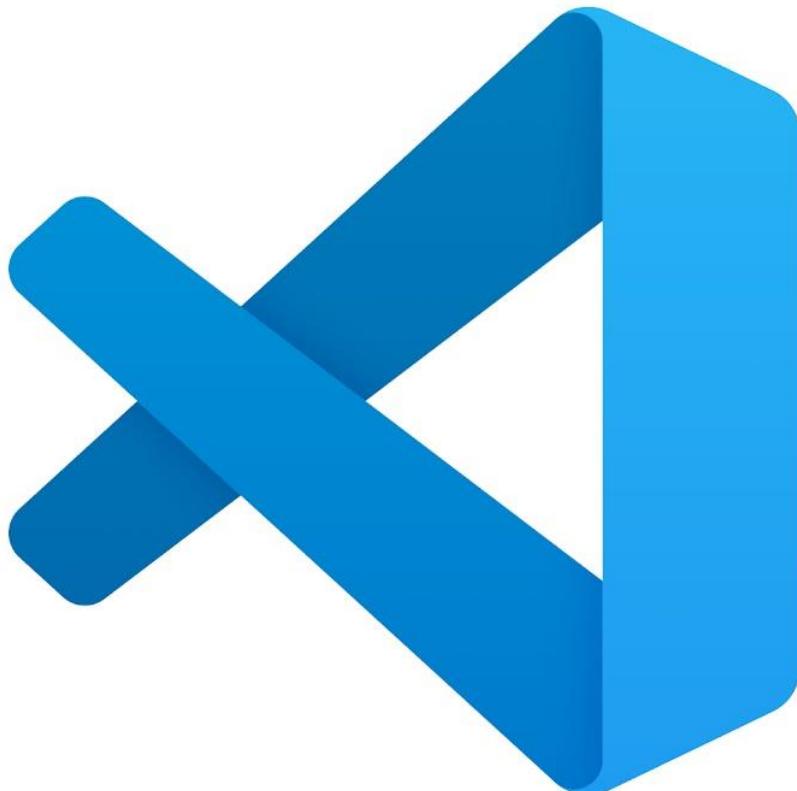
Programming can be written using varies programming languages. Some popular programming languages are:

- 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 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.

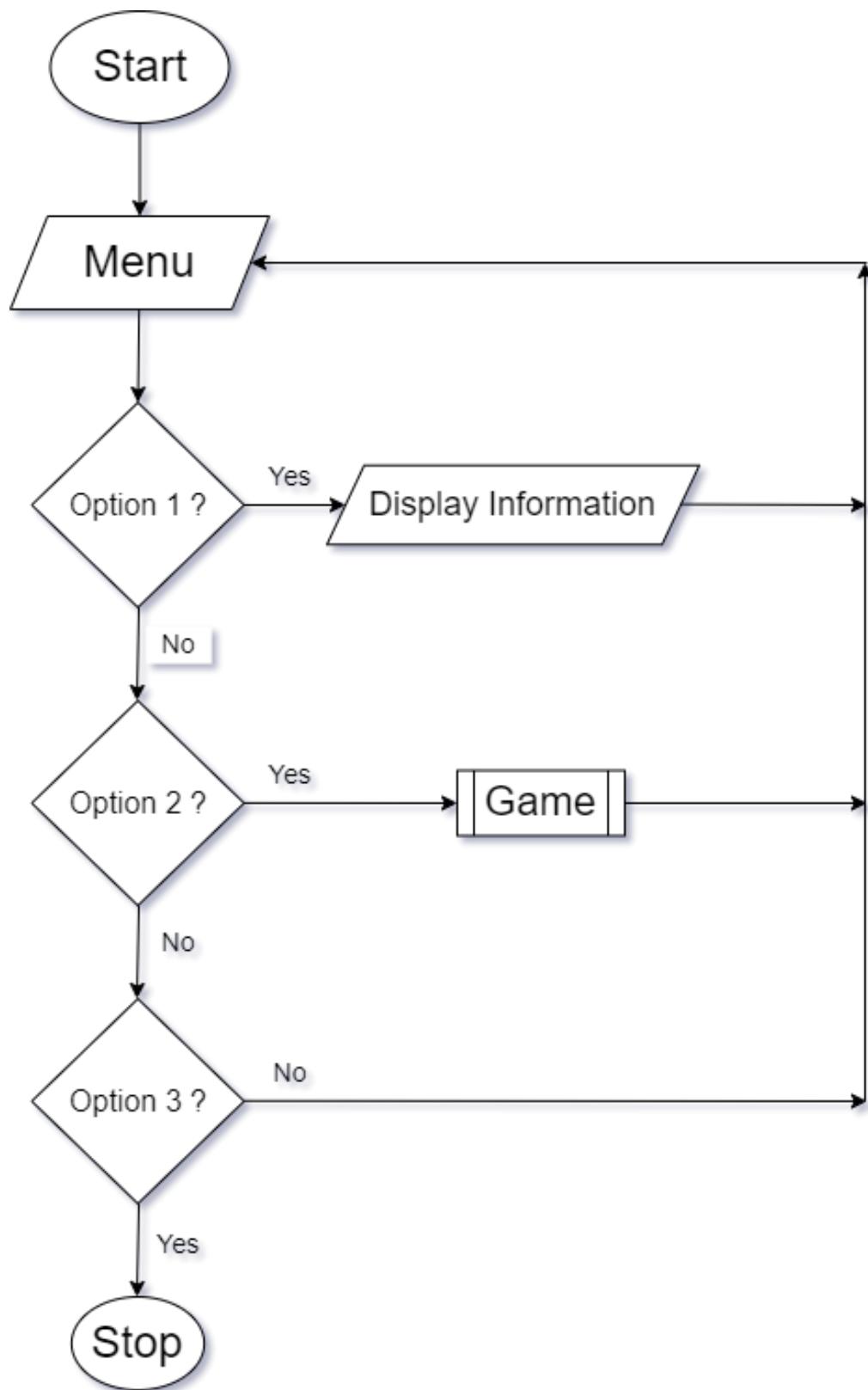
Chosen Problem: Problem 2

Problem Solution – Game Description

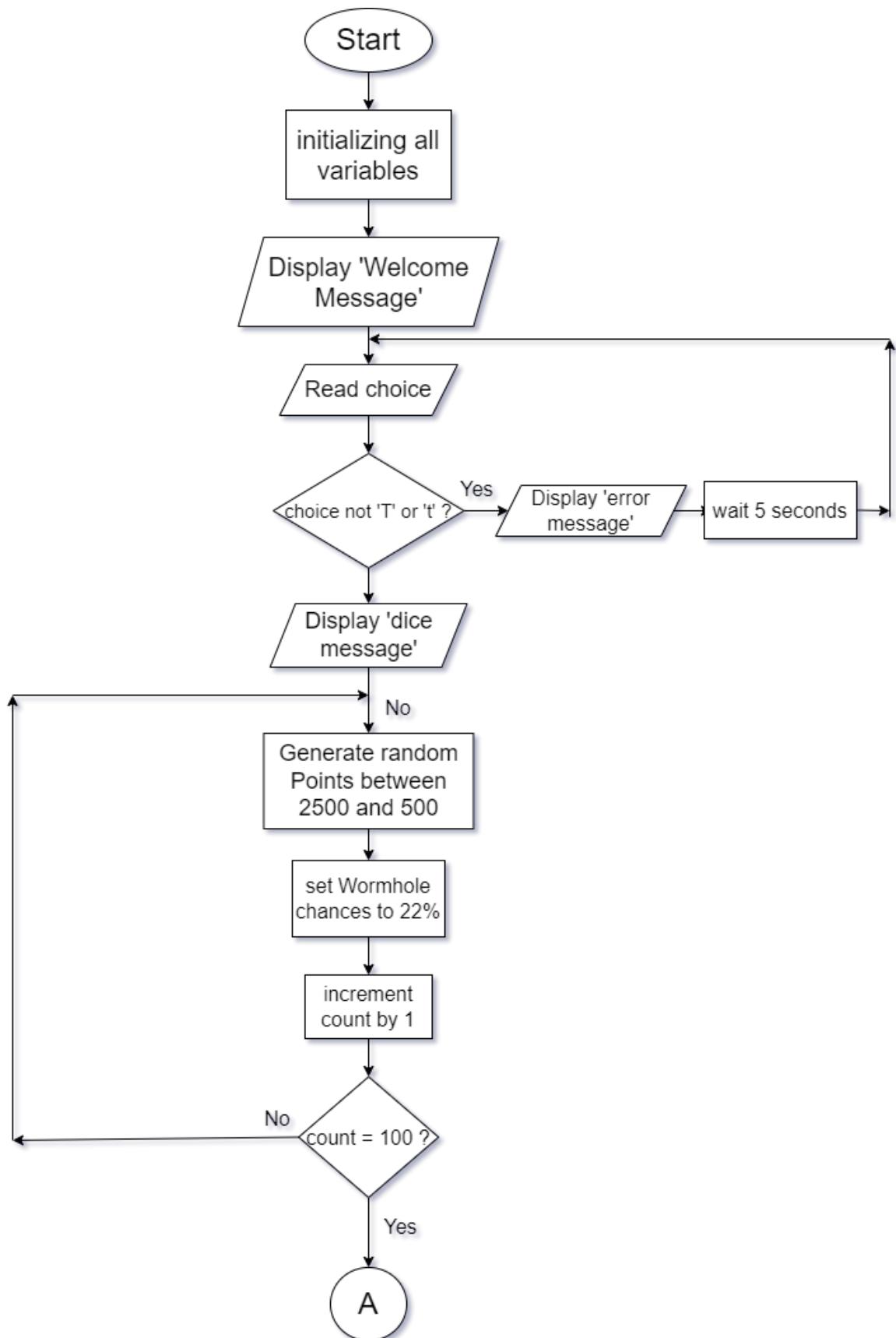
The program (game) that I developed is a space game about a space traveler who has a mission. His mission is to make it and land on Europa (Jupiter's Moon). But he first must travel from Earth to Mars and from there he has to travel across 100 asteroids, meanwhile he has several random chances of being 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!

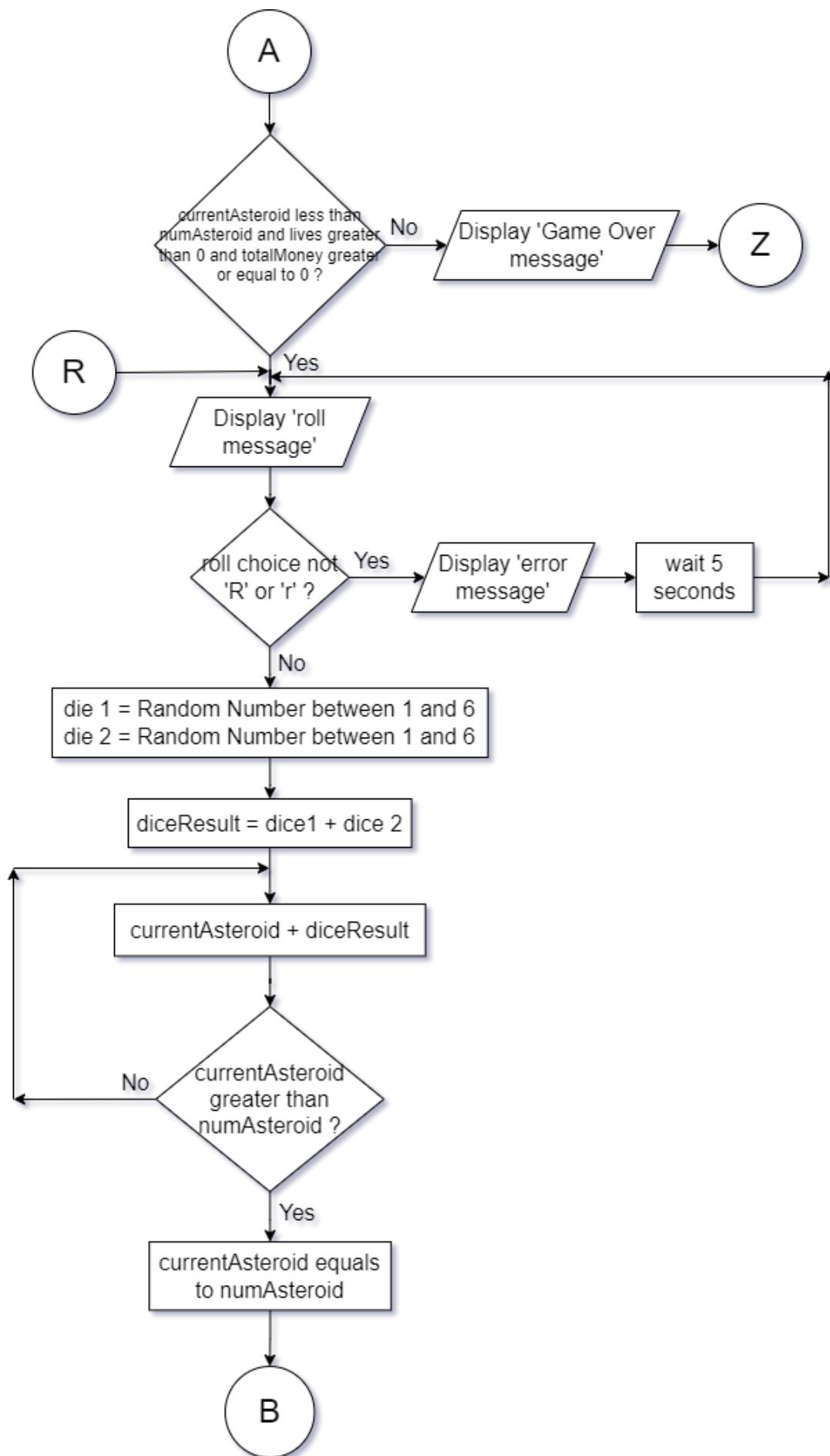


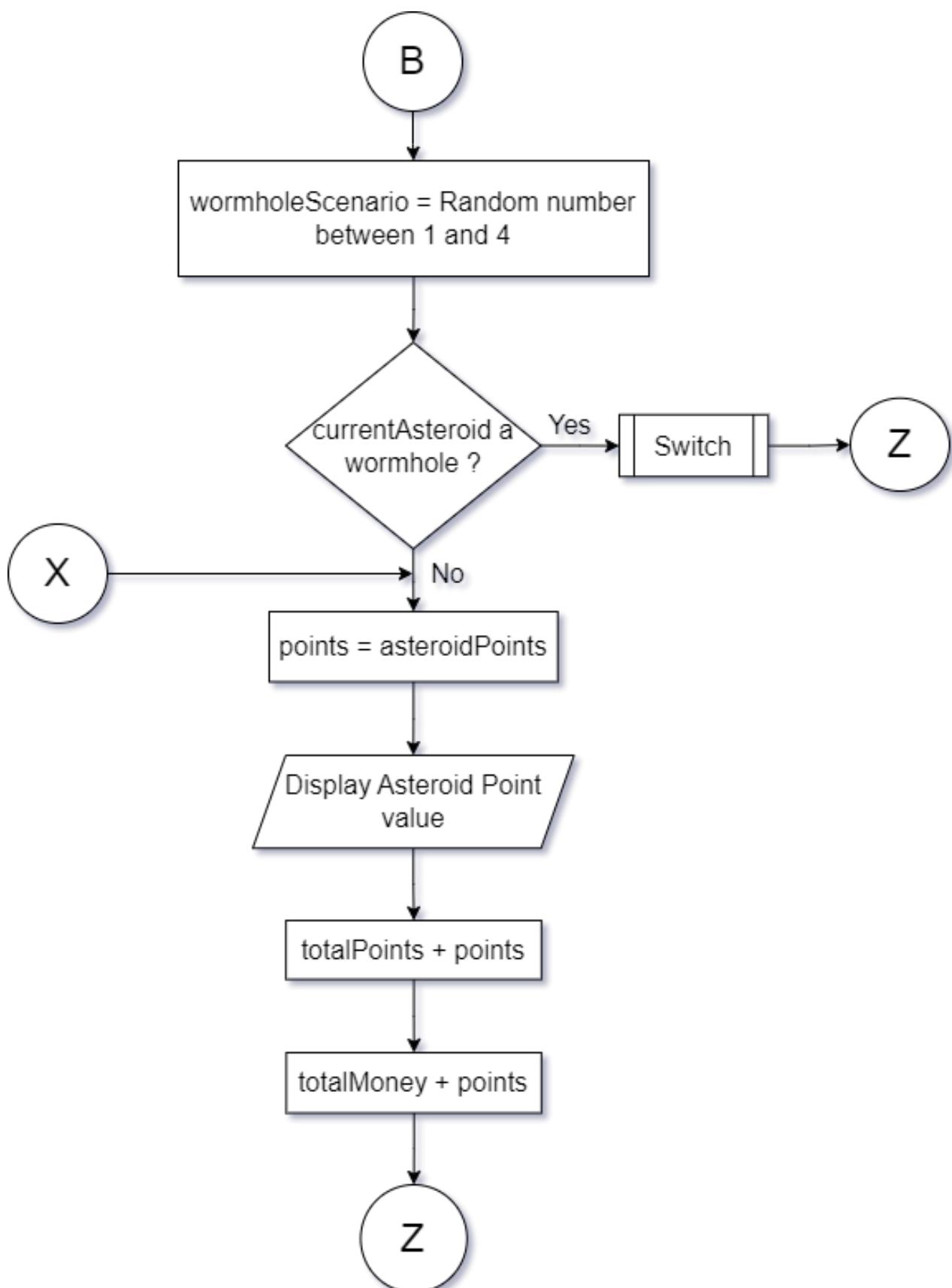
Flowchart – Main Menu



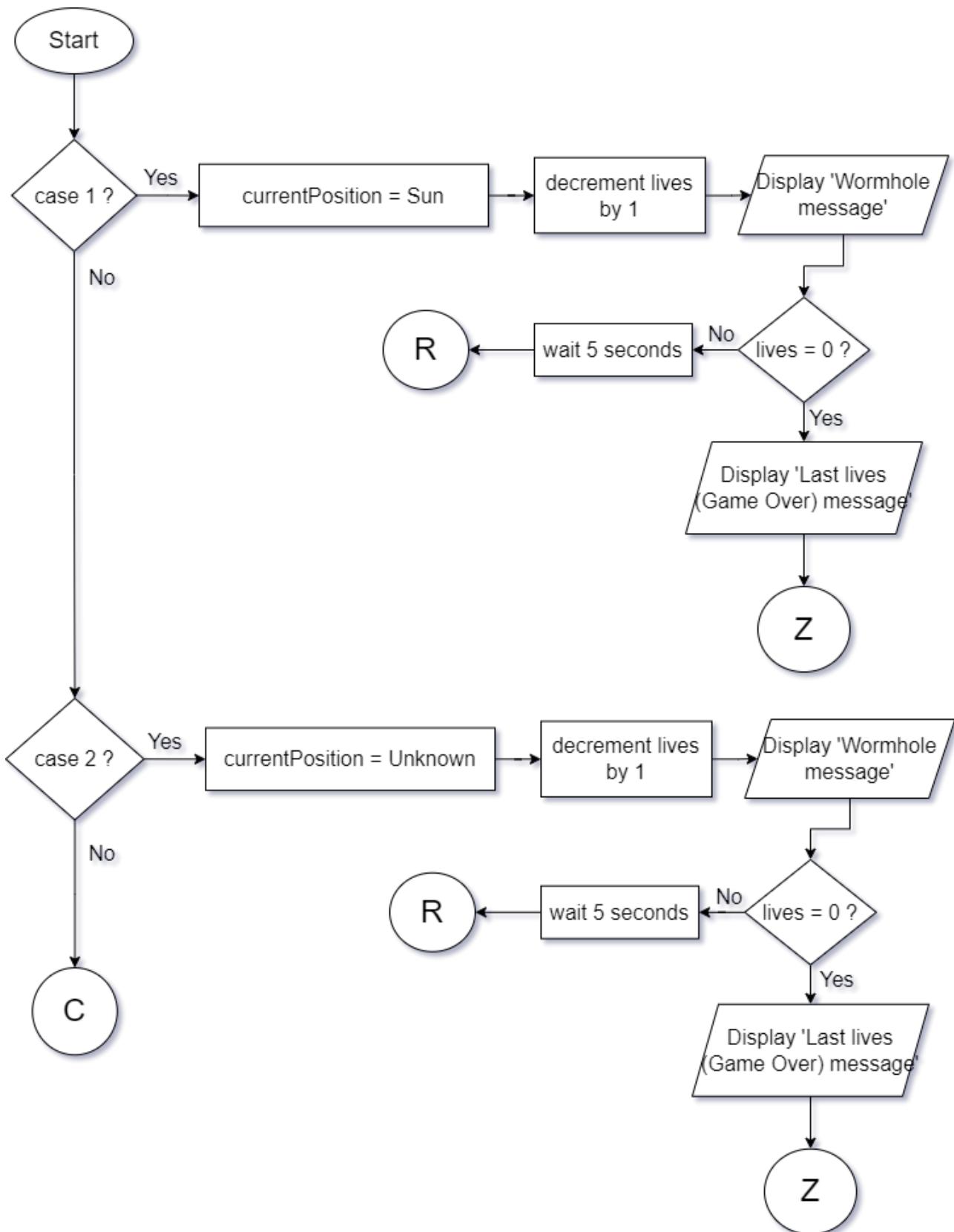
Flowchart – Game Method

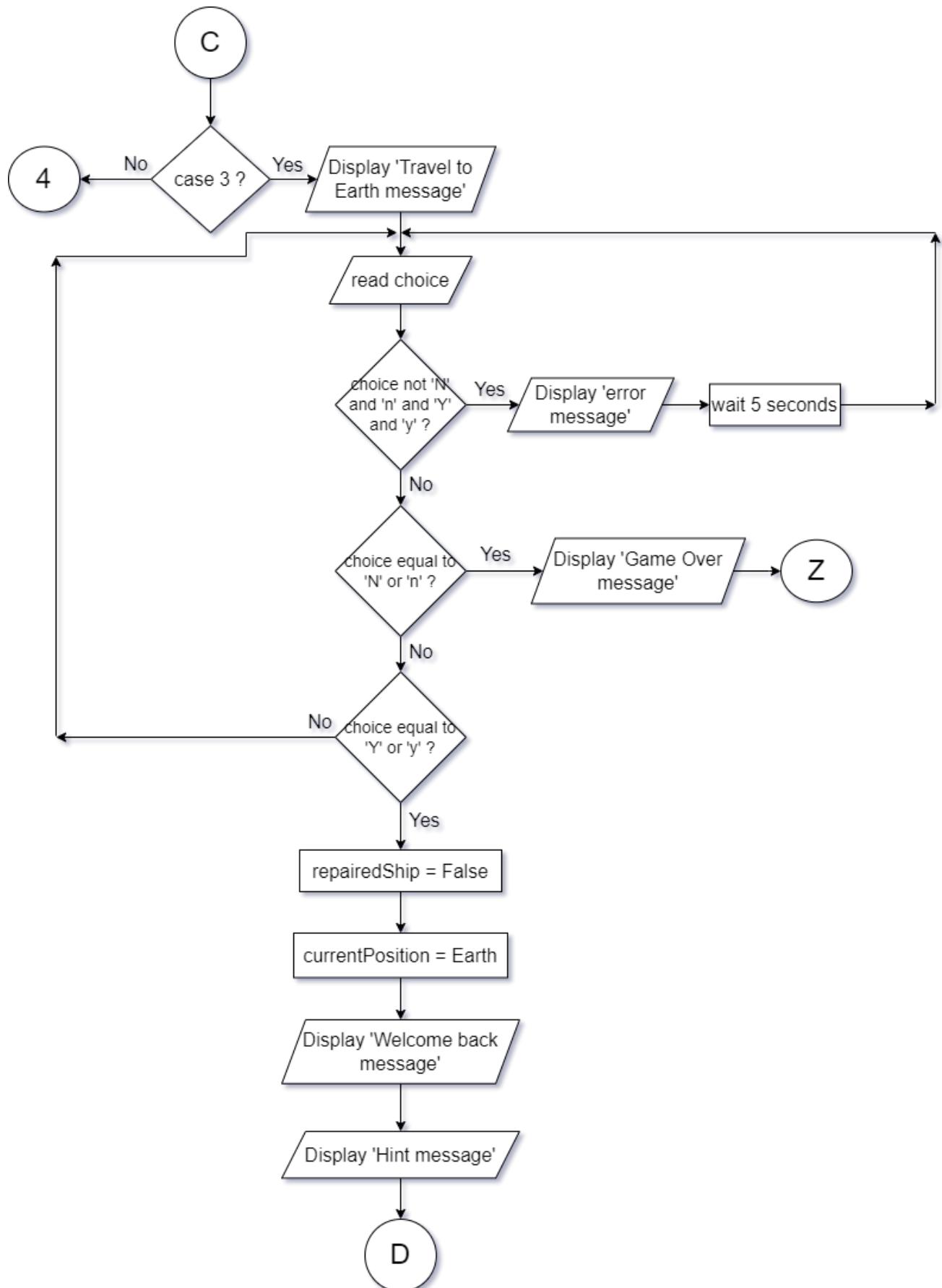


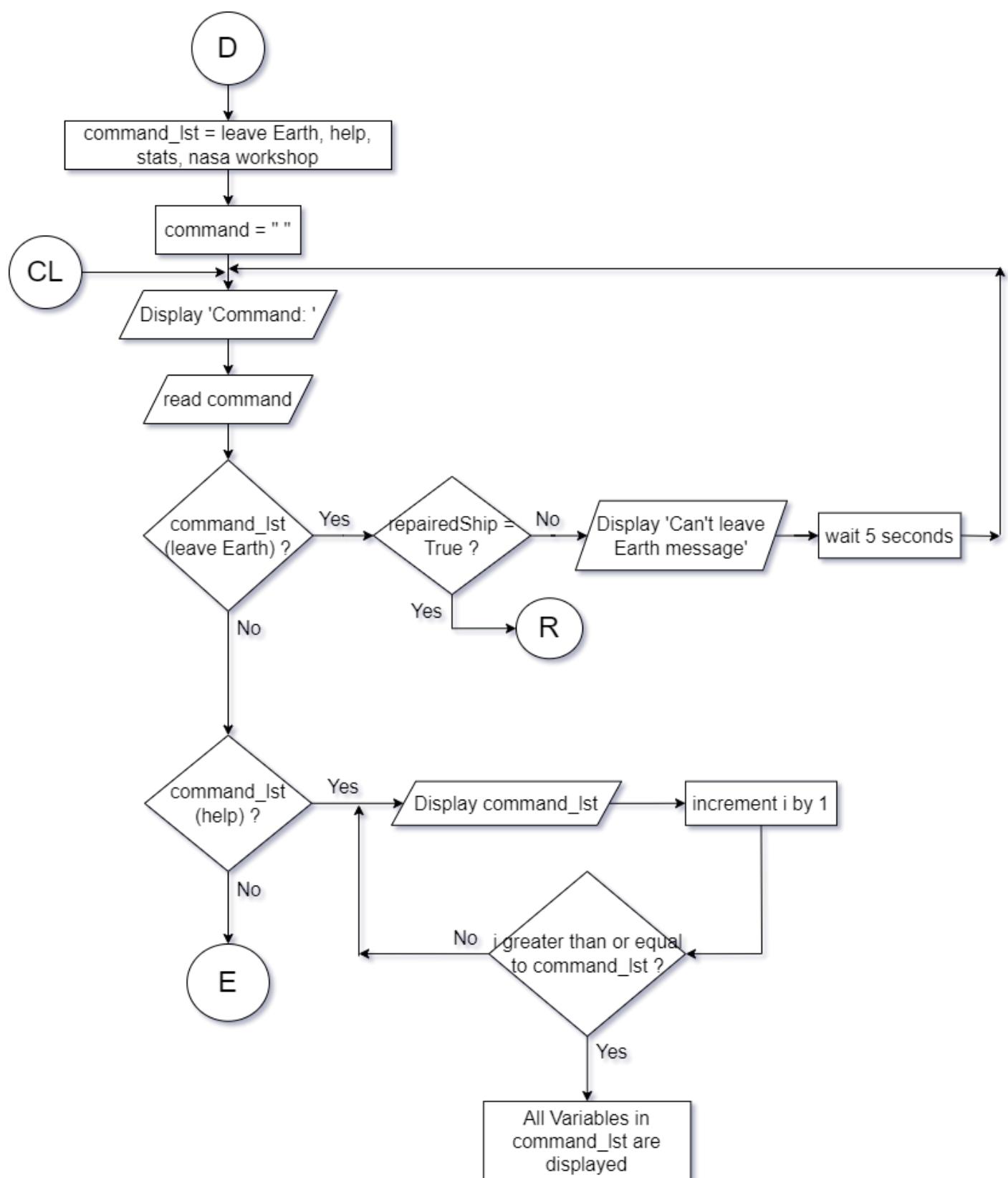


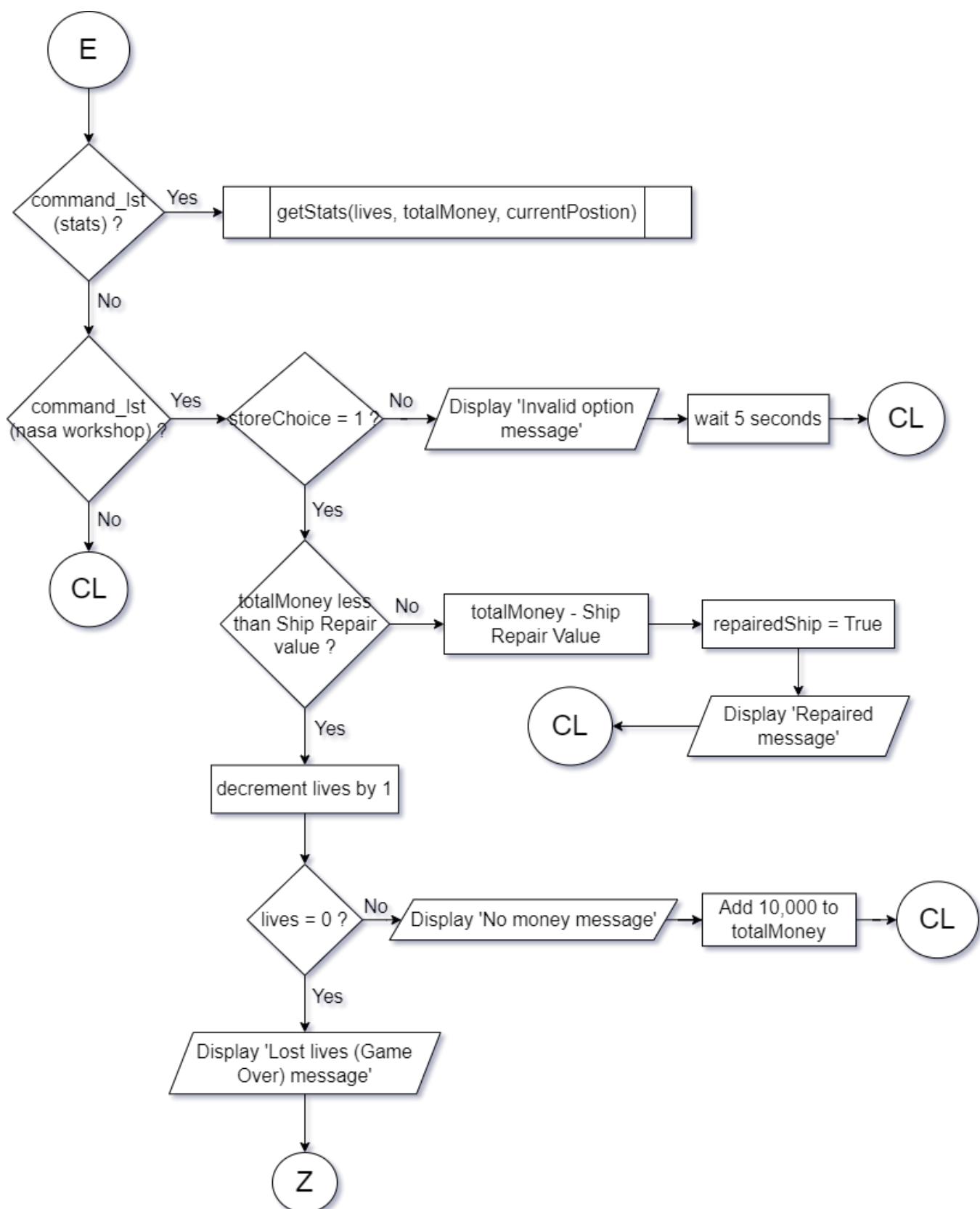


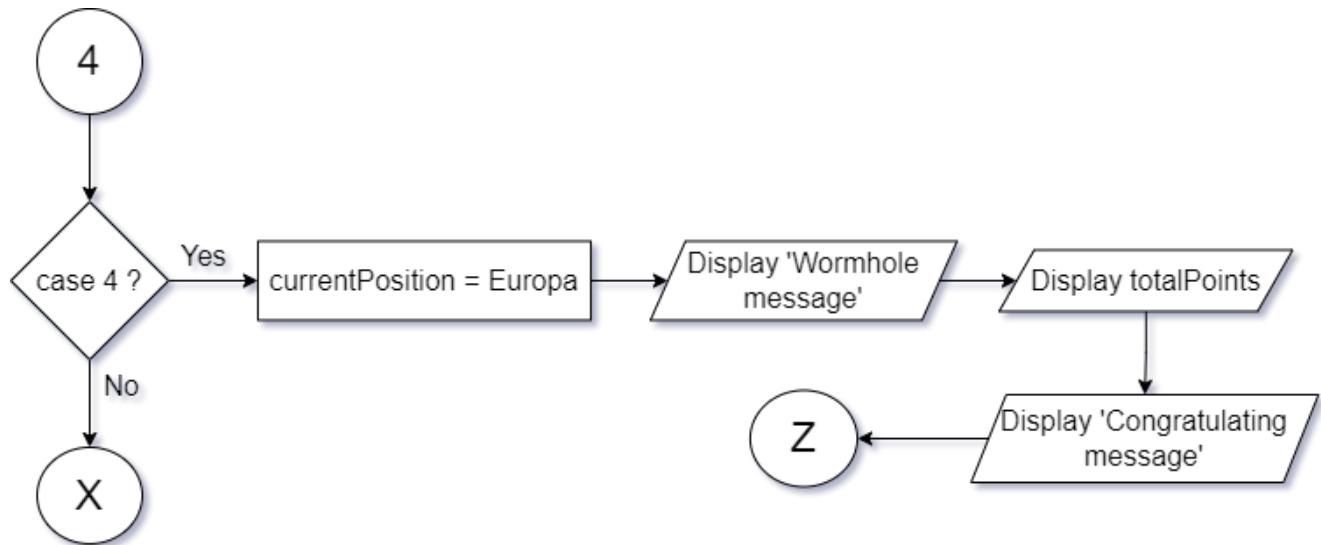
Flowchart – Switch Method











Listing of Program

C.S Form 4 Annual - Java Programming Project

```
1 //Importing extra tools for game
2 import java.io.IOException;
3 import java.util.Map;
4 import java.util.HashMap;
5
6 class SpaceRacer {
7     ****Welcome Screen*****
8     public static int mainMenu() throws IOException, InterruptedException {
9         new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
10
11        System.out.println(x:" _____");
12        System.out.println(x:"| WELCOME TO SPACE RACER |");
13        System.out.println(x:"| _____");
14        System.out.println(x:"| 1. Game Instructions |");
15        System.out.println(x:"| 2. Play Game |");
16        System.out.println(x:"| 3. Exit |");
17        System.out.println(x:"| _____ |");
18        System.out.println();
19
20        System.out.print(s:"Enter a choice: ");
21        int choice = Keyboard.readInt();
22        System.out.println();
23
24        return choice;
25    }
26    ****Game Methods & Functions*****
27
28 //This method is used to display the users' stats
29 public static void getStats(int lives, int totalMoney, String currentPosition) {
30     if (totalMoney == 0) { //This if statement checks how much money the user has and outputs the table required
31         System.out.println(x:" _____");
32         System.out.println(x:"| Lives | Money | Location |");
33         System.out.printf(format:"| %d | $-%d | %s |\n", lives, totalMoney, currentPosition);
34         System.out.println(x:"| _____ |");
35     }
36     else if (totalMoney >= 1000 && totalMoney <= 9999) {
37         System.out.println(x:" _____");
38         System.out.println(x:"| Lives | Money | Location |");
39         System.out.printf(format:"| %d | $-%d | %s |\n", lives, totalMoney, currentPosition);
40         System.out.println(x:"| _____ |");
41     }
42     else if (totalMoney >= 10000 && totalMoney <= 99999) {
43         System.out.println(x:" _____");
44         System.out.println(x:"| Lives | Money | Location |");
45         System.out.printf(format:"| %d | $-%d | %s |\n", lives, totalMoney, currentPosition);
46         System.out.println(x:"| _____ |");
47     }
48     else if (totalMoney >= 100000 && totalMoney <= 999999) {
49         System.out.println(x:" _____");
50         System.out.println(x:"| Lives | Money | Location |");
51         System.out.printf(format:"| %d | $-%d | %s |\n", lives, totalMoney, currentPosition);
52         System.out.println(x:"| _____ |");
53     }
54 }
55
56 }
```

C.S Form 4 Annual - Java Programming Project

```
57 //This method is used to assign the different values of different stuff available in the NASA Workshop
58 public static char getWorkshop(Map<String, Integer> inventory) {
59     System.out.println(x:" _____");
60     System.out.println(x:"| NASA Workshop |");
61     System.out.println(x:"| _____ |");
62     System.out.printf(format:"| 1. Repair Ship:      $-%d |\n", inventory.get(key:"Ship Repairs"));
63     System.out.println(x:"| _____ |");
64     System.out.println(x:"| _____ |");
65
66     System.out.println();
67     System.out.println(x:"Pick what you need Lieutenant.");
68     System.out.print(s:"Choice: ");
69     char storeChoice = Keyboard.readChar();
70
71     return storeChoice;
72 }
73
74 //This method is used to generate a random number
75 public static int getRandom(int max, int min) {
76     int random = (int)(Math.random() * (max - min) + min); //Generates a random number depending on the number assigned to the variable
77     return random;
78 }
79
80 //This method will print out a message depending on the Wormhole Scenario generated
81 public static void getWormholeScenario(boolean scenario3, int diceResult, String currentPosition, int lives, String message, String customMessage) {
82     if (!scenario3) { //This if statement will check whether the generated scenario was 3, if so the below code will be executed
83         if (customMessage.equals(anObject:"")) { //If customMessage was used then the below code will execute
84             System.out.println();
85             System.out.printf(format:"You rolled a %d and arrived at a wormhole!\n", diceResult);
86
87             System.out.printf(format:"Oh no! The wormhole led to %s%s\n", currentPosition, message);
88             System.out.printf(format:"Lives left: %d\n", lives);
89         }
90         else //If the above if statement does not execute then the below code will
91             System.out.printf(format:"\n%s", customMessage);
92     }
93     else { //If the above if statement does not execute then the below code will instead
94         System.out.println();
95         System.out.println("You rolled a " + diceResult + " and you got hit by a meteoroid!");
96         System.out.println(x:"You have to go back to Earth to repair your ship or else you will die and LOSE!!");
97     }
98 }
99
100 //This method will generate a message depending on what it was assigned to during coding
101 public static void generateMessage(String message, int miliSec) {
102     System.out.println(message);
103     try {
104         Thread.sleep(miliSec);
105     } catch (InterruptedException e) {
106         Thread.currentThread().interrupt();
107     }
108 }
109 }
```

C.S Form 4 Annual - Java Programming Project

```
110 //This method will execute whenever the user doesn't win, wins or to show them their points
111 public static void gameResult(boolean won, int totalPoints, String message, int miliSec) throws IOException, InterruptedException {
112     new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
113
114     if (!won) { //This if statement will check whether the user has not won, if so the below code will be executed
115         System.out.printf(format:"GAME OVER!\n%s", message);
116
117         try {
118             Thread.sleep(miliSec);
119         } catch (InterruptedException e) {
120             Thread.currentThread().interrupt();
121         }
122     }
123     else { //If the above if statement does not execute then the below code will
124         if (message.equals(anObject:"")) { //If Message was used then the below code will execute
125             System.out.println(x:"CONGRATULATIONS!");
126             System.out.println(x:"You have reached Europa!");
127             System.out.println("Your total score is: " + totalPoints + " points.");
128         }
129         else //If the above if statement does not execute then the below code will
130             System.out.print(message);
131
132         try {
133             Thread.sleep(miliSec);
134         } catch (InterruptedException e) {
135             Thread.currentThread().interrupt();
136         }
137     }
138 }
139
140 //This method will have a for loop running until all variables in command_lst[i] will be displayed
141 public static void getCommandLst(String[] command_lst) {
142     System.out.print(s:"\n| ");
143     for (int i = 0; i < command_lst.length; i++) {
144         System.out.print(command_lst[i] + " : ");
145     }
146     System.out.print(s:"|\n");
147 }
148 ****
149 
```

C.S Form 4 Annual - Java Programming Project

```
150 //*****Main Game Method*****
151 public static void game() throws IOException, InterruptedException {
152     new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
153
154 //*****Variable Initialization*****
155 int currentAsteroid = 1; //Declaring a variable named currentAsteroid to 1
156 int numAsteroids = 100; //Declaring a variable named numAsteroids to 100
157
158 int[] asteroidPoints = new int[numAsteroids]; //Declaring an array named asteroidPoints to the number of asteroids
159 boolean[] isWormhole = new boolean[numAsteroids]; //Declaring an array named isWormhole to the number of asteroids
160
161 int totalPoints = 0; //Declaring a variable named totalPoints to 0
162 int totalMoney = 10000; //Declaring a variable named totalMoney to 10000
163 int lives = 3; //Declaring a variable named lives to 3
164
165 Map<String, Integer> inventory = new HashMap<>(); //This map type String, String is connected to the method getStore()
166 inventory.put(key:"Ship Repairs", value:15000); //Declaring a map type String, String to "Repair Ship", "5-$15,000"
167
168 String[] position = {"Sun", "Mercury", "Venus", "Earth", "Mars", "Europa",
169 "Jupiter", "Saturn", "Uranus", "Neptune", "Pluto", "UNKNOWN"}; //Declaring an array named position and it stores all of the positions used in this game
170 String currentPosition = position[3]; //Starting position is set to position 3 (Earth)
171 //*****
172
173 char choice = ' ';
174 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
175     new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
176
177     System.out.println("Welcome Lieutenant!");
178     System.out.println("You are currently on " + currentPosition + ".");
179
180     System.out.print("Press T to travel to Mars: ");
181     choice = Keyboard.readChar();
182
183     if (choice != 'T' && choice != 't') //If user entered another letter than 'T' or 't', the game will print an error message
184         generateMessage(message:"\nError. Press T to travel to Mars.", milliSec:2000);
185     else { //If user entered 'T' or 't' in the variable choice
186         generateMessage(new String("\nYou have arrived on " + position[4] + "!\nUse the dice to travel between asteroids and reach Europa!"), milliSec:0);
187
188         //Randomizing Asteroid & Wormhole Chances
189         for (int i = 0; i < numAsteroids; i++) {
190             asteroidPoints[i] = getRandom(max:2500,min:500);
191             isWormhole[i] = Math.random() < 0.22;
192         }
193
194         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
195             System.out.println();
196             System.out.print("Press R to roll the dice and travel to a new asteroid: ");
197             char roll = Keyboard.readChar();
198
199             if (roll != 'R' && roll != 'r') //If user entered another letter than 'R' or 'r', the game will print an error message
200                 generateMessage(message:"\nError. Press R to roll the dice and travel to a new asteroid.\n", milliSec:2000);
201             else { //Checks if user entered 'R' or 'r' in the variable choice
202                 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
203                 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
204                 int diceResult = die1 + die2;
205                 currentAsteroid += diceResult;
206             }
207         }
208     }
209 }
```

C.S Form 4 Annual - Java Programming Project

```
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

    //This if will run everytime until currentAsteroid is equal to numAsteroids
    if (currentAsteroid > numAsteroids) {
        currentAsteroid = numAsteroids;
    }

    String wormhole_scenario = Integer.toString(((int)(Math.random() * 4) + 1)); //Variable challengeType is created to randomize 4 chances of different types of challenges
    if (isWormhole[currentAsteroid-1]) { //This if statement checks if the currentAsteroid is a Wormhole, if so it will execute
        switch (wormhole_scenario) {
            //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
            case "1":
                currentPosition = position[0];
                lives--;

                //The below code uses the getWormholeScenario() method
                getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:"Oops you burned up..", customMessage:"");

                if (lives != 0) {
                    try {
                        Thread.sleep(millis:5000);
                    } catch (InterruptedException e) {
                        Thread.currentThread().interrupt();
                    }
                }
                else {
                    //The below code uses the getResults() method
                    gameResult(won:false, totalPoints, message:"You lost all of your lives!", millisec:5000);
                    return;
                }
            break;

            //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
            case "2":
                currentPosition = position[11];
                lives--;

                //The below code uses the getWormholeScenario() method
                getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:"location in the universe and you're lost!", customMessage:"");

                if (lives != 0) {
                    try {
                        Thread.sleep(millis:5000);
                    } catch (InterruptedException e) {
                        Thread.currentThread().interrupt();
                    }
                }
                else {
                    //The below code uses the getResult() method
                    gameResult(won:false, totalPoints, message:"You lost all of your lives!", millisec:8000);
                    return;
                }
            break;
        }
    }
}
```

C.S Form 4 Annual - Java Programming Project

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

263     choice = ' ';
264     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'
265         System.out.println();
266         System.out.print($:"Do you want to travel back to Earth? (Y/N): ");
267         choice = Keyboard.readChar();

268         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
269             System.out.print($:"Error. Press (Y/N) to continue.");
270         }
271
272         try {
273             Thread.sleep(millis:2000);
274         } catch (InterruptedException e) {
275             Thread.currentThread().interrupt();
276         }
277     }

278     else if (choice == 'n' || choice == 'N') { //Checks if user entered 'n' or 'N', if so the below code will execute
279         //The below code uses the getResult() method
280         gameResult(won:false, totalPoints, message:"\\nYou did not repair your ship, therefore you couldn't continue the mission!", millisec:8000);
281         return;
282     }

283     else if (choice == 'y' || choice == 'Y') { //Checks if user entered 'y' or 'Y', if so the below code will execute
284         boolean repairedShip = false; //Resets the value of repairedShip to false everytime it runs again
285         currentPosition = position[3]; //Updates user's current position to position[3] (Earth)
286
287         new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
288
289         //The below code uses the generateMessage() method
290         generateMessage(new String("Welcome back to " + currentPosition + " Lieutenant!"), millisec:0);
291         generateMessage(message:"We noticed your ship needs a few repairs.", millisec:0);
292         generateMessage(message:"\\nHINT: Type 'help' for a list of commands.", millisec:0);

293         String command_lst[] = {"leave Earth", "help", "stats", "nasa workshop"};
294
295         String command = "";
296         while(!command.equals(command_lst[0]) || !repairedShip) { //This while loop will run until user enters "Leave Earth" and ship must be repaired
297             System.out.print($:"\\ncCommand: ");
298             command = Keyboard.readString();

299             if (command.equals(command_lst[0])) {
300                 if (repairedShip) //Checks whether the user repaired their ship, if yes it will clear screen
301                     new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
302
303                 else //Checks whether the user repaired their ship, if no below message will execute
304                     generateMessage(message:"\\nCan't Leave Earth, your ship is still broken ;)", millisec:5000);
305             }
306             else if (command.equals(command_lst[1])) {
307                 getCommandLst(command_lst); //This code uses the getCommandLst() method
308             }
309             else if (command.equals(command_lst[2])) {
310                 getStats(lives, totalMoney, currentPosition); //This code uses the getStats() method
311             }
312         }
313     }
314
315 }
316 }
```

C.S Form 4 Annual - Java Programming Project

```
317
318     else if (command.equals(command_lst[3])) {
319         char storeChoice = getWorkshop(inventory); //This code uses the getWorkshop() method
320
321         if (storeChoice == '1') { //Checks if user entered '1', if so the below code will execute
322             if (totalMoney < inventory.get(key:"Ship Repairs")) {
323                 lives--;
324                 if (lives == 0) {
325                     System.out.println();
326                     System.out.println(x:"It looks like you lost all of your lives!!!");
327                     System.out.println("Lives left: " + lives);
328                     System.out.println(x:"GAME OVER, Lieutenant!");
329
330                     try {
331                         Thread.sleep(millis:5000);
332                     } catch (InterruptedException e) {
333                         Thread.currentThread().interrupt();
334                     }
335                     return;
336                 }
337             else {
338                 //The below code uses the generateMessage() method
339                 generateMessage(message:"\nYou don't have enough money to buy this!", miliSec:0);
340                 generateMessage(message:"Therefore you lost a life.", miliSec:0);
341                 System.out.println("Lives left: " + lives);
342                 generateMessage(message:"", miliSec:3000);
343                 generateMessage(message:"In return..", miliSec:2500);
344                 generateMessage(message:"We have given you S-$1000 since you used up all of your money.", miliSec:0);
345                 totalMoney += 1000;
346             }
347         }
348         else {
349             //Subtracts the cost of ship repairs from totalMoney(current money value)
350             totalMoney -= inventory.get(key:"Ship Repairs");
351             repairedShip = true;
352
353             //The below code uses the generateMessage() method
354             generateMessage(message:"\nLieutenant, we have just repaired your ship.", miliSec:0);
355             generateMessage(new String("That costed you " + inventory.get(key:"Ship Repairs") + " Starfleet Dollars!"), miliSec:2500);
356             generateMessage(new String("\nYou have " + totalMoney + " Starfleet Dollars left!"), miliSec:0);
357             generateMessage(message:"Lieutenant, now you can keep going on your journey!\nGood Luck and don't DIE!", miliSec:5000);
358         }
359     }
360     else { //If user entered another number than '1', the game will print an error message
361         System.out.println();
362         System.out.println(x:"Invalid Option. Type 'Store' again to retry...");
363         System.out.println();
364     }
365 }
366
367
368
369 }
```

C.S Form 4 Annual - Java Programming Project

```
370 //Case 4 will send the user directly to Europa. Catch is will the user have collected a lot of points or not.
371 case "4":
372     currentPosition = position[5];
373
374     //The below code uses the getWormholeScenario() method
375     getWormholeScenario(scenario3:false, diceResult, currentPosition, lives, message:"", new String("Congratulations you won!\nThis wormhole led straight to " + currentPosition + "!"));
376
377     try {
378         Thread.sleep(millis:3000);
379     } catch (InterruptedException e) {
380         Thread.currentThread().interrupt();
381     }
382
383     //The below code uses the generateMessage() method
384     generateMessage(message:"\nLet's hope you managed to get as many points as you could then, Lieutenant!", millis:2000);
385     generateMessage(message:"Because...", millis:2000);
386     generateMessage(new String("Your total score is: " + totalPoints + " points."), millis:3000);
387
388     //The below code uses the gameResult() method
389     gameResult(won:true, totalPoints, message:"CONGRATULATIONS Again Lieutenant, Excellent Job!", millis:7000);
390
391     return;
392 }
393 else { //If challengeType isn't executed then user is told the dice result and that they moved to the next asteroid
394     int points = asteroidPoints[currentAsteroid-1];
395
396     //The below code uses the generateMessage() method
397     generateMessage(new String("\nYou rolled a " + diceResult + " and arrived at asteroid " + currentPosition + "."), millis:0);
398     generateMessage(new String("This asteroid is worth " + points + " points!"), millis:1000);
399
400     totalPoints += points; //Adds all points and stores them in the variable totalPoints
401     totalMoney += points; //Adds all points and stores them in the variable totalMoney
402
403 }
404
405
406 //Win Game method
407 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
408     //The below code uses the gameResult() method
409     gameResult(won:true, totalPoints, message:"", millis:7000);
410
411     return;
412 }
413
414 }
415 ****
416
417 }
```

C.S Form 4 Annual - Java Programming Project

```
418 //*****Instructions Menu*****/
419 public static void instructions() throws IOException, InterruptedException {
420     char choice = ' ';
421
422     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)
423         new ProcessBuilder(...command:"cmd", "/c", "cls").inheritIO().start().waitFor();
424
425         System.out.println("~-~-~-~-~-~-~-~-~-");
426         System.out.println("~-~-~-~-~-~-~-~-~-");
427         System.out.println("~-~-~-~-~-~-~-~-~-");
428         System.out.println("~-~-~-~-~-~-~-~-~-");
429         System.out.println("~-~-~-~-~-~-~-~-~-");
430         System.out.println("~-~-~-~-~-~-~-~-~-");
431         System.out.println("~-~-~-~-~-~-~-~-~-");
432         System.out.println("~-~-~-~-~-~-~-~-~-");
433         System.out.println("~-~-~-~-~-~-~-~-~-");
434         System.out.println("~-~-~-~-~-~-~-~-~-");
435         System.out.println("~-~-~-~-~-~-~-~-~-");
436         System.out.println("~-~-~-~-~-~-~-~-~-");
437         System.out.println("~-~-~-~-~-~-~-~-~-");
438         System.out.println("~-~-~-~-~-~-~-~-~-");
439         System.out.println("~-~-~-~-~-~-~-~-~-");
440         System.out.println("~-~-~-~-~-~-~-~-~-");
441         System.out.println("~-~-~-~-~-~-~-~-~-");
442         System.out.println("~-~-~-~-~-~-~-~-~-");
443         System.out.println("~-~-~-~-~-~-~-~-~-");
444         System.out.println("~-~-~-~-~-~-~-~-~-");
445         System.out.println("~-~-~-~-~-~-~-~-~-");
446         System.out.println("~-~-~-~-~-~-~-~-~-");
447         System.out.println("~-~-~-~-~-~-~-~-~-");
448         System.out.println("~-~-~-~-~-~-~-~-~-");
449         System.out.println("~-~-~-~-~-~-~-~-~-");
450         System.out.println("~-~-~-~-~-~-~-~-~-");
451         System.out.println("~-~-~-~-~-~-~-~-~-");
452         System.out.println("~-~-~-~-~-~-~-~-~-");
453         System.out.println("~-~-~-~-~-~-~-~-~-");
454         System.out.println("~-~-~-~-~-~-~-~-~-");
455         System.out.println("~-~-~-~-~-~-~-~-~-");
456         System.out.println("~-~-~-~-~-~-~-~-~-");
457         System.out.println("~-~-~-~-~-~-~-~-~-");
458         System.out.println("~-~-~-~-~-~-~-~-~-");
459         System.out.println("~-~-~-~-~-~-~-~-~-");
460         System.out.println("~-~-~-~-~-~-~-~-~-");
461         System.out.println("~-~-~-~-~-~-~-~-~-");
462         System.out.println("~-~-~-~-~-~-~-~-~-");
463         System.out.println("~-~-~-~-~-~-~-~-~-");
464         System.out.println("~-~-~-~-~-~-~-~-~-");
465         System.out.println("~-~-~-~-~-~-~-~-~-");
466
467         System.out.println();
468         System.out.print("Press B to continue: ");
469         choice = Keyboard.readChar();
470
471         if (choice != 'B' && choice != 'b') {
472             System.out.print("\nError. Press B to continue.");
473         }
474     }
475 }
```

```
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("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.

C.S Form 4 Annual - Java Programming Project

Year 10 – Programming Coursework – Marking Sheet

Criteria for Assessment	Maximum Mark	Actual Mark
<i>Problem Definition:</i>		
Description of the scope of the problem to be tackled	2	
<i>Solution of the problem:</i>		
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: _____