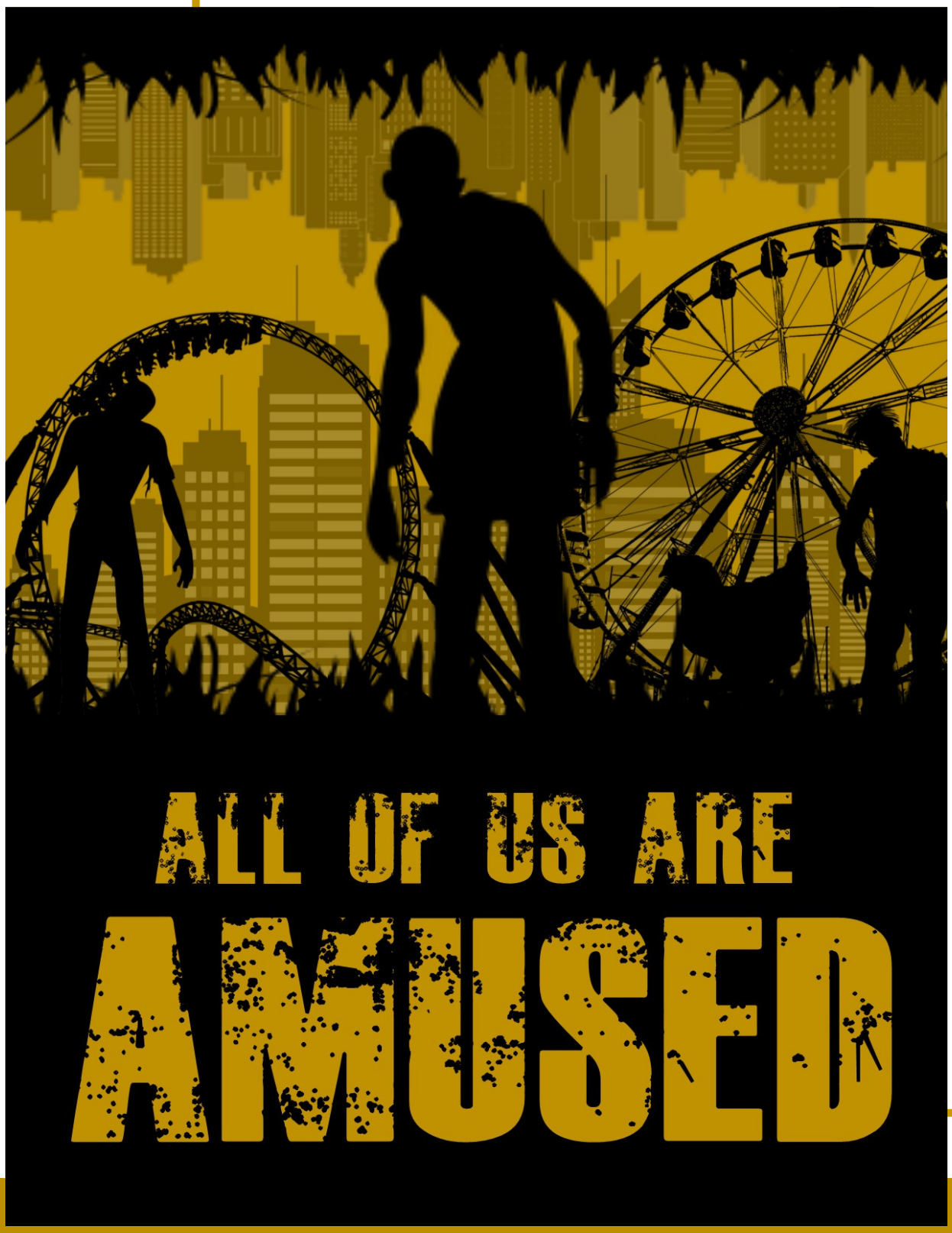
BSIT-1A GROUP 1

PROPOSAL

02 / 28 / 2022



Submitted by:

Aguilar, Josh Juri D.
Cartel, Marco Paulo D.
Espina, Eileen Hariette H.
Espina, Ellaine Hazelle H.
Evidor, Edward A.
Heraldo, Brylle Justin L.
Norial, Ma. Angelica D
Pile, John Lloyd C.
Pinca, Racheal Anne E.
Tan, Jayson Adrian Q.

DESCRIPTION and

This **mystery or adventure game** revolves around a zombie apocalypse, hidden clues, survival, and challenges. The goal is to find clues around a deserted amusement park about the zombie cure hinted at by a mysterious scientist. Nevertheless, it won't be easy as zombies are hanging around the park that the player needs to kill or escape from to advance on their adventure and solve the mysterious case of the game.

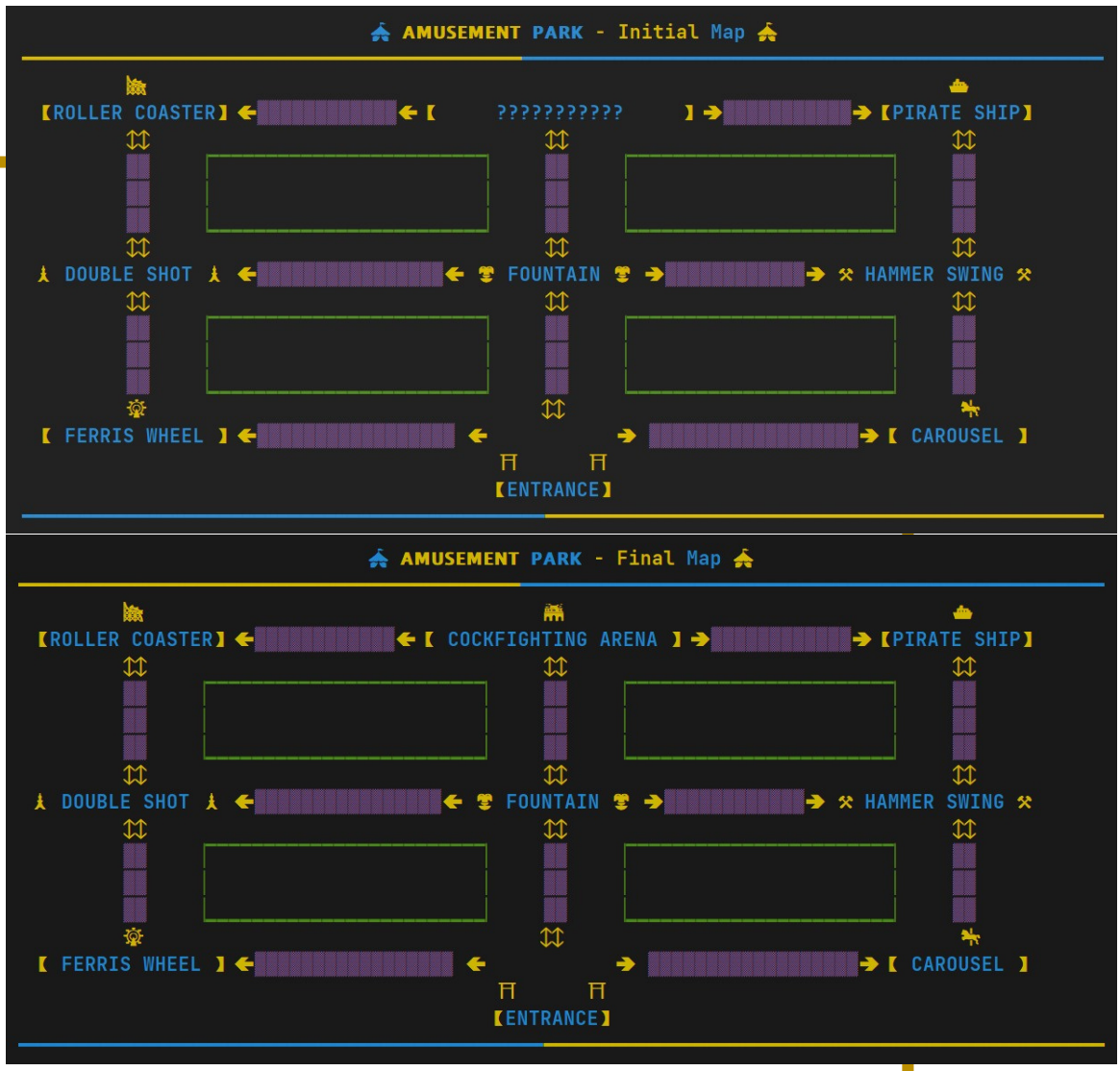
Backstory

100 years ago, a scientist was experimenting with a chicken for unknown reasons. Unfortunately, the chicken escaped. Meanwhile, a man fond of cockfighting saw the chicken and took it home. He entered a cockfighting match and won. But then, the chicken suddenly attacked a bystander. A day later, the chicken died. However, the person bitten started to feel sick.

The person started coughing uncontrollably and passed it down to his family and friends. Everyone who existed at that time paid no mind to the sudden increase in coughing every day. Nevertheless, the coughing worsened to reddish eyes, excessive hair loss, cracked skin, and a thrilling thirst for human flesh just a day later. Eventually, the people bitten or killed horrifyingly came back to life as mindless zombies. Due to the virus' mystery, the government failed to find the cure and opted to build walls in a town far from the cities.

Centuries later, when humanity is on the brink of extinction, a single detective arrives at the place where the apocalypse started. An abandoned amusement park where the cockfighting took place. Surprisingly, he found a century-old letter left by the chicken's scientist. But then, the scientist only wrote one thing:

Ride all those attractions and survive.



How to play the game

1. The player should be careful and attentive to **survive** and not get killed by the zombies.
2. The player will **obtain a map of the amusement park** to choose their target location.
3. The **starting position** of the player and zombies inside the amusement park will be **random** each time (two zombies per road).
4. The player will have a **pistol as a primary weapon** (four bullets), so they must get the **baseball bat** (lifespan of four) for **security and extra self-defense.**
5. There will be **nine locations** and, specifically, **seven rides** that the player needs to **visit to gain a set of letters per ride**, namely: Roller Coaster (1), Pirate Ship (2), Double Shot (3), Hammer Swing (5), Ferris Wheel (6), Carousel [8], and Cockfighting Arena (9).

6 To gain the set of letters, the player must **answer the designated questions** (four incorrect answers) per ride concerning the game's backstory.

7 After getting all clues, the player must **decode the letters** to find the hidden mystery of the cure (four incorrect answers).

Objective

The **objective of this game** is to primarily fulfill the requirements of Output 1 for the Fundamentals of Data Structures and Algorithm lecture. Specifically, this game displays a map of a **zombie-infested amusement park** where the player must go to the best route to solve the mystery of the zombie apocalypse. Moreover, the program stores and suggests **accurate calculations for the distance and shortest path among the rides.** All in all, the proponents resorted to this fictional map concept to creatively showcase the **implementation of 1D array for data storage and stimulate the players with the game's mystery**.

FLOWCHART LINK

https://docs.google.com/drawings/d/1T1bNlAsRH_ITcZdAc-FBp10fY_ywl6zDGD85InytP2A/edit?usp=sharing

Timeline

February 9

Ideation

Ideation session for the title and concept of the proposal project.

February 10

Task Allocation

Division of tasks and completion of the description.

February 11-13

Draft

Created a draft of the map and flowchart.

February 14 -16

Finalization of Proposal

Finalized the initial write-up proposal.

February 17

Guide

Research and division of tasks prior to program coding.

February 18-20

Coding

The group started coding for their assigned parts.

February 21

Proposal Submission

Submission of initial write-up proposal on LMS.

February 22-26

Checking and Revising

Error-checking and completion of the program.

February 27

Finalization

Filmed and edited the final output.

February 28

Output Submission

Submission of final output and presentation.

Output Requirements

Given the discussion on Introduction to Data Structures and Algorithm, **create a simple program** that will satisfy the given requirements below:

1. Create a simple program using python programming language under console interface.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

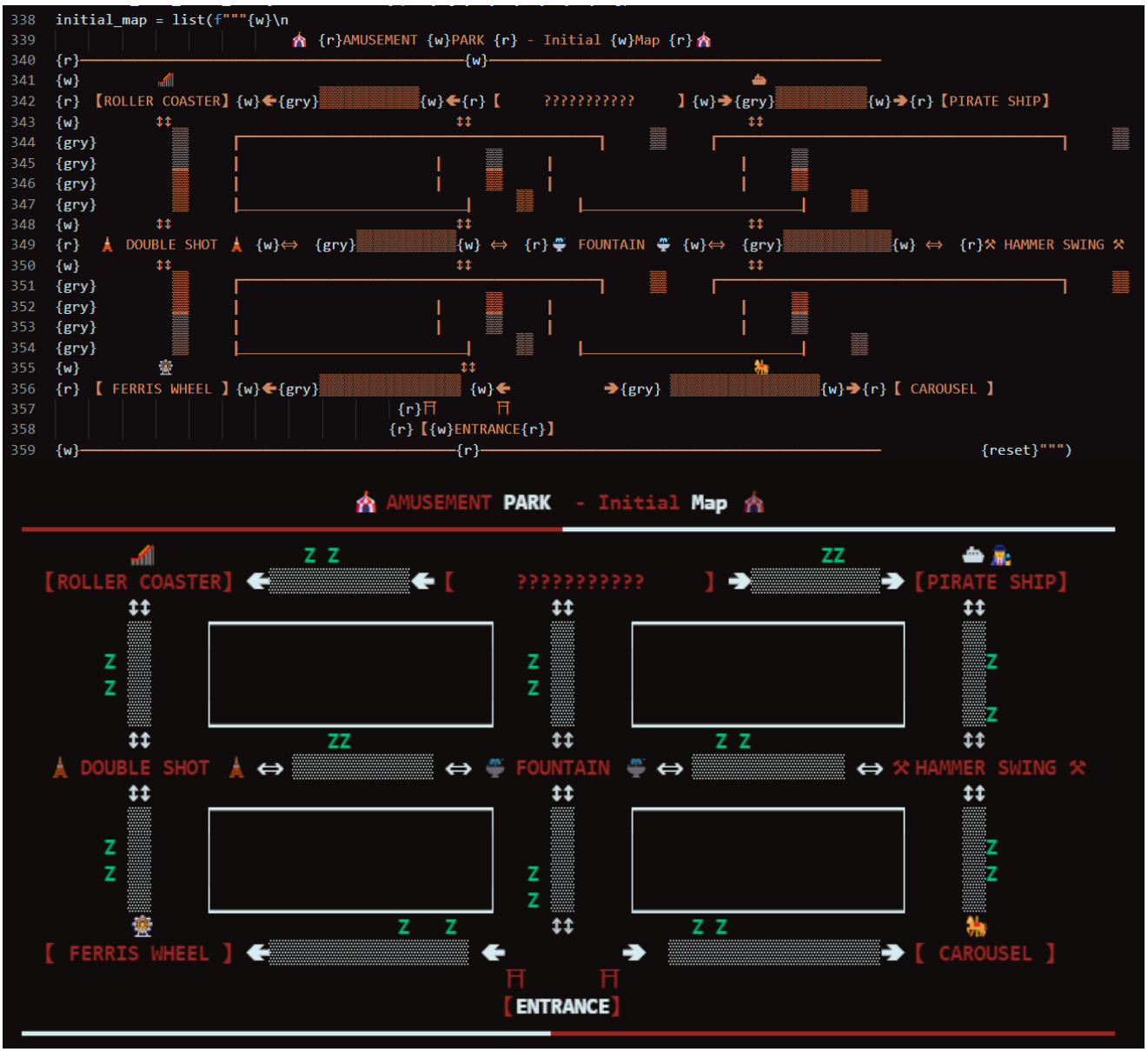
```
209 Title = '''
218
219 # Timer
220 def timer():
221     time.sleep(.35)
222
223 # Colors
224 c = ['\033[0;31m', '\033[1;91m', '\033[0;37m']
225 print(f'\n{c[0]}{Title}')
226
227 # Objective
228 while True:
229     Continue = input(f'\n                                                    Press ENTER key for MECHANICS ')
230     if not Continue:
231         break
232     else:
233         continue
234 print(f'\n\t\t{c[2]}1. Be careful and attentive to survive and not get killed by the zombies.\n');
235 timer()
236 print(f'\t\t2. You will obtain a map of the amusement park to choose your target location.\n');
237 timer()
238 print(f'\t\t3. The starting position of the player and zombies inside the amusement park will be random each time.\n');
239 timer()
240 print(f'\t\t4. You will have a pistol as a primary weapon (life span of four), then you must get the baseball bat (lifespan of four) \n\t\t
241 timer()
242 print(f'\t\t5. There will be nine locations and, specifically, seven rides that you need to visit to gain a set of letters per ride, \n\t\t
243 timer()
244 print(f"\t\t6. To gain the set of letters, you must answer the designated questions (four incorrect answers) per ride concerning the \n\t\t
245 timer()
246 print(f'\t\t7. After getting all clues, you must decode the letters to find the hidden mystery of the cure (four incorrect answers).\n')
247 timer()
248
```

```
 ALL OF US ARE AMUSED

                    Press ENTER key for MECHANICS

    1. Be careful and attentive to survive and not get killed by the zombies.

    2. You will obtain a map of the amusement park to choose your target location.

    3. The starting position of the player and zombies inside the amusement park will be random each time.

    4. You will have a pistol as a primary weapon (life span of four), then you must get the baseball bat (lifespan of four)
       for security and extra self-defense.

    5. There will be nine locations and, specifically, seven rides that you need to visit to gain a set of letters per ride,
       namely: Roller Coaster (1), Pirate Ship (2), Double Shot (3), Hammer Swing (5), Ferris Wheel (6), Carousel [8], and
       Cockfighting Arena (9).

    6. To gain the set of letters, you must answer the designated questions (four incorrect answers) per ride concerning the
       game's backstory.

    7. After getting all clues, you must decode the letters to find the hidden mystery of the cure (four incorrect answers).


                    Press ENTER key to view BACKSTORY
```

2. and 4. The program should display a detailed map based from Image 1 : Map Reference.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

```
initial_map = list(f"""{w}\n
                    {r}AMUSEMENT {w}PARK {r} - Initial {w}Map {r}
{r}                                                {w}
{w}
{r} [ROLLER COASTER] {w}<{gry}          {w}<{r} [     ??????????     ] {w}>{gry}          {w}>{r} [PIRATE SHIP]
{w}
{gry}
{gry}
{gry}
{gry}
{w}
{r}  DOUBLE SHOT  {w}<>  {gry}          {w} <> {r} FOUNTAIN  {w}<>  {gry}          {w} <> {r} HAMMER SWING
{w}
{gry}
{gry}
{gry}
{gry}
{w}
{r} [ FERRIS WHEEL ] {w}<{gry}          {w}<          >{gry}          {w}>{r} [ CAROUSEL ]
                                {r}
                                {r} [{w}ENTRANCE{r}]
{w}                                                {r}                                                {reset}""")
```



3. At initial program execution:

a. generate random data (meters) from (10- 100 value) from and to of each location.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

```
roads_distances = array("i",[random.choice([i for i in range(10, 101)]) for i in range(12)])
```

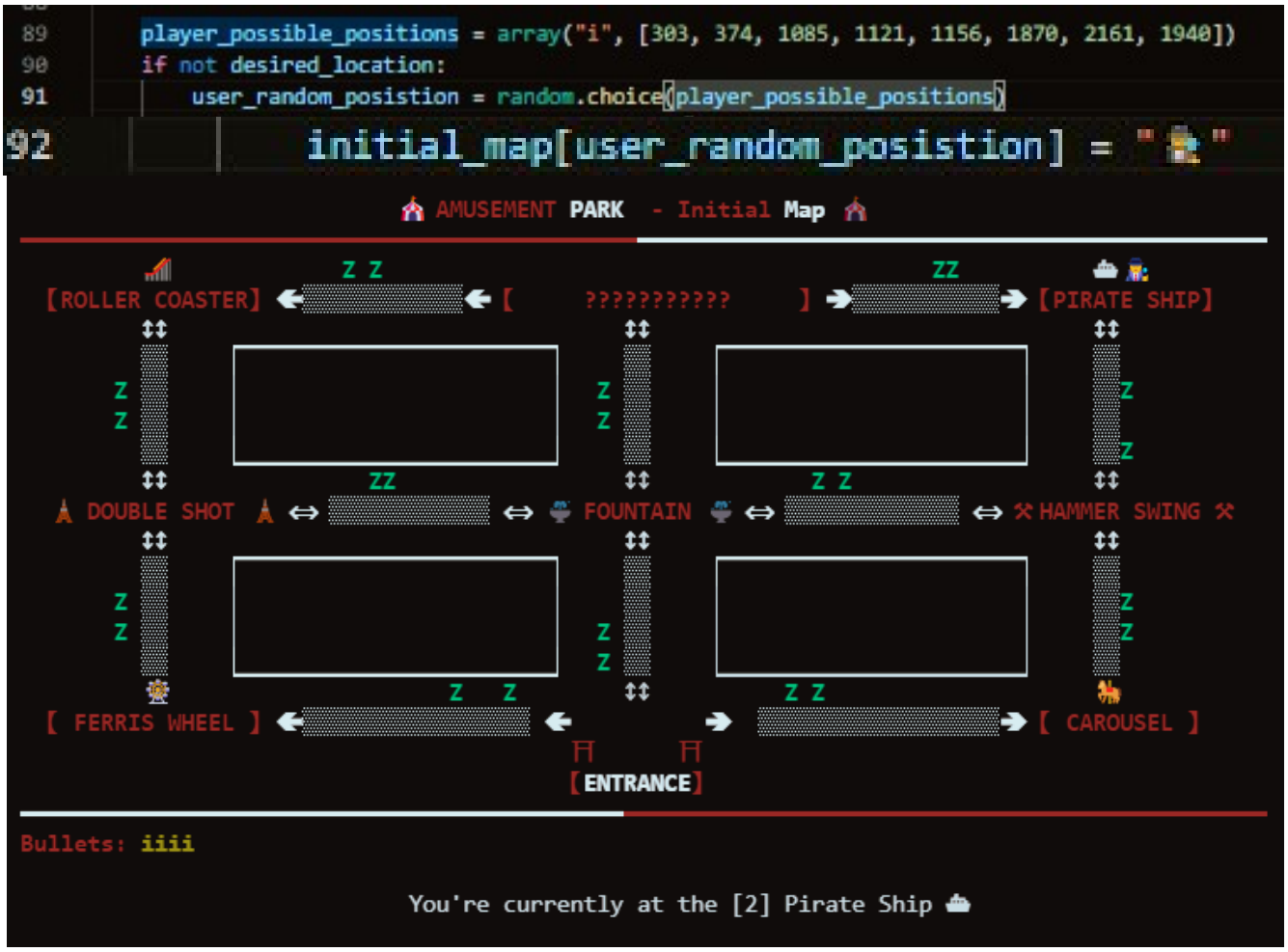
Roller Coaster → ??????? - 74m	??????? → Pirate Ship - 16m
Roller Coaster → Double Shot - 82m	??????? → Fountain - 80m
Pirate Ship → Hammer Swing - 85m	Double Shot → Fountain - 50m
Fountain → Hammer Swing - 28m	Double Shot → Ferris Wheel - 39m
Fountain → Entrance - 29m	Hammer Swing → Carousel - 33m
Ferris Wheel → Entrance - 91m	Entrance → Carousel - 59m

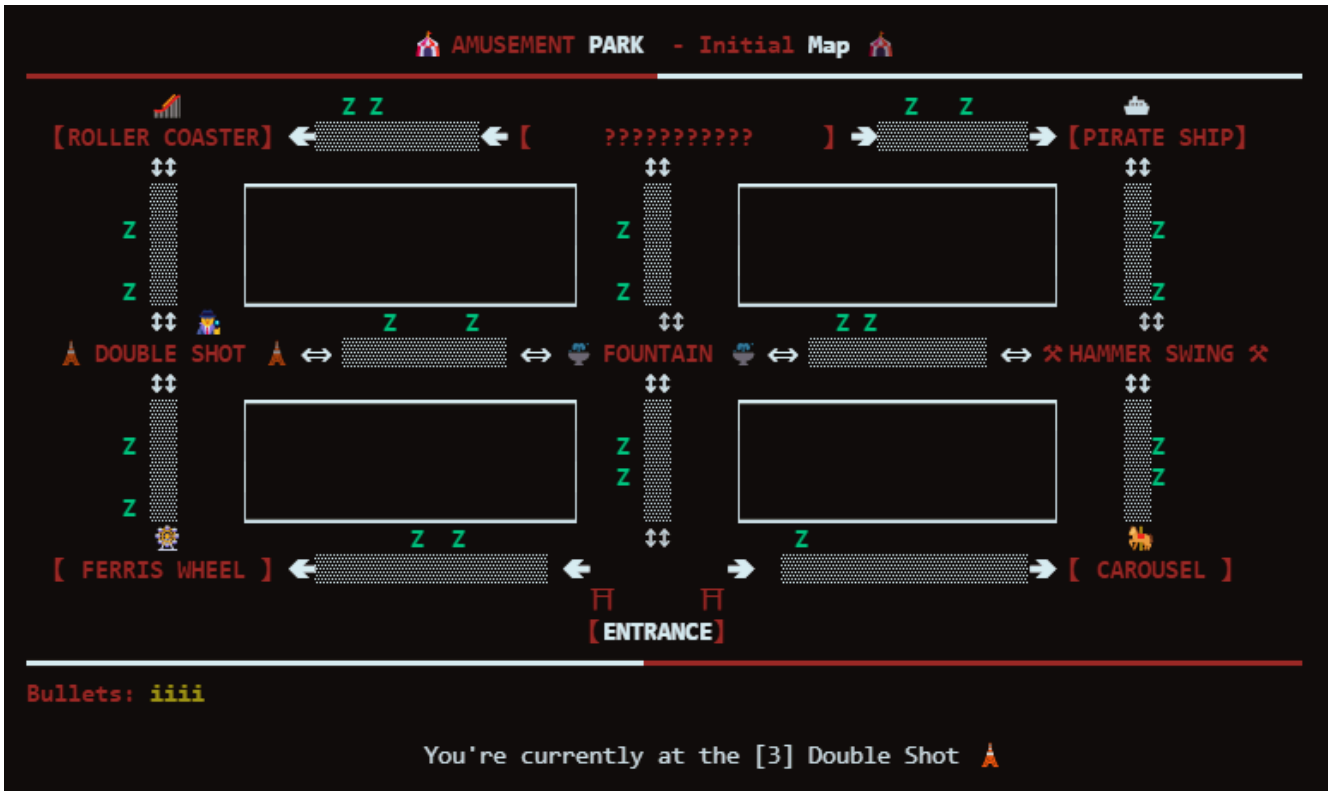
DSA WRITE-UP PROPOSAL

3. At initial program execution:
b. randomize your position at the map
5. Display your current location in the map.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

```
89      player_possible_positions = array("i", [303, 374, 1085, 1121, 1156, 1870, 2161, 1940])
90      if not desired_location:
91          user_random_posistion = random.choice(player_possible_positions)
92              initial_map[user_random_posistion] = "🚶"
```





DSA WRITE-UP PROPOSAL

6. Create an option/menu that will prompt the user to select which location he/she would like to visit.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

```
159     if answered_questions == 6:
160         print(f"{r}\nAvailable Locations:{reset}\n\n [9] Cockfighting Arena 🏟️")
161         available_locations_asInt = array("i", [9, desired_location])
162     else:
163         available_locations = ['[1] Roller Coaster 🎢', "[2] Pirate Ship 🏴‍☠️", "[3] Double Shot 🗼", "[4] Fountain ⛲",
164                                "[5] Hammer Swing ⚒", "[6] Ferris Wheel 🎡", "[7] Entrance ⛩", "[8] Carousel 🐎"]
165         available_locations_asInt = array("i", [i + 1 for i in range(len(available_locations)) if
166                                                 i != available_locations.index(user_current_position)]) + array("i", [
167             available_locations.index(user_current_position) + 1])
168         available_locations.remove(user_current_position)
```

```
Available Locations:

[1] Roller Coaster 🎢
[2] Pirate Ship 🏴‍☠️
[4] Fountain ⛲
[5] Hammer Swing ⚒
[6] Ferris Wheel 🎡
[7] Entrance ⛩
[8] Carousel 🐎

                    Your current location has a set of clues.

Would you like to answer the question to get them YES|NO?
```

7. - 10. Calculate and display :

- Total distance from location A to location B for all possible routes.
- Shortest path or Best route
- Estimated arrival time given the speed of .1 meter per min. Display time needed in hour.

SCREENSHOTS OF CODE + CONSOLE DEMONSTRATION

```
ALL ROUTES:
Double Shot → Ferris Wheel = 94m
Double Shot → Fountain → Entrance → Ferris Wheel = 244m
```

```
ALL ROUTES:
 Double Shot = 52m

BEST ROUTE: Double Shot = 52m
Estimated Time of Arrival: 8.67 hours
```

```
 8  def calculate_time(total_distance):
 9      minutes = total_distance * 10
10      return f"{round(minutes/60, 2)} hours"
11  def get_route(roads_distances,current_loc, destination, answered_questions):
12      if answered_questions == 6: cockfighting_arena = "Cockfighting Arena"
13      else: cockfighting_arena = "????????"
14      if current_loc == 1:
15          if destination == 2:
16              route1 = [roads_distances[2],roads_distances[5],roads_distances[3],roads_distances[1]]
17              route2 = [roads_distances[2],roads_distances[5],roads_distances[6],roads_distances[4]]
18              route3 = [roads_distances[2],roads_distances[5],roads_distances[8],roads_distances[11],roads_distances[9],roads_distances[4]]
19              routes = [eval("+".join(list(map(str,route1)))),eval("+".join(list(map(str,route2)))),eval("+".join(list(map(str,route3))))]
20              route1_asString = f" Double Shot → Fountain → {cockfighting_arena} → Pirate Ship = {r}{str(routes[0])}m{reset}"
21              route2_asString = f" Double Shot → Fountain → Hammer Swing → Pirate Ship = {r}{str(routes[1])}m{reset}"
22              route3_asString = f" Double Shot → Fountain → Entrance → Carousel → Hammer Swing → Pirate Ship = {r}{str(routes[2])}m{reset}"
23              print(f"{r}\nALL ROUTES:{reset}\n"+"\n".join([route1_asString,route2_asString,route3_asString]))
24              best_route_distance = min(routes)
25              final_bestRoute = []
26              for i  in range(len(routes)):
27                  if best_route_distance == routes[i]:
28                      final_bestRoute.append(eval(f"route{i+1}_asString"))
29              final_bestRoute = min(final_bestRoute)
30              print(f"{r}\nBEST ROUTE: {reset}",final_bestRoute)
31              print(f"{r}Estimated Time of Arrival:{reset} {calculate_time(best_route_distance)}")
```

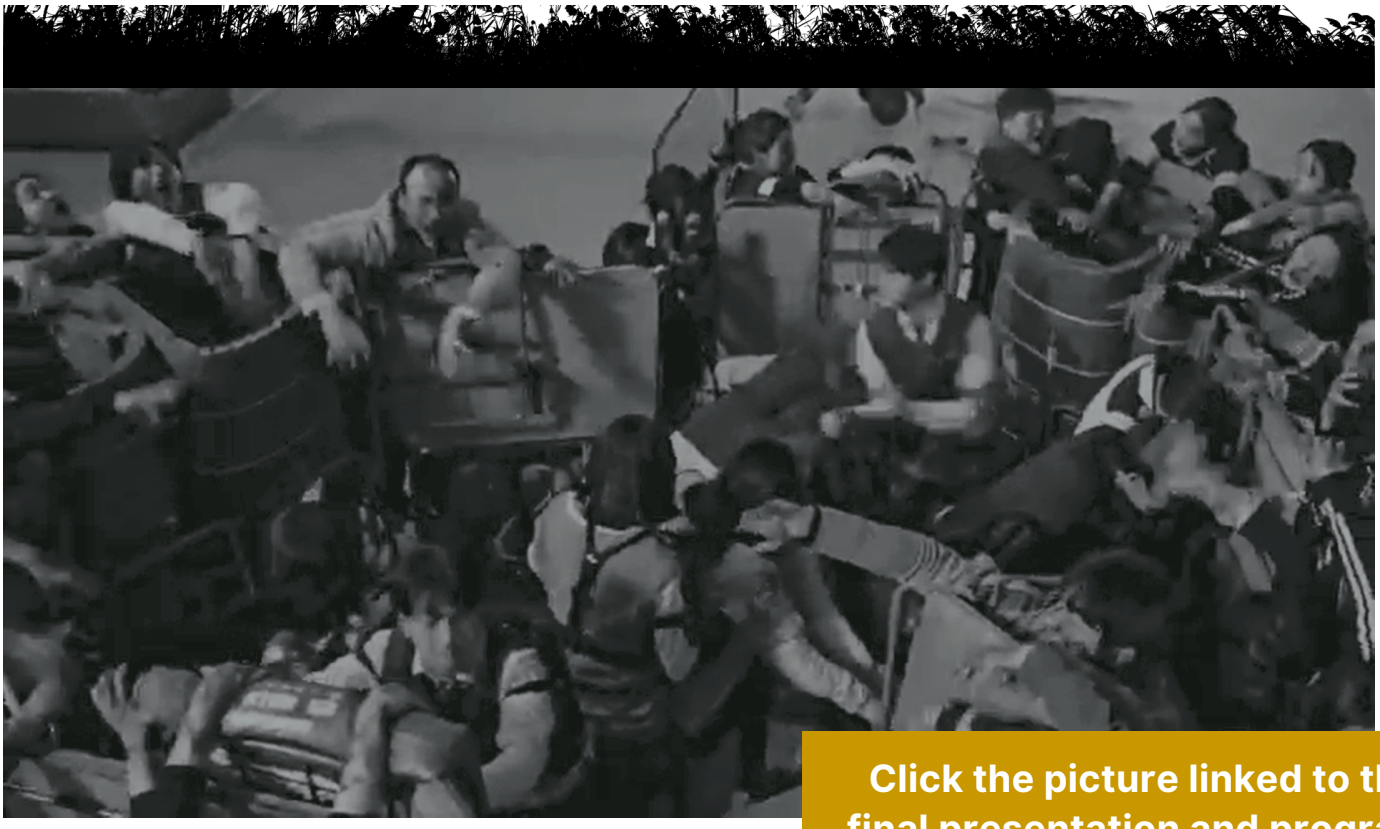
DSA WRITE-UP PROPOSAL

11. Implement 1D array for storing data and values needed in the program.

GENERAL SCREENSHOT OF PYTHON CODE

```
382   road1 = array("i", [i for i in range(313, 324)])
383   road2 = array("i", [i for i in range(354, 364)])
384   road3 = array("i", [682, 771, 872, 961])
385   road4 = array("i", [718, 807, 908, 997])
386   road5 = array("i", [758, 847, 948, 1037])
387   road6 = array("i", [i for i in range(1095, 1107)])
388   road7 = array("i", [i for i in range(1130, 1141)])
389   road8 = array("i", [1469, 1558, 1659, 1748])
390   road9 = array("i", [1505, 1594, 1695, 1784])
391   road10 = array("i", [1545, 1634, 1735, 1824])
392   road11 = array("i", [i for i in range(1880, 1896)])
393   road12 = array("i", [i for i in range(1914, 1931)])
```

G1 Video Presentation



Click the picture linked to the final presentation and program!

FINAL RATING

The rating is per the overall contribution and quality work of the members judged by the group leader.

BSIT-1A GROUP 1

MEMBERS AND RATING

10%

ACTIVENESS

10%

TIME MANAGEMENT

20%

COOPERATION

30%

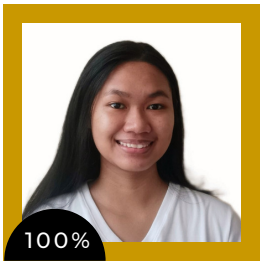
TASK COMPLETION

30%

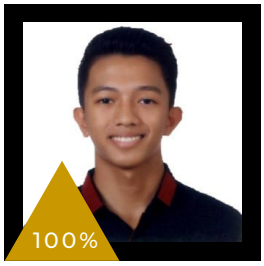
QUALITY WORK

Eileen Hariette H. Espina
GROUP 1 LEADER

John Lloyd C. Pile
GROUP 1 MEMBER

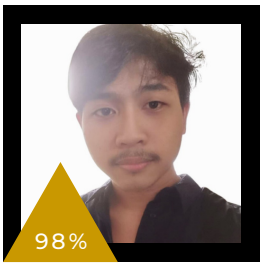


Ma. Angelica D. Norial
GROUP 1 MEMBER

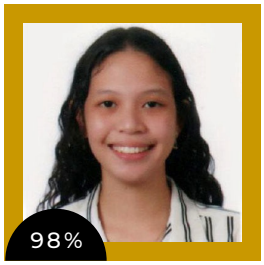


Marco Paulo D. Cartel
GROUP 1 MEMBER

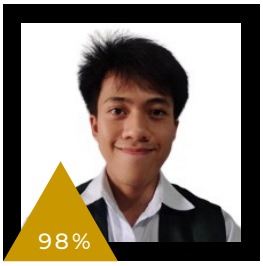
Ellaine Hazelle H. Espina
GROUP 1 MEMBER



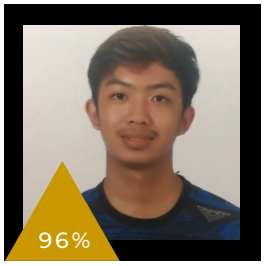
Brylle Justin L. Heraldo
GROUP 1 MEMBER



Racheal Anne E. Pinca
GROUP 1 MEMBER



Jayson Adrian Q. Tan
GROUP 1 MEMBER



Edward A. Evidor
GROUP 1 MEMBER

Josh Juri D. Aguilar
GROUP 1 MEMBER

Prepared For :

Bryan Arellano

DSA WRITE-UP PROPOSAL