



Republic of the Philippines  
**BATANGAS STATE UNIVERSITY**  
Main Campus II  
Alangilan, Batangas City



**COLLEGE OF INFORMATICS AND COMPUTING SCIENCES**

**FINAL PROJECT REPORT**  
**CS 121: Advanced Computer Programming**  
**First Semester A.Y. 2020-2021**

**PROTECT THE PLANET Z**

**Submitted by**

**DE CHAVEZ, ALFREDERICK**  
**RONQUILLO, JULIUS**  
**SERRANO, LEMUEL**

**Under Supervision of:**  
**RICHELLE M. SULIT**

**DECEMBER 2020**

## ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to our teacher, Ms.

Richelle Sulit for the continuous support of our project, for her patience, motivation, enthusiasm and immense knowledge. Her guidance help us in all the time of writing our code. She surely is a great advisor.

We thank our fellow students that also guide us when we need help in doing our work.

We thank the youtube channel TokyoEdTech that tells a lot about the module we use which is turtle and tkinter.

This assignment couldn't be completed without the effort and cooperation of our group members.

Last but not the least we would like to thank our family and friends for their constant source of inspiration.

## TABLE OF CONTENTS

Title	Page
Title Page	1
Acknowledgement	2
Table of Contents	3
List of Tables and Figures	4
Overview	4
Objective	6
Methodology	7
Flowchart	8
Functions	10
Program Output	
Appendices	
Source Code	17
Members Duties And Responsibilities	37
Member's Profile	38

## LIST OF TABLES AND FIGURES

### List of Figures

Figure No.	Name	Page
1	Get Started!	10
2	Lecture	11
3	Quiz	12
4	Quiz (Result)	13
5	Gameplay	14
6	Win Result	15
7	Game Over Result	16

### OVERVIEW

The title of our project is “Protect the Planet Z”. It is a program or game that provides some knowledge about the importance of protecting the nature. At the first part of the project, there are some lecture and ideas about the climate change. The next part is the actual game where the user needs to protect a certain thing from different enemies. The

thing to be protected symbolize the planet and the enemies are like storms, bad weather or other disasters. When a certain number of enemies enter the area around the planet, then it the game is over. On the other hand, when the character that is protecting the planet destroy the enemies then the user can proceed to another category.

### **Importance of the Program**

This program will be help the users to be disciplined and responsible on our surroundings. There are a lot of calamities we experienced and will experience every year, this program contains ideas on how can we response to those calamities or minimize the effect of it.

The users can also help others and share what lessons they have learned from the program.

## OBJECTIVE

The world as we all know is experiencing different calamities. This year gives us a warning that we cannot ignore. It reminds us that we need to protect the world now. We never know what's going to happen next. What if the disasters that happened was just a small part of a greater disaster that may happen in the future?

We create our project called "Protect the Planet Z", for a certain purpose of putting in the minds of the user that there is no second planet, this is the last one, that is why it is best to protect it.

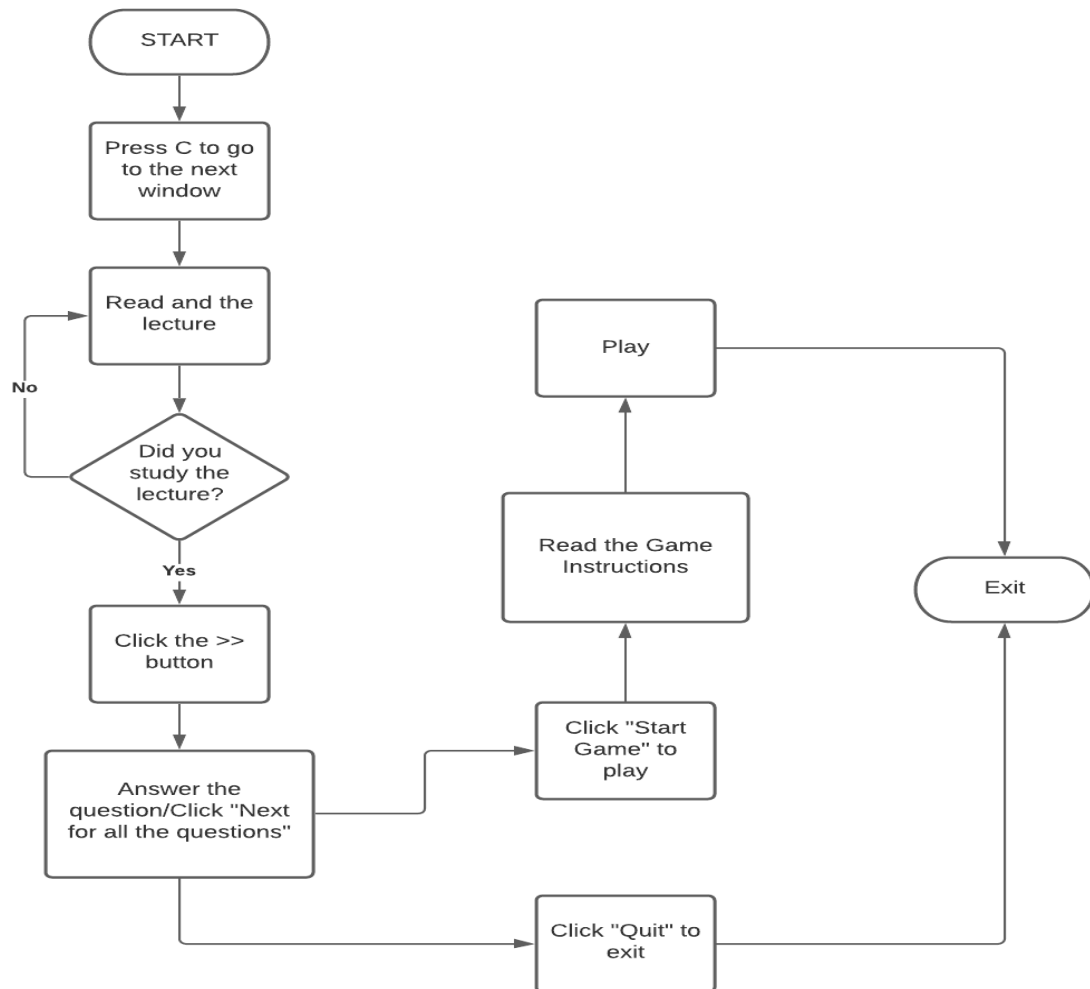
We build this project to share awareness about the cause and effect of climate change, and some ways to contribute in protecting the nature.

Awareness will be a huge factor in preserving our planet. If we know our responsibilities on our surroundings and what will be the effects of every action we made, we will be able to apply it considering what is best for the planet.

## METHODOLOGY

### FLOWCHART

The flowchart shows the process of executing the program and protecting the planet.



## FUNCTIONS

### 1. **next**

This function is used for importing new window for the next output.

### 2. **quiz**

This function is used for importing the quiz window.

### 3. **startgame**

This function is used for importing the gameplay window.

### 4. **question**

This function is used for displaying the questions that is significant to the quiz.

### 5. **radiobtns**

This function is used for displaying the radio buttons when choosing the correct answer in the questions.

### 6. **display\_options**

This function is used for displaying the choices for finding the correct answer.

### 7. **buttons**

This function is used for displaying the command buttons. There are three command buttons displayed are "Next", "Start Game", "Quit".

### 8. **display**

This function displays the result of the quiz including the number of correct answers and the number of wrong answers.



**9. up**

This function is used to let the player move upward

**10. down**

This function is used to let the player move downward

**11. right**

This function is used to let the player move right

**12. left**

This function is used to let the player move left

**13. move**

This is used to let the user or player to move.

**14. distance**

This is used to let the measure the distance between the enemy and the player and then check for collision.

**15. render**

This function is used for drawing the player in the window.

**16. fire**

This function is used for fire the weapon towards the enemy, using the space bar. The player will be able to fire a missile.

## PROGRAM OUTPUT

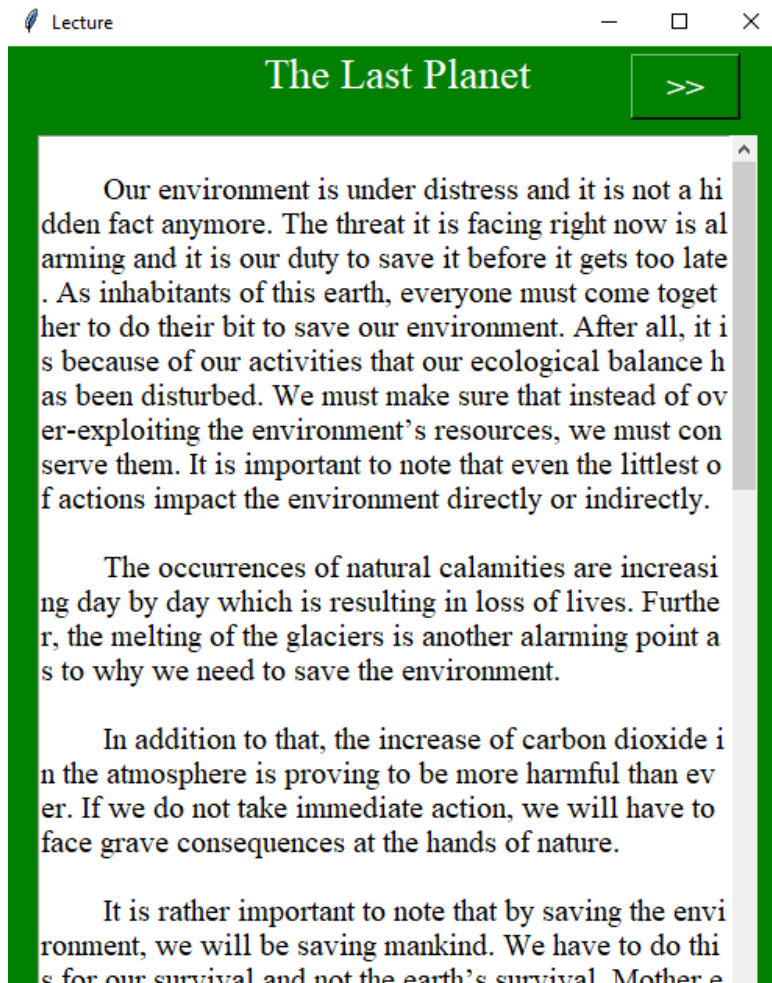
### 1. Get Started!

It is the main screen of the program.



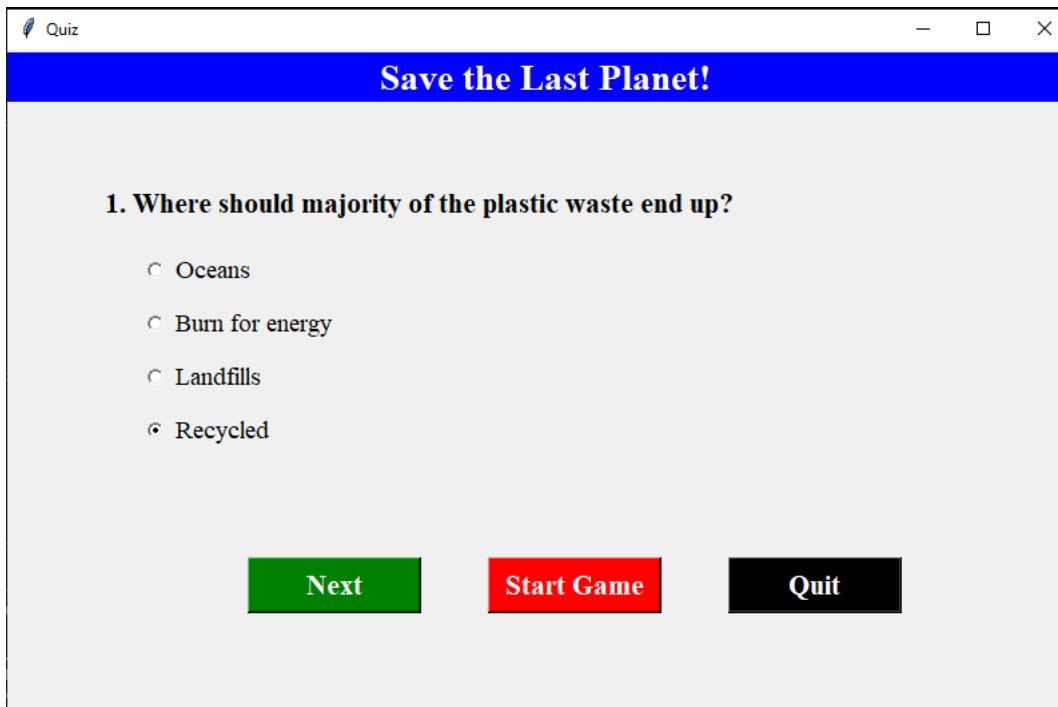
## 2. Lecture

It is the window to read the lecture that will be needed through the game.



### 3. Quiz

It is the window for answering the questions based on the lecture you read.



The image shows a web browser window titled "Quiz". The main heading of the page is "Save the Last Planet!". Below this, the first question is "1. Where should majority of the plastic waste end up?". There are four radio button options: "Oceans", "Burn for energy", "Landfills", and "Recycled". The "Recycled" option is selected. At the bottom of the page, there are three buttons: "Next" (green), "Start Game" (red), and "Quit" (black).

Quiz

## Save the Last Planet!

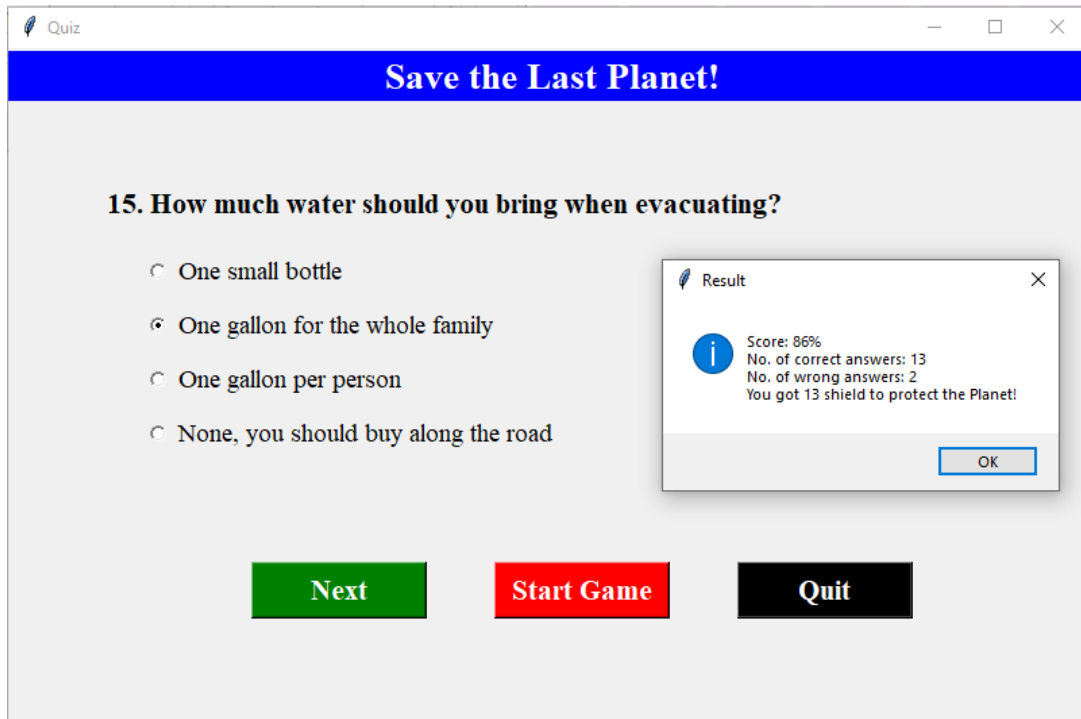
1. Where should majority of the plastic waste end up?

- ☐ Oceans
- ☐ Burn for energy
- ☐ Landfills
- ☒ Recycled

Next Start Game Quit

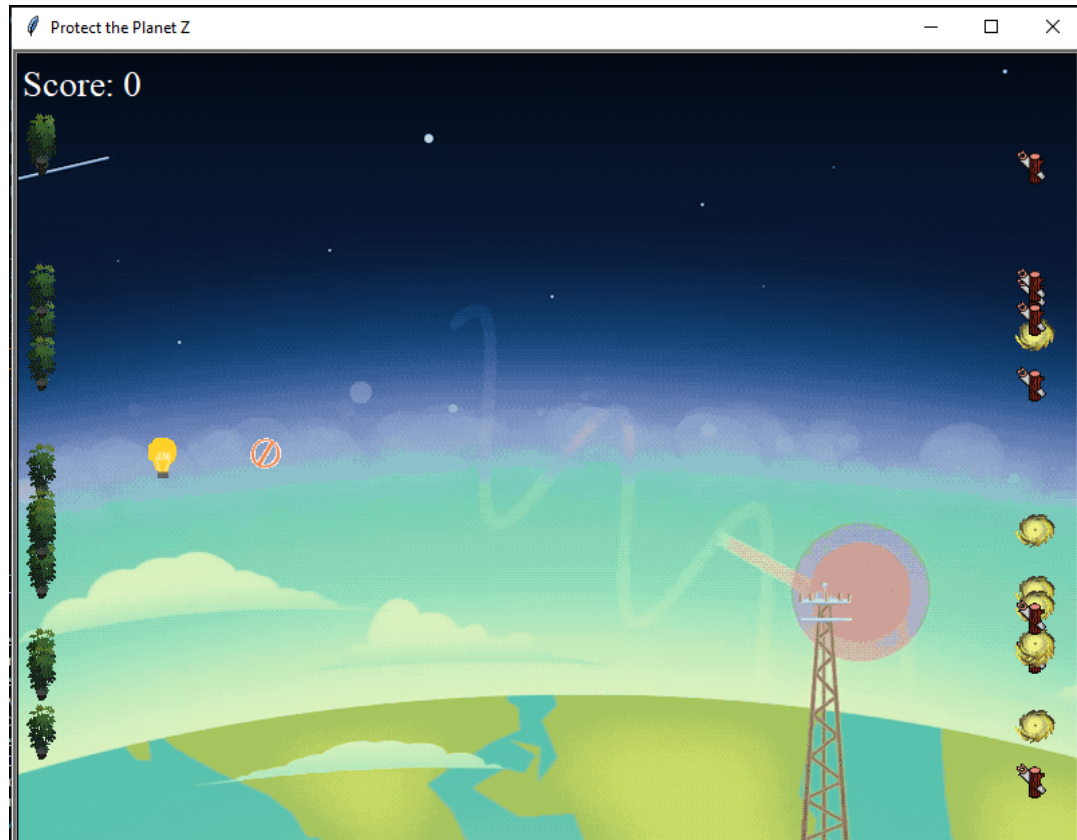
#### 4. Quiz (Result)

Displays the score you get after you answer the quiz.



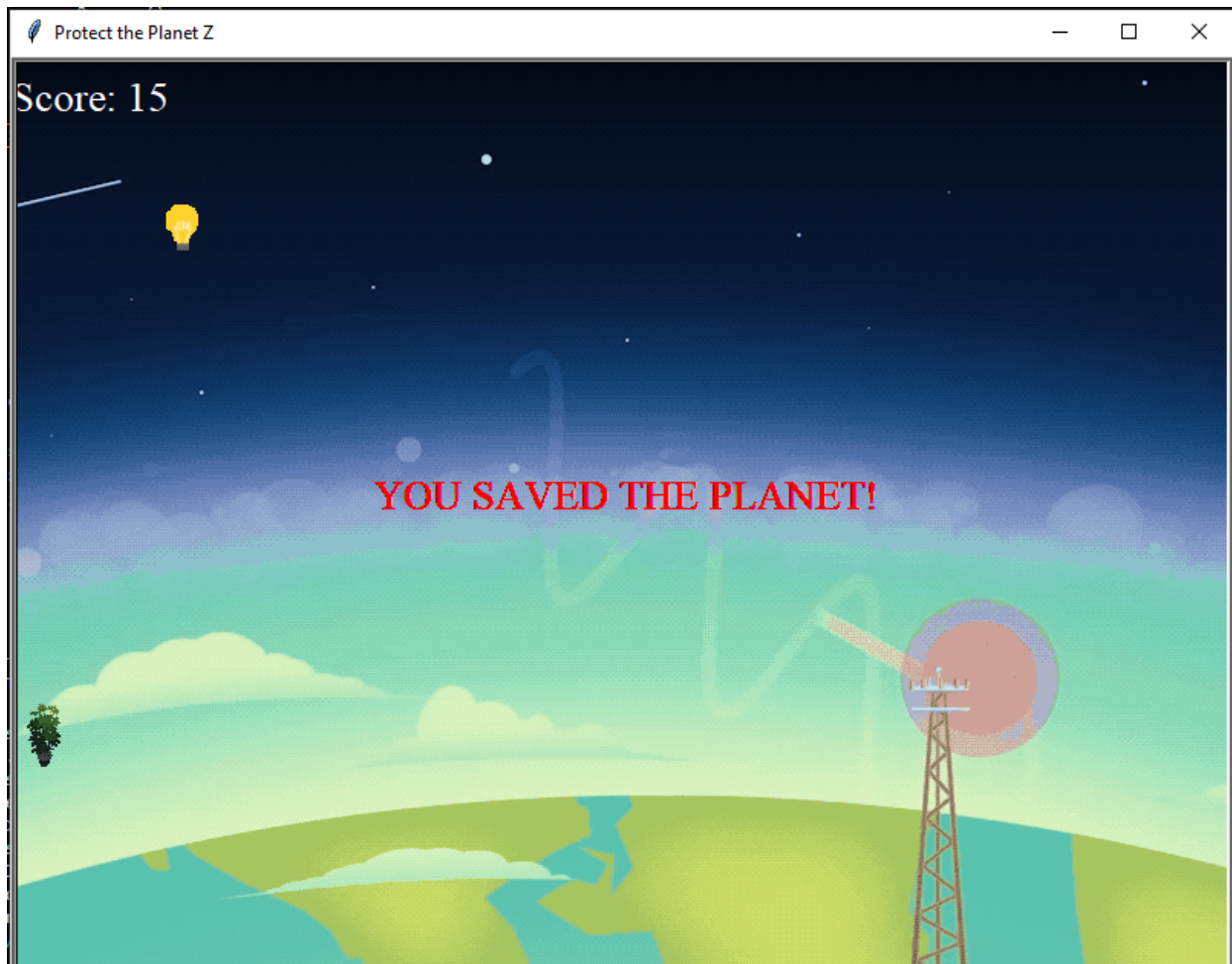
## 5. Gameplay

It is the actual game where you need to beat the enemies trying to destroy or enter the planet.



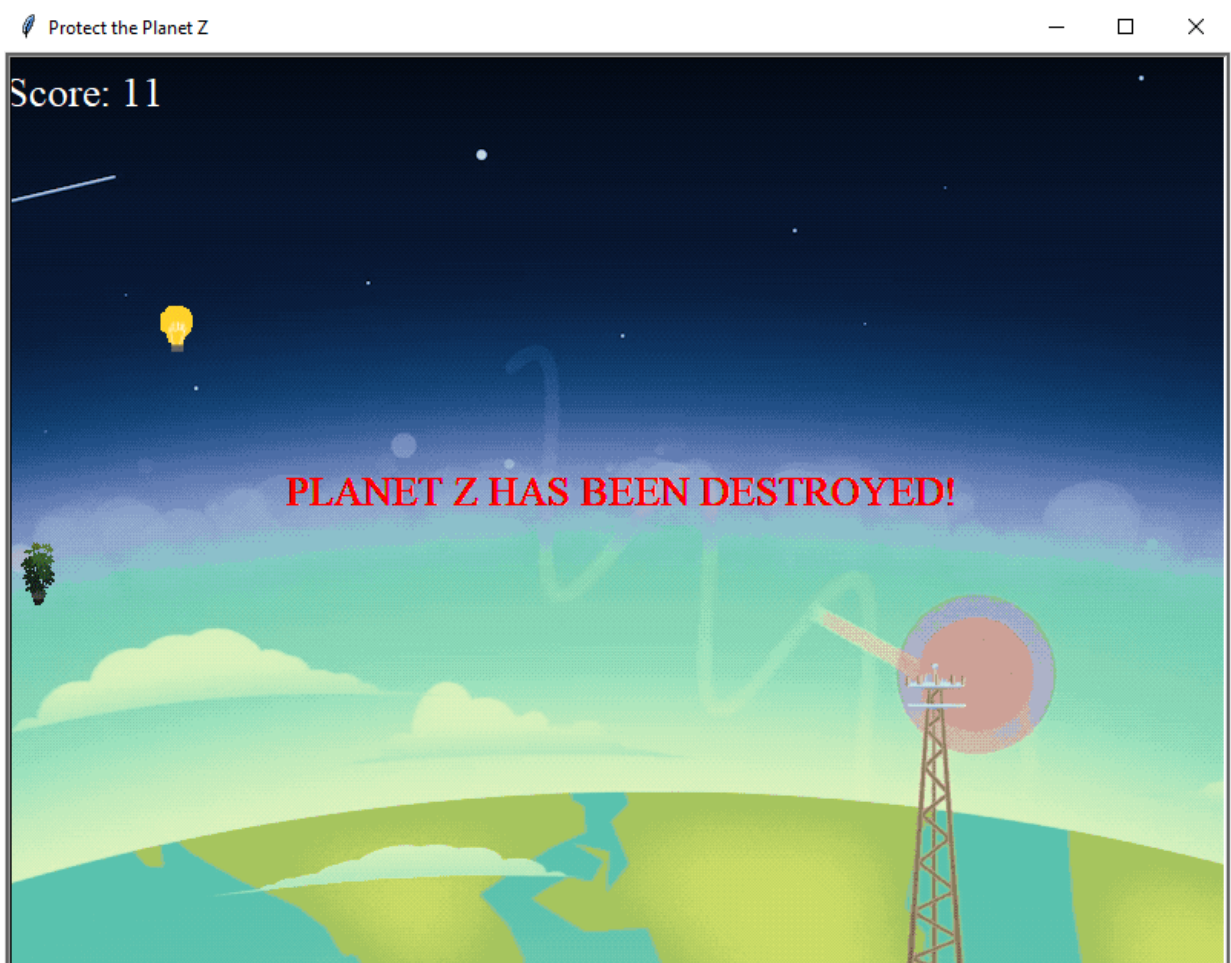
## 6. Win Result

Displays the message when you beat the enemies and protected the planet.



## 7. Game Over Result

Displays the message when you do not beat all the enemies and they entered the planet.





## APPENDICES

### SOURCE CODE

#### ProtectThePlanetZ.py (Main)

```
1     import turtle
2
3     import random
4
5     def next():
6
7         import Lecture
8
9
10    #Create the main screen
11
12    wn = turtle.Screen()
13
14    wn.title("Get Started!")
15
16    wn.setup(800, 600)
17
18    wn.bgpic("PPZbg.gif")
19
20
21
22    wn.listen()
23
24    wn.onkeypress(next, "c")
25
26    wn.onkeypress(next, "C")
27
28
29
30    wn.mainloop()
```

## Lecture.py

```
1  from tkinter import*
2
3  import tkinter as tk
4
5  import tkinter.scrolledtext as st
6
7  def quiz():
8
9      import Quiz
10
11     # Creating tkinter window
12
13     win = tk.Tk()
14
15     win.title("Lecture")
16
17     win.geometry("500x600")
18
19     win.configure(bg="green")
20
21     # Title Label
22
23     tk.Label(win, text = "The Last Planet", font = ("Times New Roman", 20),
24             background = 'green', foreground = "white").grid(column = 0, row = 10)
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
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
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

19     text_area = st.ScrolledText(win, width = 44, height = 25, font = ("Times New
    Roman", 15))

20

21     text_area.grid(column = 0, pady = 20, padx = 20)

22

23     # Inserting Text which is read only

    text_area.insert(tk.INSERT, """ Insert text here""")

24     # Making the text read only

25     text_area.configure(state = 'disabled')

26     win.mainloop()

```

### **Quiz.py**

```

1     from tkinter import*

2     from tkinter import messagebox as mb

3     import json #You need to create a json file

4

5     root = Tk()

6     root.geometry("800x500")

7     root.title("Quiz")

8

9     with open('quiz.json') as f:

```

---

```

10     obj = json.load(f)

11     q = (obj['ques'])

12     options = (obj['options'])

13     a = (obj['ans'])

14

15     def startgame():

16         import GameInstruction

17

18     class Quiz:

19         def __init__(self):

20             self.qn = 0

21             self.ques = self.question(self.qn)

22             self.opt_selected = IntVar()

23             self.opts = self.radiobtns()

24             self.display_options(self.qn)

25             self.buttons()

26             self.correct = 0

27

28         def question(self, qn):

29             t = Label(root, text = "Save the Last Planet!", width=50, bg="blue",
                        fg="white", font=("times", 20, "bold"))

```

---

```

30         t.place(x=0, y=2)

31         qn = Label(root, text=q[qn], width=60, font=("times", 16, "bold"),
anchor="w")

32         qn.place(x=70, y=100)

33         return qn

34

35     def radiobtns(self):

36         val = 0

37         b = []

38         yp = 150

39         while val < 4:

40             btn = Radiobutton(root, text="", variable=self.opt_selected,
value=val+1, font=("times", 14))

41             b.append(btn)

42             btn.place(x=100, y=yp)

43             val+=1

44             yp+=40

45         return b

46

47     def display_options(self, qn):

48         val = 0

```

```

49         self.opt_selected.set(0)

50         self.ques['text'] =q[qn]

51         for op in options[qn]:

52             self.opts[val]['text'] = op

53             val+=1

54

55     def buttons(self):

56         nbutton = Button(root, text="Next", command=self.nextbtn, width=10,
bg="green", fg="white", font=("times", 16, "bold"))

57         nbutton.place(x=180, y=380)

58         startbutton = Button(root, text="Start Game", command=startgame,
width=10, bg="red", fg="white", font=("times", 16, "bold"))

59         startbutton.place(x=360, y=380)

60         quitbutton = Button(root, text="Quit", command=root.destroy, width=10,
bg="black", fg="white", font=("times", 16, "bold"))

61         quitbutton.place(x=540, y=380)

62

63     def checkans(self, qn):

64         if self.opt_selected.get() == a[qn]:

65             return True

66

```

---

```

67     def nextbtn(self):
68         if self.checkans(self.qn):
69             self.correct+=1
70             self.qn +=1
71             if self.qn == len(q):
72                 self.display_result()
73             else:
74                 self.display_options(self.qn)
75
76     def display_result(self):
77         score = int(self.correct/len(q) * 100)
78         result = "Score: " + str(score) + "%"
79         wc = len(q) - self.correct
80         correct = "No. of correct answers: " + str(self.correct)
81         wrong = "No. of wrong answers: " + str(wc)
82         warn = "You got "+ str(self.correct)+ " shield to protect the Planet!"
83         mb.showinfo("Result", "\n".join([result, correct, wrong, warn]))
84
85     quiz = Quiz()
86     root.mainloop()

```

### **PvZlike.py**

```
1     import turtle

2     import random

3

4     #Create the screen

5

6     wn = turtle.Screen()

7     wn.title("Protect the Planet Z")

8     wn.setup(800, 600)

9     #wn.bgcolor("black")

10    wn.bgpic("mybg.gif")

11    wn.tracer(0)

12

13    wn.register_shape("dont.gif")

14    wn.register_shape("saw.gif")

15    wn.register_shape("bulb.gif")

16    wn.register_shape("storm.gif")

17    wn.register_shape("plant.gif")

18

19    pen = turtle.Turtle()

20    pen.speed(0)
```

---



```
21     pen.penup()

22

23     #Create the classes

24     #class Player(turtle.Turtle):

25     class Player():

26         def __init__(self):

27             #turtle.Turtle.__init__(self)

28             self.color = "red"

29             #self.penup()

30             self.x = -350

31             self.y = 0

32             #self.goto(-350, random.randint(-290, 290))

33             self.shape = "bulb.gif"

34             #self.speed(0)

35             self.dy = 0

35             self.dx = 0

37

38         def up(self):

39             self.dy = 1

40             self.dx = 0

41
```

---

```
42     def down(self):
43         self.dy = -1
44         self.dx = 0
45
46     def left(self):
47         self.dx = -1
48         self.dy = 0
49
50     def right(self):
51         self.dx = 1
52         self.dy = 0
53
54     def move(self):
55         #self.sety(self.ycor() + self.dy)
56         self.y = self.y + self.dy
57         self.x = self.x + self.dx
58
59         #Check for border collision
60         if self.y > 280:
61             self.y = 280
62             self.dy = 0
```

---

```
63         elif self.y < -280:
64             self.y = -280
65             self.dy = 0
66
67         if self.x < -390:
68             self.x = -390
69             self.dx = 0
70
71         if self.x < -290:
72             self.x = -290
73             self.dx = 0
74
75     def distance(self, other):
76         d = ((self.x - other.x) ** 2 + (self.y - other.y) ** 2) ** 0.5
77         return d
78
79     def render(self, pen):
80         pen.goto(self.x, self.y)
81         pen.shape(self.shape)
82         pen.shapesize(1, 1, 0)
83         pen.color(self.color)
```

---

```
84         pen.stamp()

85

86     #class Missile(turtle.Turtle):

87     class Missile():

88

89         def __init__(self):

90             self.color = "black"

91             #self.penup()

92             self.x = -450

93             self.y = 0

94             self.shape = "dont.gif"

95             self.size = random.randint(1, 9) / -10

96             self.dx = 0

97

98         def fire(self):

99             self.x = player.x

100             self.y = player.y

101             self.dx = 2

102

103         def move(self):

104             self.x = self.x + self.dx
```

---

```
105
106     def distance(self, other):
107         d = ((self.x - other.x) ** 2 + (self.y - other.y) ** 2) ** 0.5
108         return d
109
110     def render(self, pen):
111         pen.goto(self.x, self.y)
112         pen.shape(self.shape)
113         pen.shapesize(0.4, 0.4, 0)
114         pen.color(self.color)
115         pen.stamp()
116
117 class Enemy():
118     def __init__(self):
119         shapes = ["saw.gif", "storm.gif"]
120         colors = ["blue", "black", "purple", "pink", "gray"]
121         self.color = random.choice(colors)
122         #self.penup()
123         self.x = 380
124         self.y = random.randint (-250, 250)
125         self.shape = random.choice(shapes)
```

---

```
126         self.dx = -0.1

127

128     def move(self):

129         self.x = self.x + self.dx

130         #Border Check

131         if self.x < -400:

132             self.x = 400

133             self.y = random.randint(-250, 250)

134             self.dx *= 1.1

135

136     def distance(self, other):

137         d = ((self.x - other.x) ** 2 + (self.y - other.y) ** 2) ** 0.5

138         return d

139

140     def render(self, pen):

141         pen.goto(self.x, self.y)

142         pen.shape(self.shape)

143         pen.shapesize(1, 1, 0)

144         pen.color(self.color)

145         pen.stamp()
```

```
146 class Shield():
147     def __init__(self):
148         self.color = "green"
149         self.x = -380
150         self.y = random.randint(-250, 250)
151         self.shape = "plant.gif"
152
153     def distance(self, other):
154         d = ((self.x - other.x) ** 2 + (self.y - other.y) ** 2) ** 0.5
155         return d
156
157     def render(self, pen):
158         pen.goto(self.x, self.y)
159         pen.shape(self.shape)
160         pen.shapesize(1, 1, 0)
161         pen.color(self.color)
162         pen.stamp()
163
164 score = 0
```

```
165     #Display the score

166     pen1 = turtle.Turtle()

167     pen1.hideturtle()

168     pen1.speed(0)

169     pen1.shape("circle")

170     pen1.color("white")

171     pen1.penup()

172     pen1.goto(-350, 260)

173     font = ("Times New Roman", 20, "normal")

174     pen1.write("Score: {}".format(score), align="center", font=font)

175

176     #Create game objects

177

178     player = Player()

179     missile = Missile()

180     shield = Shield()

181

182     #Create a list

183     enemies = []

184     noofe = 15

185     for _ in range(noofe):
```

---



```
186     enemies.append(Enemy())

187

188     shields = []

189     for _ in range(15):

190         shields.append(Shield())

191

192     #Keyboard Bindings

193

194     wn.listen()

195     wn.onkeypress(player.up, "Up")

196     wn.onkeypress(player.down, "Down")

197     wn.onkeypress(player.left, "Left")

198     wn.onkeypress(player.right, "Right")

199     wn.onkeypress(missile.fire, "space")

200

201     while True:

202

203         wn.update()

204         pen.clear()

205         player.move()
```

---

```
206     missile.move()

207     player.render(pen)

208     missile.render(pen)

209

210     for enemy in enemies:

211         enemy.move()

212

213         #Check for collsion

214         if enemy.distance(missile) < 13:

215             enemy.x = 1000000

216             enemy.y = random.randint(-250, 250)

217             missile.dx = 0

218             missile.x = 0

219             missile.y = 1000

220             score +=1

221             pen1.clear()

222             pen1.write("Score: {}".format(score), align="center", font=font)

223

224             if enemy.x < -350:

225                 pen2 = turtle.Turtle()

226                 pen2.hideturtle()
```

---

```
227         pen2.speed(0)

228         pen2.shape("circle")

229         pen2.color("red")

230         pen2.penup()

231         pen2.goto(0, 0)

232         font = ("Times New Roman", 20, "normal")

233         pen2.write("PLANET Z HAS BEEN DESTROYED!", align="center",
234 font=font)

235     if score == 15:

236         pen3 = turtle.Turtle()

237         pen3.hideturtle()

238         pen3.speed(0)

239         pen3.shape("circle")

240         pen3.color("red")

241         pen3.penup()

242         pen3.goto(0, 0)

243         font = ("Times New Roman", 20, "normal")

244         pen3.write("YOU SAVED THE PLANET!", align="center", font=font)

245

246     enemy.render(pen)
```

---

```
247     for shield in shields:
248         shield.render(pen)
249         if enemy.distance(shield) < 13:
250             enemy.x = 1000000
251             enemy.y = random.randint(-250, 250)
252             shield.x = 1000000
253             shield.y = 1000000
254             score +=1
255             pen1.clear()
256             pen1.write("Score: {}".format(score), align="center", font=font)
```

## MEMBERS DUTIES AND RESPONSIBILITIES

Names of Members	Duty/ Responsibility	Description
De Chavez, Alfrederick M.	Program, Ideas, Documentation.	Write the code of the program including the algorithm, etc.
Serrano, Lemuel	Program, Ideas	Provide the content needed such as lecture, quiz. Help in coding.
Ronquillo, Julius	Documentation, Design	Visualize the idea of design. Help in coding.

## MEMBERS' PROFILE

### DE CHAVEZ, ALFREDERICK M.

Address: Anus, San Jose, Batangas

Email: [Alfrederick.dechavez@g.batstate-u.edu.ph](mailto:Alfrederick.dechavez@g.batstate-u.edu.ph)

Contact number: 09756133054



### Personal Information

Sex	: Male
Date of Birth	: January 19, 2001
Age	: 19
Place of Birth	: Anus, San Jose, Batangas

### Educational Background

Tertiary	:	Batangas State University
Secondary	:	Dr. Bonifacio A. Masilungan NHS
Primary	:	Tuqtug Elementary School

## MEMBERS' PROFILE

### RONQUILLO, JULIUS P.

Address: Delas Alas, Batangas City, Batangas

Email: Julius.ronquillo@g.batstate-u.edu.ph

Contact number: 09217899170

PHOTO

### Personal Information

Sex	: Male
Date of Birth	: October 3, 2000
Age	: 20
Place of Birth	: Oriental Mindoro

### Educational Background

Tertiary	:	Batangas State University
Secondary	:	BNHS
Primary	:	JAPMES

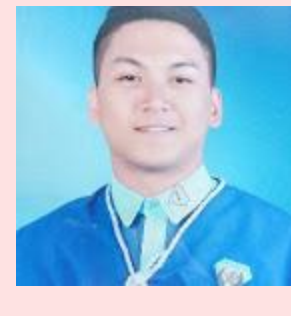
## MEMBERS' PROFILE

### SERRANO, LEMUEL L.

Address: Taal, Batangas

Email: lemuel.serrano@g.batstate-u.edu.ph

Contact number: 09631640743



#### Personal Information

Sex	: Male
Date of Birth	: September 11, 2001
Age	: 19
Place of Birth	: Batangas Provincial Hospital

#### Educational Background

Tertiary	:	Batangas State University
Secondary	:	Fame Academy of Science and Technology
Primary	:	Lemery Pilot Elementary School



## SOFTWARE PROJECT RUBRICS

CRITERIA	VERY GOOD	GOOD	FAIR	POOR	POINTS
	4	3	2	1	
<b>POINT DISTRIBUTION</b>	<b>50</b>	<b>25</b>	<b>10</b>	<b>0</b>	
<b>Program Functionality</b>	Program always works correctly and meets the specifications	Minor details of the program specification are violated, program functions incorrectly on some inputs.	Attempts to define purpose which adequately does not provide ideas and evidence that support the project concept	Significant details of specification are violated, or the program often exhibits incorrect behavior.	50
<b>POINT DISTRIBUTION</b>	<b>25</b>	<b>15</b>	<b>10</b>	<b>0</b>	
<b>Program Structure</b>	Code appropriately uses loops and methods for repeated code, and there is minimal hardcoding.	Code uses a poorly chosen approach in at least one place, for example, hard coding something that could be implemented through a for loop	Many instances where code could have used easier/faster/better approach.	Never used a better approach to minimize program structure	25
<b>POINT DISTRIBUTION</b>	<b>15</b>	<b>10</b>	<b>5</b>	<b>0</b>	
<b>Code Documentation</b>	Code is well commented.	One or two places could benefit from comments, or the code is overly commented	Major lack of comments makes it difficult to understand code.	No comments.	15
<b>POINT DISTRIBUTION</b>	<b>10</b>	<b>10</b>	<b>5</b>	<b>0</b>	

<b>Readability</b>	Code is clean, understandable, well-organized	Minor issues such as inconsistent indentation, variable naming, general organization	At least one major issue that makes it difficult to read	Several major issues that make it difficult to read	10
<b>TOTAL:</b>					100