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  Overview  Objective  Methodology  Flowchart  Functions  Program Output  Appendices  Source Code  Members Duties And Responsibilities  Member’s Profile | 4  4  6  7  8  10  17  37  38 |

**LIST OF TABLES AND FIGURES**

**List of Figures**

**Figure No.**

**Name**

**Page**

1

13

2

14

3

15

4

16

5

6

7

10

11

12

Game Over Result

Win Result

Gameplay

Quiz (Result)

Quiz

Lecture

Get Started!

**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

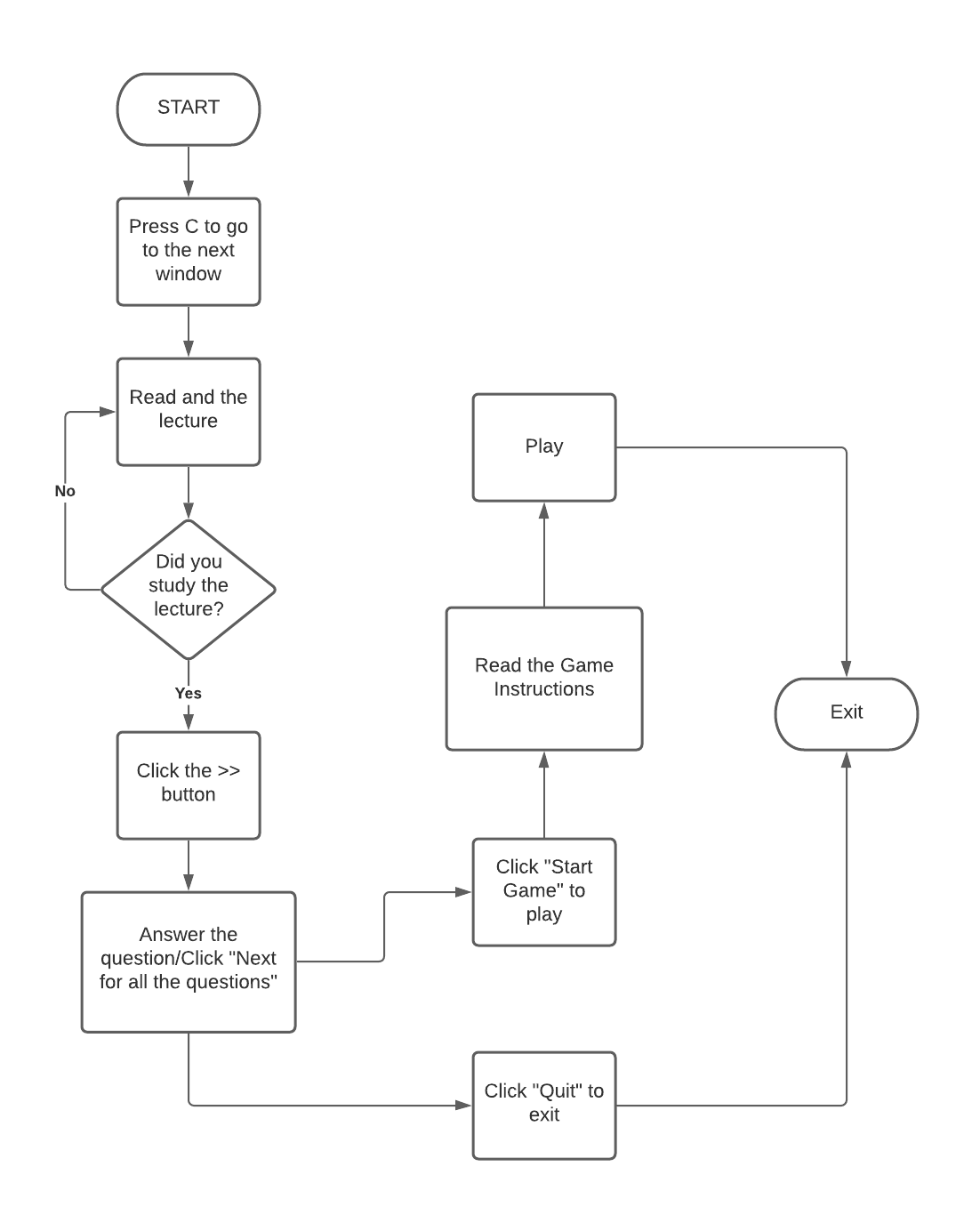
The world as we all know is experiencing different calamities. This year gives us a warning that we cannot ignore. It reminds us that we need to protect the world now. We never know what’s going to happen next. What if the disasters that happened was just a small part of a greater disaster that may happen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

1. **quiz**

This function is used for importing the quiz window.

1. **startgame**

This function is used for importing the gameplay window.

1. **question**

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

1. **radiobtns**

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

1. **display\_options**

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

1. **buttons**

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

1. **display**

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

1. **up**

This function is used to let the player move upward

1. **down**

This function is used to let the player move downward

1. **right**

This function is used to let the player move right

1. **left**

This function is used to let the player move left

1. **move**

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

1. **distance**

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

1. **render**

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

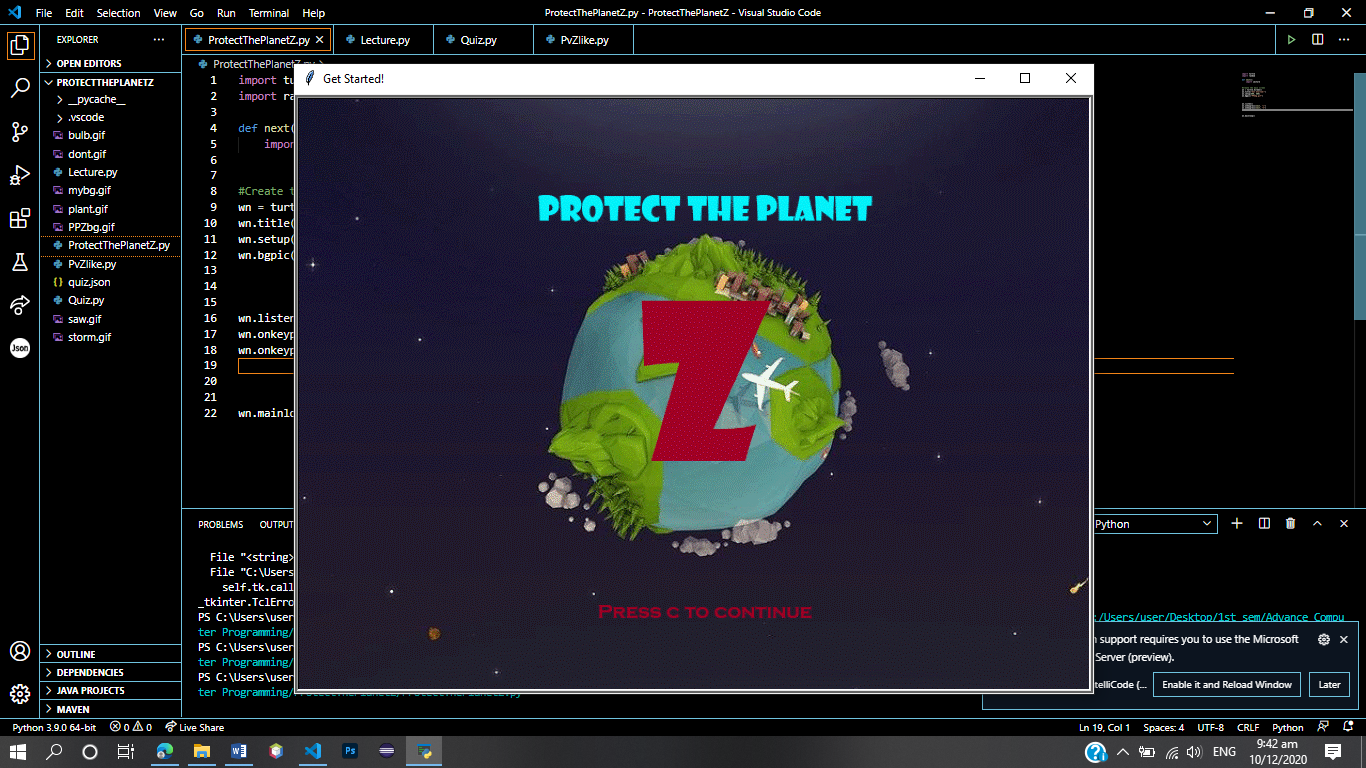
1. **fire**

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

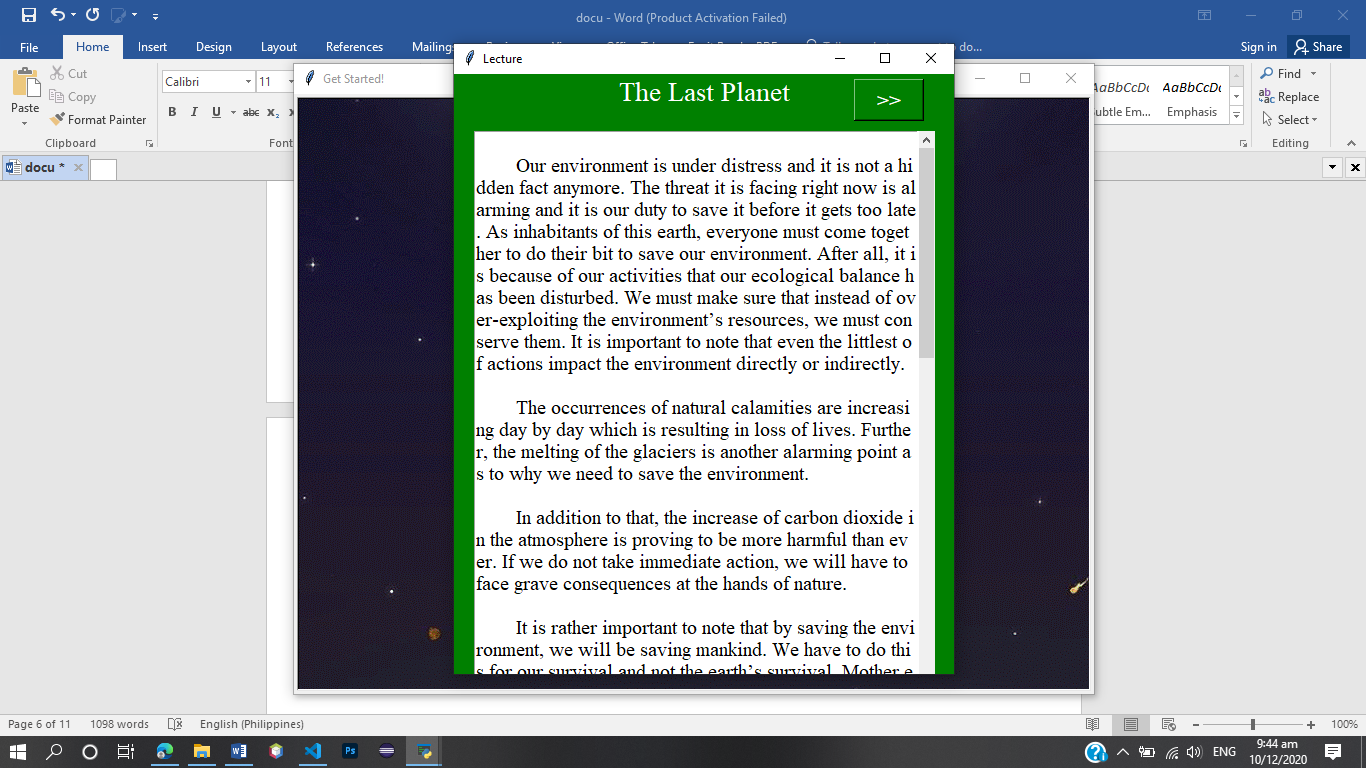
1. **Get Started!**

It is the main screen of the program.



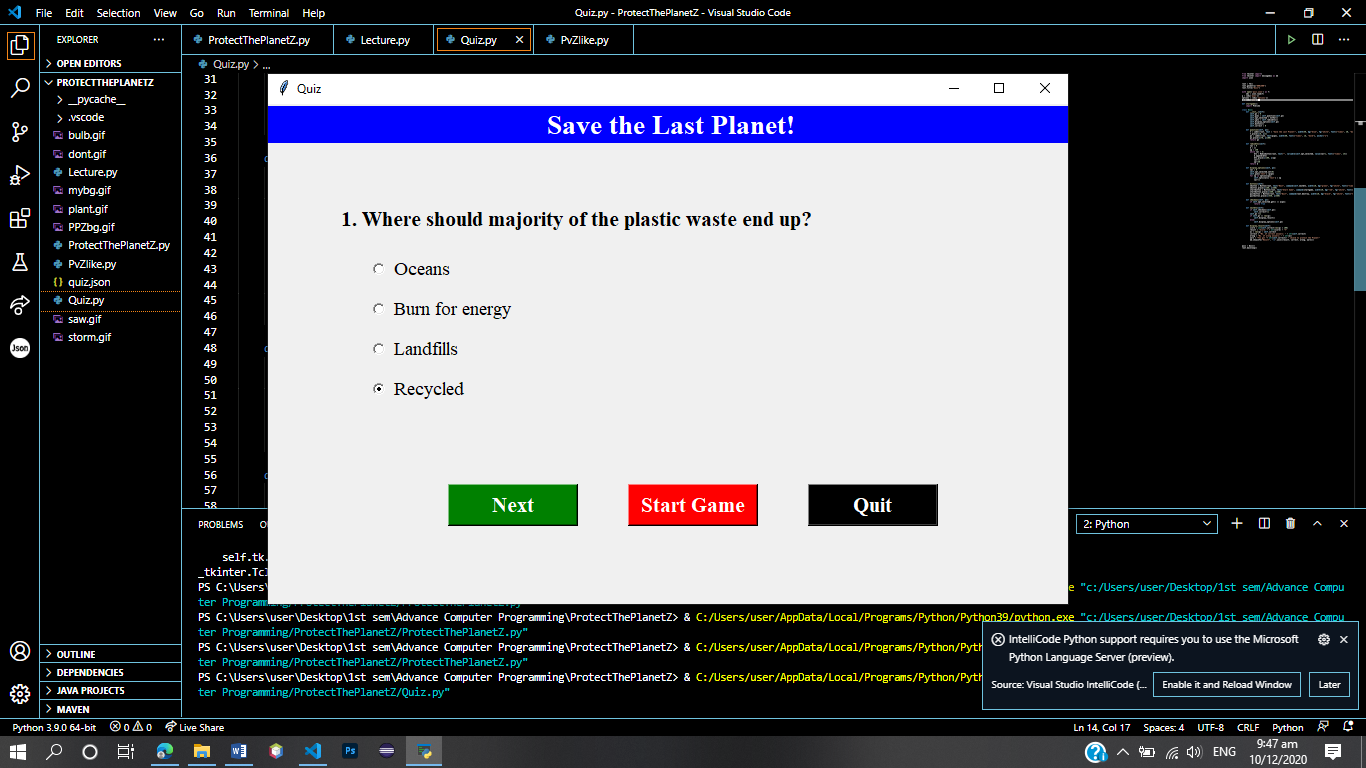
1. **Lecture**

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



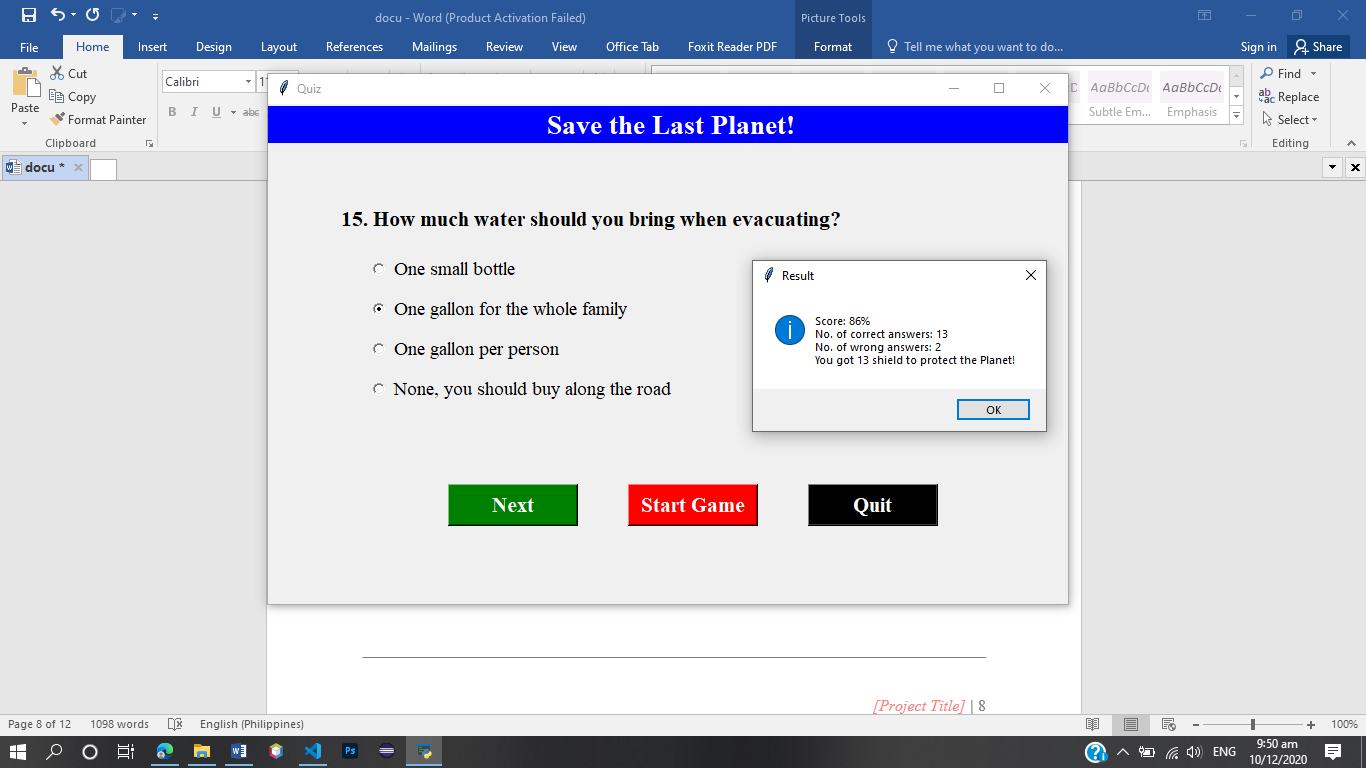
1. **Quiz**

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



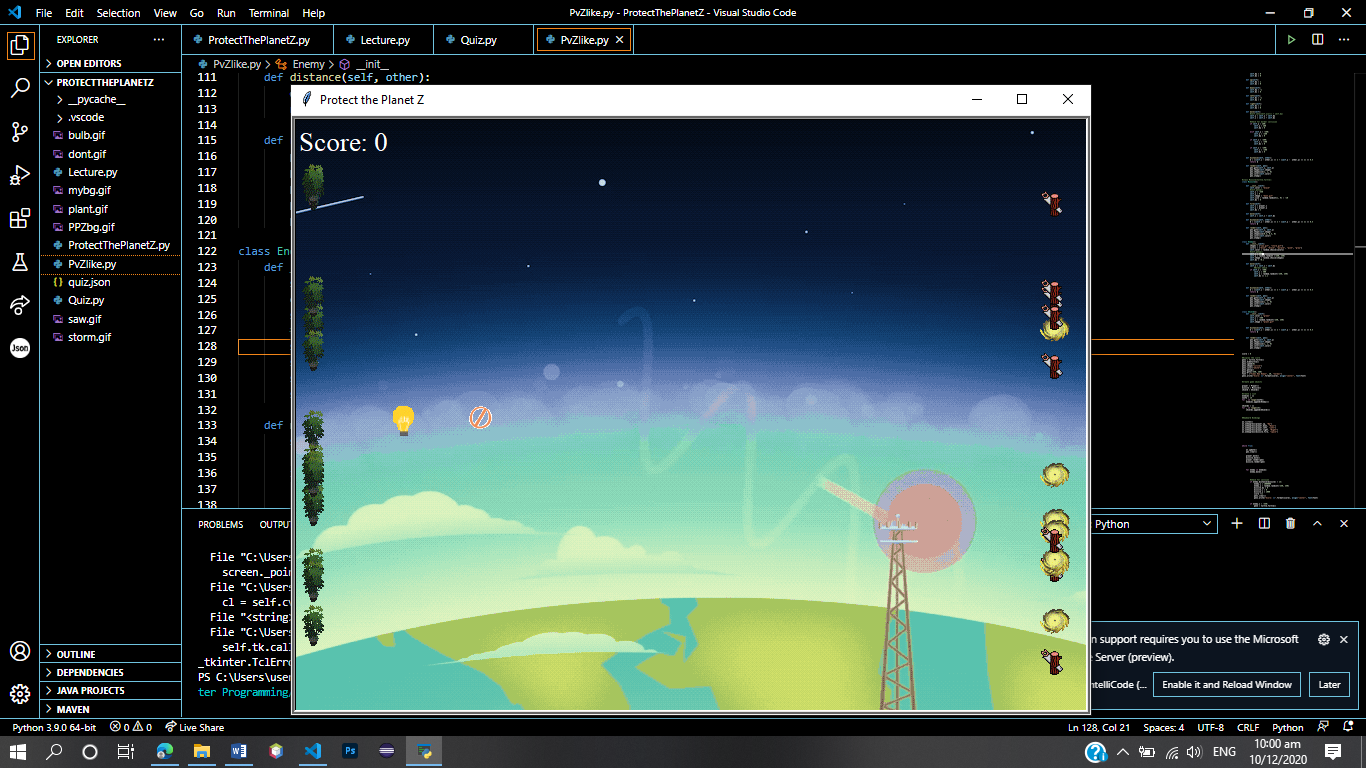
1. **Quiz (Result)**

Displays the score you get after you answer the quiz.



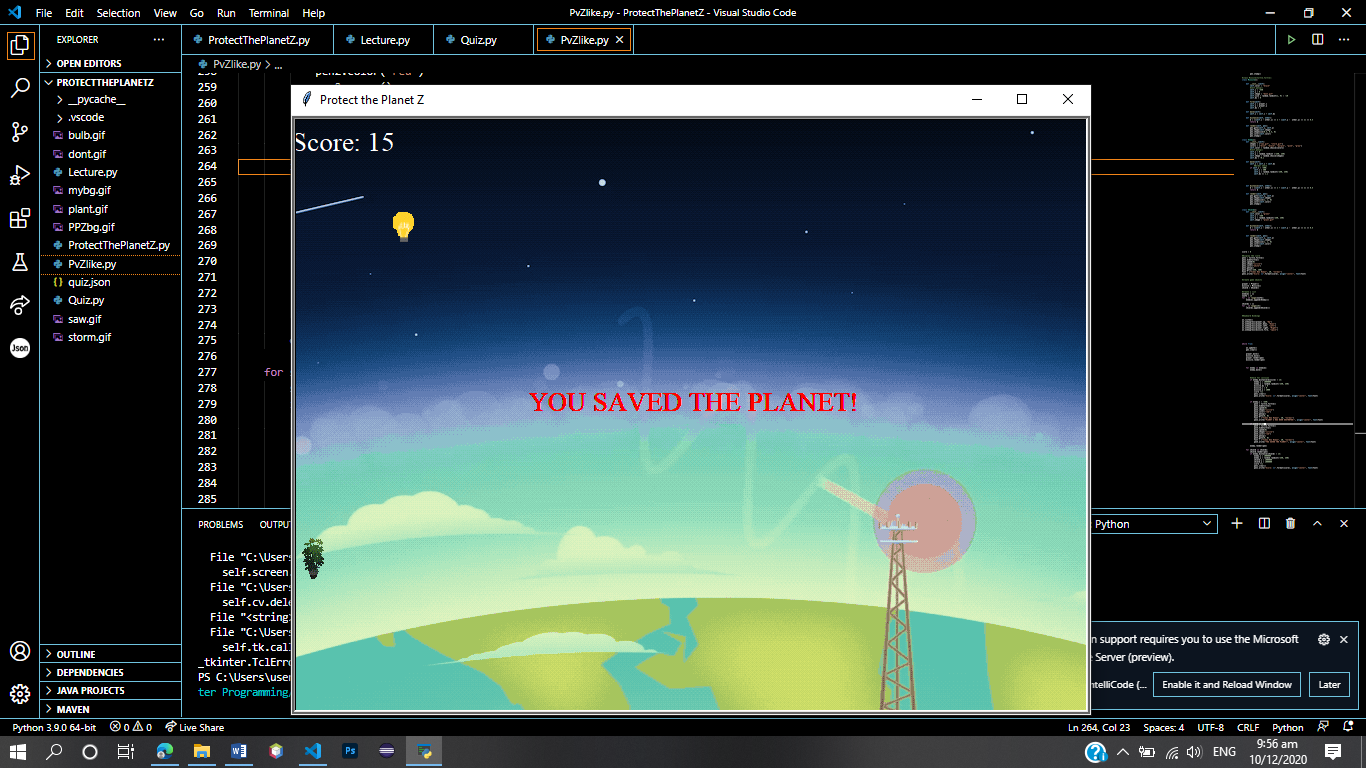
1. **Gameplay**

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



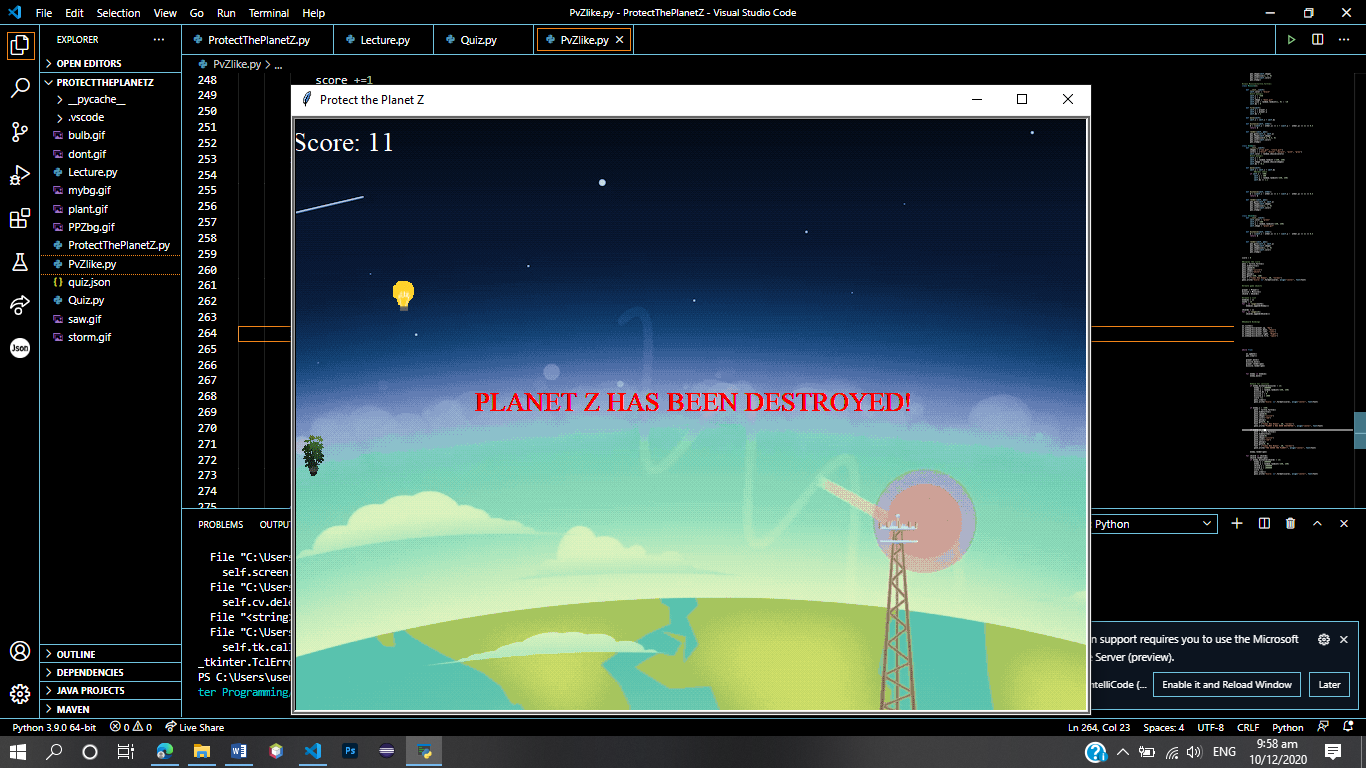
1. **Win Result**

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



1. **Game Over Result**

Displays the message when you do no beat all the enemies and they entered the planet.



**APPENDICES**

# SOURCE CODE

# ProtectThePlanetZ.py (Main)

1 import turtle

2 import random

3 def next():

4 import Lecture

5

6 #Create the main screen

7 wn = turtle.Screen()

8 wn.title("Get Started!")

9 wn.setup(800, 600)

10 wn.bgpic("PPZbg.gif")

11

12 wn.listen()

13 wn.onkeypress(next, "c")

14 wn.onkeypress(next, "C")

15

16 wn.mainloop()

**Lecture.py**

1 from tkinter import\*

2 import tkinter as tk

3 import tkinter.scrolledtext as st

4

5 def quiz():

6 import Quiz

7 # Creating tkinter window

8 win = tk.Tk()

9 win.title("Lecture")

10 win.geometry("500x600")

11 win.configure(bg="green")

12 # Title Label

13 tk.Label(win, text = "The Last Planet", font = ("Times New Roman", 20), background = 'green', foreground = "white").grid(column = 0, row = 10)

14

15 nbutton = Button(win, text=">>", command=quiz, width=5,height=1, bg="green", fg="white", font=("times", 16, "bold"))

16 nbutton.place(x=400, y=5)

17

18 # Creating scrolled text area

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", font=font)

234

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**

|  |
| --- |
| PHOTO |

# DE CHAVEZ, ALFREDERICK M.

Address: Anus, San Jose, Batangas

Email: 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 : Secondary : Primary :

Dr. Bonifacio A. Masilungan NHS

Tugtug Elementary School

Batangas State University

**MEMBERS’ PROFILE**

|  |
| --- |
| PHOTO |

# RONQUILLO, JULIUS P.

Address: Delas Alas, Batangas City, Batangas

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

Contact number: 09217899170

|  |  |
| --- | --- |
| **Personal Information** |  |
| Sex | : Male |
| Date of Birth | : October 3, 2000 |
| Age | : 20 |
| Place of Birth | : Oriental Mindoro |
| **Educational Background** |  |

Tertiary : Secondary : Primary :

JAPMES

Batangas State University

BNHS

**MEMBERS’ PROFILE**

|  |
| --- |
| PHOTO |

# 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** | Batangas State University |

Tertiary : Secondary : Primary :

Fame Academy of Science and Technology

Lemery Pilot Elementary School

**SOFTWARE PROJECT RUBRICS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CRITERIA | VERY GOOD | GOOD | FAIR | POOR |  |
| 4 | 3 | 2 | 1 | POINTS |
| **POINT DISTRIBUTION** | **50** | **25** | **10** | **0** |  |
| **Program**  **Functionality** | 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 |
| **POINT DISTRIBUTION** | **25** | **15** | **10** | **0** |  |
| **Program Structure** | 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 |
| **POINT DISTRIBUTION** | **15** | **10** | **5** | **0** |  |
| **Code Documentation** | 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 |
| **POINT DISTRIBUTION** | **10** | **10** | **5** | **0** |  |
| **Readability** | 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 |
|  |  |  |  | **TOTAL:** | 100 |