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

LIST OF TABLES AND FIGURES

List of Figures

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.

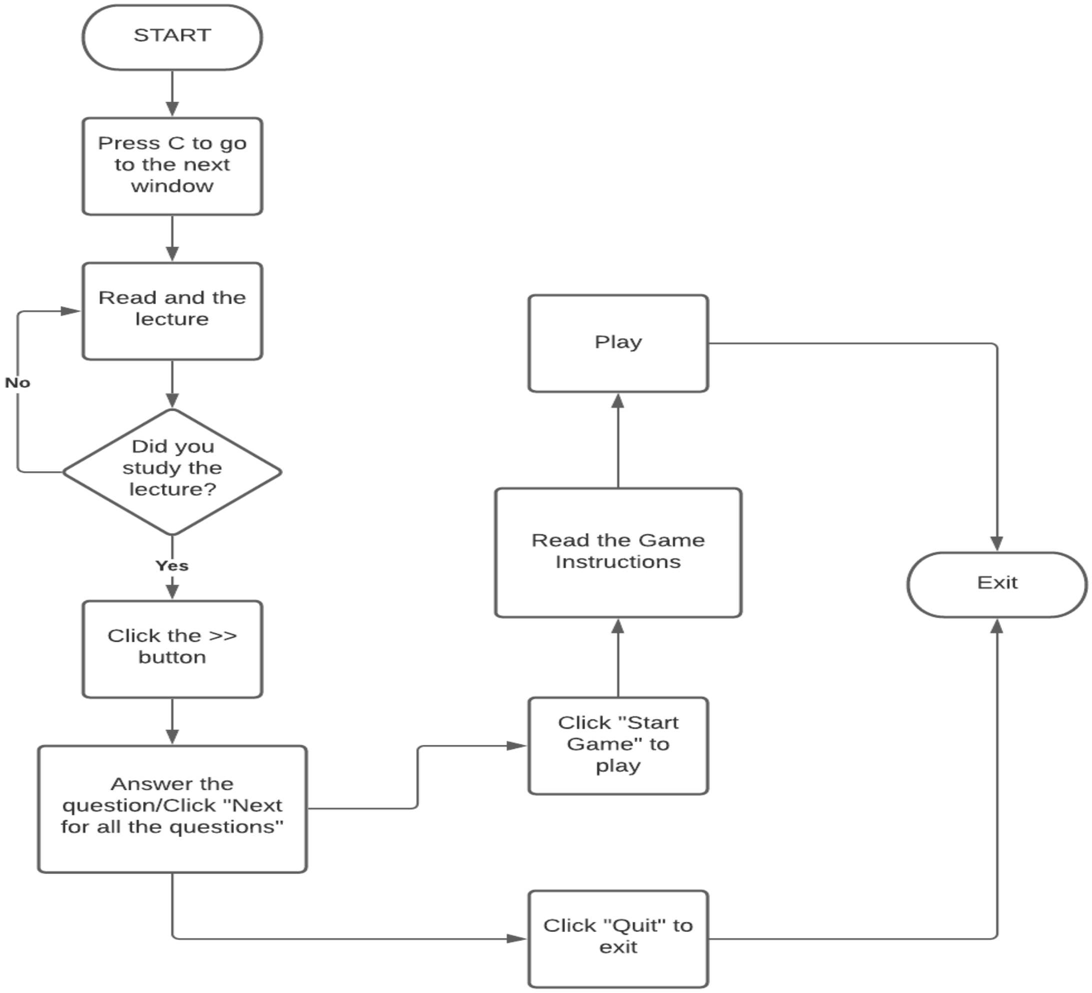
We build this project to share awareness about the cause and effect of climate change, and some ways to contribute in protecting the nature.

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.


START
Press C to go to the next window
Read and the lecture
No
Did you study the lecture?
Yes
Click the >> button
Answer the question/Click "Next for all the questions"
Click "Start Game" to play
Read the Game Instructions
Play
Click "Quit" to exit
Exit

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.

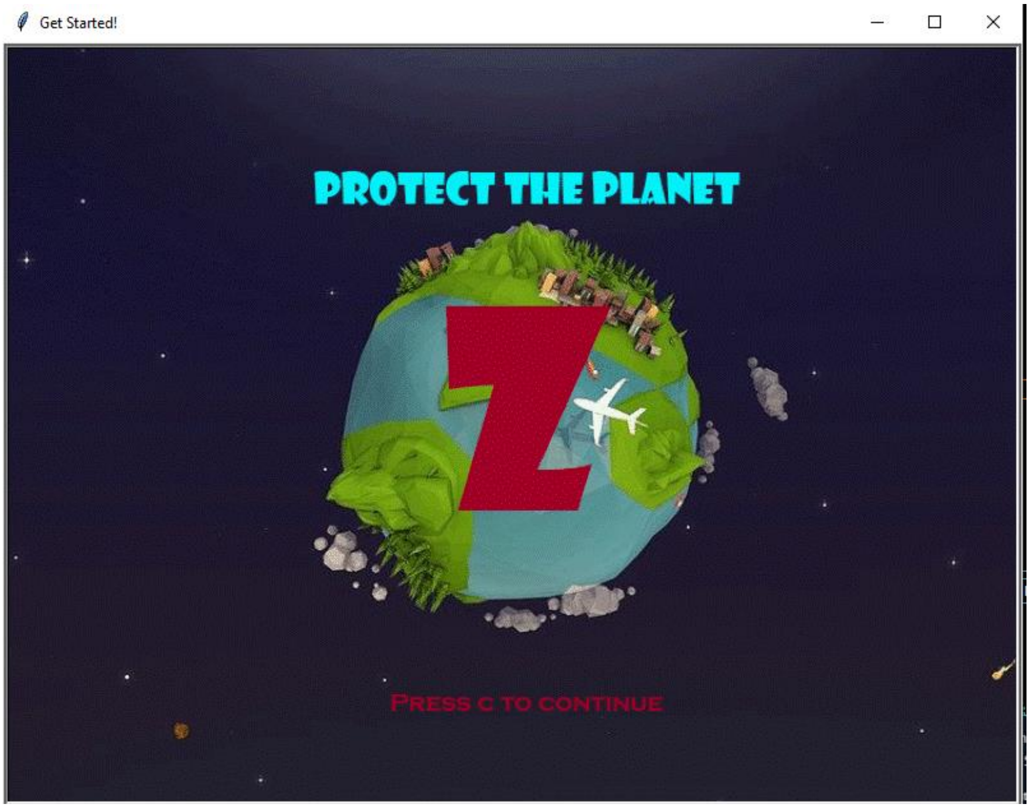
16. **fire**

    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

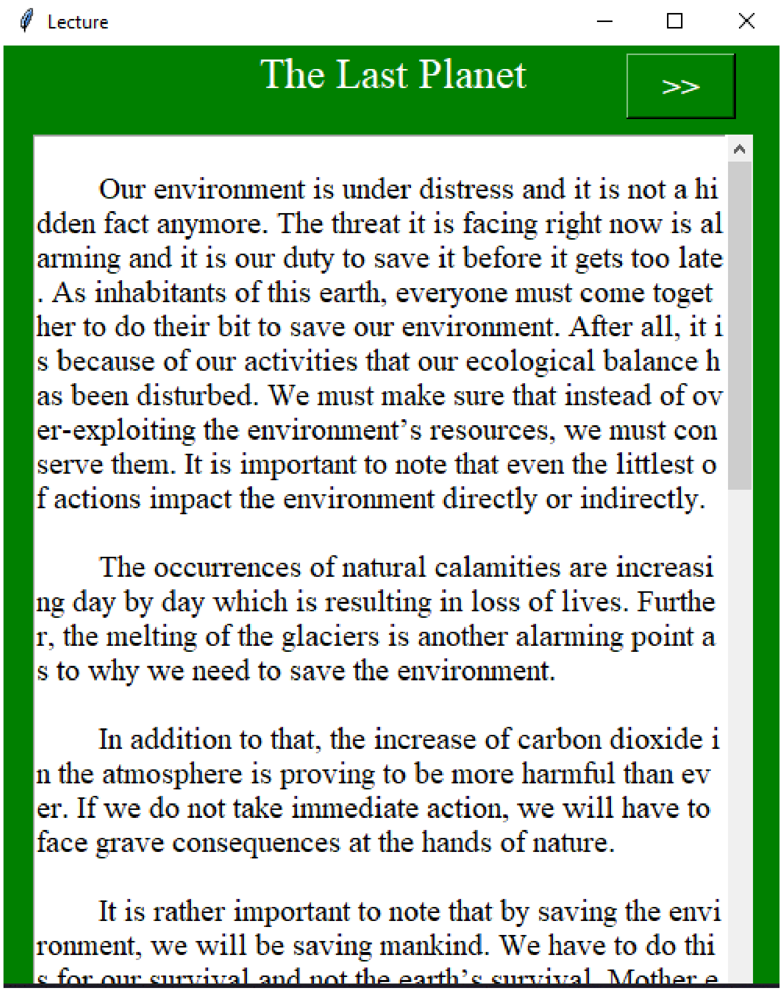
1. Get Started!

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.



The Last Planet

Our environment is under distress and it is not a hidden fact anymore. The threat it is facing right now is alarming and it is our duty to save it before it gets too late. As inhabitants of this earth, everyone must come together to do their bit to save our environment. After all, it is because of our activities that our ecological balance has been disturbed. We must make sure that instead of over-exploiting the environment's resources, we must conserve them. It is important to note that even the littlest of actions impact the environment directly or indirectly.

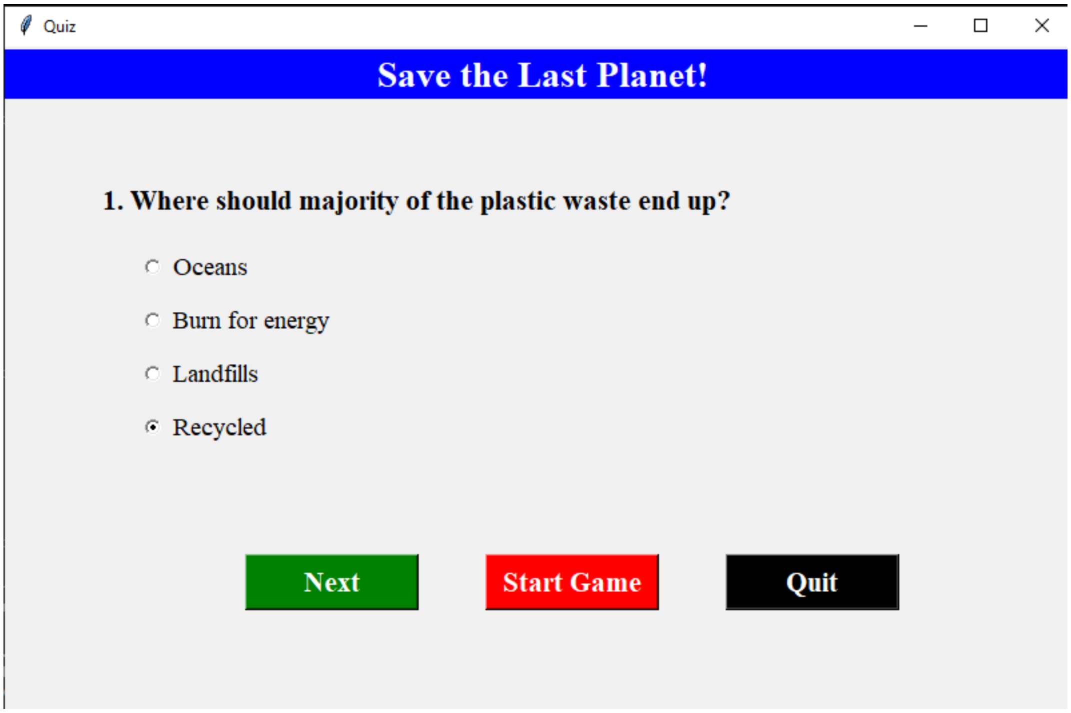
The occurrences of natural calamities are increasing day by day which is resulting in loss of lives. Further, the melting of the glaciers is another alarming point as to why we need to save the environment.

In addition to that, the increase of carbon dioxide in the atmosphere is proving to be more harmful than ever. If we do not take immediate action, we will have to face grave consequences at the hands of nature.

It is rather important to note that by saving the environment, we will be saving mankind. We have to do this for our survival and not the earth's survival. Mother e

3. Quiz

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



Save the Last Planet!

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

○ Oceans

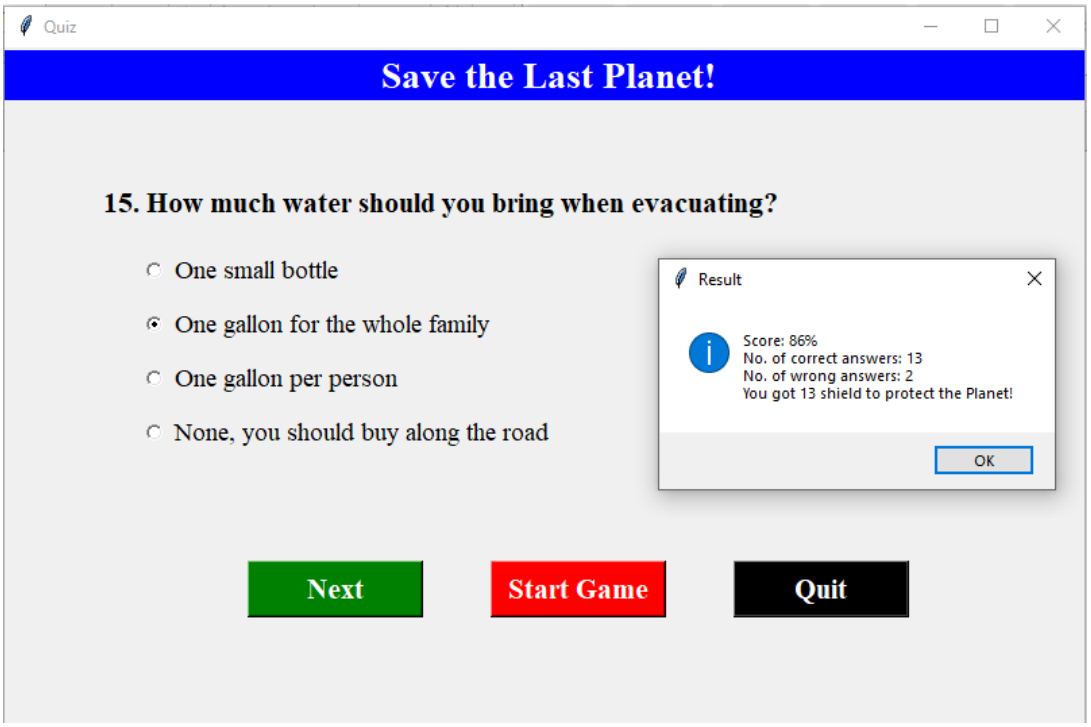
○ Burn for energy

○ Landfills

● Recycled

4. Quiz (Result)

Displays the score you get after you answer the quiz.



Save the Last Planet!

15. How much water should you bring when evacuating?

○ One small bottle

● One gallon for the whole family

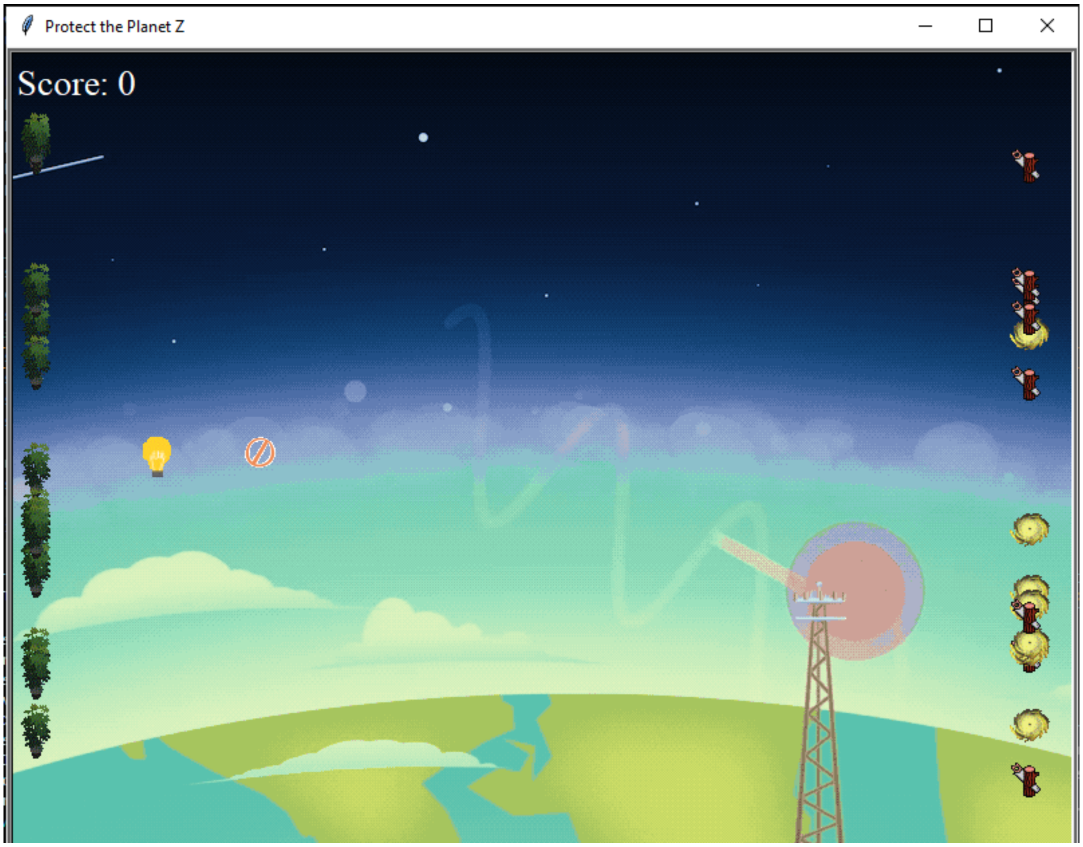
○ One gallon per person

○ None, you should buy along the road

Score: 86%
No. of correct answers: 13
No. of wrong answers: 2
You got 13 shield to protect the Planet!

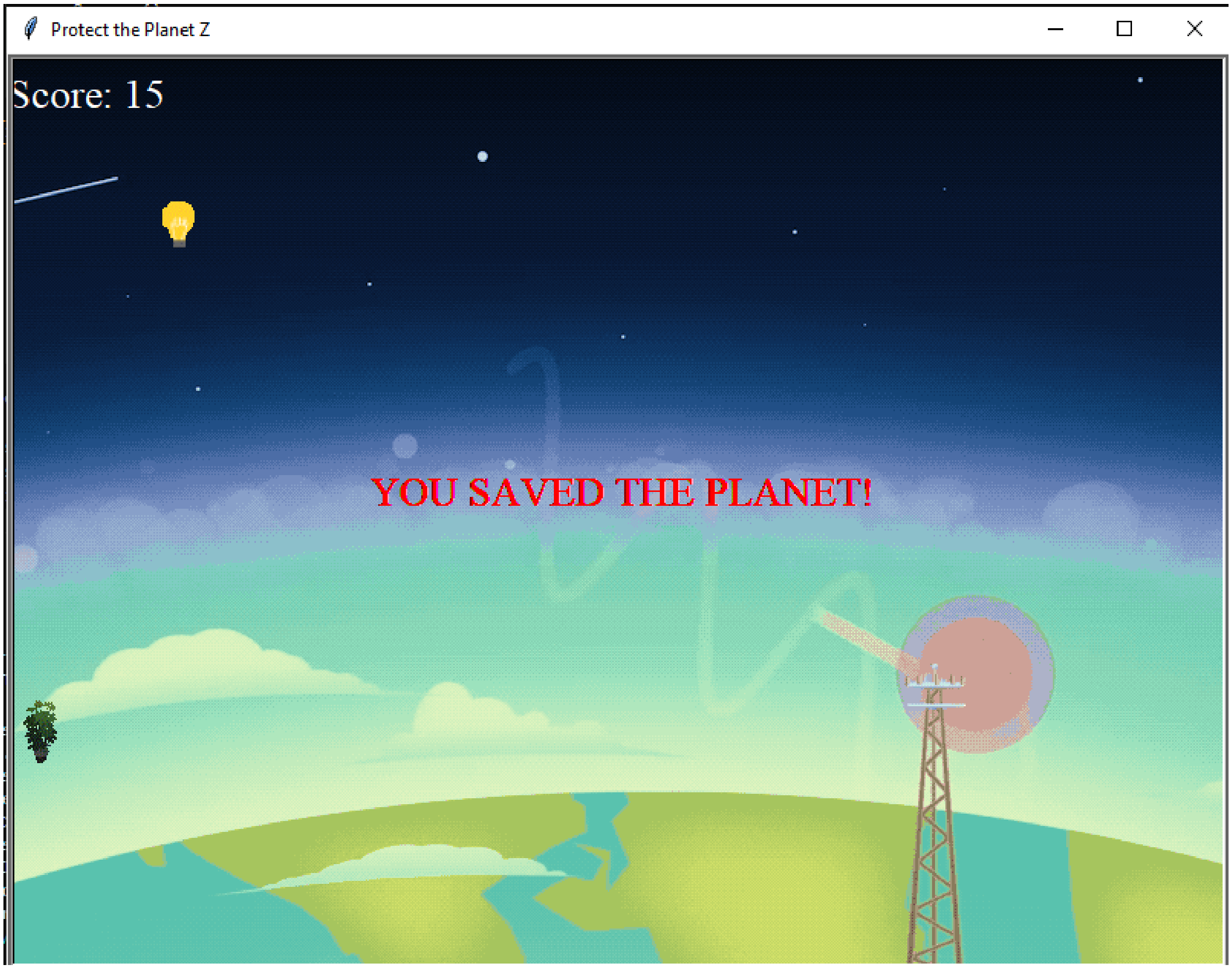
5. Gameplay

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.

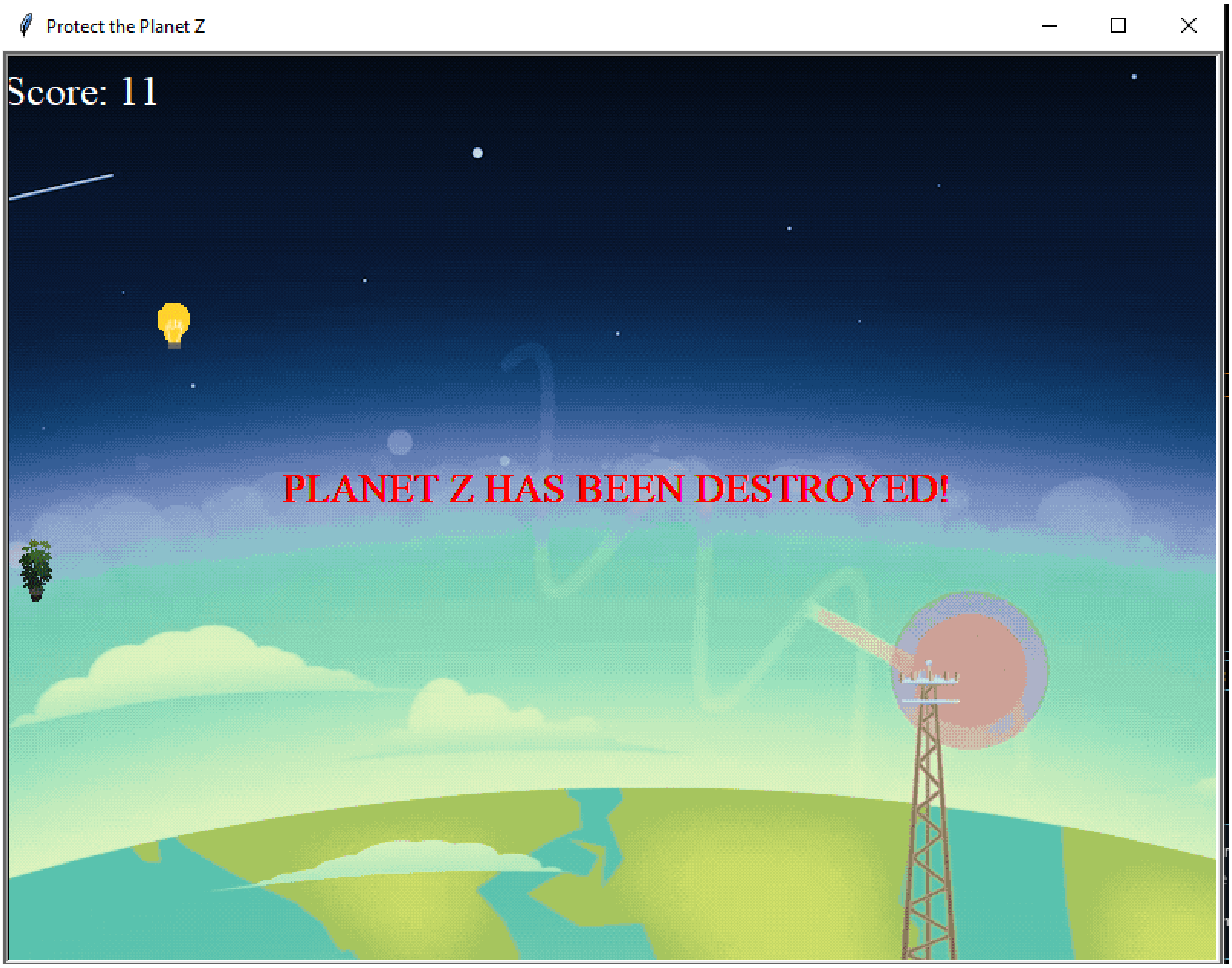


Score: 15

YOU SAVED THE PLANET!

7. Game Over Result

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



Score: 11

PLANET Z HAS BEEN DESTROYED!

APPENDICES

SOURCE CODE

ProtectThePlanetZ.py (Main)

```
import turtle
import random
def next():
  import Lecture

#Create the main screen
wn = turtle.Screen()
wn.title("Get Started!")
wn.setup(800, 600)
wn.bgpic("PPZbg.gif")

wn.listen()
wn.onkeypress(next, "c")
wn.onkeypress(next, "C")

wn.mainloop()
```

Lecture.py

```
from tkinter import*
import tkinter as tk
import tkinter.scrolledtext as st

def quiz():
    import Quiz
# Creating tkinter window
win = tk.Tk()
win.title("Lecture")
win.geometry("500x600")
win.configure(bg="green")
# Title Label
tk.Label(win,  text = "The Last Planet",  font = ("Times New Roman", 20), background = 'green',  foreground = "white").grid(column = 0, row = 10)

nbutton = Button(win, text=">>", command=quiz, width=5,height=1, bg="green", fg="white", font=("times", 16, "bold"))
nbutton.place(x=400, y=5)

# Creating scrolled text area
```

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

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

# Inserting Text which is read only
text_area.insert(tk.INSERT,””” Insert text here””)
  # Making the text read only
text_area.configure(state ='disabled')
win.mainloop()
```

Quiz.py

```
from tkinter import*
from tkinter import messagebox as mb
import json  #You need to create a json file

root = Tk()
root.geometry("800x500")
root.title("Quiz")

with open('quiz.json') as f:
    obj = json.load(f)
q = (obj['ques'])
options = (obj['options'])
a = (obj['ans'])

def startgame():
    import GameInstruction

class Quiz:
    def __init__(self):
        self.qn = 0
        self.ques = self.question(self.qn)
        self.opt_selected = IntVar()
        self.opts = self.radiobtns()
        self.display_options(self.qn)
        self.buttons()
        self.correct = 0

    def question(self, qn):
        t = Label(root, text = "Save the Last Planet!", width=50, bg="blue", fg="white", font=("times", 20, "bold"))
        t.place(x=0, y=2)
        qn = Label(root, text=q[qn], width=60, font=("times", 16, "bold"), anchor="w")
        qn.place(x=70, y=100)
        return qn

    def radiobtns(self):
        val = 0
        b = []
        yp = 150
        while val < 4:
            btn = Radiobutton(root, text="", variable=self.opt_selected, value=val+1, font=("times", 14))
            b.append(btn)
            btn.place(x=100, y=yp)
            val+=1
            yp+=40
        return b

    def display_options(self, qn):
        val = 0
        self.opt_selected.set(0)
        self.ques['text'] =q[qn]
        for op in options[qn]:
            self.opts[val]['text'] = op
            val+=1

    def buttons(self):
        nbutton = Button(root, text="Next", command=self.nextbtn, width=10, bg="green", fg="white", font=("times", 16, "bold"))
        nbutton.place(x=180, y=380)
        startbutton = Button(root, text="Start Game", command=startgame, width=10, bg="red", fg="white", font=("times", 16, "bold"))
        startbutton.place(x=360, y=380)
        quitbutton = Button(root, text="Quit", command=root.destroy, width=10, bg="black", fg="white", font=("times", 16, "bold"))
       quitbutton.place(x=540, y=380)

    def checkans(self, qn):
        if self.opt_selected.get() == a[qn]:
            return True

    def nextbtn(self):
        if self.checkans(self.qn):
            self.correct+=1
        self.qn +=1
        if self.qn == len(q):
            self.display_result()
        else:
            self.display_options(self.qn)

    def display_result(self):
        score = int(self.correct/len(q) * 100)
        result = "Score: " + str(score) + "%"
        wc = len(q) - self.correct
        correct = "No. of correct answers: " + str(self.correct)
        wrong = "No. of wrong answers: " + str(wc)
        warn = "You got "+ str(self.correct)+ " shield to protect the Planet!"
        mb.showinfo("Result", "\n".join([result, correct, wrong, warn]))

quiz = Quiz()
root.mainloop()
```

PvZlike.py

```
import turtle
import random

#Create the screen

wn = turtle.Screen()
wn.title("Protect the Planet Z")
wn.setup(800, 600)
#wn.bgcolor("black")
wn.bgpic("mybg.gif")
wn.tracer(0)

wn.register_shape("dont.gif")
wn.register_shape("saw.gif")
wn.register_shape("bulb.gif")
wn.register_shape("storm.gif")
wn.register_shape("plant.gif")

pen = turtle.Turtle()
pen.speed(0)
pen.penup()

#Create the classes
#class Player(turtle.Turtle):
class Player():
    def __init__(self):
        #turtle.Turtle.__init__(self)
        self.color = "red"
        #self.penup()
        self.x = -350
        self.y = 0
        #self.goto(-350, random.randint(-290, 290))
        self.shape = "bulb.gif"
        #self.speed(0)
        self.dy = 0
        self.dx = 0

    def up(self):
        self.dy = 1
        self.dx = 0

    def down(self):
        self.dy = -1
        self.dx = 0

    def left(self):
        self.dx = -1
        self.dy = 0

    def right(self):
        self.dx = 1
        self.dy = 0

    def move(self):
        #self.sety(self.ycor() + self.dy)
        self.y = self.y + self.dy
        self.x = self.x + self.dx

        #Check for border collision
        if self.y > 280:
            self.y = 280
            self.dy = 0
        elif self.y < -280:
            self.y = -280
            self.dy = 0

        if self.x < -390:
            self.x = -390
            self.dx = 0

        if self.x < -290:
            self.x = -290
            self.dx = 0

    def distance(self, other):
        d = ((self.x - other.x) ** 2 + (self.y -  other.y) ** 2) ** 0.5
        return d

    def render(self, pen):
        pen.goto(self.x, self.y)
        pen.shape(self.shape)
        pen.shapesize(1, 1, 0)
        pen.color(self.color)
        pen.stamp()

#class Missile(turtle.Turtle):
class Missile():

    def __init__(self):
        self.color = "black"
        #self.penup()
        self.x = -450
        self.y = 0
        self.shape = "dont.gif"
        self.size = random.randint(1, 9) / -10
        self.dx = 0

    def fire(self):
        self.x = player.x
        self.y = player.y
        self.dx = 2

    def move(self):
        self.x = self.x + self.dx

    def distance(self, other):
        d = ((self.x - other.x) ** 2 + (self.y -  other.y) ** 2) ** 0.5
        return d

    def render(self, pen):
        pen.goto(self.x, self.y)
        pen.shape(self.shape)
        pen.shapesize(0.4, 0.4, 0)
        pen.color(self.color)
        pen.stamp()

class Enemy():
    def __init__(self):
        shapes = ["saw.gif", "storm.gif"]
        colors = ["blue", "black", "purple", "pink", "gray"]
        self.color = random.choice(colors)
        #self.penup()
        self.x = 380
        self.y = random.randint (-250, 250)
        self.shape = random.choice(shapes)
        self.dx = -0.1

    def move(self):
        self.x = self.x + self.dx
            #Border Check
        if self.x < -400:
            self.x = 400
            self.y = random.randint(-250, 250)
            self.dx *= 1.1

    def distance(self, other):
        d = ((self.x - other.x) ** 2 + (self.y -  other.y) ** 2) ** 0.5
        return d

    def render(self, pen):
        pen.goto(self.x, self.y)
        pen.shape(self.shape)
        pen.shapesize(1, 1, 0)
        pen.color(self.color)
        pen.stamp()
```

```
class Shield():
    def __init__(self):
        self.color = "green"
        self.x = -380
        self.y = random.randint(-250, 250)
        self.shape = "plant.gif"

    def distance(self, other):
        d = ((self.x - other.x) ** 2 + (self.y -  other.y) ** 2) ** 0.5
        return d

    def render(self, pen):
        pen.goto(self.x, self.y)
        pen.shape(self.shape)
        pen.shapesize(1, 1, 0)
        pen.color(self.color)
        pen.stamp()

score = 0
```

```
#Display the score
pen1 = turtle.Turtle()
pen1.hideturtle()
pen1.speed(0)
pen1.shape("circle")
pen1.color("white")
pen1.penup()
pen1.goto(-350, 260)
font = ("Times New Roman", 20, "normal")
pen1.write("Score: {}".format(score), align="center", font=font)

#Create game objects

player = Player()
missile = Missile()
shield = Shield()

#Create a list
enemies = []
noofe = 15
for _ in range(noofe):
 enemies.append(Enemy())

shields = []
for _ in range(15):
    shields.append(Shield())

#Keyboard Bindings

wn.listen()
wn.onkeypress(player.up, "Up")
wn.onkeypress(player.down, "Down")
wn.onkeypress(player.left, "Left")
wn.onkeypress(player.right, "Right")
wn.onkeypress(missile.fire, "space")

while True:

    wn.update()
    pen.clear()


    player.move()
    missile.move()
    player.render(pen)
    missile.render(pen)

    for enemy in enemies:
        enemy.move()

        #Check for collsion
        if enemy.distance(missile) < 13:
            enemy.x = 1000000
            enemy.y = random.randint(-250, 250)
            missile.dx = 0
            missile.x = 0
            missile.y = 1000
            score +=1
            pen1.clear()
            pen1.write("Score: {}".format(score), align="center", font=font)

        if enemy.x < -350:
            pen2 = turtle.Turtle()
            pen2.hideturtle()
            pen2.speed(0)
            pen2.shape("circle")
            pen2.color("red")
            pen2.penup()
            pen2.goto(0, 0)
            font = ("Times New Roman", 20, "normal")
            pen2.write("PLANET Z HAS BEEN DESTROYED!", align="center", font=font)

        if score == 15:
            pen3 = turtle.Turtle()
            pen3.hideturtle()
            pen3.speed(0)
            pen3.shape("circle")
            pen3.color("red")
            pen3.penup()
            pen3.goto(0, 0)
            font = ("Times New Roman", 20, "normal")
            pen3.write("YOU SAVED THE PLANET!", align="center", font=font)

        enemy.render(pen)
    for shield in shields:
        shield.render(pen)
        if enemy.distance(shield) < 13:
            enemy.x = 1000000
            enemy.y = random.randint(-250, 250)
            shield.x = 1000000
            shield.y = 1000000
            score +=1
            pen1.clear()
            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

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	:	Tugtug Elementary School

MEMBERS' PROFILE

RONQUILLO, JULIUS P.

PHOTO

Address: Delas Alas, Batangas City, Batangas

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

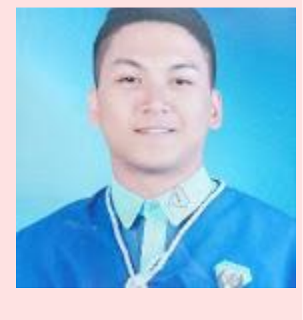
Contact number: 09217899170

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

Educational Background		
Tertiary	:	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	
	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