# GOVERNMENT POLYTECHNIC, PUNE

**(An Autonomous Institute Of Government Of Maharashtra)**

**A Micro-Project Report on**

# Tile Matching Game

**Submitted by :**

**AISHWARYA AUTI - 1906003**

**PRIYANKA BALIVADA - 1906008**

**MINAL CHHATRE – 1906016**

**Under the guidance of**

**Smt. Megha Yawalkar**

**THIRD YEAR – COMPUTER ENGINEERING**

**ACADEMIC YEAR – ODD 2021-22**

**CLASS :- G3**

## Government Polytechnic, Pune-16

**(An Autonomous Institute Of Government of Maharashtra)**

## CERTIFICATE

This is to certify that **Minal Chhatre, Priyanka Balivada and Aishwarya Auti** of class Third Year(2021-2022) have successfully completed project demonstration on **Gaming Application** under the guidance of **Mrs. Megha Yawalkar** Mam inpartial fulfillment of the requirement for the award of diploma in computer engineering from Government Polyetchnic, Pune.

| | | |
|---|---|---|
| **Mrs.** Megha Yawalkar | **Prof. S.B.Nikam** | **Dr. V.S. Bandal** |
| (Course Guide) | (Head Of Department) | (Principal) |

## INDEX

## ACKNOWLEDMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to havegot this all along the completion of my project. All that We have done is only due to such supervision and assistance and We would not forget tothank them.

I respect and thank Smt. Megha Yawalkar, for providing us an opportunity to do the project work and giving us all support and guidance which made us complete the project duty. We are extremely thankful to her forproviding such a nice support and guidance . It is our immense pleasureto express our gratitude to our guide who provided us constructive and positive feedback during the preparation of this project.

Last but not the least, we are thankful to our friends whose encouragement and suggestions helped us to complete our project.
We will also thankful to our PARENTS whose best wishes are always with us.

Thanking You.

# Abstract

Python is a modern, easy-to-learn, object-oriented programming language. It has a powerful set of built-in data types and easy-to-use control constructs. Since Python is aninterpreted language, it is most easily reviewed by simply looking at and describing interactive sessions. It is used in vast number of applications due to the various standard libraries that come along with it and its capacity to integrate with other languages and use their features. Python can be used for scripting, web scraping, and creating data sets. It’s popular in the scientific community for scientific computing; there are libraries that make it easy to share academic codeprojects in Python. Python is a web programming language, so it interfaces with the internet. It knows how to receive andsend web requests and talk to databases. This paperdescribes the main features of Python programming, loops and control statements in python then discusses applications of Python programming

# Introduction

## Tile-Matching Game

A **tile-matching video game** is a type of puzzle video game where the player manipulates tiles in order to make them disappear according to a matching criterion. In many tile-matching games, that criterion is to place a given number of tiles of the same type so that they adjoin each other. That number is often three, and these games are called **match-three games**.

The core challenge of tile-matching games is the identification of patterns on a seemingly chaotic board. Their origins lie in puzzle games from the 1980s such as *Tetris*, *Chain Shot!* (*SameGame*) and *Puzznic*. Tile-matching games were made popular in the 2000s.

# Case Study :

## Goals

The tile-matching is a game where one can play the game withonly some keyboard keys. It can be viewed as refreshment and mood changing tool for those who spends most of the working in front of computer. The game has engaging UI which makes gaming a fantasticexperience.

**The goal of the game is to get all the tiles flipped face up (i.e., find all the matching image pairs) in the least number of tries. That means that lower number of tries are better scores.**

## Working

- When the game starts, all tiles are turned face down.
- The player then flips over two cards, selecting them by clicking on them.
- If the two tiles have the same image, they remain face up. If not, they should be flipped face down again after a short delay

## Technologies used

- **Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

## Packages and modules used :

- ### pygame

pygame is a free and open-source cross-platform library for the development of multimedia applications like video games using Python. It uses the Simple DirectMedia Layer library and several other popular libraries to abstract the most common functions, making writing these programs a more intuitive task.

```
import pygame
```

- ### random

Python Random module is an in-built module of Pythonwhich is used to generate random numbers. These arepseudo-random numbers means these are not truly random. This module can be used to perform randomactions such as generating random numbers, print random a value for a list or string, etc.

```
import random
```

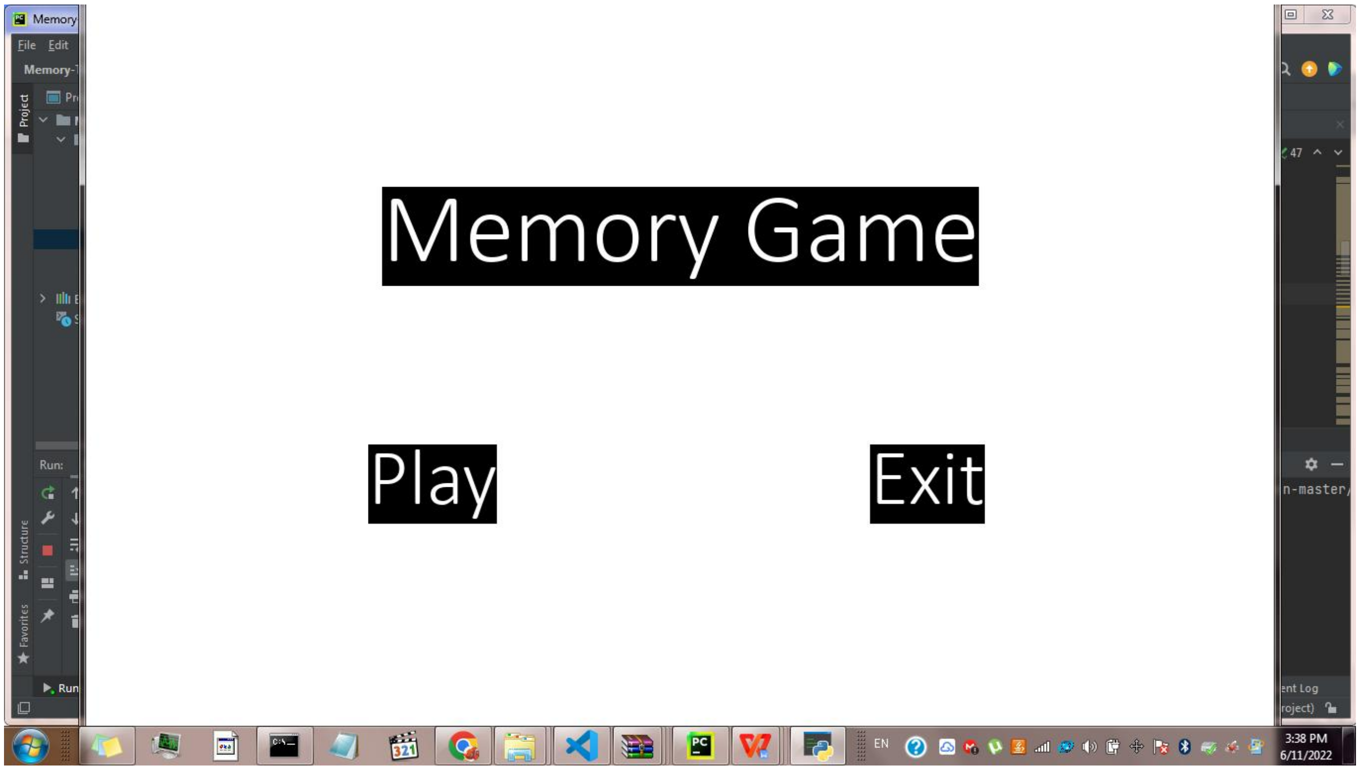
## Pros and Cons

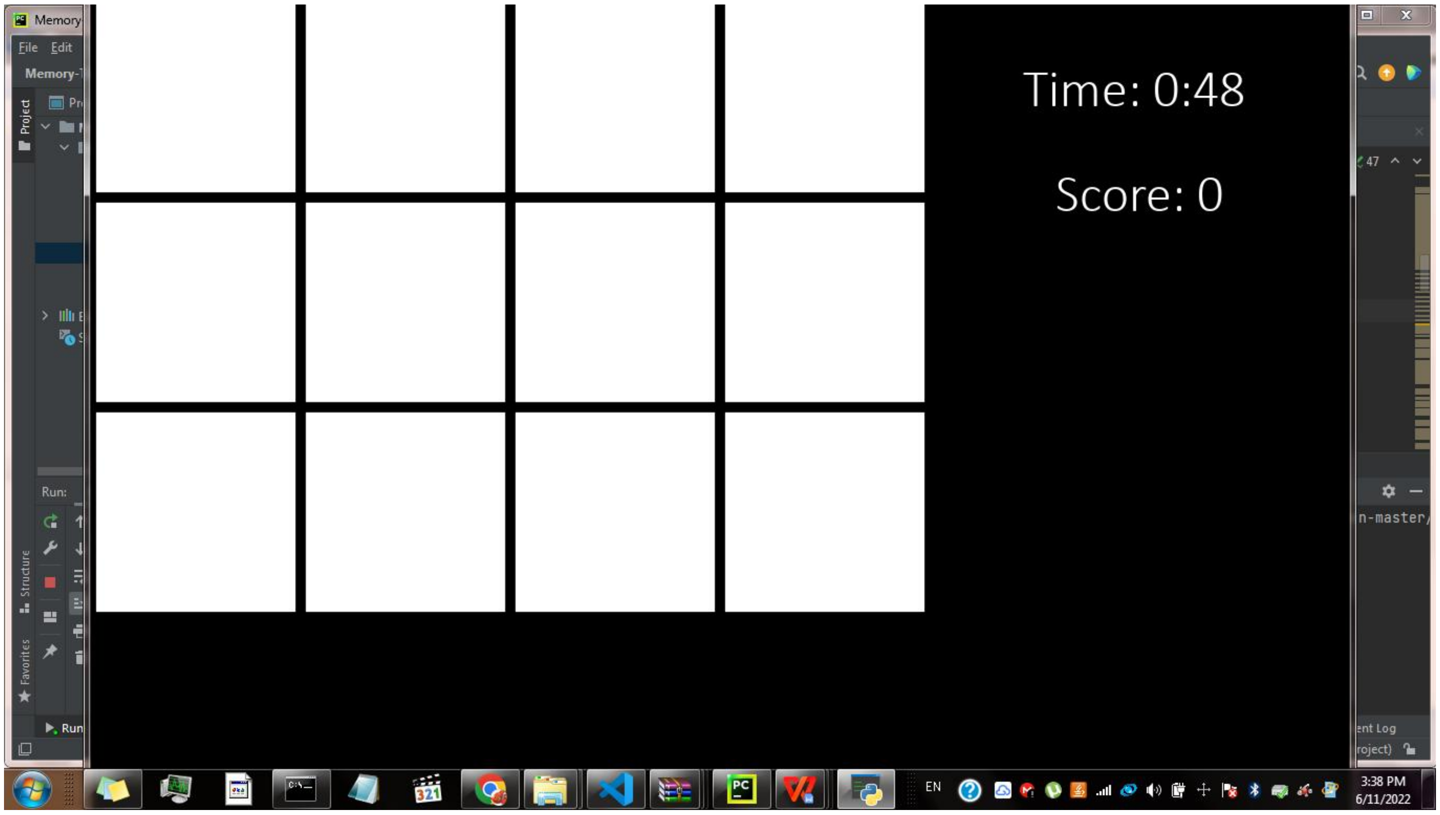
The game has engaging and easy to use UI .The game uses only four keys of keyboard (up arrow, down arrow, left arrow and right arrow) which makes it veryuser friendly and engaging. Talking about drawback, the game doesn't save previous scores of the player, because of this game becomes less challenging.

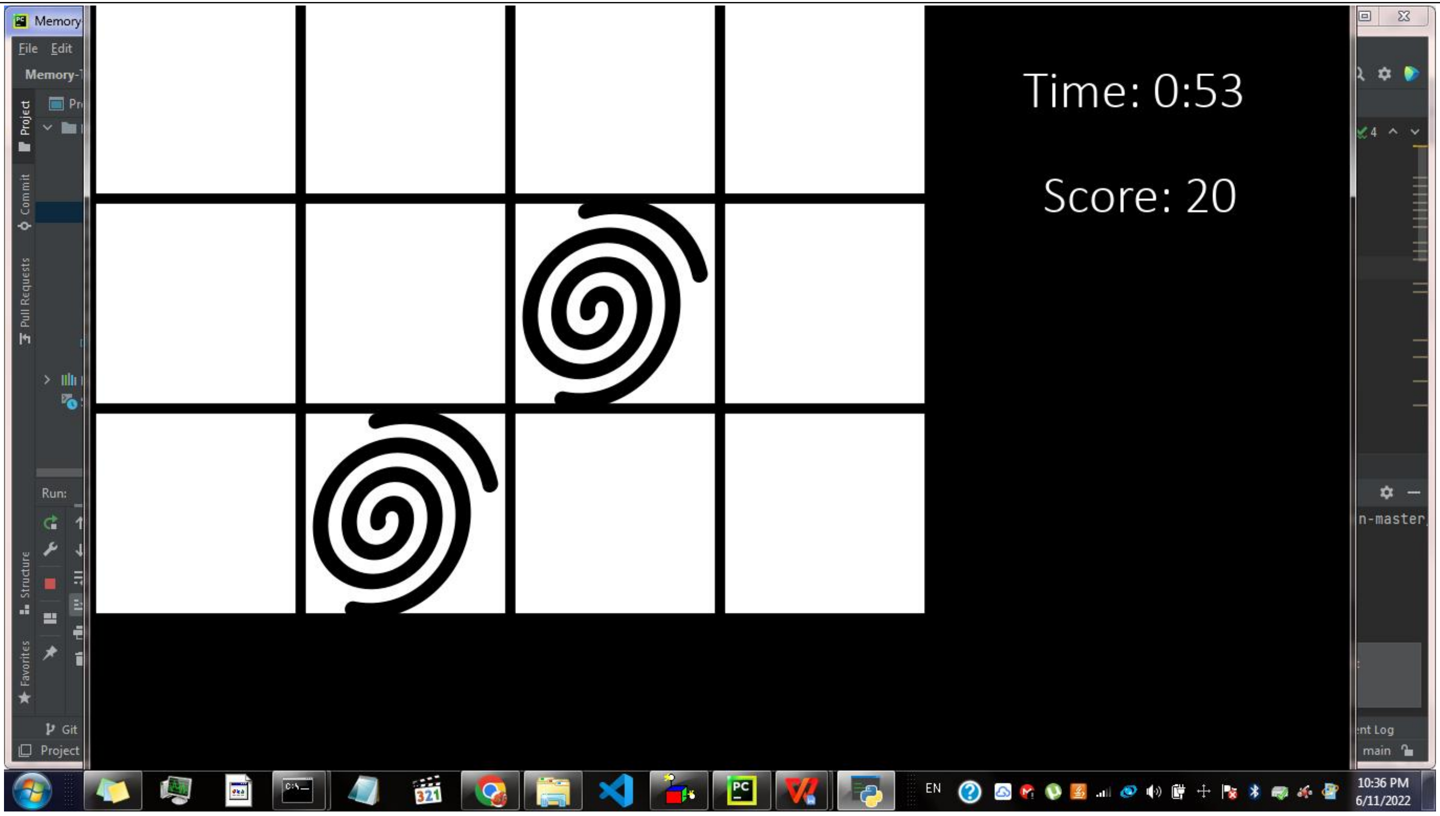
## Code :

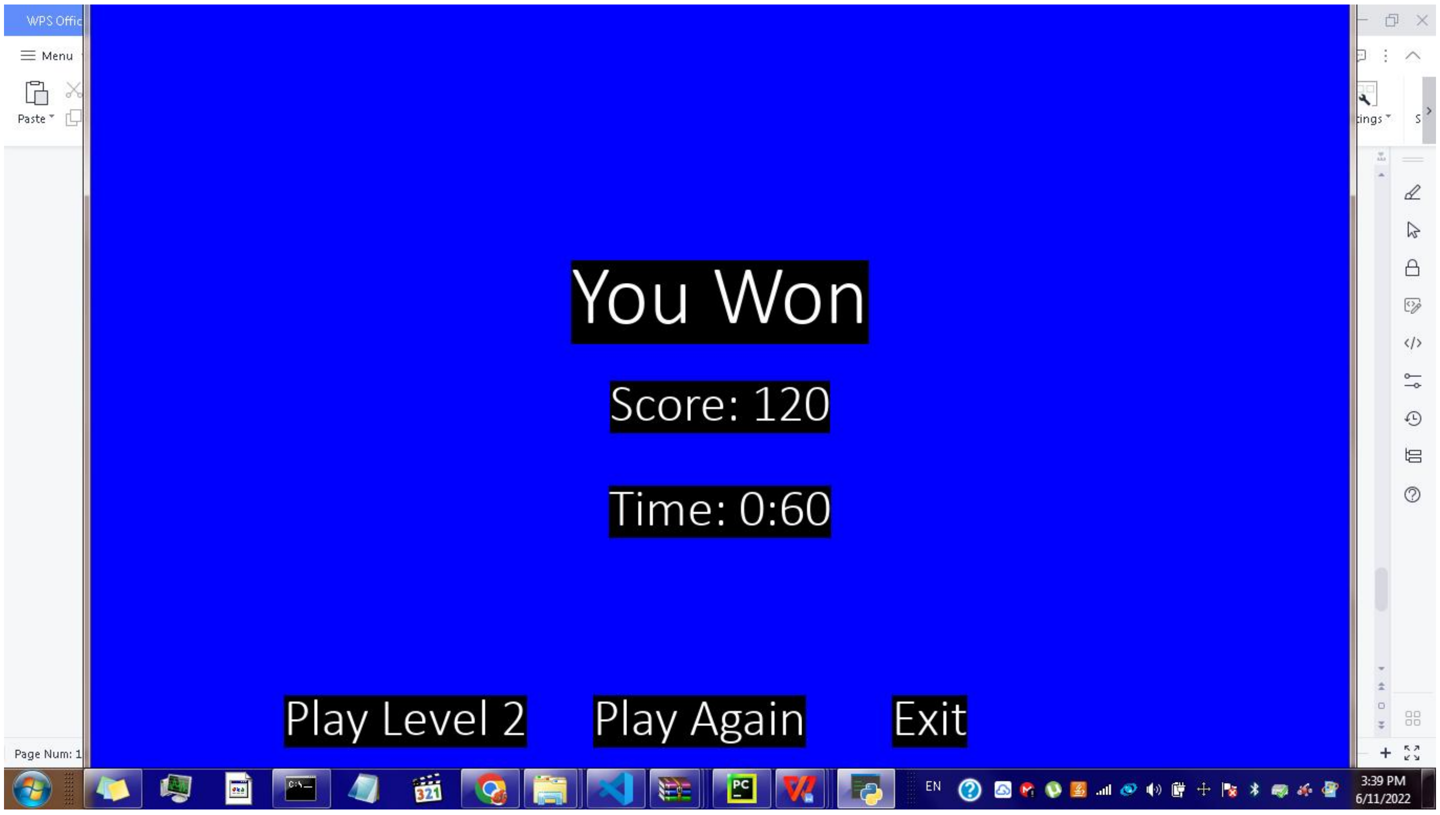
Code on Github

## Look of the game:






Time: 0:53
Score: 20

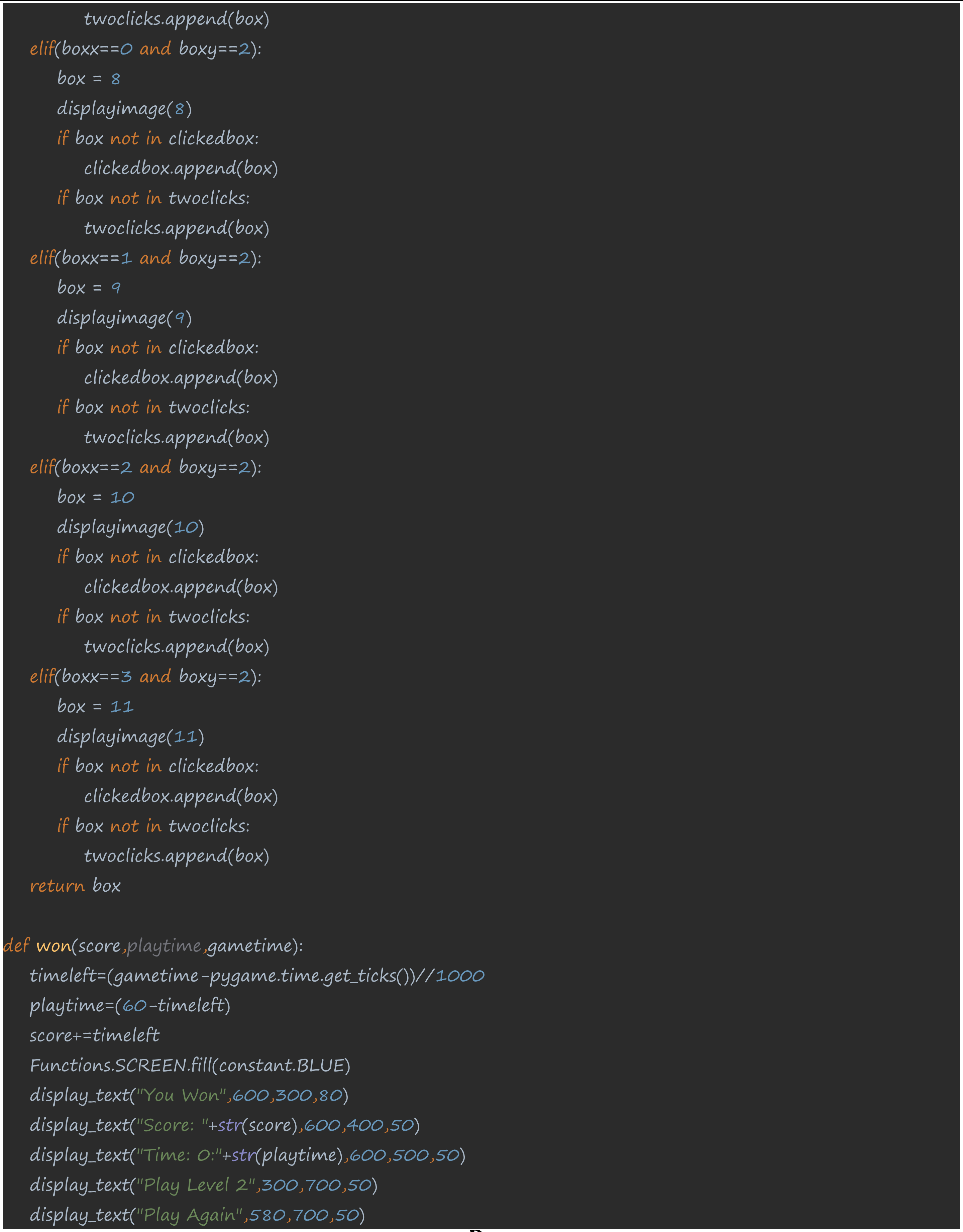

You Won
Score: 120
Time: 0:60
Play Level 2
Play Again
Exit

# Code:

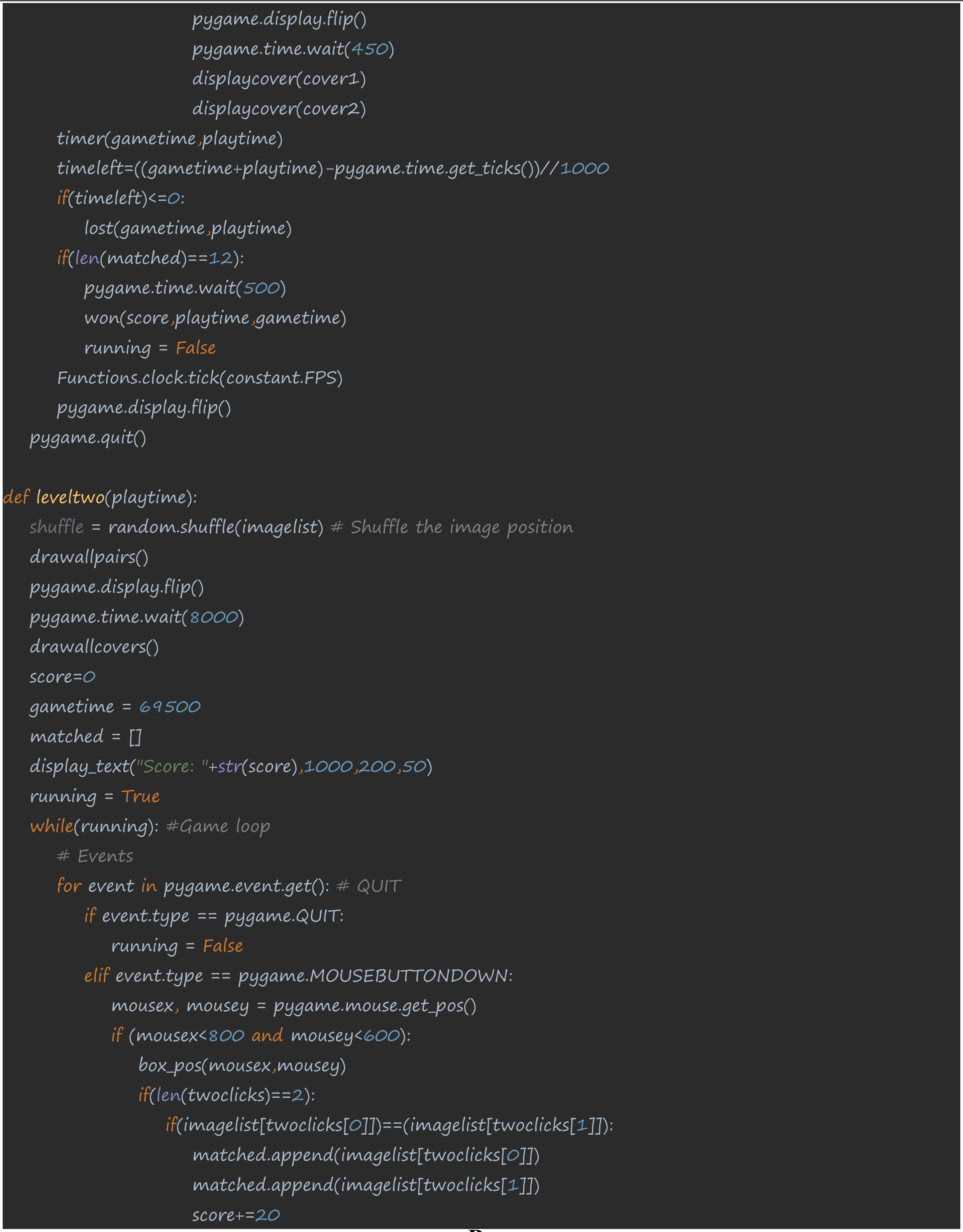
## Maingame.py

```python
import pygame, random
import constant
import Functions

playtime = 0
mousex = 0
mousey = 0
imagelist = []
tilelist = []
twoclicks = []
clickedbox = []
matched = []

# Adding images to a list
imagelist.append(constant.aard)
imagelist.append(constant.axii)
imagelist.append(constant.igni)
tilelist.append(constant.tile)

# Making 6 pairs
imagelist *=4
tilelist*=12

def display_text(writetext,textx,texty,size):
    "Loading text to be used in other functions"
    font = pygame.font.Font(constant.fontname,size)
    text = font.render(writetext,True,constant.WHITE,constant.BLACK)
    textRect = text.get_rect()
    textRect.center = (textx, texty)
    Functions.SCREEN.blit(text, textRect)
    pygame.display.flip()

def timer(gametime,playtime):
    "Countdown timer"
    timeleft=((playtime+gametime)-pygame.time.get_ticks())//1000
    font = pygame.font.Font(constant.fontname, 50)
    time = font.render(' Time: 0:' + str(timeleft)+"  ",True,constant.WHITE, constant.BLACK)
    timeRect = time.get_rect()
    timeRect.center = (1000,100)
    Functions.SCREEN.blit(time, timeRect)
    pygame.display.flip()
    return(timeleft)

def drawallcovers():
    "Drawing all cover tiles"
    tX=0  # tile row x coordinate
    tY=0  # tile row y coordinate
    j=0   # tilelist index

    for i in range(0,4):     # Creating 1st row of tiles
        i+=j
        Functions.SCREEN.blit(tilelist[j], (tX,tY))
        tX+=200
    for i in range(5,9):  # Creating 2nd row of tiles
        i+=j
        Functions.SCREEN.blit(tilelist[j], (tX-800,tY+200))
        tX+=200
    for i in range(9,13): # Creating 3rd row of tiles
        i+=j
        Functions.SCREEN.blit(tilelist[j], (tX-1600,tY+400))
        tX+=200

def drawallpairs():
    "Drawing all image pairs"
    iX=0 # image row x coordinate
    iY=0 # image row y coordinate
    k=0  # imagelist index

    for i in range(0,4): # Creating 1st row of images
        Functions.SCREEN.blit(imagelist[k], (iX,iY))
        k+=1
        iX+=200
    for i in range(5,9):  # Creating 2nd row of images
        Functions.SCREEN.blit(imagelist[k], (iX-800,iY+200))
        k+=1
        iX+=200
    for i in range(9,13): # Creating 3rd row of images
        Functions.SCREEN.blit(imagelist[k], (iX-1600,iY+400))
        k+=1
        iX+=200

def displaycover(index):
    "Drawing one cover at a time"
    if(index==0):
        Functions.SCREEN.blit(tilelist[0], (0,0))
    elif(index==1):
        Functions.SCREEN.blit(tilelist[1], (200,0))
    elif(index==2):
        Functions.SCREEN.blit(tilelist[2], (400,0))
    elif(index==3):
        Functions.SCREEN.blit(tilelist[3], (600,0))
    elif(index==4):
        Functions.SCREEN.blit(tilelist[4], (0,200))
    elif(index==5):
        Functions.SCREEN.blit(tilelist[5], (200,200))
    elif(index==6):
        Functions.SCREEN.blit(tilelist[6], (400,200))
    elif(index==7):
        Functions.SCREEN.blit(tilelist[7], (600,200))
    elif(index==8):
        Functions.SCREEN.blit(tilelist[8], (0,400))
    elif(index==9):
        Functions.SCREEN.blit(tilelist[9], (200,400))
    elif(index==10):
        Functions.SCREEN.blit(tilelist[10], (400,400))
    elif(index==11):
        Functions.SCREEN.blit(tilelist[11], (600,400))

def displayimage(index):
    "Drawing one image at a time"
    if(index==0):
        Functions.SCREEN.blit(imagelist[0], (0,0))
    elif(index==1):
        Functions.SCREEN.blit(imagelist[1], (200,0))
    elif(index==2):
        Functions.SCREEN.blit(imagelist[2], (400,0))
    elif(index==3):
        Functions.SCREEN.blit(imagelist[3], (600,0))
    elif(index==4):
        Functions.SCREEN.blit(imagelist[4], (0,200))
    elif(index==5):
        Functions.SCREEN.blit(imagelist[5], (200,200))
    elif(index==6):
        Functions.SCREEN.blit(imagelist[6], (400,200))
    elif(index==7):
        Functions.SCREEN.blit(imagelist[7], (600,200))
    elif(index==8):
        Functions.SCREEN.blit(imagelist[8], (0,400))
    elif(index==9):
        Functions.SCREEN.blit(imagelist[9], (200,400))
    elif(index==10):
        Functions.SCREEN.blit(imagelist[10], (400,400))
    elif(index==11):
        Functions.SCREEN.blit(imagelist[11], (600,400))

def box_pos(mousex,mousey):
    "Getting position of clicked box and giving it a number"
    boxx = int(mousex/constant.BOXSIZE)
    boxy = int(mousey/constant.BOXSIZE)

    if len(twoclicks)==2:
        del twoclicks[:3]

    if(boxx==0 and boxy==0):
        box = 0
        displayimage(0)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
    elif(boxx==1 and boxy==0):
        box = 1
        displayimage(1)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
```

```
elif(boxx==2 and boxy==0):
    box = 2
    displayimage(2)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
        twoclicks.append(box)
elif(boxx==3 and boxy==0):
    box = 3
    displayimage(3)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
        twoclicks.append(box)
elif(boxx==0 and boxy==1):
    box = 4
    displayimage(4)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
        twoclicks.append(box)
elif(boxx==1 and boxy==1):
    box = 5
    displayimage(5)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
        twoclicks.append(box)
elif(boxx==2 and boxy==1):
    box = 6
    displayimage(6)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
        twoclicks.append(box)
elif(boxx==3 and boxy==1):
    box = 7
    displayimage(7)
    if box not in clickedbox:
        clickedbox.append(box)
    if box not in twoclicks:
```



```
            twoclicks.append(box)
    elif(boxx==0 and boxy==2):
        box = 8
        displayimage(8)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
    elif(boxx==1 and boxy==2):
        box = 9
        displayimage(9)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
    elif(boxx==2 and boxy==2):
        box = 10
        displayimage(10)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
    elif(boxx==3 and boxy==2):
        box = 11
        displayimage(11)
        if box not in clickedbox:
            clickedbox.append(box)
        if box not in twoclicks:
            twoclicks.append(box)
    return box

def won(score,playtime,gametime):
    timeleft=(gametime-pygame.time.get_ticks())//1000
    playtime=(60-timeleft)
    score+=timeleft
    Functions.SCREEN.fill(constant.BLUE)
    display_text("You Won",600,300,80)
    display_text("Score: "+str(score),600,400,50)
    display_text("Time: 0:"+str(playtime),600,500,50)
    display_text("Play Level 2",300,700,50)
    display_text("Play Again",580,700,50)
    display_text("Exit",800,700,50)
    pygame.display.flip()
    pygame.time.wait(1000)
    running = True
    while(running):
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
                pygame.quit()
                quit()
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mousex, mousey = pygame.mouse.get_pos()
                if (mousex>175 and mousex<420) and (mousey>675 and mousey<725):
                    timeleft=(gametime-pygame.time.get_ticks())//1000
                    timeleftwin = timeleft
                    playtime=(60-timeleftwin)*1000
                    Functions.SCREEN.fill(constant.BLACK)
                    leveltwo(playtime)
                if (mousex>475 and mousex<685) and (mousey>650 and mousey<750):
                    timeleft=(gametime-pygame.time.get_ticks())//1000
                    timeleftwin = timeleft
                    playtime=(60-timeleftwin)*1000
                    print(playtime)
                    Functions.SCREEN.fill(constant.BLACK)
                    levelone(playtime)
                elif (mousex>760 and mousex<840) and (mousey>675 and mousey<725):
                    pygame.quit()
                    quit()

def lost(gametime,playtime):
    Functions.SCREEN.fill(constant.RED)
    display_text("You Lost",600,250,80)
    display_text("Play Again",350,500,50)
    display_text("Exit",850,500,50)
    pygame.display.flip()
    running = True
    while(running):
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
                pygame.quit()
                quit()
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mousex, mousey = pygame.mouse.get_pos()
                if (mousex>280 and mousex<420) and (mousey>450 and mousey<550):
                    timeleft=(gametime-pygame.time.get_ticks())//1000
                    timeleftwin = timeleft
                    playtime=(60-timeleftwin)*1000
                    Functions.SCREEN.fill(constant.BLACK)
                    levelone(playtime)
                elif (mousex>780 and mousex<915) and (mousey>455 and mousey<550):
                    pygame.quit()
                    quit()

def levelone(playtime):
    drawallcovers()
    shuffle = random.shuffle(imagelist) # Shuffle the image position
    score=0
    gametime = 61500
    matched = []
    display_text("Score: "+str(score),1000,200,50)
    running = True
    while(running): #Game loop
        # Events
        for event in pygame.event.get(): # QUIT
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mousex, mousey = pygame.mouse.get_pos()
                if (mousex<800 and mousey<600):
                    box_pos(mousex,mousey)
                    if(len(twoclicks)==2):
                        if(imagelist[twoclicks[0]])==(imagelist[twoclicks[1]]):
                            matched.append(imagelist[twoclicks[0]])
                            matched.append(imagelist[twoclicks[1]])
                            score+=20
                            display_text("Score: "+str(score),1000,200,50)
                        else:
                            cover1 = int(twoclicks[0])
                            cover2 = int(twoclicks[1])
                            displayimage(cover1)
                            displayimage(cover2)
```



```
                            pygame.display.flip()
                            pygame.time.wait(450)
                            displaycover(cover1)
                            displaycover(cover2)
        timer(gametime,playtime)
        timeleft=((gametime+playtime)-pygame.time.get_ticks())//1000
        if(timeleft)<=0:
            lost(gametime,playtime)
        if(len(matched)==12):
            pygame.time.wait(500)
            won(score,playtime,gametime)
            running = False
        Functions.clock.tick(constant.FPS)
        pygame.display.flip()
    pygame.quit()

def leveltwo(playtime):
    shuffle = random.shuffle(imagelist) # Shuffle the image position
    drawallpairs()
    pygame.display.flip()
    pygame.time.wait(8000)
    drawallcovers()
    score=0
    gametime = 69500
    matched = []
    display_text("Score: "+str(score),1000,200,50)
    running = True
    while(running): #Game loop
        # Events
        for event in pygame.event.get(): # QUIT
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.MOUSEBUTTONDOWN:
                mousex, mousey = pygame.mouse.get_pos()
                if (mousex<800 and mousey<600):
                    box_pos(mousex,mousey)
                    if(len(twoclicks)==2):
                        if(imagelist[twoclicks[0]])==(imagelist[twoclicks[1]]):
                            matched.append(imagelist[twoclicks[0]])
                            matched.append(imagelist[twoclicks[1]])
                            score+=20
                            display_text("Score: "+str(score),1000,200,50)
                        else:
                            cover1 = int(twoclicks[0])
                            cover2 = int(twoclicks[1])
                            displayimage(cover1)
                            displayimage(cover2)
                            pygame.display.flip()
                            pygame.time.wait(450)
                            displaycover(cover1)
                            displaycover(cover2)
        timer(gametime,playtime)
        timeleft=((gametime+playtime)-pygame.time.get_ticks())//1000
        if(timeleft)<=0:
            lost(gametime,playtime)
        if(len(matched)==12):
            pygame.time.wait(500)
            won(score,playtime,gametime)
            running = False
        Functions.clock.tick(constant.FPS)
        pygame.display.flip()
    pygame.quit()


Functions.start_game()
```

## Functions.py

```
import pygame, random
import maingame
import constant

def start_game():
    global SCREEN, clock
    pygame.init() #run pygame
    SCREEN = pygame.display.set_mode((constant.WIDTH, constant.HEIGHT)) #Window
    pygame.display.set_caption('Memory Game') #Title
    clock = pygame.time.Clock() # Track time
    bg = pygame.Surface((constant.WIDTH,constant.HEIGHT))
    SCREEN.fill(constant.WHITE)
    maingame.display_text('Memory Game',600,250,100)
    maingame.display_text('Play',350,500,80)
    maingame.display_text('Exit',850,500,80)
    pygame.display.flip()
    running = True
    while(running):
        for event in pygame.event.get(): # QUIT
            if event.type == pygame.QUIT:
                running = False
                pygame.quit()
                quit()
            elif event.type == pygame.MOUSEBUTTONDOWN:
                global mousex, mousey
                mousex, mousey = pygame.mouse.get_pos()
                if (mousex>280 and mousex<420) and (mousey>450 and mousey<550):
                    SCREEN.fill(constant.BLACK)
                    maingame.levelone(maingame.playtime)
                elif (mousex>780 and mousex<915) and (mousey>455 and mousey<550):
                    pygame.quit()
                    quit()
    pygame.quit()
```

**Constants.py**

```
import pygame

# Variables
WIDTH = 1200
HEIGHT = 800
FPS = 60
BOXSIZE = 200
fontname = pygame.font.match_font('calibri')

# Colours
BLACK = (  0,   0,   0)
WHITE = (255, 255, 255)
RED   = (255,   0,   0)
GREEN = (  0, 255,   0)
BLUE  = (  0,   0, 255)

# Load images
aard = pygame.image.load("loop4.png","aard")
axii = pygame.image.load("loop5.png")
igni = pygame.image.load("loop6.png")
tile = pygame.image.load("tile.png")
```

## References :

- Github
- Stackoverflow

## Conclusion:

We built a game together wherein players can play it very easily, using python and its packagesand modules