Odd Semester (2021)

BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:		**Surname**	**Given Names**	**Student ID Number**
	1.	**Nicolas**	**Nicolas**	**2101704520**

Course Code	**: COMP6502**	**Course Name**	**: Introduction to Programming**
Class	**: L1AC**	**Name of Lecturer(s)**	**:** 1. Ida Bagus Kerthyayana Manuaba 2. Tri Asih Budiono
Major	**: CS**		
Title of Assignment	: Whack an Ant		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 6-11-2017**	**Submission Date**	**: 6-11-2017**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. The assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating, and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)
1. Nicolas

“Interest and Depreciation”

Name :Nicolas

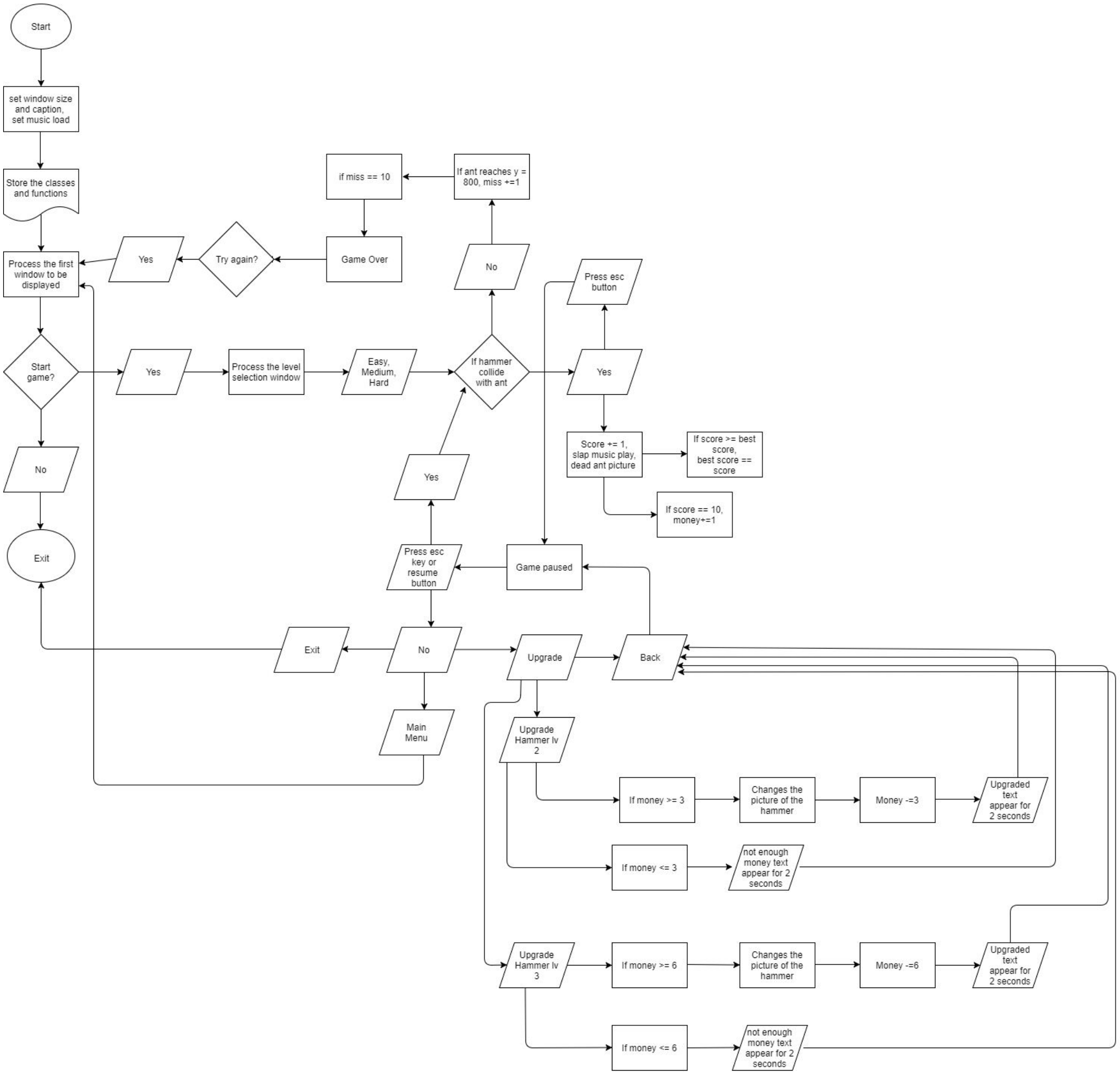
ID :2101704520

I. Description

The function of this program:

This program is a game developed using python, pygame meant to help people to relief their stress by only making them to play this amazing clicking game which is an ant game. This game has variety features to let the user play such as levels, and also upgrades. In this game, there is a lot more features to let the user interact and have fun.

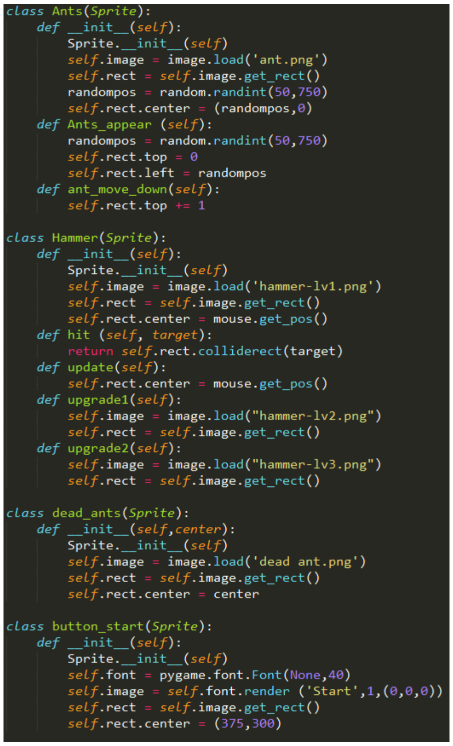
II.a. Design/Plan


Start
set window size and caption, set music load
Store the classes and functions
Process the first window to be displayed
Yes
Try again?
Game Over
if miss == 10
If ant reaches y = 800, miss +=1
No
Press esc button
Start game?
Yes
Process the level selection window
Easy, Medium, Hard
If hammer collide with ant
Yes
Score += 1, slap music play, dead ant picture
If score >= best score, best score == score
If score == 10, money+=1
No
Exit
Yes
Press esc key or resume button
Game paused
Exit
No
Upgrade
Back
Main Menu
Upgrade Hammer lv 2
If money >= 3
Changes the picture of the hammer
Money -=3
Upgraded text appear for 2 seconds
If money <= 3
not enough money text appear for 2 seconds
Upgrade Hammer lv 3
If money >= 6
Changes the picture of the hammer
Money -=6
Upgraded text appear for 2 seconds
If money <= 6
not enough money text appear for 2 seconds

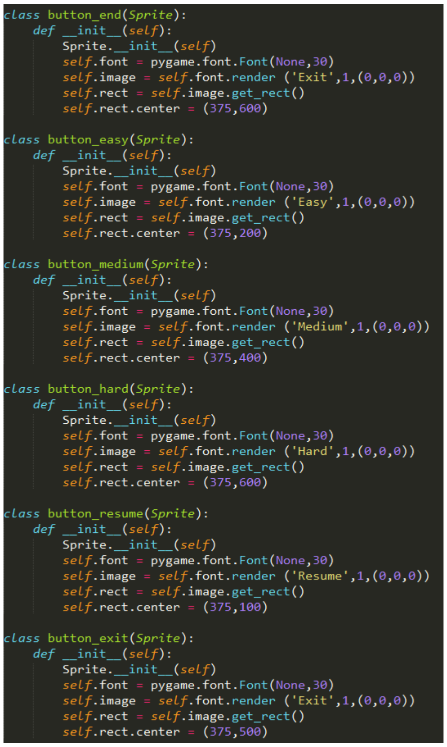
II.b Explanation of Each function

1. Classes:

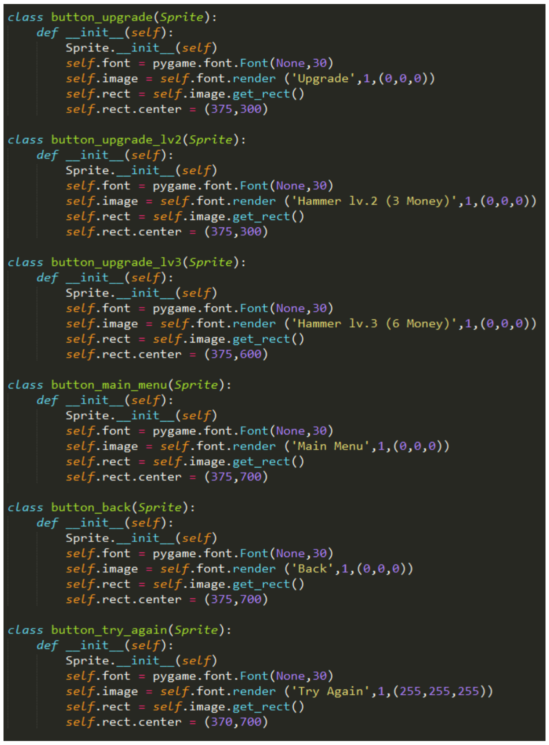
- Holds the sprites as images, and text to be pressed as a button.
- Class ants - is to load an image which is an ant image in order to load the image on the program's screen. It also has two functions inside the class which are def ant move down and def ant appear which is basically ant move down is a code that allows the image of the ant to move down from the top of the screen. And the ant appear holds the function of the ant to appear from the top of the screen which is a y = 0 and x = random.
- Class hammer - is there to also load an image as the replacement of the cursor to make the game look and play better.
- Class dead ants - is in the class also to load an image of a dead ant after clicking the moving ant.
- Class buttons (start, end, easy, medium, hard, resume, exit, upgrade, back, try again) - is in the class as a sprite to make a text to the window game and make it as clickable by the collidepoint function.



```
class Ants(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.image = image.load('ant.png')
        self.rect = self.image.get_rect()
        randompos = random.randint(50,750)
        self.rect.center = (randompos,0)
    def Ants_appear (self):
        randompos = random.randint(50,750)
        self.rect.top = 0
        self.rect.left = randompos
    def ant_move_down(self):
        self.rect.top += 1

class Hammer(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.image = image.load('hammer-lv1.png')
        self.rect = self.image.get_rect()
        self.rect.center = mouse.get_pos()
    def hit (self, target):
        return self.rect.colliderect(target)
    def update(self):
        self.rect.center = mouse.get_pos()
    def upgrade1(self):
        self.image = image.load("hammer-lv2.png")
        self.rect = self.image.get_rect()
    def upgrade2(self):
        self.image = image.load("hammer-lv3.png")
        self.rect = self.image.get_rect()

class dead_ants(Sprite):
    def __init__(self,center):
        Sprite.__init__(self)
        self.image = image.load('dead ant.png')
        self.rect = self.image.get_rect()
        self.rect.center = center

class button_start(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,40)
        self.image = self.font.render ('Start',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)
```

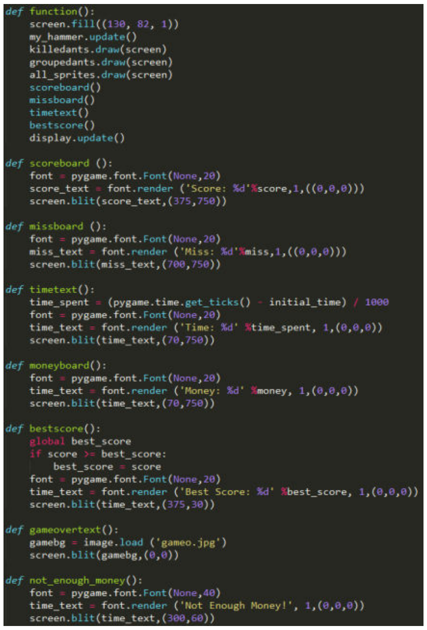


```
class button_end(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Exit',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_easy(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Easy',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,200)

class button_medium(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Medium',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,400)

class button_hard(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hard',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_resume(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Resume',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,100)

class button_exit(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Exit',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,500)
```

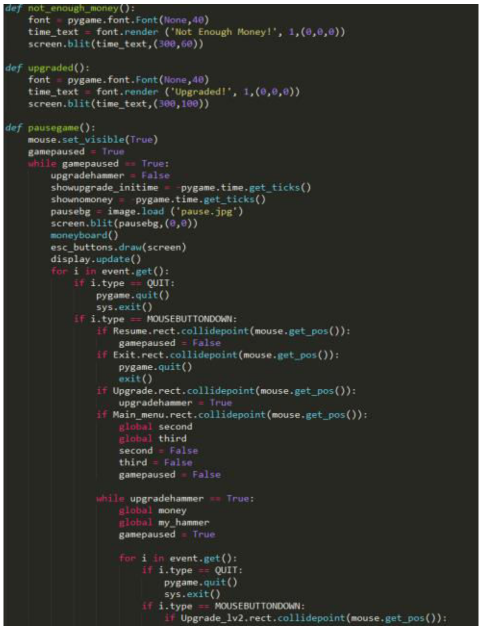


```
class button_upgrade(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Upgrade',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)

class button_upgrade_lv2(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hammer lv.2 (3 Money)',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)

class button_upgrade_lv3(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hammer lv.3 (6 Money)',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_main_menu(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Main Menu',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,700)

class button_back(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Back',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,700)

class button_try_again(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Try Again',1,(255,255,255))
        self.rect = self.image.get_rect()
        self.rect.center = (370,700)
```

2. Function (def):
    - Holds all of the necessary functions in the program as:
        1. A function that calls all of the necessary function in order to run the program and tells the program what to do.
    - Def function – calls all the functions that need to be in the game’s window
    - Def texts (score, miss, time, money, best score, not enough money, upgraded)– creates a text to be displayed in the game and it also can be updated by the code in the main game and updated by def function by calling the text again and display update.
    - Def game over text – is a function where if there is 10 misses, the game will stop and displays the game over which is a picture.
    - Def pausegame – a while loop for pausing the game to be able to give an actions to the game such as resume (can be done by clicking it or pressing esc button), exit (exit the game), main menu (go back to main menu), and upgrade for the weapon which is the hammer from level 1 to level 2 and level 3 which is basically change the picture of the hammer to a bigger hammer and for the texts in the pausegame layer, it shows the money which is how much money you have currently and the money is used to upgrade the hammer. And as if the user clicks on the upgrade button, the game will have another while loop to access the upgrade hammer and by having the upgrade hammer, we will need to global the variable for the hammer lv 1 and also the money in order to make the use of it in this layer.

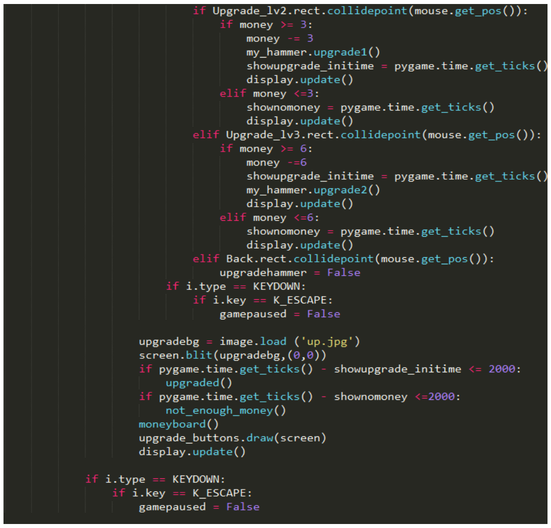


```
def function():
    screen.fill((130, 82, 1))
    my_hammer.update()
    killedants.draw(screen)
    groupedants.draw(screen)
    all_sprites.draw(screen)
    scoreboard()
    missboard()
    timetext()
    bestscore()
    display.update()

def scoreboard ():
    font = pygame.font.Font(None,20)
    score_text = font.render ('Score: %d'%score,1,((0,0,0)))
    screen.blit(score_text,(375,750))

def missboard ():
    font = pygame.font.Font(None,20)
    miss_text = font.render ('Miss: %d'%miss,1,((0,0,0)))
    screen.blit(miss_text,(700,750))

def timetext():
    time_spent = (pygame.time.get_ticks() - initial_time) / 1000
    font = pygame.font.Font(None,20)
    time_text = font.render ('Time: %d' %time_spent, 1,(0,0,0))
    screen.blit(time_text,(70,750))

def moneyboard():
    font = pygame.font.Font(None,20)
    time_text = font.render ('Money: %d' %money, 1,(0,0,0))
    screen.blit(time_text,(70,750))

def bestscore():
    global best_score
    if score >= best_score:
        best_score = score
    font = pygame.font.Font(None,20)
    time_text = font.render ('Best Score: %d' %best_score, 1,(0,0,0))
    screen.blit(time_text,(375,30))

def gameovertext():
    gamebg = image.load ('gameo.jpg')
    screen.blit(gamebg,(0,0))

def not_enough_money():
    font = pygame.font.Font(None,40)
    time_text = font.render ('Not Enough Money!', 1,(0,0,0))
    screen.blit(time_text,(300,60))
```



```
def not_enough_money():
    font = pygame.font.Font(None,40)
    time_text = font.render ('Not Enough Money!', 1,(0,0,0))
    screen.blit(time_text,(300,60))

def upgraded():
    font = pygame.font.Font(None,40)
    time_text = font.render ('Upgraded!', 1,(0,0,0))
    screen.blit(time_text,(300,100))

def pausegame():
    mouse.set_visible(True)
    gamepaused = True
    while gamepaused == True:
        upgradehammer = False
        showupgrade_initime = -pygame.time.get_ticks()
        shownomoney = -pygame.time.get_ticks()
        pausebg = image.load ('pause.jpg')
        screen.blit(pausebg,(0,0))
        moneyboard()
        esc_buttons.draw(screen)
        display.update()
        for i in event.get():
            if i.type == QUIT:
                pygame.quit()
                sys.exit()
            if i.type == MOUSEBUTTONDOWN:
                if Resume.rect.collidepoint(mouse.get_pos()):
                    gamepaused = False
                if Exit.rect.collidepoint(mouse.get_pos()):
                    pygame.quit()
                    exit()
                if Upgrade.rect.collidepoint(mouse.get_pos()):
                    upgradehammer = True
                if Main_menu.rect.collidepoint(mouse.get_pos()):
                    global second
                    global third
                    second = False
                    third = False
                    gamepaused = False

                while upgradehammer == True:
                    global money
                    global my_hammer
                    gamepaused = True

                    for i in event.get():
                        if i.type == QUIT:
                            pygame.quit()
                            sys.exit()
                        if i.type == MOUSEBUTTONDOWN:
                            if Upgrade_lv2.rect.collidepoint(mouse.get_pos()):
```

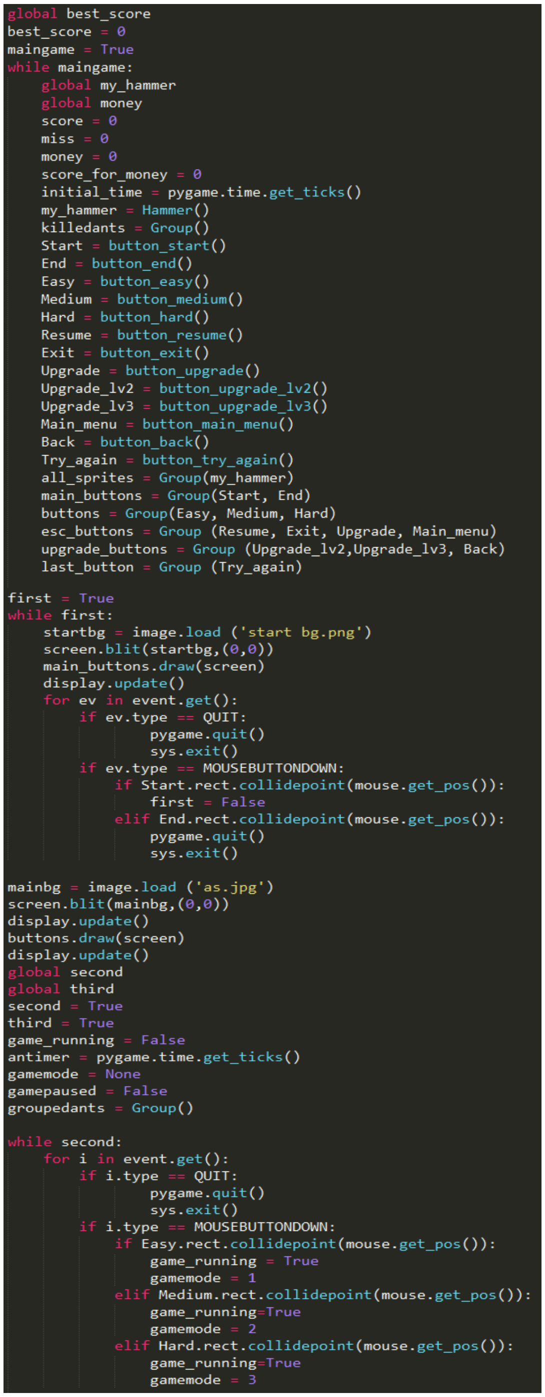
3.



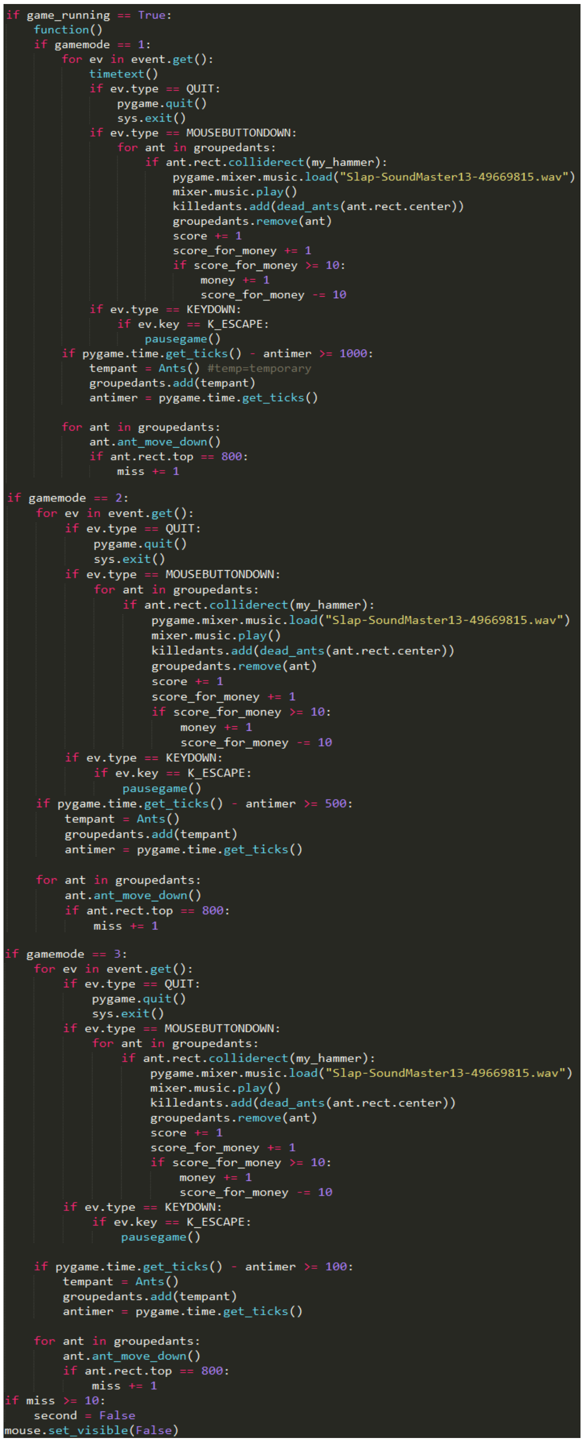
```
                            if Upgrade_lv2.rect.collidepoint(mouse.get_pos()):
                                if money >= 3:
                                    money -= 3
                                    my_hammer.upgrade1()
                                    showupgrade_initime = pygame.time.get_ticks()
                                    display.update()
                                elif money <=3:
                                    shownomoney = pygame.time.get_ticks()
                                    display.update()
                            elif Upgrade_lv3.rect.collidepoint(mouse.get_pos()):
                                if money >= 6:
                                    money -=6
                                    showupgrade_initime = pygame.time.get_ticks()
                                    my_hammer.upgrade2()
                                    display.update()
                                elif money <=6:
                                    shownomoney = pygame.time.get_ticks()
                                    display.update()
                            elif Back.rect.collidepoint(mouse.get_pos()):
                                upgradehammer = False
                        if i.type == KEYDOWN:
                            if i.key == K_ESCAPE:
                                gamepaused = False

                    upgradebg = image.load ('up.jpg')
                    screen.blit(upgradebg,(0,0))
                    if pygame.time.get_ticks() - showupgrade_initime <= 2000:
                        upgraded()
                    if pygame.time.get_ticks() - shownomoney <=2000:
                        not_enough_money()
                    moneyboard()
                    upgrade_buttons.draw(screen)
                    display.update()

            if i.type == KEYDOWN:
                if i.key == K_ESCAPE:
                    gamepaused = False
```

Main game

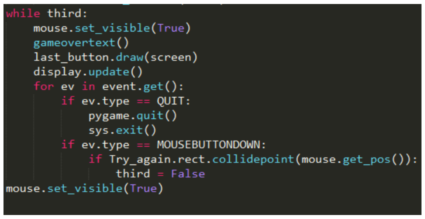
- In this section of the code, there are four while loops that runs the game which the first loop contains all the variables from the class to be declared and grouped by their individual function in the game such as the first screen that pops when you run the game will be the grouped button start which make the second loop = True and button exit which exits the game with a background inside the first loop of the game, into the second loop, which is the gamemode that contains a background and the three buttons for the level select which are easy, medium, and hard where the if the easy button collides with the cursor, the variable, game running will be True and will set the variable game mode to number 1 to use a different timing for the ant to appear which is the longest time to make the ant to appear to the display of the game with the time of exactly 1 second. Where the if the medium button collides with the cursor, game running will be True and will set game mode to number 2 to use a different timing for the ant to appear which is the second longest time to make the ant to appear to the display of the game with the time of exactly 0.5 seconds. Where the if the hard button collides with the cursor, game running will be True and will set game mode to number 3 to use a different timing for the ant to appear which is the second longest time to make the ant to appear to the display of the game with the time of exactly 0.1 seconds. The ant is grouped by the variable grouped ant in the first loop. Once the ant is clicked, there will be a smack music played in the background and also there is the function in all of the game modes from east, medium, and hard to remove the clicked ant and to replace the clicked ant by the dead ant described as the class dead ant by the code of killedants.add(dead_ants (ant.rect.center)) which killedants is a variable to group the dead_ant (the class) and with the ant.rect.center is the position of the clicked ant to be replaced. For the money, the user needs to have ten scores to get one money for all levels which the code will need to have a different variable for a score which displays the score in the main game and the score for the money to go up and minus the price for the upgrade. Once the user clicks on the upgrade button with enough money for upgrade hammer level 2 or 3, the game will decrease the money of the user with an upgraded hammer and there will be a pop up text above that says upgraded as if the user does not have enough money, there will also be a pop up text above saying not enough money! And the hammer will not upgrade. Once one ant touches the coordinate of y = 800 px because the window is 800 X 800 pixel, the miss will add by one and if the miss reaches ten miss, the second loop will be False and the third loop comes which is the function game over that displays the game over picture with a try again button below it to make the game return to the main menu, which is the first loop that shows the start and end button.



```
global best_score
best_score = 0
maingame = True
while maingame:
    global my_hammer
    global money
    score = 0
    miss = 0
    money = 0
    score_for_money = 0
    initial_time = pygame.time.get_ticks()
    my_hammer = Hammer()
    killedants = Group()
    Start = button_start()
    End = button_end()
    Easy = button_easy()
    Medium = button_medium()
    Hard = button_hard()
    Resume = button_resume()
    Exit = button_exit()
    Upgrade = button_upgrade()
    Upgrade_lv2 = button_upgrade_lv2()
    Upgrade_lv3 = button_upgrade_lv3()
    Main_menu = button_main_menu()
    Back = button_back()
    Try_again = button_try_again()
    all_sprites = Group(my_hammer)
    main_buttons = Group(Start, End)
    buttons = Group(Easy, Medium, Hard)
    esc_buttons = Group (Resume, Exit, Upgrade, Main_menu)
    upgrade_buttons = Group (Upgrade_lv2,Upgrade_lv3, Back)
    last_button = Group (Try_again)

first = True
while first:
    startbg = image.load ('start bg.png')
    screen.blit(startbg,(0,0))
    main_buttons.draw(screen)
    display.update()
    for ev in event.get():
        if ev.type == QUIT:
                pygame.quit()
                sys.exit()
        if ev.type == MOUSEBUTTONDOWN:
            if Start.rect.collidepoint(mouse.get_pos()):
                first = False
            elif End.rect.collidepoint(mouse.get_pos()):
                pygame.quit()
                sys.exit()

mainbg = image.load ('as.jpg')
screen.blit(mainbg,(0,0))
display.update()
buttons.draw(screen)
display.update()
global second
global third
second = True
third = True
game_running = False
antimer = pygame.time.get_ticks()
gamemode = None
gamepaused = False
groupedants = Group()

while second:
    for i in event.get():
        if i.type == QUIT:
                pygame.quit()
                sys.exit()
        if i.type == MOUSEBUTTONDOWN:
            if Easy.rect.collidepoint(mouse.get_pos()):
                game_running = True
                gamemode = 1
            elif Medium.rect.collidepoint(mouse.get_pos()):
                game_running=True
                gamemode = 2
            elif Hard.rect.collidepoint(mouse.get_pos()):
                game_running=True
                gamemode = 3
```

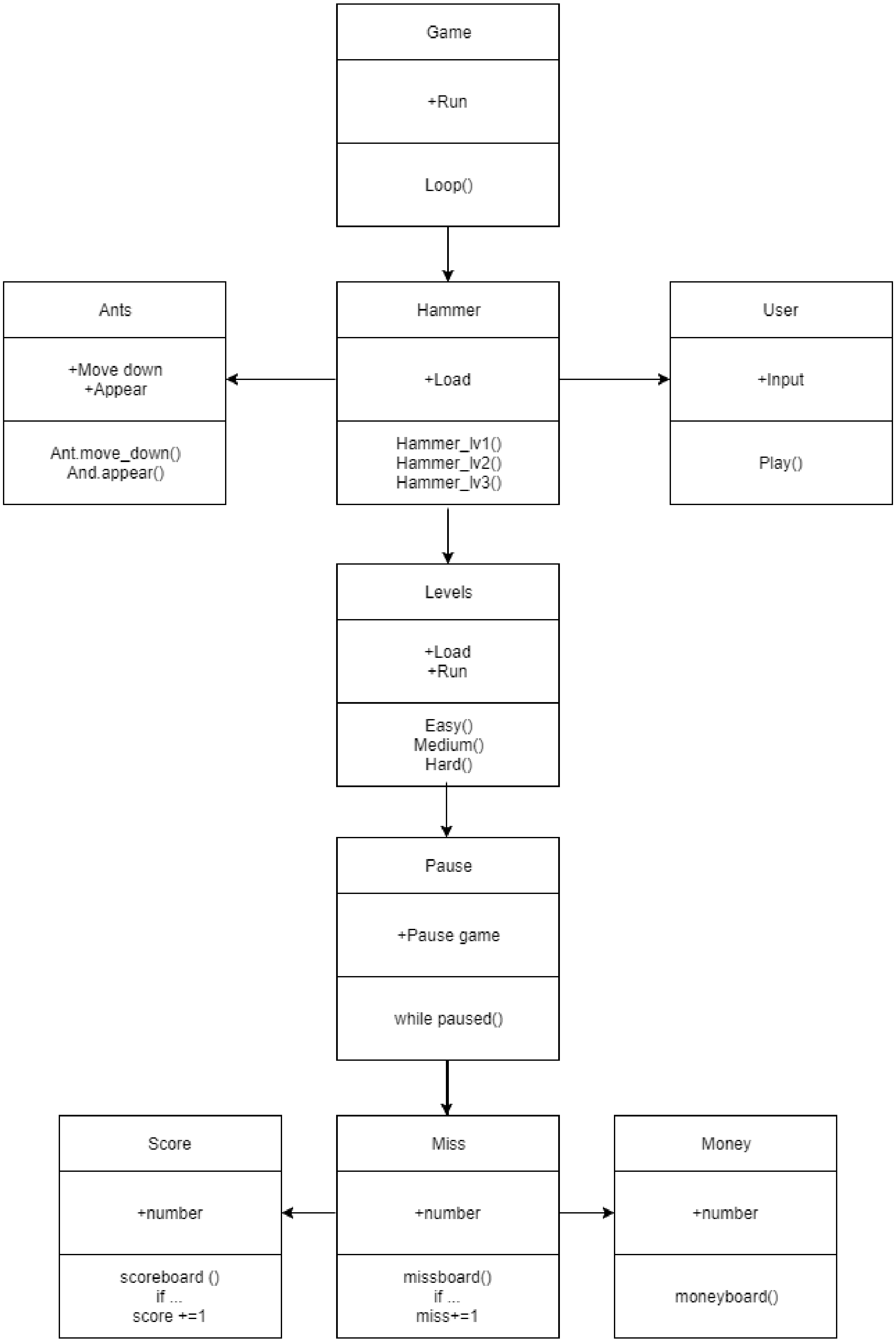


```
if game_running == True:
    function()
    if gamemode == 1:
        for ev in event.get():
            timetext()
            if ev.type == QUIT:
                pygame.quit()
                sys.exit()
            if ev.type == MOUSEBUTTONDOWN:
                for ant in groupedants:
                    if ant.rect.colliderect(my_hammer):
                        pygame.mixer.music.load("Slap-SoundMaster13-49669815.wav")
                        mixer.music.play()
                        killedants.add(dead_ants(ant.rect.center))
                        groupedants.remove(ant)
                        score += 1
                        score_for_money += 1
                        if score_for_money >= 10:
                            money += 1
                            score_for_money -= 10
            if ev.type == KEYDOWN:
                if ev.key == K_ESCAPE:
                    pausegame()
        if pygame.time.get_ticks() - antimer >= 1000:
            tempant = Ants() #temp=temporary
            groupedants.add(tempant)
            antimer = pygame.time.get_ticks()

        for ant in groupedants:
            ant.ant_move_down()
            if ant.rect.top == 800:
                miss += 1

if gamemode == 2:
    for ev in event.get():
        if ev.type == QUIT:
            pygame.quit()
            sys.exit()
        if ev.type == MOUSEBUTTONDOWN:
            for ant in groupedants:
                if ant.rect.colliderect(my_hammer):
                    pygame.mixer.music.load("Slap-SoundMaster13-49669815.wav")
                    mixer.music.play()
                    killedants.add(dead_ants(ant.rect.center))
                    groupedants.remove(ant)
                    score += 1
                    score_for_money += 1
                    if score_for_money >= 10:
                        money += 1
                        score_for_money -= 10
        if ev.type == KEYDOWN:
            if ev.key == K_ESCAPE:
                pausegame()
    if pygame.time.get_ticks() - antimer >= 500:
        tempant = Ants()
        groupedants.add(tempant)
        antimer = pygame.time.get_ticks()

    for ant in groupedants:
        ant.ant_move_down()
        if ant.rect.top == 800:
            miss += 1

if gamemode == 3:
    for ev in event.get():
        if ev.type == QUIT:
            pygame.quit()
            sys.exit()
        if ev.type == MOUSEBUTTONDOWN:
            for ant in groupedants:
                if ant.rect.colliderect(my_hammer):
                    pygame.mixer.music.load("Slap-SoundMaster13-49669815.wav")
                    mixer.music.play()
                    killedants.add(dead_ants(ant.rect.center))
                    groupedants.remove(ant)
                    score += 1
                    score_for_money += 1
                    if score_for_money >= 10:
                        money += 1
                        score_for_money -= 10
        if ev.type == KEYDOWN:
            if ev.key == K_ESCAPE:
                pausegame()

    if pygame.time.get_ticks() - antimer >= 100:
        tempant = Ants()
        groupedants.add(tempant)
        antimer = pygame.time.get_ticks()

    for ant in groupedants:
        ant.ant_move_down()
        if ant.rect.top == 800:
            miss += 1
if miss >= 10:
    second = False
mouse.set_visible(False)
```



```
while third:
    mouse.set_visible(True)
    gameovertext()
    last_button.draw(screen)
    display.update()
    for ev in event.get():
        if ev.type == QUIT:
            pygame.quit()
            sys.exit()
        if ev.type == MOUSEBUTTONDOWN:
            if Try_again.rect.collidepoint(mouse.get_pos()):
                third = False
mouse.set_visible(True)
```

UML Diagram


Game
+Run
Loop()
Ants
+Move down
+Appear
Ant.move_down()
And.appear()
Hammer
+Load
Hammer_lv1()
Hammer_lv2()
Hammer_lv3()
User
+Input
Play()
Levels
+Load
+Run
Easy()
Medium()
Hard()
Pause
+Pause game
while paused()
Score
+number
scoreboard ()
if ...
score +=1
Miss
+number
missboard()
if ...
miss+=1
Money
+number
moneyboard()

III.a. Lessons that Have Been Learned

In the first file that I made, this ant game was still very simple by the game play, the code also still simple including the classes, functions, and the main code. As I learned more in class, I was able to make a lot more improvements to it, such as the pause menu, which is exactly like buttons inside buttons fixed by loop inside a loop, None function to declare something later on from the user's action, screen blit to overlap the layer and I also implement the screen blit function to the game in order to make the background of an image appear so it is not boring just like a screen fill (RGB), set the caption of the program to ant by display set caption, after that comes the set the pixel by pixel which I choose to do 800 x 800 by the function display set mode. The timer also counts as a learning progress in class as I tried to input the code which is pygame.time.get_ticks() to my game so that it will make the game more interesting by spawning the ant in a specific amount of time. The code Group is also learned in class and is used to make the buttons or sprites to be grouped so that later on in the program, the developer will be easier to access the buttons that needs to be drawn on screen by the (buttons).draw code. The buttons also can be formed by using sprite in pygame. And the interface, in order for the user to be able to make their input to the game, we need to use event.get to get the input from the user. By this event.get, it will get what the user had inputed, it's either mouse click, as known as MOUSEBUTTONDOWN, or a keyboard type as known as KEYDOWN. To make the game more interesting for the user to interact, I decided to add images and musics that I have learned in class from the code image.load for the image and pygame.mixer.music.load continued with pygame.mixer.music.play by importing the code import pygame.mixer. by that said, there also is a function built in the python to make a ramdom position of the ant to be appeared which is it needed to use a import random function and later on, the random also needs to be called to make the appearance of the ant to be random by the x axis. By that said, the variables that are declared in the while loop also needs to be globaled to make the variable to be visible from the code above. After so may tries and fails with a lot of documents in all of the changes needed in the game window, we also need to put display.update() function in order to make the display of the game update.

III.b. Problems that Have Been Overcome

There is a lot of problems that have been overcome in this project, one of the biggest and major problem that myself found is the question ‘How to make this game even working?’ At first, I was able to fix the problem that the text editor provides which is the simple problem such as type error or indent error, but when it comes to the code itself, and the logic, It started to be a lot more challenging for me. Whenever I edit the code to develop it more, more and more errors came. At this point, I took a lot of time to think of the code if there is something undeclared, not imported, undefined. And by rechecking it, I was able to manage the game to work however I found the game not running smoothly as I wanted it to be and also the game has not met my expectation. So then I started to think that making a simple game like this clicking game is not easy for a beginner like me who does not know anything about coding and programming. Once I also found out that there is an error in the early stage of the development of this game, which is the code that does not work or does not exist, but I made it myself by looking at the previous codes that we did in class. As I discover more and more, I realize that the timer built in the game window was not ticking, as I test the program and decided to play, I also found out that there is a major problem with the program which is the gameplay including the hammer is not moving with the cursor, the ant does not move down, when the cursor collides with the hammer, the score updates and increases by one point, and the upgrade button in the pause menu does not work as well, for the upgrade menu, I figure the problem out by only adding another while loop inside a loop. But at the end, I discussed the all of the errors with my friends, It took us a lot of time to figure out the error in my code but when we discovered the location of the error, I immediately fix the error which is only a small mistake that I did which is an error of the placement, and the error disappeared and now I have a perfectly working game as I expected it to be.

V. Source Code

```
import random
import sys
import pygame.mixer
from pygame import *
from pygame.locals import *
from pygame.sprite import *
pygame.init()
pygame.mixer.music.load("Patakas World.wav")
pygame.mixer.music.play()
display.set_caption ('Ants')
screen = pygame.display.set_mode((800, 800))

class Ants(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.image = image.load('ant.png')
        self.rect = self.image.get_rect()
        randompos = random.randint(50,750)
        self.rect.center = (randompos,0)
    def Ants_appear (self):
        randompos = random.randint(50,750)
        self.rect.top = 0
        self.rect.left = randompos
    def ant_move_down(self):
        self.rect.top += 1

class Hammer(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.image = image.load('hammer-lv1.png')
        self.rect = self.image.get_rect()
        self.rect.center = mouse.get_pos()
    def hit (self, target):
        return self.rect.colliderect(target)
    def update(self):
        self.rect.center = mouse.get_pos()
    def upgrade1(self):
        self.image = image.load("hammer-lv2.png")
        self.rect = self.image.get_rect()
    def upgrade2(self):
        self.image = image.load("hammer-lv3.png")
        self.rect = self.image.get_rect()

class dead_ants(Sprite):
    def __init__(self,center):
        Sprite.__init__(self)
        self.image = image.load('dead ant.png')
        self.rect = self.image.get_rect()
        self.rect.center = center

class button_start(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,40)
        self.image = self.font.render ('Start',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)

class button_end(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Exit',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_easy(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Easy',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,200)

class button_medium(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Medium',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,400)

class button_hard(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hard',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_resume(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Resume',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,100)

class button_exit(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Exit',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,500)

class button_upgrade(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Upgrade',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)

class button_upgrade_lv2(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hammer lv.2 (3 Money)',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,300)

class button_upgrade_lv3(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Hammer lv.3 (6 Money)',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,600)

class button_main_menu(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Main Menu',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,700)

class button_back(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Back',1,(0,0,0))
        self.rect = self.image.get_rect()
        self.rect.center = (375,700)

class button_try_again(Sprite):
    def __init__(self):
        Sprite.__init__(self)
        self.font = pygame.font.Font(None,30)
        self.image = self.font.render ('Try Again',1,(255,255,255))
        self.rect = self.image.get_rect()
        self.rect.center = (370,700)

def function():
    screen.fill((130, 82, 1))
    my_hammer.update()
    killedants.draw(screen)
    groupedants.draw(screen)
    all_sprites.draw(screen)
    scoreboard()
    missboard()
    timetext()
    bestscore()
    display.update()

def scoreboard ():
    font = pygame.font.Font(None,20)
    score_text = font.render ('Score: %d'%score,1,((0,0,0)))
    screen.blit(score_text,(375,750))

def missboard ():
    font = pygame.font.Font(None,20)
    miss_text = font.render ('Miss: %d'%miss,1,((0,0,0)))
    screen.blit(miss_text,(700,750))

def timetext():
    time_spent = (pygame.time.get_ticks() - initial_time) / 1000
    font = pygame.font.Font(None,20)
    time_text = font.render ('Time: %d' %time_spent, 1,(0,0,0))
    screen.blit(time_text,(70,750))

def moneyboard():
    font = pygame.font.Font(None,20)
    time_text = font.render ('Money: %d' %money, 1,(0,0,0))
    screen.blit(time_text,(70,750))
```

```
def bestscore():
    global best_score
    if score >= best_score:
        best_score = score
    font = pygame.font.Font(None,20)
    time_text = font.render ('Best Score: %d' %best_score, 1,(0,0,0))
    screen.blit(time_text,(375,30))

def gameovertext():
    gamebg = image.load ('gameo.jpg')
    screen.blit(gamebg,(0,0))

def not_enough_money():
    font = pygame.font.Font(None,40)
    time_text = font.render ('Not Enough Money!', 1,(0,0,0))
    screen.blit(time_text,(300,60))

def upgraded():
    font = pygame.font.Font(None,40)
    time_text = font.render ('Upgraded!', 1,(0,0,0))
    screen.blit(time_text,(300,100))

def pausegame():
    mouse.set_visible(True)
    gamepaused = True
    while gamepaused == True:
        upgradehammer = False
        showupgrade_initime = -pygame.time.get_ticks()
        shownomoney = -pygame.time.get_ticks()
        pausebg = image.load ('pause.jpg')
        screen.blit(pausebg,(0,0))
        moneyboard()
        esc_buttons.draw(screen)
        display.update()
        for i in event.get():
            if i.type == QUIT:
                pygame.quit()
                sys.exit()
            if i.type == MOUSEBUTTONDOWN:
                if Resume.rect.collidepoint(mouse.get_pos()):
                    gamepaused = False
                if Exit.rect.collidepoint(mouse.get_pos()):
                    pygame.quit()
                    exit()
                if Upgrade.rect.collidepoint(mouse.get_pos()):
                    upgradehammer = True
                if Main_menu.rect.collidepoint(mouse.get_pos()):
                    global second
                    global third
                    second = False
                    third = False
                    gamepaused = False

                while upgradehammer == True:
                    global money
                    global my_hammer
                    gamepaused = True

                    for i in event.get():
                        if i.type == QUIT:
                            pygame.quit()
                            sys.exit()
                        if i.type == MOUSEBUTTONDOWN:
                            if Upgrade_lv2.rect.collidepoint(mouse.get_pos()):
                                if money >= 3:
                                    money -= 3
                                    my_hammer.upgrade1()
                                    showupgrade_initime = pygame.time.get_ticks()
                                    display.update()
                                elif money <=3:
                                    shownomoney = pygame.time.get_ticks()
                                    display.update()
                            elif Upgrade_lv3.rect.collidepoint(mouse.get_pos()):
                                if money >= 6:
                                    money -=6
                                    showupgrade_initime = pygame.time.get_ticks()
                                    my_hammer.upgrade2()
                                    display.update()
                                elif money <=6:
                                    shownomoney = pygame.time.get_ticks()
                                    display.update()
                            elif Back.rect.collidepoint(mouse.get_pos()):
                                upgradehammer = False
                        if i.type == KEYDOWN:
                            if i.key == K_ESCAPE:
                                gamepaused = False

                    upgradebg = image.load ('up.jpg')
                    screen.blit(upgradebg,(0,0))
                    if pygame.time.get_ticks() - showupgrade_initime <= 2000:
                        upgraded()
                    if pygame.time.get_ticks() - shownomoney <=2000:
                        not_enough_money()
                    moneyboard()
                    upgrade_buttons.draw(screen)
                    display.update()

            if i.type == KEYDOWN:
                if i.key == K_ESCAPE:
                    gamepaused = False

global best_score
best_score = 0
maingame = True
while maingame:
    global my_hammer
    global money
    score = 0
    miss = 0
    money = 0
    score_for_money = 0
    initial_time = pygame.time.get_ticks()
    my_hammer = Hammer()
    killedants = Group()
    Start = button_start()
    End = button_end()
    Easy = button_easy()
    Medium = button_medium()
    Hard = button_hard()
    Resume = button_resume()
    Exit = button_exit()
    Upgrade = button_upgrade()
    Upgrade_lv2 = button_upgrade_lv2()
    Upgrade_lv3 = button_upgrade_lv3()
    Main_menu = button_main_menu()
    Back = button_back()
```

```
Try_again = button_try_again()
all_sprites = Group(my_hammer)
main_buttons = Group(Start, End)
buttons = Group(Easy, Medium, Hard)
esc_buttons = Group (Resume, Exit, Upgrade, Main_menu)
upgrade_buttons = Group (Upgrade_lv2,Upgrade_lv3, Back)
last_button = Group (Try_again)

first = True
while first:
    startbg = image.load ('start bg.png')
    screen.blit(startbg,(0,0))
    main_buttons.draw(screen)
    display.update()
    for ev in event.get():
        if ev.type == QUIT:
                pygame.quit()
                sys.exit()
        if ev.type == MOUSEBUTTONDOWN:
            if Start.rect.collidepoint(mouse.get_pos()):
                first = False
            elif End.rect.collidepoint(mouse.get_pos()):
                pygame.quit()
                sys.exit()

mainbg = image.load ('as.jpg')
screen.blit(mainbg,(0,0))
display.update()
buttons.draw(screen)
display.update()
global second
global third
second = True
third = True
game_running = False
antimer = pygame.time.get_ticks()
gamemode = None
gamepaused = False
groupedants = Group()

while second:
    for i in event.get():
        if i.type == QUIT:
                pygame.quit()
                sys.exit()
        if i.type == MOUSEBUTTONDOWN:
            if Easy.rect.collidepoint(mouse.get_pos()):
                game_running = True
                gamemode = 1
            elif Medium.rect.collidepoint(mouse.get_pos()):
                game_running=True
                gamemode = 2
            elif Hard.rect.collidepoint(mouse.get_pos()):
                game_running=True
                gamemode = 3

    if game_running == True:
        function()
        if gamemode == 1:
            for ev in event.get():
                timetext()
                if ev.type == QUIT:
                    pygame.quit()
```

```
                        sys.exit()
                    if ev.type == MOUSEBUTTONDOWN:
                        for ant in groupedants:
                            if ant.rect.colliderect(my_hammer):
                                pygame.mixer.music.load("Slap-SoundMaster13-
49669815.wav")

                                mixer.music.play()
                                killedants.add(dead_ants(ant.rect.center))
                                groupedants.remove(ant)
                                score += 1
                                score_for_money += 1
                                if score_for_money >= 10:
                                    money += 1
                                    score_for_money -= 10
                    if ev.type == KEYDOWN:
                        if ev.key == K_ESCAPE:
                            pausegame()
                if pygame.time.get_ticks() - antimer >= 1000:
                    tempant = Ants()#temp=temporary
                    groupedants.add(tempant)
                    antimer = pygame.time.get_ticks()

                for ant in groupedants:
                    ant.ant_move_down()
                    if ant.rect.top == 800:
                        miss += 1

            if gamemode == 2:
                for ev in event.get():
                    if ev.type == QUIT:
                        pygame.quit()
                        sys.exit()
                    if ev.type == MOUSEBUTTONDOWN:
                        for ant in groupedants:
                            if ant.rect.colliderect(my_hammer):
                                pygame.mixer.music.load("Slap-SoundMaster13-
49669815.wav")

                                mixer.music.play()
                                killedants.add(dead_ants(ant.rect.center))
                                groupedants.remove(ant)
                                score += 1
                                score_for_money += 1
                                if score_for_money >= 10:
                                    money += 1
                                    score_for_money -= 10
                    if ev.type == KEYDOWN:
                        if ev.key == K_ESCAPE:
                            pausegame()
                if pygame.time.get_ticks() - antimer >= 500:
                    tempant = Ants()
                    groupedants.add(tempant)
                    antimer = pygame.time.get_ticks()

                for ant in groupedants:
                    ant.ant_move_down()
                    if ant.rect.top == 800:
                        miss += 1

            if gamemode == 3:
                for ev in event.get():
                    if ev.type == QUIT:
                        pygame.quit()
                        sys.exit()
                    if ev.type == MOUSEBUTTONDOWN:
                        for ant in groupedants:
                            if ant.rect.colliderect(my_hammer):
                                pygame.mixer.music.load("Slap-SoundMaster13-
49669815.wav")
                                mixer.music.play()
                                killedants.add(dead_ants(ant.rect.center))
                                groupedants.remove(ant)
                                score += 1
                                score_for_money += 1
                                if score_for_money >= 10:
                                    money += 1
                                    score_for_money -= 10
                    if ev.type == KEYDOWN:
                        if ev.key == K_ESCAPE:
                            pausegame()

                if pygame.time.get_ticks() - antimer >= 100:
                    tempant = Ants()
                    groupedants.add(tempant)
                    antimer = pygame.time.get_ticks()

                for ant in groupedants:
                    ant.ant_move_down()
                    if ant.rect.top == 800:
                        miss += 1
            if miss >= 10:
                second = False
            mouse.set_visible(False)
    while third:
        mouse.set_visible(True)
        gameovertext()
        last_button.draw(screen)
        display.update()
        for ev in event.get():
            if ev.type == QUIT:
                pygame.quit()
                sys.exit()
            if ev.type == MOUSEBUTTONDOWN:
                if Try_again.rect.collidepoint(mouse.get_pos()):
                    third = False
    mouse.set_visible(True)
```