Odd Semester (2020)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |
| **Student Information**: **Surname** | | | | | **Given Names**  **Raharja** | | **Student ID Number**  **2101731351** | |
| 30. | | **William** |  | |
|  |  |
| **Course Code** | **: COMP6056** |  |  | | **Course Name** | | **: Introduction to Programming** | |
| **Class** | **: L1AC** |  |  | | **Name of Lecturer(s)** | | **:** 1. Bagus Kerthyayana | |
|  |  |  |  | |  | | 2. Tri Asih Budiono | |
| **Major** | **: CS** |  |  | |  | |  | |
| **Title of Assignment**  (if any) | : **Hit the Fuhrer** | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  | |  | |  | |
| **Due Date** | **: 06.11.2017** |  |  | | **Submission Date** | | **: 05.11.2017** | |

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. 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:

William Raharja

**“Hit the Führer”**

**Name :William Raharja**

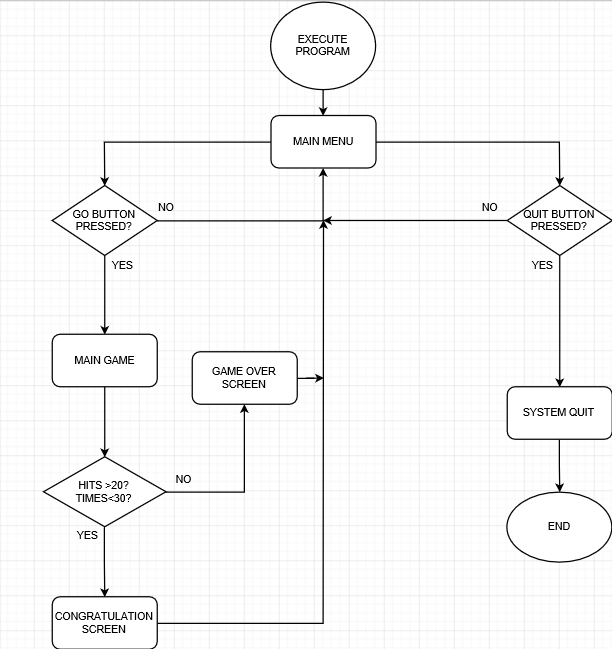
**ID :2101731351**

1. **Description**

This program is a “Whack a Mole” game concept that is integrated with the theme alternate history. It was made for educational purposes, specifically in Pygame. The objective of this game is to shoot the Führer at least 21 times in 30 seconds or less in order to beat the game itself. It also has an installer to convert the game from python file into executable application.

**II.a. Design/Plan**

**Project’s Flow Chart**

****

**II.b. Explanation of Each function**

**Class Hitler:**

This class is used to describe what Hitler will do in the game, which consist of three functions, namely:

1. **Def \_\_init\_\_ ():**

* This function role is to initialize the class Hitler as a mole
* Declaring the variable of self.image so that we can call the picture inside the game
* Declaring the variable self.rect to store and manipulate the rectangular window’s area

1. **Def flee ():**

* This function uses random integer of the x and y axis so that Hitler could move to another coordinate.

1. **Def draw():**

* This function plays an important role so that Hitler’s picture can emerged in the game screen (to be exact, on the background)

**Class Gun:**

The class Gun plays an important role as the cursor or mouse pointer in the game. It consist of four functions :

1. **Def \_\_init\_\_ ():**

* This function role is to initialize the class Gun as shovel
* Declaring the variable of self.image so that we can call the picture inside the game
* Declaring the variable self.rect to store and manipulate the rectangular window’s area

1. **Def hit ():**

* This function determines whether the mouse pointer hit Hitler or not
* If one hit Hitler by using the mouse pointer, one will score a point
* This function will also change the mouse pointer from Animation 1 to Animation 2 (when shots are fired)

1. **Def update():**

* This function update the mouse cursor back into Animation1 (normal pistol when shot is not fired)

1. **Def draw():**

* This function plays an important role so that the gun is visible in the game window

**Functions Outside of the Class**

1. **Def center\_Image(screen, im):**

This function plays an important role so that the “Hits” table will surface in the game screen.

1. **Def center\_screen(screen, im):**

This function will make any message emerge at the center of the screen.

1. **Def text\_object(text, font):**

This function declare the variable text surface as the font.render function so that one can select which font one wanted to use.

1. **Def message\_display(text)**

This function will display any text that is typed inside the colon or () with the font “Free Sans Bold”. This function also plays an important role to display the “Game Over” screen and “Congratulations” screen.

1. **Def button(msg, x, y, w, h, ic, ac, action=None):**

This is a universal button function which role is very important for the front menu screen. It defines the button size, position, color, the color when the mouse cursor hover above the button.

1. **Def introduction():**

This function’s role is to display the front menu screen and to let players choose whether they want to play the game or quit from the game.

1. **Def main():**

This function is the main concept of the game. It describes the process of the game when it is executed by the player. As one click the “Go” Button, the game will redirect one to the game. Inside the game, the mouse cursor will be replaced with the picture “Animated\_Pistol.png”. Once the player fire his or her shot, the Fuhrer will start to move into another coordinates. The player has 30 seconds to shot the Fuhrer more than 20 times in order to win the game. If one doesn’t shot the Fuhrer within the time limit for more than 20 times, one will lose the game which will be directed into the “Game Over” screen. Otherwise, one will be directed to the “Congratulation” screen.

1. **installer.py**

This file plays a very important role as the installer to convert python file into executable application. It consist of files that are involved in the game (e.g image, program, and sound)

1. **Source Code**
2. *'''  
   Whack\_A\_Mole.py  
    A Shooting Simulator with Whack-A-Mole logic  
   Created by Excelino.Fernando/excelincode  
    & William.Raharja/wilraharja  
    in 21.10.2017  
   Version 0.1a 21.10.2017  
    Framework and Basic Logic  
   Improved by William.Raharja/willraharja  
    in 04.11.2017  
    Version 0.9  
   Special Thanks to Excelino, Adrian Pratama, and Georgius Kurli  
   '''***import** pygame, random, time  
   **from** pygame.locals **import** \*  
   **from** pygame.font **import** \*  
     
   background\_image = **'Wolfenstein.jpg'***#RGB colours combination*BLACK = (0, 0, 0)  
   WHITE = (255, 255, 255)  
   DARKGRAY = (47, 79, 79)  
   BRIGHTRED = (255, 0, 0)  
   RED = (200, 0, 0)  
   GREEN = (0, 200, 0)  
   BRIGHTGREEN = (0, 255, 0)  
   DARKGREEN = (0, 155, 0)  
   BLUE = (0, 0, 255)  
     
   scrWidth = 800  
   scrHeight = 600  
     
   screen = pygame.display.set\_mode((scrWidth, scrHeight), HWSURFACE)  
     
   *# ---------------------------------------------------------***class** Hitler(pygame.sprite.Sprite):  
    **def** \_\_init\_\_(self):  
    super().\_\_init\_\_()  
    self.image = pygame.image.load(**"adolf.png"**).convert\_alpha()  
    self.rect = self.image.get\_rect()  
     
    *# move Hitler to a new random location when it gets hit* **def** flee(self):  
    x = random.randint(0, scrWidth-1-self.rect.width)  
    y = random.randint(0, scrHeight-1-self.rect.height)  
    self.rect.topleft = (x,y)  
     
    **def** draw(self, screen):  
    screen.blit(self.image, self.rect)  
     
   *# ---------------------------------------------------------***class** Gun(pygame.sprite.Sprite):  
     
    **def** \_\_init\_\_(self):  
    super().\_\_init\_\_()  
    self.image = pygame.image.load(**"Animated\_Pistol1.png"**)  
    self.rect = self.image.get\_rect()  
     
     
    *# did the shovel hit the mole?* **def** hit(self, target):  
    self.image = pygame.image.load(**"Animated\_Pistol2.png"**)  
    **return** self.rect.colliderect(target)  
     
    *# follows the mouse cursor* **def** update(self, pt):  
    self.image = pygame.image.load(**"Animated\_Pistol1.png"**)  
    self.rect.center = pt  
     
    **def** draw(self, screen):  
    screen.blit(self.image, self.rect)  
     
   *# -----------------------------------***def** center\_Image(screen, im):  
    x = (scrWidth - im.get\_width() - 10)  
    y = (scrHeight - im.get\_height())  
    screen.blit(im, (x, 0))  
     
   **def** center\_Screen(screen, im):  
    x = (scrWidth/4)  
    y = (scrHeight/2)  
    screen.blit(im, (x, y))  
     
     
   *#-----------------------------------  
     
   #variables*pygame.init()  
   screen = pygame.display.set\_mode([655,365])  
   screen.fill(BLACK)  
   pygame.display.set\_caption(**"Shoot-the-Fuhrer"**)  
   background = pygame.image.load(background\_image).convert()  
   scrWidth, scrHeight = screen.get\_size()  
   bigFont = pygame.font.Font(**None**, 50)  
     
     
     
   *#hide the mouse curs or*font = pygame.font.Font(**None**, 40)  
     
   hitSnd = pygame.mixer.Sound(**'Gunfire.wav'**)  
   hitSnd.set\_volume(1)  
     
   *# create sprites and a group*mole = Hitler()  
   shovel = Gun()  
     
   *# game variables*mousePos = (scrWidth/2, scrHeight/2)  
   DELAY = 800  
   clock = pygame.time.Clock()  
     
   **def** text\_objects(text, font):  
    textSurface = font.render(text, **True**, BLACK)  
    **return** textSurface, textSurface.get\_rect()  
     
   **def** message\_display(text):  
    largeText = pygame.font.Font(**'freesansbold.ttf'**,115)  
    TextSurf, TextRect = text\_objects(text, largeText)  
    TextRect.center = ((scrWidth/2),(scrHeight/2))  
    screen.blit(TextSurf, TextRect)  
     
    pygame.display.update()  
     
     
     
   *#Universal\_Button***def** button(msg,x,y,w,h,ic,ac,action=**None**):  
     
    mouse = pygame.mouse.get\_pos()  
    click = pygame.mouse.get\_pressed()  
    print(click)  
    **if** x+w > mouse[0] > x **and** y+h > mouse[1] > y:  
    pygame.draw.rect(screen, ac,(x,y,w,h))  
     
    **if** click[0] == 1 **and** action != **None**:  
    action()  
    **else**:  
    pygame.draw.rect(screen, ic,(x,y,w,h))  
     
    smallText = pygame.font.SysFont(**None**, 20)  
    textSurf, textRect = text\_objects(msg, smallText)  
    textRect.center = ( (x+(w/2)), (y+(h/2)) )  
    screen.blit(textSurf, textRect)  
     
     
   *#menu UI***def** introduction():  
    pygame.mouse.set\_visible(**True**)  
    intro = **True  
     
    while** intro:  
     
    **for** event **in** pygame.event.get():  
    *#print(event)* **if** event.type == pygame.QUIT:  
    pygame.quit()  
    quit()  
     
    screen.fill(WHITE)  
    largeText = pygame.font.Font(**"LemonMilk.otf"**,55)  
    TextSurf, TextRect = text\_objects(**"Shoot the Fuhrer"**, largeText)  
    TextRect.center = ((scrWidth/2), (scrHeight/2))  
    screen.blit(TextSurf, TextRect)  
     
    button(**"GO!"**,100,250,100,50,GREEN,BRIGHTGREEN,main)  
    button(**"Quit"**,400,250,100,50,RED,BRIGHTRED,quit)  
     
    pygame.display.update()  
    clock.tick(100)  
     
     
   *#in game concept***def** main():  
    init\_time = pygame.time.get\_ticks()  
    running = **True** score = 0  
    *#global variable to declare mousePos inside the function* **while** running:  
    *#hide the mouse cursor* pygame.mouse.set\_visible(**False**)  
     
    *# handle events* **for** event **in** pygame.event.get():  
    **if** event.type == QUIT:  
    running = **False  
    if** event.type == KEYDOWN:  
    **if** event.key == K\_ESCAPE:  
    running = **False  
    if** event.type == MOUSEMOTION:  
    mousePos = pygame.mouse.get\_pos()  
    **if** event.type == MOUSEBUTTONDOWN:  
    running = **True** *# update game* shovel.update(pygame.mouse.get\_pos())  
    ev = pygame.event.wait()  
    **if** ev.type == QUIT:  
    pygame.quit()  
    **break  
    elif** ev.type == MOUSEBUTTONDOWN:  
    **if** shovel.hit(mole):  
    hitSnd.play()  
    mole.flee()  
    score += 1  
    pygame.time.set\_timer(USEREVENT + 1, DELAY)  
    **else**:  
    hitSnd.play()  
    **elif** ev.type == USEREVENT + 1:  
    mole.flee()  
    *#time limit, approximately 30 sec* **if** (pygame.time.get\_ticks() - init\_time) >= 30000:  
    temp = pygame.time.get\_ticks()  
    **if** score < 20:  
    **while** pygame.time.get\_ticks() - temp <= 5000: *#wait 5000 ms == 5 sec* screen.fill(BLACK)  
    lose = bigFont.render(**"PLEASE TRY AGAIN"**, **True**, WHITE)  
    center\_Screen(screen, lose)  
    pygame.display.update()  
    **else**:  
    print(**"0: Alas, You Lost!"**)  
    introduction()  
    **else**:  
     
    **while** pygame.time.get\_ticks()-temp <= 5000: *#wait 5000ms* screen.fill(WHITE)  
    win = bigFont.render(**"YOU WIN!"**, **True**, BLACK)  
    center\_Screen(screen, win)  
    pygame.display.update()  
    **else**:  
    print(**"0: You Win!"**)  
    introduction()  
    *# time elapsed (in secs)  
   # -------------------------------------------------  
    # redraw game* temp\_duration = int((pygame.time.get\_ticks() - init\_time)/1000)  
    screen.blit(background, (0, 0))  
    mole.draw(screen)  
    shovel.draw(screen)  
    timeIm = font.render(str(temp\_duration), **True**, WHITE)  
    hitIm = font.render(**"Hits = "** + str(score), **True**, WHITE)  
    center\_Image(screen, hitIm)  
    screen.blit(timeIm, (10,10))  
    pygame.display.update()  
    clock.tick(80)  
     
   introduction()  
   pygame.quit()  
   quit()

**IV REFERENCES**

<https://www.youtube.com/user/sentdex>

<http://www.pygame.org/docs/>

https://pythonprogramming.net/pygame-button-function-events/

Special thanks to Adrian Pratama, Excelino, and Georgius Kurli