BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter
(Individual/Group* Work)

Student Information: 1.

Surname: Divian

Given Names: Rendy

Student ID Number : 2440096755

Course Code : Course Name : COMP6056

Class : L1AC

Name of Lecturer(s) : Jude Joseph Lamug Martinez

Major : Computer Science

Title of Assignment : Dodge Game
(if any)

Type of Assignment : Final Project

Submission Pattern :

Due Date : 13/01/2021

Submission Date : 13/01/2021

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/we* understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I/we* declare that the work contained in this assignment is my/our* 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. Rendy Divian

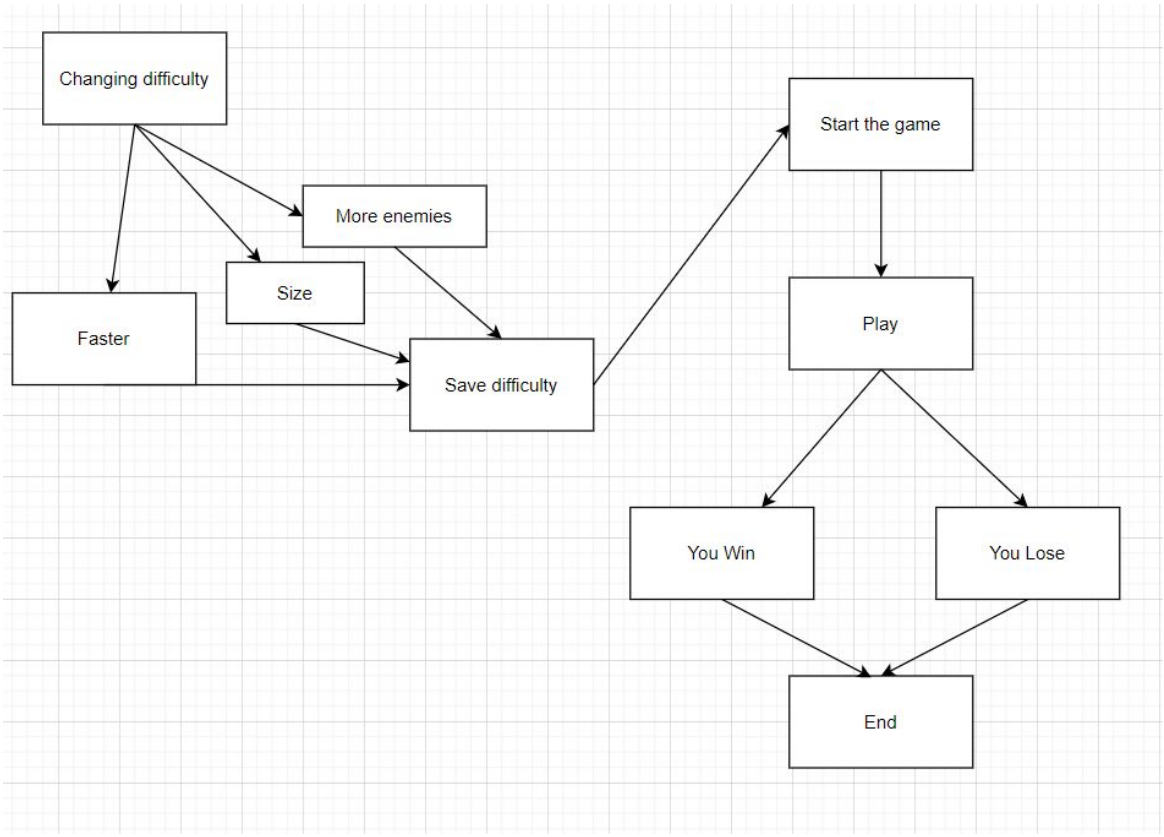
etc. *) Delete the inappropriate option

“Dodge Game”
Name: Rendy Divian
ID: 2440096755

Project Specification

This is a python based application to let people have some fun and be able to challenge themself when they have some free time, or just feeling bored. The theme of the game is dodging. They would have to go from the starting square to the ending square while evading a few fireballs or more depending on how much is added.

Implementation



Users can start the game and play the game right away but if they want it to be more difficult, they will have to change or add the difficulty in the program. Once play, they either get hit by the fireball and lose or touch the ending square and win the game where the game will end and close.

Game Appearance

Start of the game

Win screen

Lose screen

How the game works

```
import pygame, sys, random, time, math
from pygame.locals import*

pygame.init()

#Colors
BLUE = (0, 0, 255)
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
YELLOW = (255, 255, 0)
GRAY = (128, 128, 128)

screenx = 1000
screeny = 600

windowSurface = pygame.display.set_mode((screenx, screeny), 0, 0)

#The pictures used (person and green squares)
person1 = pygame.image.load("mc.bmp")
personpic = pygame.transform.scale(person1, (40, 50))
personpic.convert()

startpoint = pygame.image.load("starting.bmp")
greenImage1 = pygame.transform.scale(startpoint, (100, 100))
greenImage1.convert()

endpoint = pygame.image.load("ending.bmp")
greenImage2 = pygame.transform.scale(endpoint, (70, 70))
greenImage2.convert()
```

Some color codes that could be used to change background, text, border and transparency for the images used. The screen is set up as width is 600 and length 1000. The images were converted to bmp because it was said that it is a lot better to use bmp for pygames. The convert() makes it more smooth for pixels to appear. Also, you can change the size of the person and squares.

```
# Sets the position of the green squares
# Position in x and y coordinates (x=left, right and y=up, down)
# Starting square
greenRectA = greenImage1.get_rect()
greenRectA.centerx = 50
greenRectA.centery = 550

# Finishing square
greenRectB = greenImage2.get_rect()
greenRectB.centerx = 950
greenRectB.centery = 50

fireballpic = pygame.image.load("fireballicon.bmp")
fireball = pygame.transform.scale(fireballpic, (60, 60))
fireball.set_colorkey(WHITE) #To make the fireball background transparent
fireball.convert()

class Aperson():
    def __init__(self, Image):
        self.speed = 5
        self.directionx = 0
        self.directiony = 0
        self.image = Image
        self.rect = self.image.get_rect()

    def move(self):
        self.rect.centerx = self.rect.centerx + self.speed * self.directionx
        self.rect.centery = self.rect.centery + self.speed * self.directiony
```

This part is mainly where most things are positioned using rect. The starting square is set to the bottom left and the end square at top right. Character is given movement. The fireball like the above image can change size and colorkey to erase the white background.

```
hero = Aperson(personpic)
#Sets the position of the player
hero.rect.centerx = greenRectA.centerx
hero.rect.centery = greenRectA.centery
#Sets the speed of the player
hero.speed = 7

# Sets number of enemies
numberfireball = 3
enemy = [Aperson(fireball) for i in range(0, numberfireball)]
#Centerx and centery is the position
#Directionx and directiony show which way the enemy is moving
#Directionx=1 is right, -1 is left, directiony=1 is down, -1 is up
#One enemy
enemy[0].rect.centerx = 500
enemy[0].rect.centery = 0
enemy[0].directionx = 0
enemy[0].directiony = -1
enemy[0].speed = 10
#Second enemy
enemy[1].rect.centerx = 0
enemy[1].rect.centery = 350
enemy[1].directionx = 1
enemy[1].directiony = 0
enemy[1].speed = 6
#Third Enemy
enemy[2].rect.centerx = 300
enemy[2].rect.centery = 250
enemy[2].directionx = 0
enemy[2].directiony = 1
enemy[2].speed = 6
# Change number up above and add another enemy section to create more enemies
```

The person is set to the starting square and speed can be changed. Number of fireballs can be increased or decreased but you will have to delete or add more enemies. Also, the enemy position when they spawn and their speed can be changed.

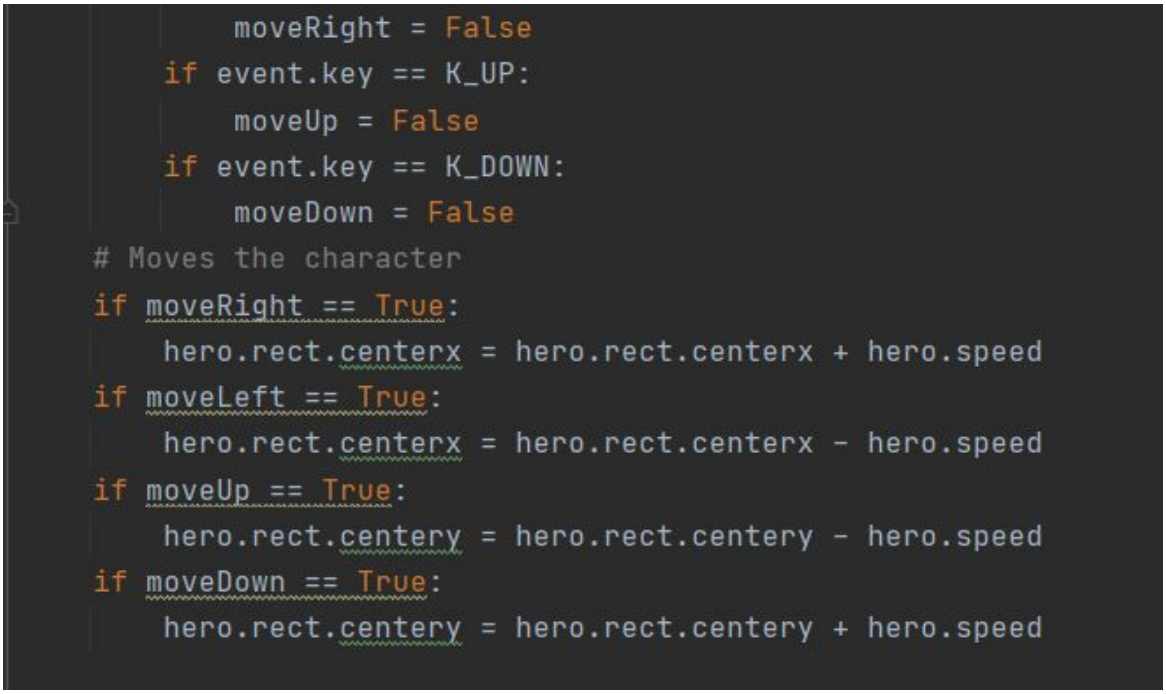
```
pygame.display.set_caption('Dodge Game')

moveRight = False
moveLeft = False
moveUp = False
moveDown = False

gameOver = False

pygame.display.update()

while gameOver == False:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    # Checks when the arrow keys or escape are pressed
    if event.type == KEYDOWN:
        if event.key == K_ESCAPE:
            gameOver = True
        if event.key == K_LEFT:
            moveLeft = True
        if event.key == K_RIGHT:
            moveRight = True
        if event.key == K_UP:
            moveUp = True
        if event.key == K_DOWN:
            moveDown = True
    # Checks when they are released
    if event.type == KEYUP:
        if event.key == K_LEFT:
            moveLeft = False
```



```
            moveRight = False
        if event.key == K_UP:
            moveUp = False
        if event.key == K_DOWN:
            moveDown = False
    # Moves the character
    if moveRight == True:
        hero.rect.centerx = hero.rect.centerx + hero.speed
    if moveLeft == True:
        hero.rect.centerx = hero.rect.centerx - hero.speed
    if moveUp == True:
        hero.rect.centery = hero.rect.centery - hero.speed
    if moveDown == True:
        hero.rect.centery = hero.rect.centery + hero.speed
```

These two pictures are movement keys for the character. The top is just a display for the game name when started.

```
    #Blocks the player from going off the screen
    if hero.rect.centerx < 0:
        hero.rect.centerx = 0
    if hero.rect.centerx > screenx:
        hero.rect.centerx = screenx
    if hero.rect.centery < 0:
        hero.rect.centery = 0
    if hero.rect.centery > screeny:
        hero.rect.centery = screeny

    windowSurface.fill(GRAY)
    #Blocks the enemy
    windowSurface.blit(greenImage1, greenRectA)
    windowSurface.blit(greenImage2, greenRectB)
    windowSurface.blit(hero.image, hero.rect)
    for i in range(0, numberfireball):
        windowSurface.blit(enemy[i].image, enemy[i].rect)
        enemy[i].move()
        if enemy[i].rect.centery < 0:
            enemy[i].directiony = 1
        if enemy[i].rect.centery > screeny:
            enemy[i].directiony = -1
        if enemy[i].rect.centerx < 0:
            enemy[i].directionx = 1
        if enemy[i].rect.centerx > screenx:
            enemy[i].directionx = -1
```

In here the player will bounce back if added and for the enemies they will loop around the path they are moving.

```
        #Checks when you collide with an enemy and says that you lost
        if enemy[i].rect.colliderect(hero.rect):
            bigFont = pygame.font.SysFont('arial', 60)
            myMessage = bigFont.render('You lost!', True, RED, BLACK)
            myMessageRect = myMessage.get_rect()
            myMessageRect.centerx = screenx / 2
            myMessageRect.centery = screeny / 2
            windowSurface.blit(myMessage, myMessageRect)
            pygame.display.update()
            time.sleep(1.5)
            gameOver = True

    #Checks when you collide with the ending square and says that you won
    if hero.rect.colliderect(greenRectB):
        bigFont = pygame.font.SysFont('arial', 60)
        myMessage = bigFont.render('You won!', True, YELLOW, BLACK)
        myMessage.set_colorkey(BLACK)
        myMessageRect = myMessage.get_rect()
        myMessageRect.centerx = screenx / 2
        myMessageRect.centery = screeny / 2
        windowSurface.blit(myMessage, myMessageRect)
        pygame.display.update()
        time.sleep(1.5)
        gameOver = True

    pygame.display.update()
    time.sleep(0.02)

pygame.quit()
```

Just the text saying you won or you lost, they are positioned to the middle from the division by two. And there will be a short delay before the game quits.

Project Link

https://github.com/RendyDivian/Final-Project

Source and Reference

- Fireball - https://pngimg.com/download/63945
- Person - https://dlpng.com/png/1294687
- Pygame tutorials - https://www.youtube.com/watch?v=1aGuhUFwvXA&list=PLzMcBGfZo4-lp3jAExUCewBfMx3UZFkh5&index=7

- Pygame documentation - https://www.pygame.org/docs/ref/rect.html
- https://htmlcolorcodes.com/