**Mục lục**

[Phần 1: Giới thiệu bài toán 2](#_Toc103254591)

[Phần 2: Thuật toán/Method 2](#_Toc103254592)

[2.1. Cấu trúc thư mục của đồ án và class Button 2](#_Toc103254593)

[2.1.1. Cấu trúc thư mục của đồ án 2](#_Toc103254594)

[2.1.2. Giải thích class Button 2](#_Toc103254595)

[2.2. Giải thích file MainGame.py 4](#_Toc103254596)

[2.2.1. Giải thích các biến: 4](#_Toc103254597)

[2.2.2. Các class, funtion quan trọng 6](#_Toc103254598)

[2.2.3. Giải thích các class, funtion khác 15](#_Toc103254599)

[Phần 3: Đánh giá 17](#_Toc103254600)

[Phần 4: Demo 17](#_Toc103254601)

[4.1. Trình bày hình ảnh khi bắt đầu trò chơi 17](#_Toc103254602)

[4.2. Trình bày hình ảnh khi trò chơi đang chạy 18](#_Toc103254603)

[4.3. Trình bày hình ảnh khi trò chơi kết thúc 18](#_Toc103254604)

# Phần 1: Giới thiệu bài toán

Hiện nay, ngành phát triển trò chơi điện tử đang phát triển nhanh, trở nên xu hướng trong xã hội. Các trò chơi điện tử đang đem lại một doanh thu lớn cho các công ty, cá nhân.

Lĩnh vực phát triển trò chơi điện tử đang tạo ra nhiều trò chơi hay, hấp dẫn. Chúng giúp cho người chơi giải trí sau những giờ làm việc căng thẳng, tạo động lực cho lần làm việc tiếp theo.

Trong tình hình đó, nhóm em quyết định xây dựng trò chơi rắn săn mồi. Trò chơi được viết bằng ngôn ngữ Python và chủ yếu sử dụng thư viện pygame. Trò chơi có cách chơi đơn giản: sử dụng bàn phím để điều khiển con rắn và ăn những trái táo để tăng điểm số, có chức năng lưu lại điểm cao mỗi khi người chơi đạt được một điểm cao mới.

# Phần 2: Thuật toán/Method

## 2.1. Cấu trúc thư mục của đồ án và class Button

### 2.1.1. Cấu trúc thư mục của đồ án

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 1: Cấu trúc thư mục của đồ án

- Cây thư mục đồ án gồm:

+ Thư mục images gồm các hình ảnh được sử dụng.

+ File Button.py để định nghĩa class button.

+ File MainGame.py gồm các class, funtion quan trọng.

+ File score.dat để lưu điểm cao nhất của người chơi.

### 2.1.2. Giải thích class Button

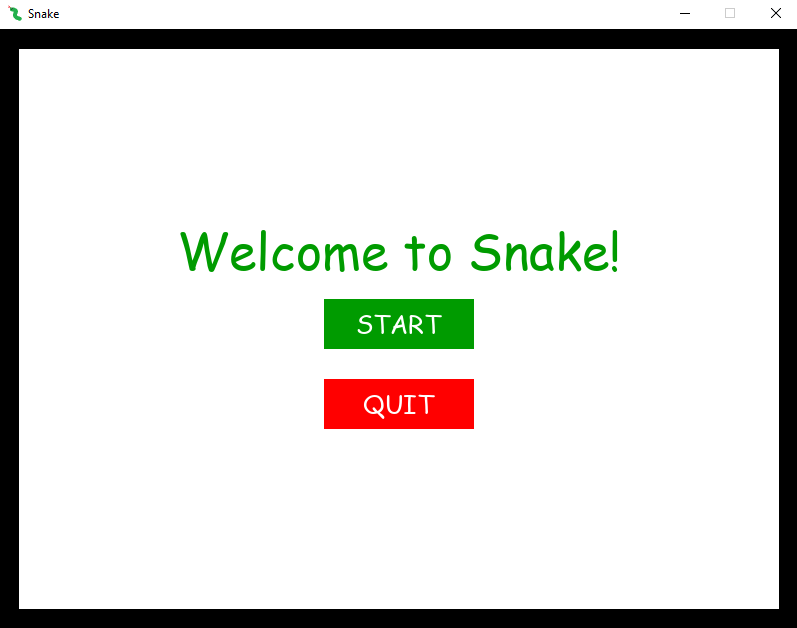
- Dùng để định nghĩa button để sử dụng trong màn hình trò chơi, bao gồm:

+ Hàm init để khởi tạo button.

+ Hàm displayText để chèn chữ vào button.

+ Hàm showButton để hiển thị button lên màn hình.

+ Hàm isHovered để đổi màu button khi di chuyển chuột vào button.



Hình 2: Các button

class button(*object*):

    def \_\_init\_\_(*self*, *colour*, *hoverColour*, *display*, *text*, *left*, *top*, *width*, *height*, *textColour*, *offset*, *centerWidth*,

*centerHeight*, *font*):

*self*.colour = colour

*self*.hoverColour = hoverColour

*self*.display = display

*self*.text = text

*self*.top = top

*self*.left = left

*self*.width = width

*self*.height = height

*self*.textColour = textColour

*self*.offset = offset

*self*.centerWidth = centerWidth

*self*.centerHeight = centerHeight

*self*.font = font

    def displayText(*self*):

        displayText = *self*.font.render(*self*.text, True, *self*.textColour)

*self*.display.blit(displayText, [*self*.centerWidth - (displayText.get\_rect().width / 2), *self*.centerHeight + (*self*.height / 2) - (displayText.get\_rect().height / 2) + *self*.offset])

    def showButton(*self*):

*self*.display.fill(*self*.colour, (*self*.left, *self*.top, *self*.width, *self*.height))

*self*.displayText()

    def isHovered(*self*, *cursor*):

        if *self*.left < cursor[0] < *self*.left + *self*.width and *self*.top < cursor[1] < *self*.top + *self*.height:

*self*.display.fill(*self*.hoverColour, (*self*.left, *self*.top, *self*.width, *self*.height))

*self*.displayText()

            return True

## 2.2. Giải thích file MainGame.py

- Dùng để tạo đầy đủ màn hình khi mới mở trò chơi, khi đang chơi và khi kết thúc trò chơi.

### 2.2.1. Giải thích các biến:

- Định nghĩa các màu để dùng

# Color definitions

white = (255, 255, 255)

black = (0, 0, 0)

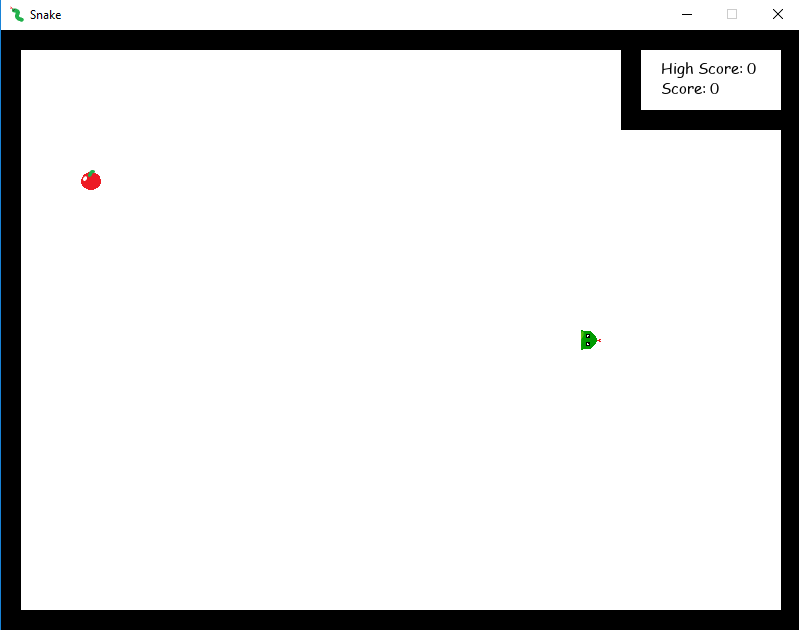
red = (255, 0, 0)

lightRed = (249, 52, 52)

green = (0, 155, 0)

lightGreen = (74, 196, 74)

- Định nghĩa property (thuộc tính) constants (hằng số) của trò chơi gồm: kích thước của màn hình, kích thước của ô hiển thị điểm số, kích thước để hiển thị thân hình con rắn.



Hình 3: Khởi tạo các biến của trò chơi

# Game property constants

blockSize = 20

displayWidth = 800

displayHeight = 600

centerDisplayWidth = displayWidth / 2

centerDisplayHeight = displayHeight / 2

boundX = displayWidth - (blockSize \* 2)

boundY = displayHeight - (blockSize \* 2)

scoreOffsetX = 140

scoreOffsetY = 27

scoreBoundWidth = displayWidth - 180

scoreBoundHeight = 100 – blockSize

FPS = 13

- Định nghĩa variables (các biến) của trò chơi:

+ Đặc biệt là biến degrees để xử lý góc quay của đầu con rắn (với góc quay bằng 0 là góc quay hướng xuống phía dưới trọng lực; góc quay của đầu con rắn khi bắt đầu chơi là góc quay bằng 270).

# Game variables

degrees = 270

randAppleX, randAppleY = (0,) \* 2

goldenApple = random.randint(1, 10) == 10

leadX = centerDisplayWidth

leadY = centerDisplayHeight

leadXChange = blockSize

leadYChange = 0

appleCounter = 0

highScore = 0

buttonWidth = 150

buttonHeight = 50

snakeList = []

+ Chèn font chữ

# Importing font

bodyFont = pygame.font.SysFont("comicsansms", 50)

buttonFont = pygame.font.SysFont("comicsansms", 25)

+ Chèn các ảnh để sử dụng

# Importing images

snakeHeadImage = pygame.image.load("images/SnakeHead.png")

snakeBodyImage = pygame.image.load("images/SnakeBody.png")

appleImage = pygame.image.load("images/Apple.png")

goldenAppleImage = pygame.image.load("images/GoldenApple.png")

icon = pygame.image.load("images/Icon.png")

+ Thiết lập thuộc tính của màn hình

# Configuring display

gameDisplay = pygame.display.set\_mode((displayWidth, displayHeight))

pygame.display.set\_caption("Snake")

pygame.display.set\_icon(icon)

clock = pygame.time.Clock()

### 2.2.2. Các class, funtion quan trọng

- Mở file score.dat để đọc và ghi điểm cao nhất

# High score loading

try:

    with open('score.dat', 'rb') as file:

        highScore = pickle.load(file)

except:

    highScore = 0

    with open('score.dat', 'wb') as file:

        pickle.dump(highScore, file)

- Hiển thị điểm và điểm cao nhất

def showScores(*score*, *new*):

    screen\_text = pygame.font.SysFont("comicsansms", 15).render("Score: " + *str*(score), True, black)

    gameDisplay.blit(screen\_text, (displayWidth - scoreOffsetX, scoreOffsetY + 20))

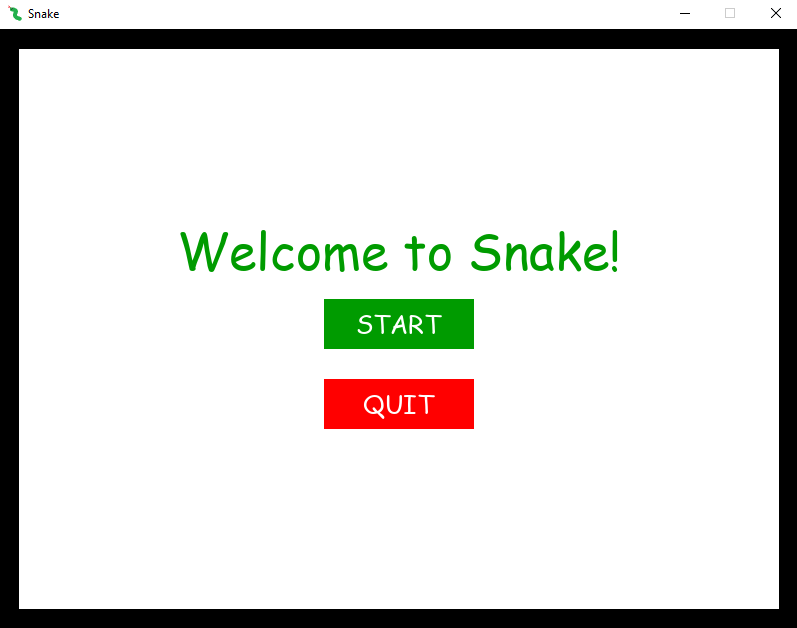
    high\_score = pygame.font.SysFont("comicsansms", 15).render("High Score: " + *str*(highScore), True, black)

    if new:

        high\_score = pygame.font.SysFont("comicsansms", 13).render("New High Score!", True, red)

    gameDisplay.blit(high\_score, (displayWidth - scoreOffsetX, scoreOffsetY))

- Màn hình khi bắt đầu gồm: lời chào và các button để bắt đầu hay thoát trò chơi.



Hình 4: Tạo màn hình khi mới mở trò chơi

def startScreen():

    while True:

        fillBackground(True)

        put\_message\_custom("Welcome to Snake!", green, -80)

        events = pygame.event.get()

        for event in events:

            if event.type == pygame.QUIT:

                quitProgram()

        startButton.showButton()

        quitButton.showButton()

        if startButton.isHovered(getCursorPos()) and isLeftMouseClicked():

            reset()

            return

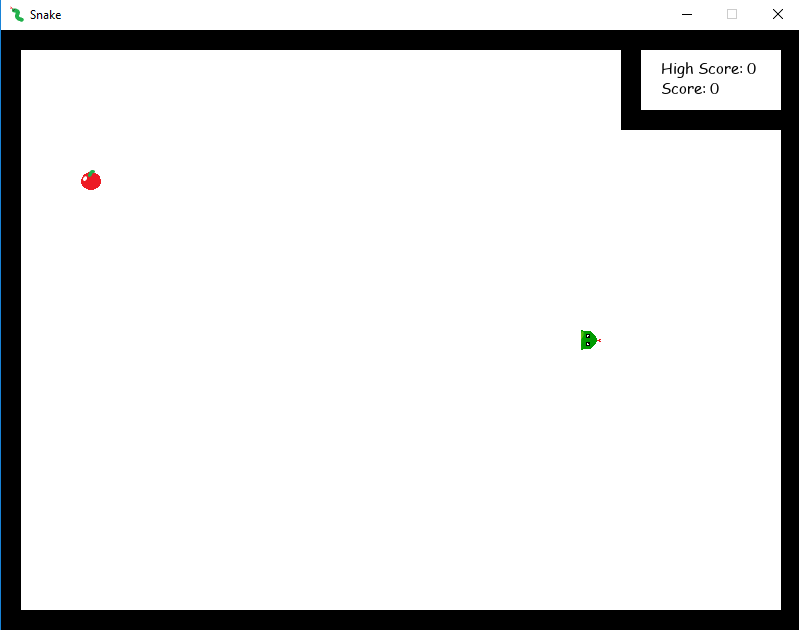
        elif quitButton.isHovered(getCursorPos()) and isLeftMouseClicked():

            quitProgram()

        pygame.display.update()

- Màn hình khi đang chơi và khi kết thúc:

+ Màn hình khi đang chơi: con rắn sẽ khởi tạo tại vị trí chính giữa màn hình và đang di chuyển sang phía bên trái màn hình.



Hình 5: Tạo màn hình khi đang chơi trò chơi

+ Màn hình khi kết thúc: khi con rắn chạm vào khung viền màu đen hoặc tự chạm vào thân của chính mình, trò chơi sẽ chuyển sang màn hình thông báo “Game over”.



Hình 6: Tạo màn hình khi trò chơi kết thúc

def gameLoop():

global appleCounter

    global degrees

    global highScore

    global leadX

    global leadY

    global leadXChange

    global leadYChange

    global snakeList

    global goldenApple

    global FPS

    leadXChange = blockSize

    leadYChange = 0

    gameOver = False

    goldenApple = generateGoldenApple()

    randomApple()

    while True:

        events = pygame.event.get()

        fillBackground(False)

        while gameOver:  # the user lost

            if highScore < appleCounter:

                # set new high score if applicable

                with open('score.dat', 'rb') as fromFile:

                    highScore = pickle.load(fromFile)

                with open('score.dat', 'wb') as fromFile:

                    pickle.dump(appleCounter, fromFile)

            events = pygame.event.get()

            for event in events:

                if event.type == pygame.QUIT:

                    quitProgram()

                if event.type == pygame.MOUSEBUTTONDOWN:

                    reset()

                    gameLoop()

            fillBackground(False)

            showScores(appleCounter, highScore < appleCounter)

            put\_message\_custom("Game Over!", red, -80)

            put\_message\_custom("Press R to play again.", black, -20)

            quitButton.showButton()

            if quitButton.isHovered(getCursorPos()) and isLeftMouseClicked():

                quitProgram()

            pygame.display.update()

        for event in events:

            if event.type == pygame.QUIT:

                quitProgram()

            if event.type == pygame.KEYDOWN:  # key presses

                if (len(snakeList) < 2 or degrees != 270) and (event.key == pygame.K\_LEFT or event.key == pygame.K\_a):

                    leadXChange = -blockSize

                    leadYChange = 0

                    degrees = 90

                elif (len(snakeList) < 2 or degrees != 90) and (event.key == pygame.K\_RIGHT or event.key == pygame.K\_d):

                    leadXChange = blockSize

                    leadYChange = 0

                    degrees = 270

                elif (len(snakeList) < 2 or degrees != 180) and (event.key == pygame.K\_UP or event.key == pygame.K\_w):

                    leadYChange = -blockSize

                    leadXChange = 0

                    degrees = 0

                elif (len(snakeList) < 2 or degrees != 0) and (event.key == pygame.K\_DOWN or event.key == pygame.K\_s):

                    leadYChange = blockSize

                    leadXChange = 0

                    degrees = 180

                elif event.key == pygame.K\_p:

                    pause()

        leadX += leadXChange

        leadY += leadYChange

        if leadX == randAppleX and leadY == randAppleY:  # if the snake has eaten the apple

            if goldenApple:

                appleCounter += 3

            else:

                appleCounter += 1

            randomApple()

        snakeHead = [leadX, leadY]  # updates the snake's head location

        # checks if a golden apple should be generated

        if goldenApple:

            gameDisplay.blit(goldenAppleImage, (randAppleX, randAppleY))

        else:

            gameDisplay.blit(appleImage, (randAppleX, randAppleY))

        # condition checking if the snake has run into itself or gone out of bounds

        if snakeHead in snakeList[:-1] or \

                (leadX > boundX or leadX < blockSize or leadY > boundY or leadY < blockSize) \

                or (leadX >= scoreBoundWidth and leadY <= scoreBoundHeight):

            gameOver = True

        snakeList.append(snakeHead)  # add the snakeHead

        snake(snakeList)  # generate the snake

        if len(snakeList) > appleCounter:  # delete the first element of the snakeList.

            del snakeList[0]

        with open('score.dat', 'rb') as fromFile:  # load high score

            highScore = pickle.load(fromFile)

        showScores(appleCounter, highScore < appleCounter)

        pygame.display.update()

        clock.tick(FPS + (appleCounter / 50))  # set FPS, scales with how many apples the user has

- Màn hình tạm ngưng khi đang chơi

def pause():

    while True:

        events = pygame.event.get()

        for event in events:

            if event.type == pygame.QUIT:

                quitProgram()

            if event.type == pygame.MOUSEBUTTONDOWN:

                return

        put\_message\_center("Game Paused", black, )

        put\_message\_custom("Click to resume..", black, *fontSize*=30, *offsetY*=50)

        pygame.display.update()

- Tạo con rắn

def snake(*snakeCoors*):

    rotatedHead = pygame.transform.rotate(snakeHeadImage, degrees)

    gameDisplay.blit(rotatedHead, (snakeCoors[-1][0], snakeCoors[-1][1]))

    for coor in snakeCoors[:-1]:

        gameDisplay.blit(snakeBodyImage, [coor[0], coor[1]])

- Tạo vị trí cho quả táo thường: nếu vị trí xuất hiện của quả táo trùng với ô điểm số hoặc trùng với thân hình con rắn thì sẽ tạo lại ở vị trí khác.

def randomApple():

    global randAppleX

    global randAppleY

    global goldenApple

    lastAppleX = randAppleX

    lastAppleY = randAppleY

    goldenApple = generateGoldenApple()

    randAppleX = round(random.randint(blockSize \* 2, boundX - (blockSize \* 4)) / blockSize) \* blockSize

    randAppleY = round(random.randint(blockSize \* 2, boundY - (blockSize \* 4)) / blockSize) \* blockSize

    while [randAppleX, randAppleY] in snakeList or randAppleX == lastAppleX or randAppleY == lastAppleY or \

            (randAppleX >= scoreBoundWidth and randAppleY <= scoreBoundHeight):

        # if the apple generates under the snake or within the high score box, regenerate it

        randAppleX = round(random.randint(blockSize \* 2, boundX - scoreBoundWidth - (blockSize \* 4)) / blockSize) \* \ blockSize

        randAppleY = round(random.randint(blockSize \* 2, boundY - scoreBoundHeight - (blockSize \* 4)) / blockSize) \* \ blockSize

- Tạo sự xuất hiện cho quả táo vàng

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 7: Tạo quả táo vàng

def generateGoldenApple():

    return random.randint(1, 15) == 1

### 2.2.3. Giải thích các class, funtion khác

- Hàm để hiển thị các thông báo lên màn hình

def put\_message\_center(*message*, *color*):

    screen\_text = bodyFont.render(message, True, color)

    gameDisplay.blit(screen\_text, [centerDisplayWidth - (screen\_text.get\_rect().width / 2), centerDisplayHeight - (screen\_text.get\_rect().height / 2)])

def put\_message\_custom(*message*, *color*, *offsetY*, *fontSize*=50):

    screen\_text = pygame.font.SysFont("comicsansms", fontSize).render(message, True, color)

    gameDisplay.blit(screen\_text, [centerDisplayWidth - (screen\_text.get\_rect().width / 2),

                                   (centerDisplayHeight - (screen\_text.get\_rect().height / 2) + offsetY)])

- Hàm thoát khỏi chương trình

def quitProgram():

    pygame.quit()

    exit()

- Hàm vẽ Background

def fillBackground(*isStartScreen*):

    gameDisplay.fill(black)

    gameDisplay.fill(white, [blockSize, blockSize, boundX, boundY])

    if not isStartScreen:

        gameDisplay.fill(black, [scoreBoundWidth, blockSize, displayWidth - 150, scoreBoundHeight])

        gameDisplay.fill(white, [(scoreBoundWidth + blockSize, blockSize), (blockSize \* 7, 100 - (blockSize \* 2))])

- Hàm reset trò chơi

def reset():

    global appleCounter

    global degrees

    global highScore

    global leadX

    global leadY

    global leadXChange

    global leadYChange

    global randAppleX

    global randAppleY

    global snakeList

    global goldenApple

    degrees = 270

    leadX = centerDisplayWidth

    leadY = centerDisplayHeight

    leadXChange = blockSize

    leadYChange = 0

    randAppleX, randAppleY, appleCounter = (0,) \* 3

    snakeList = []

    goldenApple = generateGoldenApple()

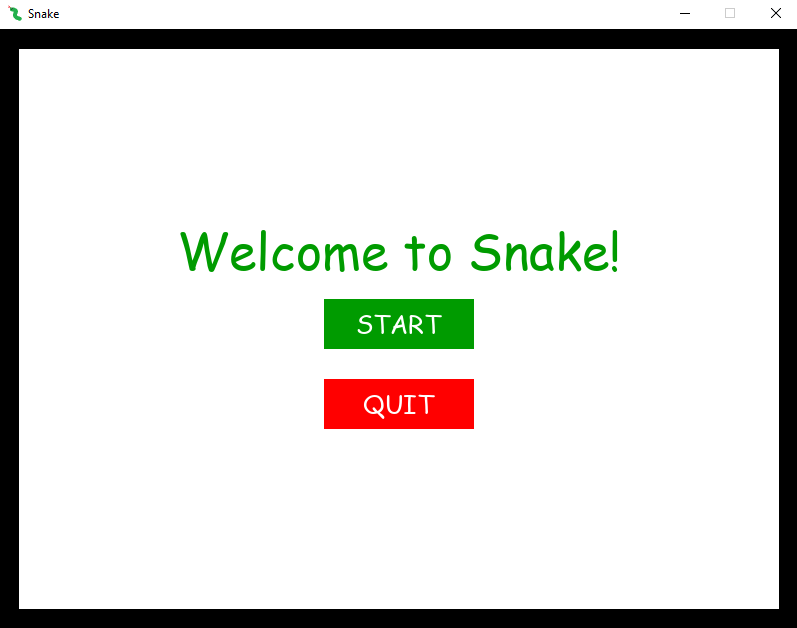
# Phần 3: Đánh giá

Trò chơi rắn săn mồi trên sẽ giúp cho người chơi giải trí sau những giờ làm việc. Tuy vậy, trò chơi vẫn còn vài hạn chế. Vì vậy, nó cần cải thiện thêm về một số chức năng: cho người dùng chọn màu con rắn với nhiều màu sắc hơn, tạo ra thêm nhiều chế độ chơi với các luật chơi khác nhau để tạo thêm sự thú vị, mới lạ cho người chơi,… Khi khắc phục các nhược điểm thành công, trò chơi sẽ trở nên hấp dẫn, thú vị hơn về đồ họa và cách chơi đối với người chơi, góp phần tạo ra sự vui vẻ, giải trí sau những giờ làm việc căng thẳng.

# Phần 4: Demo

## 4.1. Trình bày hình ảnh khi bắt đầu trò chơi

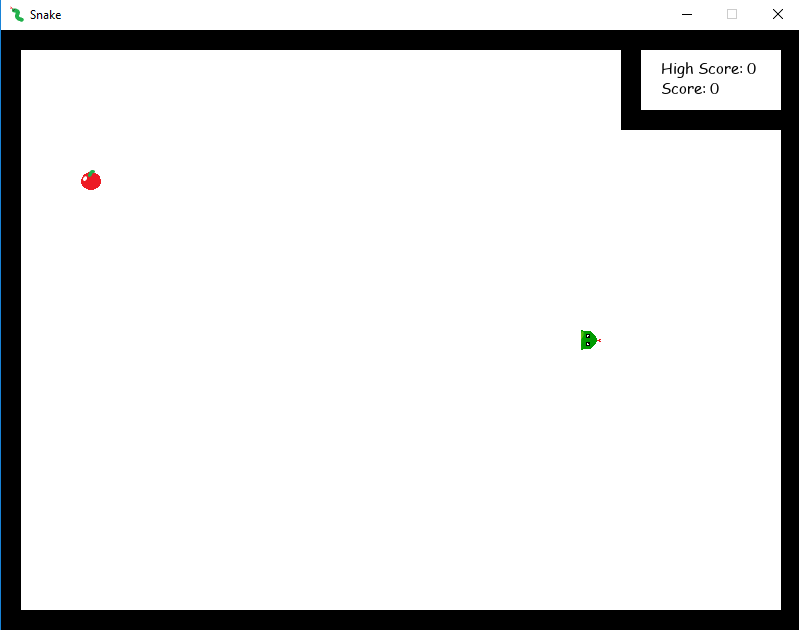
- Người chơi sẽ nhấn chuột vào nút Start để bắt đầu trò chơi hoặc nhấn nút Quit để thoát khỏi trò chơi.



Hình 8: Màn hình khi mới khởi động trò chơi

## 4.2. Trình bày hình ảnh khi trò chơi đang chạy

- Người chơi sẽ điều khiển con rắn bằng các nút mũi tên để con rắn ăn những trái táo xuất hiện trên màn hình. Nếu con rắn ăn trái táo màu đỏ thì điểm số sẽ tăng thêm một điểm, nếu là trái táo màu vàng thì điểm số sẽ tăng thêm ba điểm.



Hình 9: Màn hình trong lúc đang chơi trò chơi

## 4.3. Trình bày hình ảnh khi trò chơi kết thúc

- Khi con rắn chạm vào đường viền màu đen hoặc nó chạm vào thân của chính mình thì màn hình “Game over” sẽ được hiển thị. Người chơi sẽ nhấn phím r để chơi lại hoặc nhấn “Quit” để thoát khỏi trò chơi.



Hình 10: Màn hình khi trò chơi kết thúc

**Tài liệu tham khảo:**

1. https://www.edureka.co/blog/snake-game-with-pygame/

2.https://www.youtube.com/watch?v=K5FaGDIYaM&list=PL6gx4Cwl9DGAjkwJocj7vlc\_mFU-4wXJq