ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

BÁO CÁO ĐỒ ÁN

TÌM HIỂU THƯ VIỆN PYGAME VÀ XÂY DỰNG GAME DINO ISLAND

Sinh viên thực hiện:

Họ và tên: Phạm Thanh Thiện

MSSV: 20520027

Giảng viên: ThS. Nguyễn Thanh Sơn

Thành phố Hồ Chí Minh, tháng 12 năm 2022

MỤC LỤC

DANH MỤC HÌNH ẢNH

1. Giới thiệu đồ án

Trong báo cáo này, em muốn giới thiệu đồ án game mà em đã làm cho môn Kỹ thuật lập trình Python. Tên của game là "Dino Island", đây là một game thuộc thể loại phiêu lưu đơn giản và dễ chơi. Nhiệm vụ của người chơi là điều khiển nhân vật chạy và tránh các vật cản xuất hiện trên màn hình. Cách tính điểm số là theo quảng đường mà nhân vật chạy. Mục tiêu của người chơi là đi càng xa càng tốt để đạt điểm cao nhất. Khi nhân vật đạt được mỗi 50 điểm, nó sẽ tiến hóa lên một nhân vật khác và tăng tốc độ chạy. Trò chơi kết thúc khi nhân vật va chạm với một vật cản hoặc khi người chơi nhấn vào nút thoát. Trò chơi bao gồm các tính năng chính như: Hiển thị điểm số, lưu điểm cao nhất qua mỗi vòng chơi, chuyển đổi nhân vật khi đạt được số điểm nhất định và tạo hiệu ứng âm thanh và hình ảnh trong trò chơi. Trò chơi được xây dựng dựa trên nền tảng Pygame trong Python và được sử dụng để giúp người chơi rèn luyện khả năng tập trung và phản xạ.

Trò chơi được xây dựng dựa trên nền tảng ngôn ngữ lập trình Python và thư viện Pygame. Em sử dụng Pygame để tạo ra các hiệu ứng âm thanh và hình ảnh trong trò chơi cũng như xử lý các sự kiện khi người chơi nhập vào từ bàn phím, chuột và các hàm có sẵn trong thư viện Pygame như chức năng hiển thị (blit), xử lý va chạm (collision) và render hình ảnh. Trong quá trình làm đồ án, em đã học được nhiều kiến thức về lập trình Python và cách sử dụng Pygame. Đồ án này giúp em có cơ hội để khám phá và học hỏi thêm về lập trình game và cũng như sử dụng thư viện Pygame trong Python. Trong tương lai, em sẽ phát triển game này thêm các tính năng như nhặt vũ khí để tiêu diệt vật cản, thêm nhiều nhân vật để game có thể trở nên phong phú và sinh động hơn.

2. Tóm tắt quá trình thực hiện

Tìm hiểu các kiến thức về lập trình Python và thư viện Pygame, học các khái niệm cơ bản và cách sử dụng chúng để tạo ra game.

Tạo các lớp cần thiết cho game như lớp màn hình bắt đầu, màn hình kết thúc, lớp background, lớp nhân vật, lớp vật cản, lớp hiển thị điểm số và lớp xử lý va chạm.

Thiết kế giao diện game với các hình ảnh và âm thanh tương ứng.

Xây dựng các chức năng cần thiết cho game, bao gồm các chức năng điều khiển, xử lý trạng thái của nhân vật khi di chuyển, tính điểm, lưu trữ điểm cao nhất qua mỗi lần chơi lại, thay đổi đối tượng khi đạt số điểm nhất định, tăng tốc độ theo điểm số, xử lý va chạm và kết thúc game.

Suy nghĩ logic và tiến hành viết code cho các lớp đã tạo và kết hợp chúng với nhau.

Kiểm tra và sửa các lỗi trong quá trình chạy game.

Tạo file .exe để người chơi dễ dàng vào game.

3. Kết quả đạt được

Trong quá trình làm đồ án, em đã thực hiện đầy đủ các task đã nêu trong phiếu đăng ký và hoàn thành được game Dino Island bằng Pygame. Game được thiết kế theo dạng endless running, khi chơi người chơi sẽ điều khiển nhân vật chạy trên màn hình và phải tránh các vật cản để tiếp tục chơi. Trong quá trình chơi, người chơi có thể nhảy để tránh các vật cản và đạt điểm số cao hơn. Ngoài ra, em còn thêm vào một số tính năng khi người chơi đạt được mỗi 50 điểm, nhân vật sẽ thay đổi hình dạng và tăng tốc độ chạy khi điểm cao dần và cho phép người chơi chơi lại sau khi thua. Bên cạnh đó, em cũng thiết kế và

code màn hình bắt đầu, màn hình kết thúc và hiệu ứng âm thanh giúp em trở nên sinh động và hấp dẫn hơn.

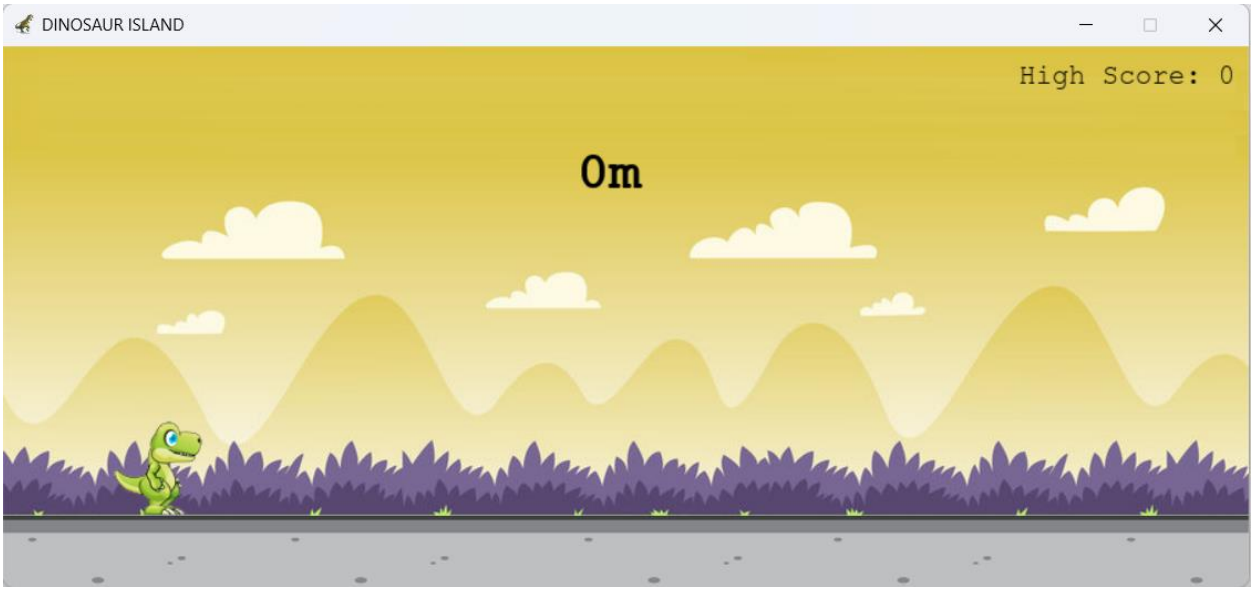
Han chế:

Chưa tạo được nhiều kích cỡ màn hình khác nhau để người chơi có thể thoải mái lựa chọn.

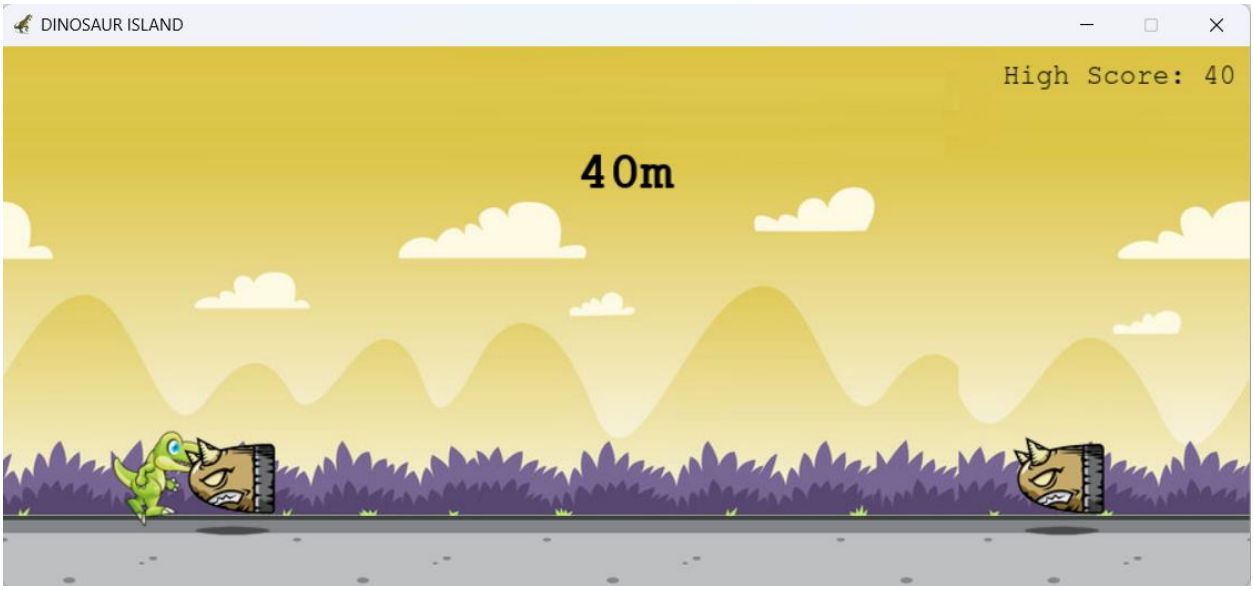
Các hình ảnh dùng để tạo nhân vật trong game bị dính bản quyền nên chất lượng hình ảnh chưa được đẹp.

4. Phụ lục 1: Giới thiệu (demo) kết quả

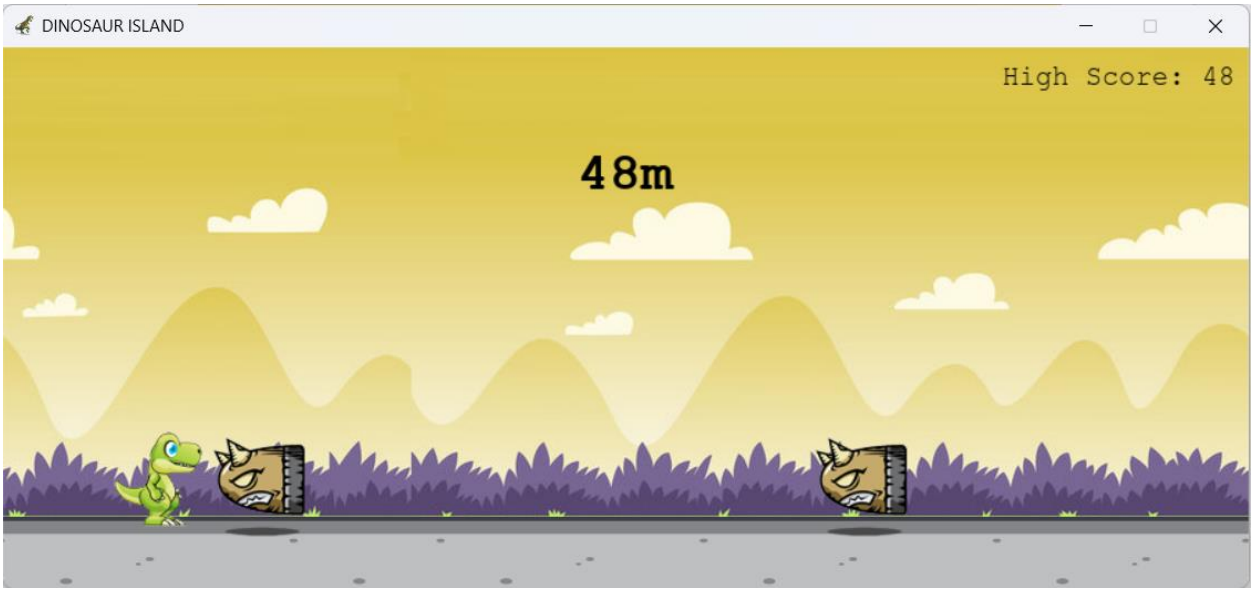
Hình 1: Giao diện màn hình khi khởi động game



Hình 2: Giao diện background và nhân vật

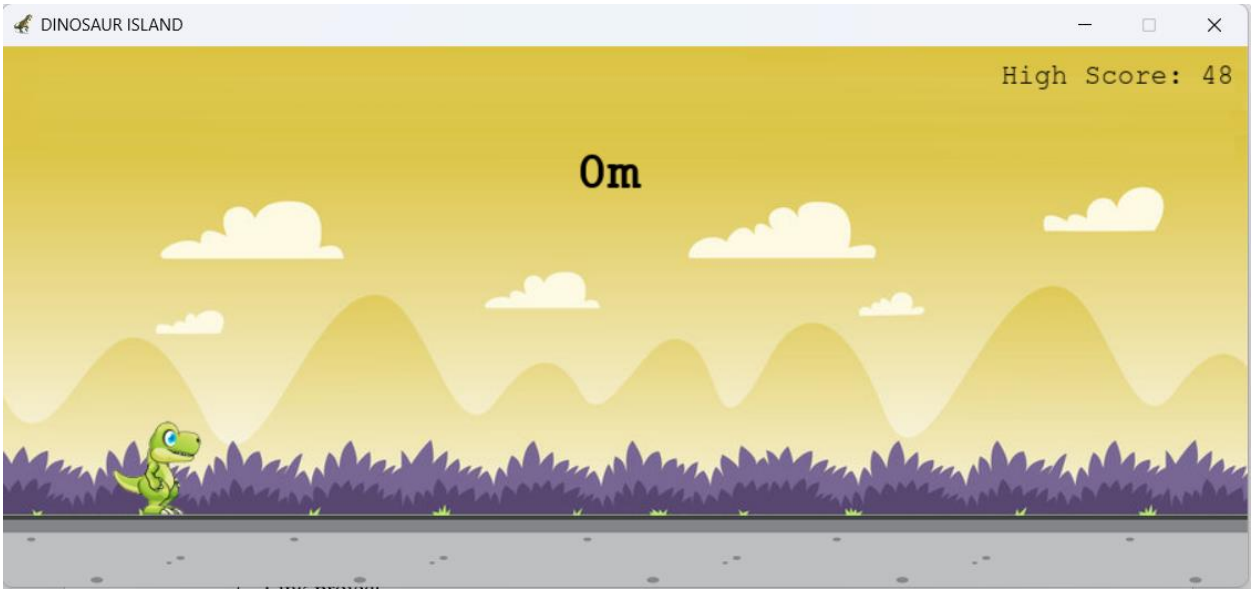


Hình 3: Giao diện vật cản

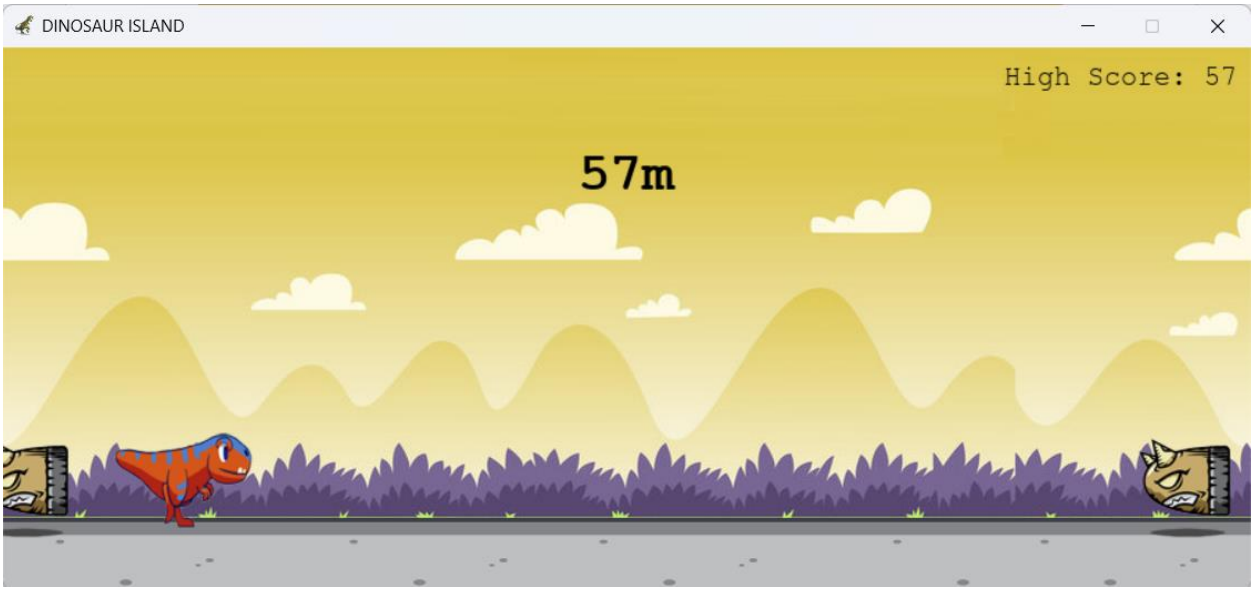


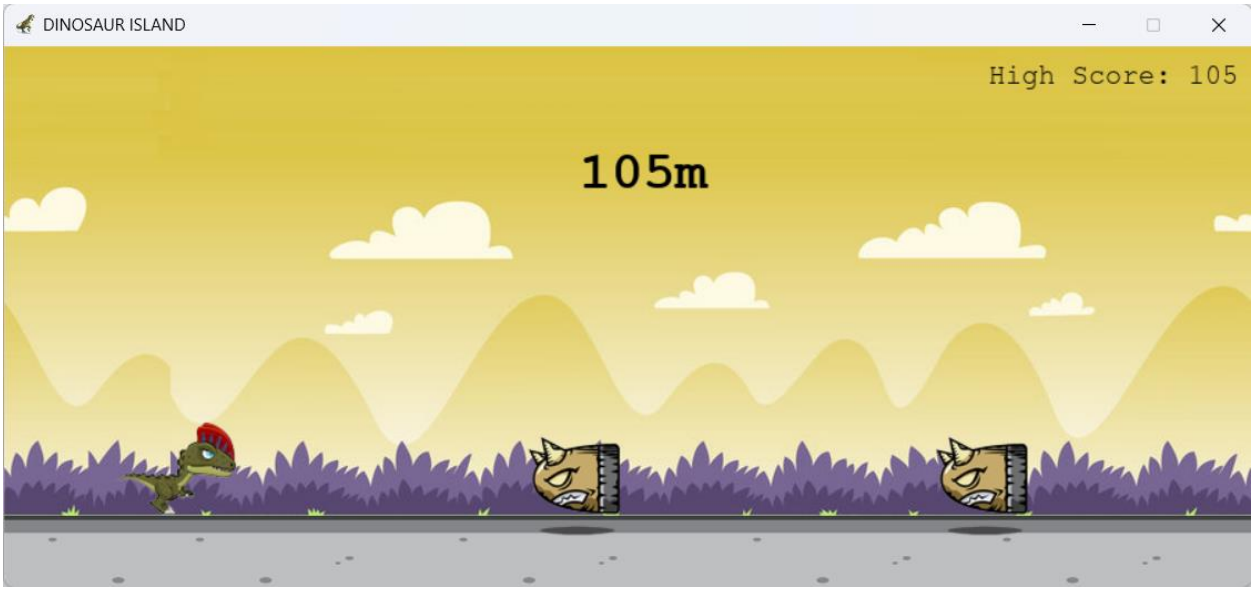
Hình 4: Giao diện điểm hiện tại và điểm cao nhất

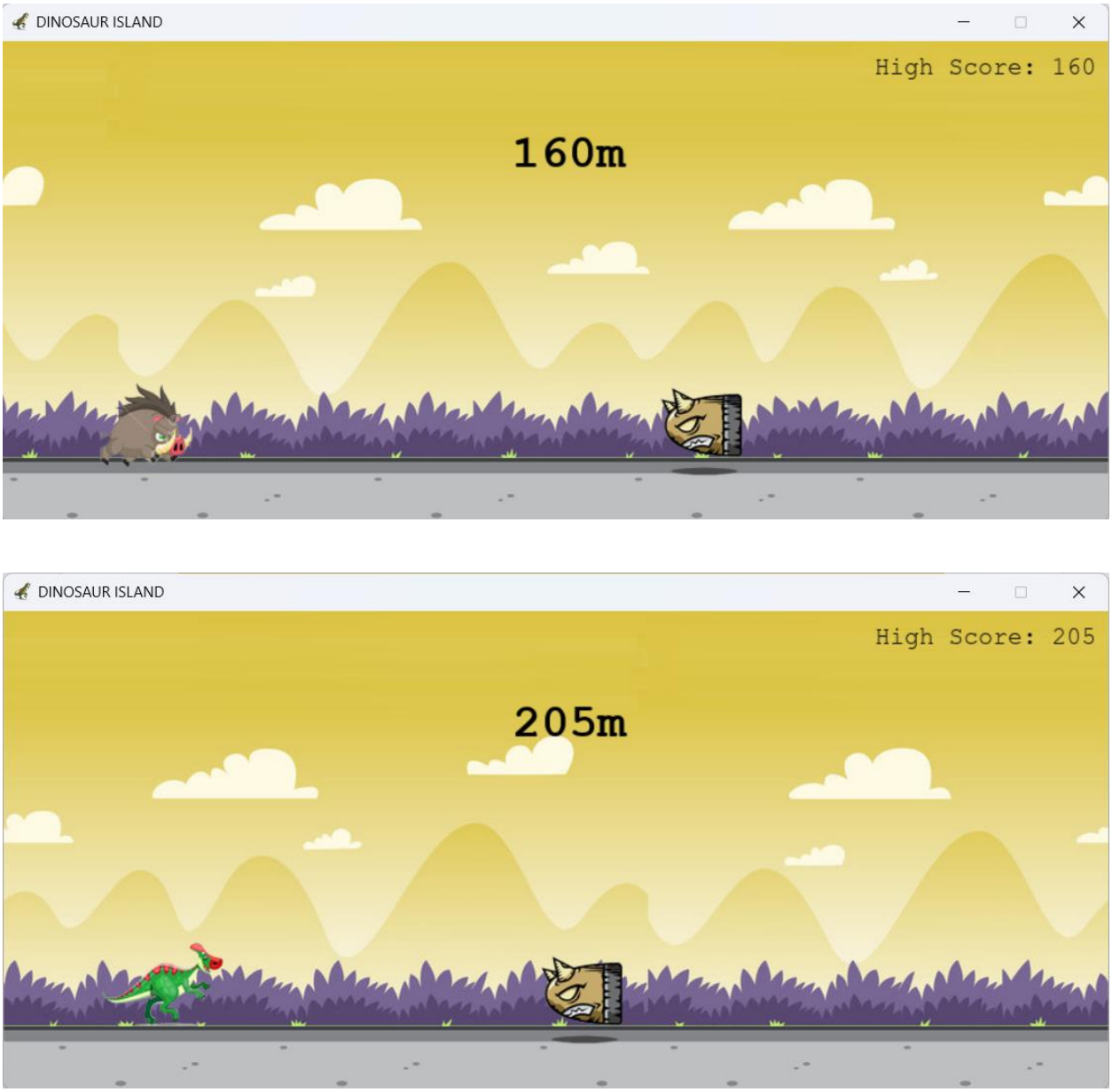
Hình 5: Giao diện màn hình game over



Hình 6: Giao diện điểm cao khi chơi lại


DINOSAUR ISLAND
High Score: 57
57m


DINOSAUR ISLAND
High Score: 105
105m



Hình 7: Giao diện các nhân vật biến đổi mỗi 50m

5. Phụ lục 2: Docstring

```
# Tạo lớp Background
class BG:

    """
    Chức năng: Tạo một hình nền cho game.

    Args:
        x: Tọa độ x của hình nền.

    Input:
        width: Chiều rộng của hình nền.
        height: Chiều cao của hình nền.
        x: Tọa độ x của hình nền.
        y: Tọa độ y của hình nền.
        texture (Surface): Đối tượng Surface tạo từ hình ảnh của hình nền.

    Output:
        update(def_x): Cập nhật trạng thái của hình nền theo biến speed được khai
báo ở class game di chuyển theo trục x.
        show(): Hiển thị hình nền lên màn hình.
        set_texture(): Tạo ra một đối tượng Surface từ hình ảnh của hình nền và
lưu trữ trong thuộc tính self.texture.
    """

    # Hàm khởi tạo
    def __init__(self, x):

        """
            Chức năng: Tạo một đối tượng hình nền với tọa độ x cho trước và tạo
ra một đối tượng Surface từ hình ảnh của hình nền.

        Args:
            x: Tọa độ x của hình nền dùng để tạo vòng lặp.

        Input:
            width: Chiều rộng của hình nền.
            height: Chiều cao của hình nền.
            x: Tọa độ x của hình nền.
            y: Tọa độ y của hình nền.
            texture (Surface): Đối tượng Surface tạo từ hình ảnh của hình nền.

        Output:
```

```
            BG: Đối tượng hình nền với thuộc tính width, height, x, y, texture
được khởi tạo.
        """

    # Hàm cập nhật
    def update(self, def_x):

        """
        Chức năng: Cập nhật tọa độ x của hình nền theo biến speed được khai báo ở
class game di chuyển theo trục x. Nếu tọa độ x của hình nền nhỏ hơn hoặc bằng -
WIDTH, thì sẽ đặt lại tọa độ x bằng WIDTH.

        Input:
            def_x: Độ dịch cần cập nhật tọa độ x.

        Output:
            x: Tọa độ x của hình nền.
        """

    # Hàm hiển thị
    def show(self):

        """
        Chức năng: Sử dụng phương thức blit() của module pygame để hiển thị đối
tượng Surface của hình nền lên màn hình tại tọa độ x và y.

        Attributes:
            x: Tọa độ x của hình nền.
            y: Tọa độ y của hình nền.
            texture (Surface): Đối tượng Surface tạo từ hình ảnh của hình nền.
        """

    # Hàm surface dùng để load hình ảnh
    def set_texture(self):

        """
        Chức năng: Phương thức này dùng để load hình ảnh background vào chương
trình.

        Input:
            Hình ảnh bg.png

        Output:
            Hình ảnh background
        """
```

```
# Tạo lớp Dino
class Dino:

    """
    Chức năng: Class đại diện cho nhân vật khủng long trong game. Nó dùng để định
nghĩa các phương thức khởi tạo, cập nhật, hiển thị
    , tạo đối tượng Surface từ hình ảnh và tạo đối tượng Sound từ âm thanh của
nhân vật khủng long, cũng như các phương thức nhảy, rơi và chạy.

    Attributes:
        width (int): Chiều rộng của nhân vật khủng long.
        height (int): Chiều cao của nhân vật khủng long.
        x (int): Tọa độ x của nhân vật khủng long.
        y (int): Tọa độ y của nhân vật khủng long.
        img_num (int): Số thứ tự của hình ảnh hiện tại của nhân vật khủng long.
        def_y (int): Độ dịch theo trục y khi nhân vật khủng long nhảy.
        gra (float): Trọng lực của nhân vật khủng long.
        running (bool): Trạng thái đang chạy trên mặt đất.
        jumping (bool): Trạng thái đang nhảy.
        falling (bool): Trạng thái đang rơi.
        jump_to (int): Độ cao tối đa mà khủng long nhảy đến.
        floor (int): Định nghĩa tọa độ y của sàn.
        lv (int): Cấp độ hiện tại của khủng long.
        texture (pygame.Surface): Đối tượng Surface đại diện cho hình ảnh của
nhân vật khủng long.
        sound (pygame.mixer.Sound): Đối tượng Sound đại diện cho âm thanh của
nhân vật khủng long.
    """

    # Hàm khởi tạo
    def __init__(self):
        """
        Chức năng: Khởi tạo các giá trị và phương thức cho nhân vật Dino
        """

    # Hàm cập nhật
    def update(self, loop):

        """
        Chức năng: Phương thức này cập nhật trạng thái nhảy, đáp đất và chạy của
đối tượng Dino dựa trên trạng thái hiện tại và tọa độ y
        . Nếu đối tượng Dino đang nhảy, tọa độ y của nó sẽ bị giảm bằng khoảng trừ
tọa độ y. Nếu đối tượng Dino đang đáp đất, tọa
```

```
        độ y của nó sẽ tăng bằng tích của trọng lực và khoảng trừ tọa độ y. Nếu đối tượng Dino đang chạy và số vòng lặp hiện tại
        chia hết cho 4, hình ảnh của đối tượng Dino sẽ thay đổi tạo ra hoạt ảnh.

        Args:
        loop (int): Số vòng lặp hiện tại.

        Returns:
            Trả về các trạng thái và thực hiện hành động ở trạng thái đó
        """

    # Hàm hiển thị
    def show(self):

        """
        Chức năng: Sử dụng phương thức blit() của module pygame để hiển thị đối tượng Surface của nhân vật Dino lên màn hình tại tọa độ x và y.

        Attributes:
            x: Tọa độ x của nhân vật Dino.
            y: Tọa độ y của nhân vật Dino.
            texture (Surface): Đối tượng Surface tạo từ hình ảnh của nhân vật Dino.
        """

    # Hàm surface
    def set_texture(self):

        """
        Chức năng: Phương thức này thiết lập hình ảnh của đối tượng Dino dựa trên số hình ảnh của nó.

        Input:
            img_num (int): Số thứ tự hình ảnh được định nghĩa từ 0 đến số cuối cùng của def init

        Output:
            Hình ảnh nhân vật Dino
        """

    def set_sound(self):

        """
        Chức năng: Phương thức này thiết lập âm thanh của đối tượng Dino.
```

```
        Input:
            Tệp âm thanh jump.wav

        Output:
            Load âm thanh jump.wav
        """

    def jump(self):

        """
        Chức năng: Phương thức này đổi trạng thái của đối tượng Dino thành trạng
thái nhảy và phát âm thanh nhảy.

        Input:
            sound: Dùng biến này để lấy âm thanh và phát
            jumping, running: Giá trị của 2 biến này là boolean và được lấy từ
init

        Output:
            Trả về biến boolean định nghĩa trạng nhảy của đối tượng Dino
        """

    def fall(self):

        """
        Chức năng: Phương thức này đổi trạng thái của đối tượng Dino thành trạng
thái rơi.

        Input:
            falling, jumping: Giá trị của 2 biến này là boolean và được lấy từ
init

        Output:
            Trả về biến boolean định nghĩa trạng rơi của đối tượng Dino
        """

    def run(self):

        """
        Chức năng: Phương thức này đổi trạng thái của đối tượng Dino thành trạng
thái chạy.

        Input:
            running, falling: Giá trị của 2 biến này là boolean và được lấy từ
init
```

```
        Output:
            Trả về biến boolean định nghĩa trạng chạy của đối tượng Dino
        """

# Tạo lớp chướng ngại vật
class Gun:

    """
    Chức năng: class này này khởi tạo đối tượng chướng ngại vật là hình viên đạn
với chiều rộng là 113, chiều cao là 90, vị trí x là một đối số
    và vị trí y là 270. Thiết lập hình dạng của súng bằng phương thức
'set_texture()' và hiển thị nó trên màn hình bằng phương thức 'show()'.

    Args:
        x: Tọa độ x của viên đạn dùng để tạo vòng lặp.
    """

    def __init__(self, x):

        """
        Chức năng: Đây là phương thức khởi tạo của lớp Gun. Nó khởi tạo các thuộc
tính của lớp như chiều rộng, chiều cao,
        vị trí x, vị trí y và gọi phương thức set_texture() để đặt load hình ảnh
viên đạn.

        Args:
            x: Vị trí x ban đầu của viên đạn trên màn hình.

        Return:
            None
        """

    def update(self, def_x):

        """
        Chức năng: Cập nhật vị trí x của viên đạn theo giá trị def_x.

        Args:
            def_x: Vị trí di chuyển của viên đạn.

        Returns:
            None
        """
```

```
    def show(self):

        """
        Chức năng: Sử dụng phương thức blit() của module pygame để hiển thị đối tượng Surface của hình viên đạn lên màn hình tại tọa độ x và y.

        Attributes:
            x: Tọa độ x của hình nền.
            y: Tọa độ y của hình nền.
            texture (Surface): Đối tượng Surface tạo từ hình ảnh của hình nền.
        """

    def set_texture(self):

        """
        Chức năng: Phương thức này dùng để load hình ảnh chướng ngại vật hình viên đạn vào chương trình.

        Input:
            Hình ảnh gun.png

        Output:
            Hình ảnh chướng ngại vật hình viên đạn
        """

# Kiểm tra va chạm
class Collision:

    """
    Chức năng: class Collision dùng để xử lý va chạm giữa các đối tượng trong game.
    """

    def between(self, obj1, obj2):

        """
        Chức năng: Phương thức này tính khoảng cách giữa obj1 và obj2 sử dụng công thức khoảng cách và trả về giá trị boolean cho biết
        khoảng cách có nhỏ hơn 30 hay không. Nếu khoảng cách nhỏ hơn 30 thì xảy ra va chạm.

        Input:
            obj1: Đối tượng khủng long.
            obj2: Đối tượng viên đạn.
```

```
        Output:
            bool: Trả về True nếu có va chạm, ngược lại trả về False.
        """

# Tạo lớp điểm số
class Score:

    """
    Chức năng: Class Score quản lý việc hiển thị điểm số và điểm cao nhất trong
game.
    """

    def __init__(self, hscore):

        """
        Chức năng: Phương thức khởi tạo của lớp Score. Nó khởi tạo các thuộc tính
của lớp như điểm số, điểm cao nhất,
        font để hiển thị điểm số, font để hiển thị điểm cao nhất và màu chữ, và
gọi phương thức 'set_sound()'
        để khởi tạo nhạc và phát khi điểm số tăng.

        Args:
            hscore (int): Lưu trữ biến điểm cao nhất và hiển thị lại những vòng
chơi tiếp theo."
        """

    def update(self, loop):

        """
        Chức năng: Phương thức này cập nhật điểm mỗi 10 vòng lặp và kiểm tra xem
điểm hiện tại có phải là điểm cao nhất hiện tại không.
        Nó phát ra một âm thanh mỗi 50 điểm.

        Args:
        loop (int): Số vòng lặp hiện tại.

        Output: Giá trị của loop
        """

    def set_sound(self):

        """
        Chức năng: Phương thức này dùng để load âm thanh khi nhân vật đạt mỗi 50
điểm.
```

```
        Input:
            File âm thanh point.wav

        Output:
            Âm thanh khi đạt mỗi 50 điểm
        """

    def check_hscore(self):

        """
        Chức năng: Phương thức này kiểm tra xem điểm hiện tại có phải là điểm cao
nhất hiện tại không.

        Output: high score hiện tại
        """

    def check_sound(self):

        """
        Chức năng: Hàm này dùng để kiểm tra xem điểm hiện tại đạt mỗi 50 thì nó
phát âm thanh.

        Output:
            sound: Âm thanh đã load ở hàm set_sound
        """

    def show(self):

        """
        Chức năng: Phương thức này hiển thị điểm cao nhất và điểm hiện tại trên
màn hình bằng cách sử dụng phương thức render() của module pygame.

        Input: None

        Output:
            label1: Là điểm số cao nhất hiển thị ở góc phải trên cùng màn hình
            label2:Là điểm số hiện tại hiển thị ở giữa màn hình
        """

# Tạo lớp trò chơi
class Game:

    def __init__(self, hscore = 0):

        """
```

```
        Chức năng: Khởi tạo các thuộc tính của game, như: mảng đối tượng
background, đối tượng nhân vật, các đối tượng vật cản,
        đối tượng xử lý va chạm, đối tượng hiển thị điểm số, trạng thái của game
(đang chơi hay game over), tốc độ của game, đối tượng
        âm thanh, và các đối tượng label.

        Input:
            hscore (int, optional): Điểm số lớn nhất mà người chơi đã đạt được
trong game (mặc định là 0).
        """

    def set_sound(self):

        """
        Chức năng: Load âm thanh
        """

    def set_labels(self):

        """
        Chức năng: Tạo font chữ cho các tiêu đề khi game over và hiển thị bằng
phương thức blit()
        """

    def start(self):

        """
        Chức năng: Khởi tạo lại giá trị playing

        Input:
            playing: Là biến nhận giá trị boolean để xác định game đang chạy hay
game over

        Output: Giá trị boolean
        """

    def end(self):

        """
        Chức năng: Hiển thị màn hình game over và ngừng chơi, đồng thời phát âm
thanh khi nhân vật va chạm với vật cản
        """

    def appear(self, loop):
```

```
        """
        Chức năng: Xác định xem có hiển thị viên đạn mới trong game hay không.
        Input:
            loop: số lần lặp hiện tại trong game

        Output: Trả về True nếu loop chia hết cho 100, ngược lại trả về False.
        """

    def appear_gun(self):

        """
        Style: Hàm xử lý chức năng
        Chức năng: Hàm này dùng để tạo ra viên đạn mới và thêm nó vào mảng
self.obstacle

        Nếu mảng self.obstacle không rỗng thì lấy vị trí viên đạn cuối cùng và
tạo ra viên đạn mới có vị trí ngẫu nhiên trong khoảng:
        x = vị trí viên đạn cuối cùng + chiều rộng của con Dino + 155
        y = chiều rộng của màn hình + vị trí viên đạn cuối cùng + chiều rộng của
con Dino + 155

        Nếu mảng self.obstacle rỗng thì tạo ra viên đạn mới có vị trí ngẫu nhiên
trong khoảng:
        x = Chiều rộng của màn hình + 300
        y = 1200
        """

    def Uplv1(self):

        """
        Chức năng: Hàm dùng để kiểm tra điểm hiện tại có lớn hơn 50 và nhỏ hơn
hoặc bằng 100 hay không. Nếu đúng, hàm sẽ trả về
        giá trị True, ngược lại sẽ trả về False. Nếu True, nhân vật sẽ biến đổi
thành đối tượng nhân vật kế tiếp

        Input: None

        Output: Giá trị boolean
        """

    def Uplv2(self):

        """
        Chức năng: Tương tự hàm Uplv1()
        """
```

```
    def Uplv3(self):

        """
        Chức năng: Tương tự hàm Uplv1()
        """

    def Uplv4(self):

        """
        Chức năng: Tương tự hàm Uplv1()
        """

    def restart(self):

        """
        Chức năng: Khởi tạo lại trạng thái ban đầu của trò chơi, trong đó giá trị điểm cao nhất (high score) lưu lại và
        truyền vào trong lần chơi kế tiếp.

        Input: None

        Output: Trạng thái ban đầu của trò chơi được khởi tạo lại.
        """

# Vòng lặp chính của game
def main():

    """
    Chức năng: Phương pháp này là vòng lặp chính của trò chơi nơi tất cả các đối tượng trò chơi được cập nhật và hiển thị trên màn hình.
    Tốc độ trò chơi tăng lên sau mỗi 50 điểm. Vòng lặp sẽ kết thúc nếu người chơi va chạm với chướng ngại vật hoặc thoát khỏi trò chơi.
    Người chơi cũng có thể khởi động lại trò chơi bằng cách nhấn phím Enter.
    """

# Tạo màn hình bắt đầu game
class Stcreen:

    """
    Chức năng: Class để hiển thị màn hình bắt đầu với hình của 2 nhân vật chính là nhân vật Dino và chướng ngại vật gun và nhấp nháy
    phông chữ "Nhấn phím bất kì để bắt đầu" để người chơi biết cách bắt đầu chơi trò chơi.
```

```
    Attributes:
        flash (bool): Trạng thái nhấp nháy của phông chữ.
        sound (pygame.mixer.Sound): Âm thanh bắt đầu khi khởi động game.
        texture (pygame.Surface): Hình nền.
    """

    def __init__(self):

        """
        Chức năng: Hàm này khởi tạo các thuộc tính của class Streen, bao gồm
trạng thái nhấp nháy của phông chữ, âm thanh,
        texture và hiển thị tiêu đề trên màn hình.
        """

    def set_sound(self):

        """
        Chức năng: Phương thức này dùng để load âm thanh khi khởi động trò chơi.

        Input:
            File âm thanh start.wav

        Output:
            Âm thanh khi khởi động trò chơi.
        """

    def title(self):

        """
        Chức năng: Phương thức này dùng để xử lý hiển thị của tiêu đề "Nhấn phím
bất kì để bắt đầu". Sau mỗi chu kì 1 giây thì tiêu đề
        sẽ hiện sau đó ẩn và liên tục trong vòng lặp như vậy tạo nên sự sinh động
cho trò chơi.

        Input: None

        Output: None
        """

    def show(self):

        """
        Chức năng: Hiển thị màn hình khi khởi động game và tiếng âm đầu màn hình
cùng với chuỗi nhấp nháy "Nhấn phím bất kì để bắt đầu"
        """
```

```
    def set_texture(self):

        """
        Chức năng: Phương thức này dùng để load hình ảnh màn hình bắt đầu vào
chương trình.

        Input:
            Hình ảnh screen.png

        Output:
            Hình ảnh màn hình bắt đầu khi khởi động game
        """

def menu():

    """
    Chức năng: Gọi hàm khởi tạo màn hình khi khởi động game. Khi người dùng nhấn
phím, hàm này sẽ dừng âm thanh menu và
    chuyển qua hàm main().
    """
```

Link project

https://github.com/ThanhThien-dev/Dinosaur-Island

Tài liệu tham khảo

- Trang chủ của Pygame, **URL:** https://www.pygame.org/docs/
- Phát triển game với Pygame, **URL:** https://viblo.asia/p/phat-trien-game-voipygame-part-1-map-al5XRBDeGqPe
- PyGame Tutorial, **URL:** https://www.geeksforgeeks.org/pygame-tutorial/
- Học lập trình game bằng Pygame thông qua Youtube, **URL:** https://www.youtube.com/watch?v=v7V2DlGBjvI&t=37s
- Lập Trình Game Flappy Bird với Python Cho Người Mới Bắt Đầu, **URL:** https://www.youtube.com/watch?v=mFbdfXWmLU8