BÁO CÁO ĐỒ ÁN MÔN HỌC
Lớp: IE221O21.CNCL

SINH VIÊN THỰC HIỆN:

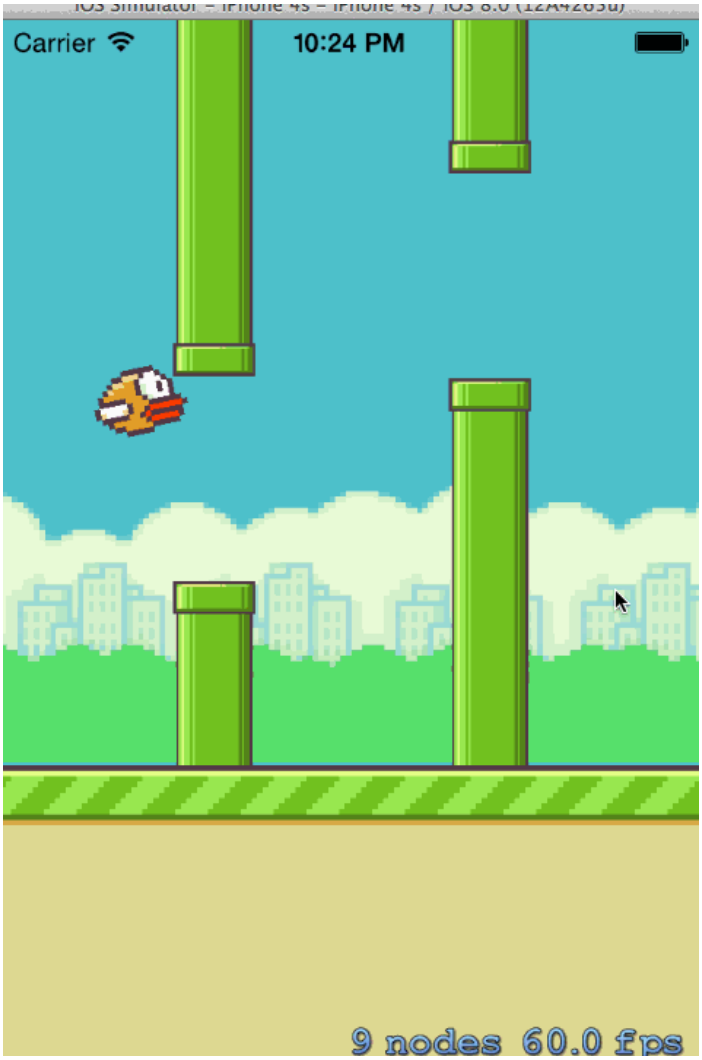
Mã sinh viên: 21520646 **Họ và tên: Đào Đại Chí**

TÊN ĐỀ TÀI: Lập trình game vượt chướng ngại vật bằng Python

CÁC NỘI DUNG CẦN BÁO CÁO:

1. Giới thiệu đồ án.

   Một trò chơi về 1 chú chim nhỏ màu vàng sẽ cùng người chơi vượt qua những chướng ngại vật.



2. Quá trình thực hiện

a. Tuần 1:

- Chuẩn bị các hình ảnh về chướng ngại vật, nhân vật chính(chú chim màu vàng), ảnh nền, nút bắt đầu, giao diện kết thúc
- Định nghĩa một số lớp cần thiết cho trò chơi như:
    + Bird: đại diện cho chú (chim nhân vật chính của trò chơi). Chú chim sẽ bay lên sau khi nhấn nút chỉ định. Việc này nhằm giúp nhân vật chính có thể vượt qua các chướng ngại vật phía trước.
    + Pipe: đại diện cho chướng ngại vật mà nhân vật chính phải vượt qua. Các chướng ngại vật sẽ được xuất hiện ngẫu

nhiên về chiều cao hay khoảng cách giữa các chướng ngại vật

+ Floor: là một cái sàn được chạy liên tục khi trò chơi bắt đầu.
+ Game: nhằm quản lý các lớp trên cũng như quản lý việc vận hành trò chơi.
+ Collision: lớp này được tạo ra để có nhận diện sự va chạm của lớp Bird và Pipe.

- Tạo ra giao diện bắt đầu trò chơi. Giao diện sẽ có một nút bắt đầu (Play button). Sau khi người chơi nhấn vào sẽ có thể bắt đầu trò chơi
- Tạo ra giao diện kết thúc.
- Source code: [DaoDaiChi/Flappy-bird (github.com)]

b. Tuần 2:

- Tìm kiếm hình ảnh cho các trang phục của chú chim (4 trang phục) , nút quay lại (back button).
- Thực hiện cắt ghép và chỉnh sửa sao cho phù hợp để đưa vào gameplay
- Tạo ra class Menu thay thế nó "Play button" ở tuần 1 và giao diện Menu này có chứa 3 nút để tương tác gồm: "Start game", "Bird skins", "Exit".
    + "Start game": để bắt đầu trò chơi
    + "Bird skins": có chứa các trang phục của chú chim mà người chơi có thể tùy chọn, có nút "OK" để xác nhận lựa chọn của mình
    + "Exit" dùng để thoát khỏi trò chơi
- Thêm nút back ở giao diện kết thúc trò chơi nhằm giúp người chơi có thể trở về giao diện bắt đầu.
- Thêm hiệu ứng âm thanh cho việc bay, va chạm hay mỗi lần vượt qua chướng ngại vật của chú chim
- Cập nhật điểm số qua từng lượt chơi và có ghi nhận kỉ lục cao nhất qua các lần chơi.

- Source code: DaoDaiChi/Flappy-bird (github.com)

c. Tuần 3:

- sửa lại lỗi sau khi va chạm giữa class bird và class pipe xảy ra nhưng bạn ấn ngay nút space để tiếp tục trò chơi ngay sau đó thì chú chim lại tiếp tục ở vị trí đó và tiếp tục bay và điểm vẫn được tính
- sửa lại lỗi phát âm thanh tăng điểm (tức là âm thanh sẽ được phát ra sau khi chú chim vượt qua chướng ngại vật)
- thêm thanh tiến trình khoảng 30s thì trò chơi sẽ kết thúc (bạn sẽ hoàn thành màn chơi)
- chỉnh lại điều kiện khi nào thì class bird sẽ vượt qua chướng ngại vật, do vẫn gặp 1 số lỗi dù class bird chưa qua chướng ngại vượt thì điểm đã được tăng
- tạo ra layout of level nơi chứa 5 mức độ trò chơi có độ khó khác nhau và:

d. Tuần 4.

- Sửa lỗi gặp phải ghi tạo ra 5 level của game. Đó là không thể chọn skin theo ý muốn.
- Tạo ra nút Autoplay và chế độ AutoPlay giúp trò chơi có thể tự động vận hành mà không cần đến sự điều khiển của người chơi.
- Thêm vào nút Start Game, nút này sẽ xuất hiện đầu tiên khi chạy source code và khi ấn vào thì sẽ chuyển tới giao diện Menu nơi có chứa các nút giúp người dùng tương tác.
- Viết báo cáo + docstring.
- Thêm âm thanh bay của chú chim.
- Source code: DaoDaiChi/Flappy-bird (github.com)

3. Kết quả đạt được.

Đã tạo ra một trò chơi Flappy bird và nâng cấp nó lên nơi người chơi sẽ điều khiển chú chim vượt qua các chướng ngại vật (ống) , qua mỗi chướng ngại vật sẽ được cộng 1 điểm. Có thêm thanh thời gian (khoảng 30s) để khi hết thời gian bạn sẽ thắng. Có thêm chế độ chọn trang phục (skins) cho nhận vật gồm 4 trang phục. Có thêm chế độ

chơi tự động (autoplay) sẽ tự động điều khiển chú chim mà không cần người dùng thao tác.

4. Tài liệu tham khảo
    a. newlinedotco/FlappySwift: swift implementation of flappy bird. More at fullstackedu.com (github.com)
    b. yenchenlin/DeepLearningFlappyBird: Flappy Bird hack using Deep Reinforcement Learning (Deep Q-learning). (github.com)
5. Phụ lục 1: Giới thiệu (demo) kết quả

   Demo
6. Phụ lọc 2: Docstring

   - Class Bird

```
class Bird:
    def __init__(self, screen, initial_skin_index=0,gravity=0.7):
        """
        Khởi tạo Bird với màn hình Pygame, chỉ số skin ban đầu, và trọng lực.

        :param screen: Màn hình Pygame nơi chim được vẽ.
        :param initial_skin_index: Chỉ số của skin chim ban đầu.
        :param gravity: Trọng lực tác động lên chim, mặc định là 0.7.
        """
def rotate_bird(self):
        """
        Xoay chim dựa trên chuyển động hiện tại.

        :return: Hình ảnh chim sau khi xoay.
        """
def bird_animation(self):
        """
        Thay đổi skin của chim theo skin hiện tại và giữ nguyên vị trí.
        """
def update_movement(self):
        """
        Cập nhật chuyển động của chim dựa trên trọng lực.
        """
def change_skin(self, new_skin_index):
        """
        Thay đổi skin của chim theo chỉ số mới.

```

```
        :param new_skin_index: Chỉ số của skin mới.
        """
```

- Class Pipe

```
class Pipe:
    def __init__(self, screen):
        """
        Khởi tạo Pipe với màn hình Pygame và tốc độ di chuyển của đường ống.

        :param screen: Màn hình Pygame nơi các đường ống được vẽ.
        :param pipe_speed: Tốc độ di chuyển của đường ống. Mặc định là 5.
        """
def play_hit_sound(self):
        """
        Phát âm thanh va chạm khi chim chạm vào đường ống.
        """
def create_pipe(self):
        """
        Tạo một cặp đường ống với vị trí ngẫu nhiên.

        :return: Một cặp đường ống (top pipe và bottom pipe).
        """
def move_pipe(self):
        """
        Di chuyển các đường ống sang trái dựa trên tốc độ di chuyển.
        """
def draw(self):
        """
        Vẽ các đường ống lên màn hình.
        """
def draw(self):
        """
        Vẽ các đường ống lên màn hình.
        """
```

- Class Floor

```
 def __init__(self, screen):
        """
        Khởi tạo sàn với màn hình và tốc độ di chuyển.

        :param screen: Màn hình Pygame nơi sàn được vẽ.
        :param speed: Tốc độ di chuyển của sàn. Mặc định là 1.
        """
def update_position(self):
        """
```

```
        Cập nhật vị trí của sàn để tạo hiệu ứng chuyển động.
        """
```

- Class Game:

```
class Game:
    def __init__(self, screen,level_number):
        """
        Khởi tạo trò chơi với màn hình và cấp độ ban đầu.

        :param screen: Màn hình pygame để hiển thị đồ họa.
        :param level_number: Cấp độ của trò chơi, dùng để điều chỉnh trọng
   lực.
        """
def reset_game(self):
        """
        Đặt lại trạng thái trò chơi về trạng thái ban đầu.
        """
def update_high_score(self):
        """
        Cập nhật điểm cao nhất (high score) nếu điểm hiện tại cao hơn.
        """
def play_score_sound(self):
        """
        Phát âm thanh ghi điểm khi chim vượt qua một cột.
        """
 def set_auto_play(self, enable):
        """
        Bật hoặc tắt chế độ tự động chơi.

              True để bật tự động chơi, False để tắt.
              """
     def check_collision(self):
              """
              Kiểm tra xem chim có va chạm với cột hoặc sàn không.

              :return: True nếu có va chạm, False nếu không.
              """
     def auto_play(self):
              """
              =Điều khiển chim tự động vượt qua các cột.
              """
     def update_movement(self):
              """Cập nhật chuyển động của chim."""
     def run_game(self):
              """
```

```
            Chạy vòng lặp chính của trò chơi, xử lý sự kiện và cập nhật giao diện.
            Vòng lặp này kiểm soát toàn bộ trò chơi, từ việc xử lý sự kiện, điều khiển chim,
            đến kiểm tra va chạm và vẽ lên màn hình.
            """
```

- Class Menu

```
    class Menu:
        def __init__(self, screen):
            """
            Khởi tạo Menu với màn hình Pygame và danh sách các tùy chọn.

            :param screen: Màn hình Pygame để hiển thị các tùy chọn menu.
            """
    def choose_bird_skin(self, BLACK = (0,0,0)):
            """
            Hiển thị menu chọn skin của chim và trả về chỉ số của skin được chọn.

            :return: Chỉ số của skin chim được chọn.
            """
    def run(self):
            """
            Chạy vòng lặp chính của menu, hiển thị các tùy chọn và xử lý sự kiện.
            """
```