

BÁO CÁO ĐỒ ÁN MÔN HỌC

Lớp: IT002.O21.TTNT

SINH VIÊN THỰC HIỆN

Mã sinh viên: 23521640

Họ và tên: Ngô Minh Trí

TÊN ĐỀ TÀI: PIRATE ADVENTURE

CÁC NỘI DUNG CẦN BÁO CÁO:

1. Giới thiệu đồ án

a. Mô tả chi tiết về nội dung và các yêu cầu:

- Đây là game 2D platformer với chủ đề cướp biển tìm kho báu, trên hành trình này tên cướp biển phải vượt qua những cạm bẫy chết người và chiến đấu với các quái vật
- Tạo ra các bản đồ cho cuộc hành trình của Hải tặc (gồm 3 bản đồ)
- Xây dựng nhân vật cướp biển với khả năng di chuyển (qua trái, qua phải, nhảy) và tấn công kẻ thù
- Tạo các quái vật (phe đối địch với nhân vật)
- + Tooth: di chuyển qua lại, khi bị nhân vật đánh sẽ quay đầu chạy hướng khác, bị đánh 2 nhát sẽ chết
- + Shell: bắn ra các Pearl khi nhận thấy có nhân vật nằm trong tầm hoạt động của mình, Shell không thể bị chết, đối với Pearl nhân vật có thể tấn công để Pearl quay trở về Shell
- Thiết lập các tính năng tương tác, giao diện cho game
- + Các vật phẩm để thu thập trên hành trình của cướp biển : silver và diamond để tăng thêm tiền cho cướp biển, nếu thu thập đủ 10000 tiền thì có thể đổi lấy một mạng, skull thu thập được sẽ tăng thêm một mạng

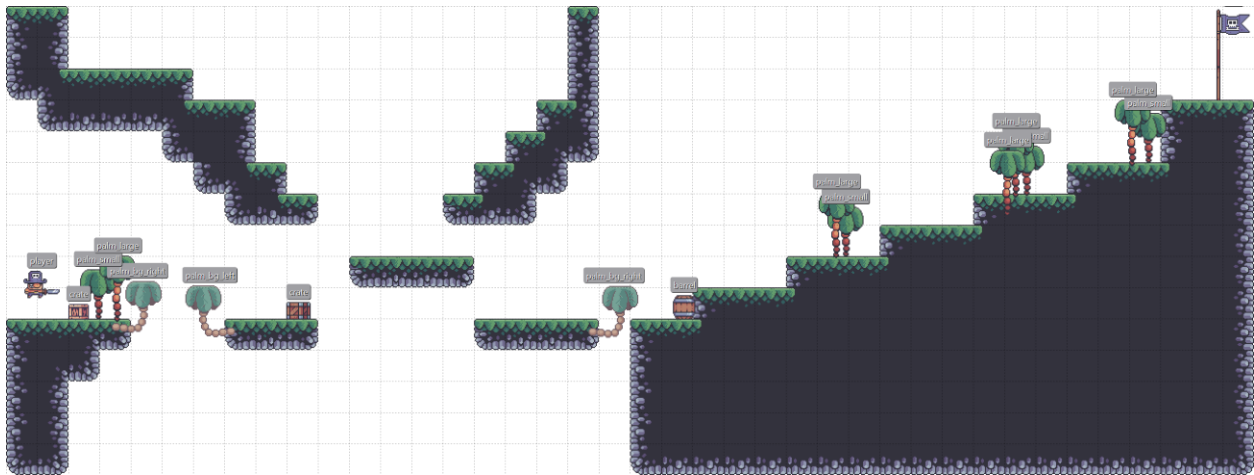
- + Thêm đồ họa cho các đối tượng trong trò chơi
- Thiết lập hiệu ứng âm thanh cho trò chơi

b. Link github: <https://github.com/chisngooo/IT002-PirateAdventure>

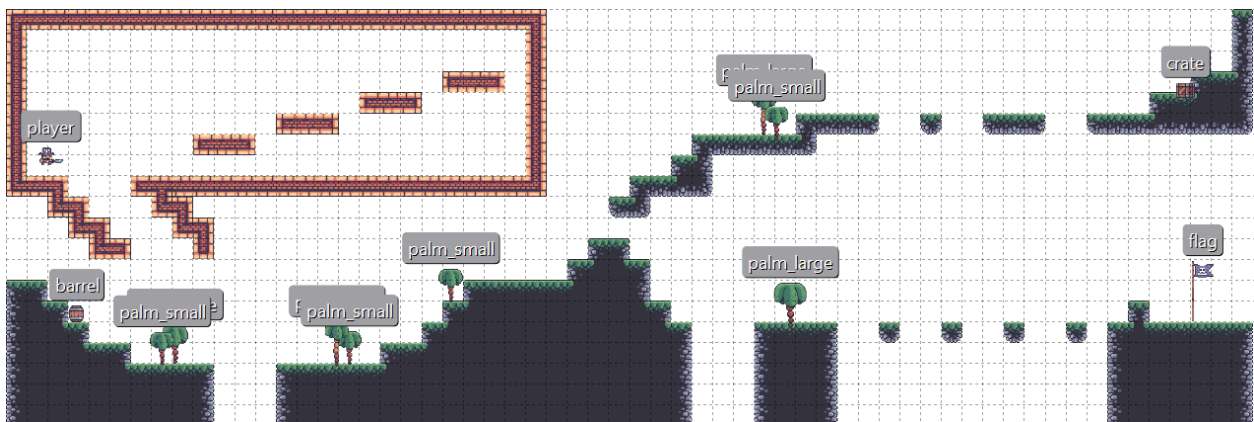
2. Quá trình thực hiện

a. Tuần 1:

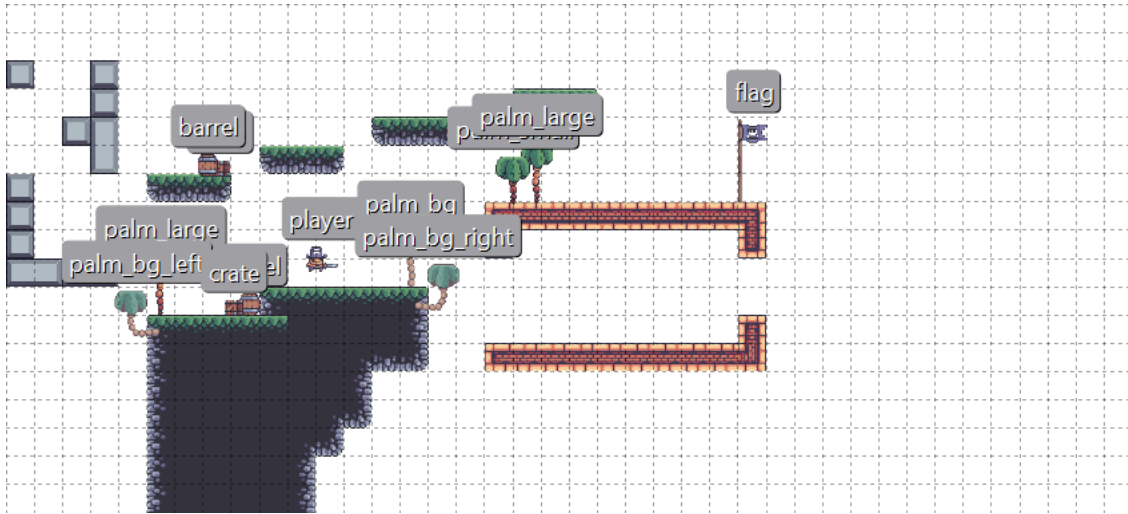
- Sử dụng asset pack: [TREASURE HUNTERS](#).
- Sử dụng phần mềm Tiled để vẽ 3 bản đồ cho trò chơi. Đây mới là phần khung của trò chơi gồm **Tile layer “Terrain”** và **Object layer “Objects”**



Bản đồ 1



Bản đồ 2



Bản đồ 3

- Làm đồ họa cho Menu bằng Photoshop và Tiled



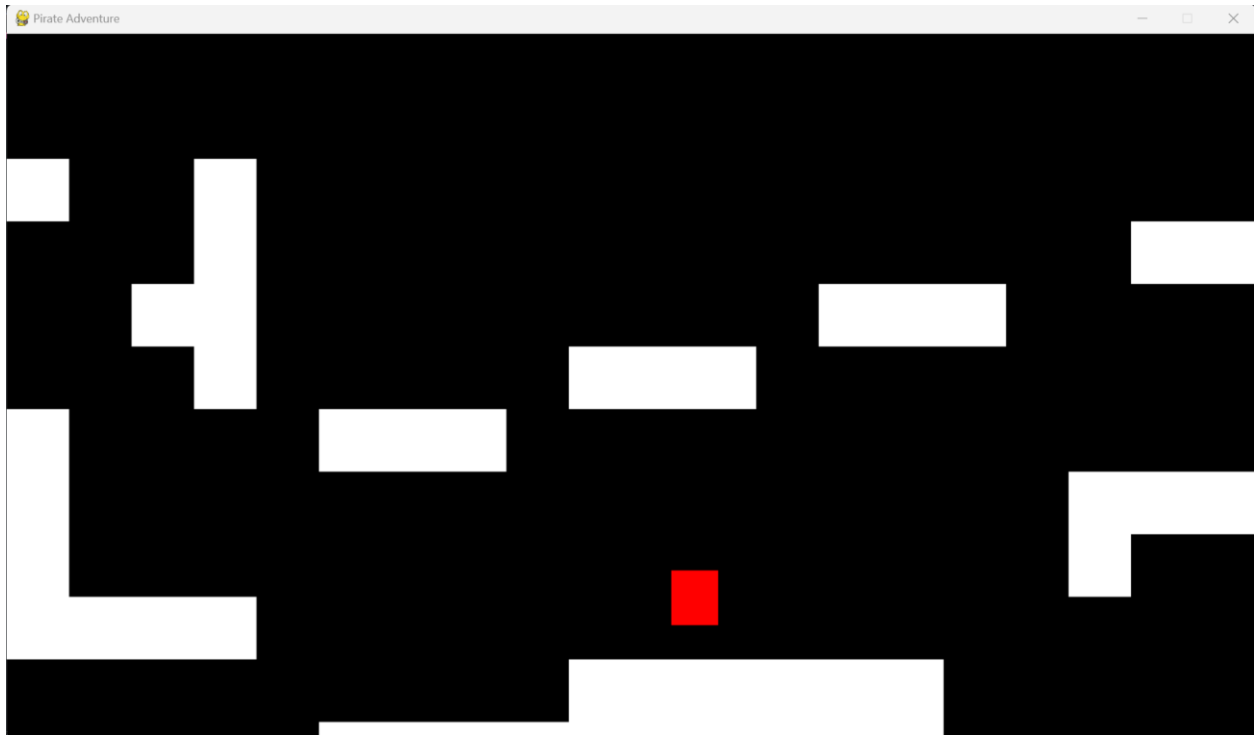
Start Menu



Tutorial Menu

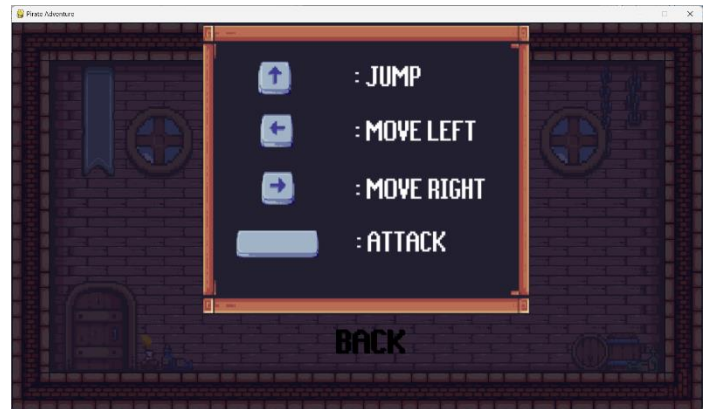
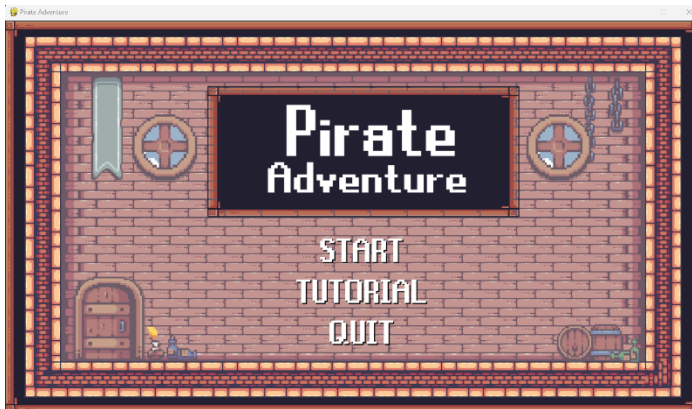
- Lập trình những bước đầu tiên cho **“Pirate Adventure”**
 - + Kích thước màn hình hiển thị trò chơi: 1280 x 720, nền trò chơi: Đen
 - + Sử dụng **Bản đồ 3** là bản đồ chính (chưa chuyển giữa các bản đồ với nhau được và **Bản đồ 3** có đầy đủ các layer để chạy thử nhất)

+ Tạo đồ họa cơ bản cho game gồm các khối chữ nhật với khối màu trắng là tile layer **Terrain** và khối màu đỏ là object layer **player**

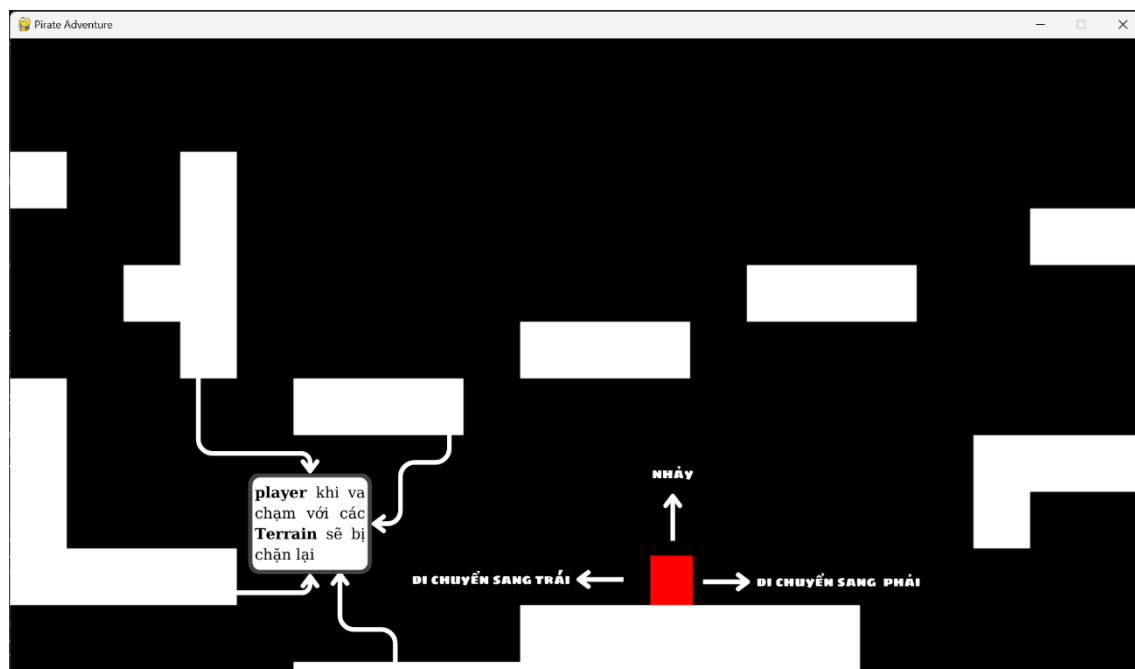


+ Tạo Menu trò chơi gồm 3 nút:

- **START:** Bắt đầu trò chơi
- **TUTORIAL:** Hướng dẫn chơi khi nhấn vào sẽ hiện màn hình hướng dẫn, khi đọc xong nhấn **BACK** để quay về Menu chính
- **QUIT:** Thoát khỏi trò chơi



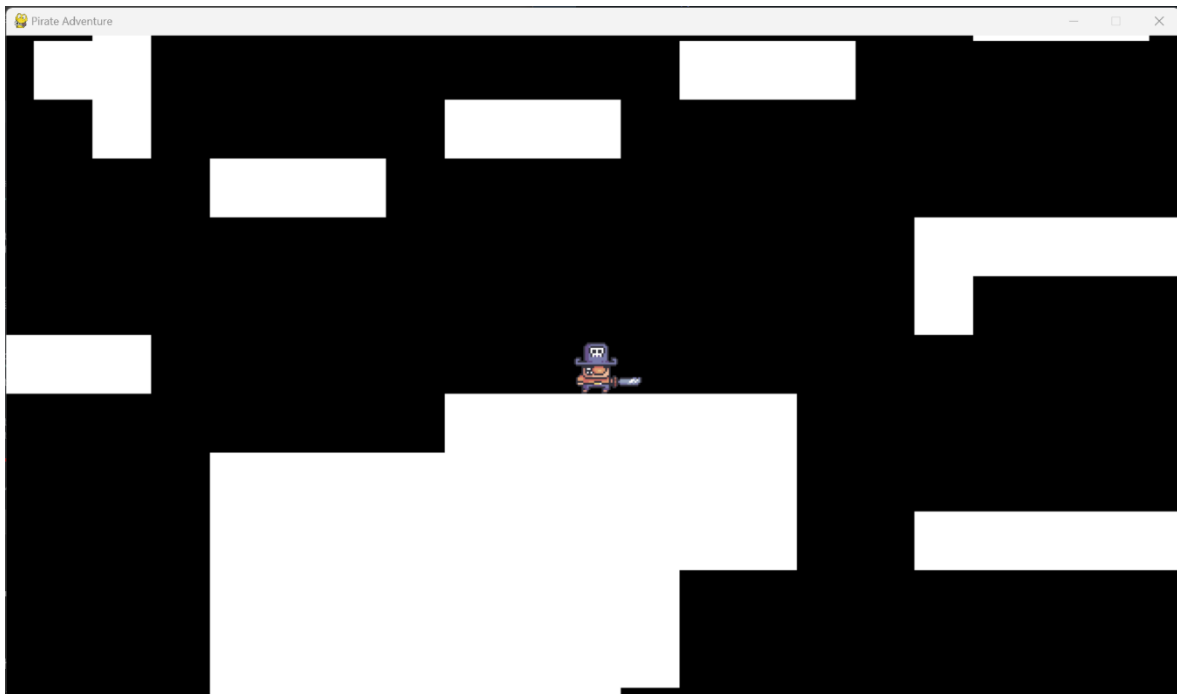
- Tạo cơ chế di chuyển cho **player**: với 3 cử động đi qua trái, qua phải, nhảy lên bằng nút di chuyển như hướng dẫn trên hình
- Thiết lập tính vật lí cho trò chơi:
 - + **Player** va chạm với các **Terrain** và bị chặn lại
 - + Trọng lực cho **player**
 - + Tốc độ di chuyển, độ rơi theo thời gian của **player**



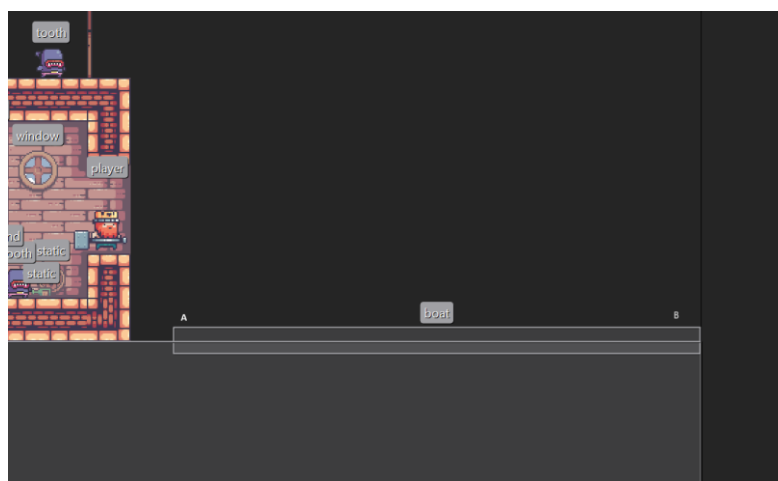
- Học cách sử dụng Github

b. Tuần 2:

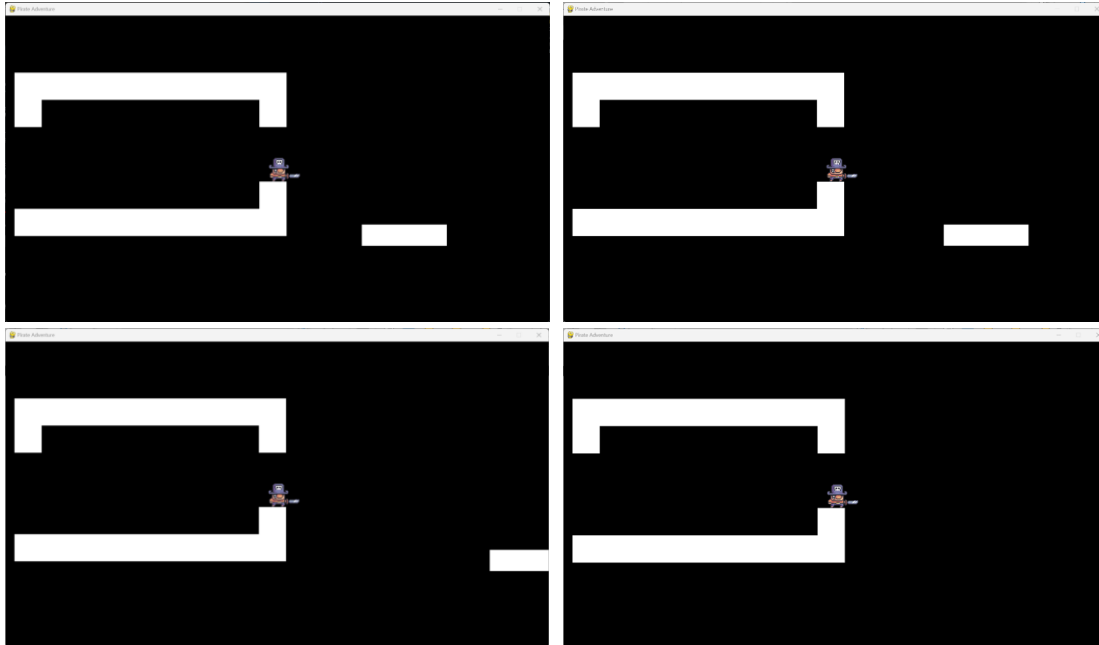
- Thêm đồ họa cho **player** (một tấm ảnh cho mọi chuyển động) đồng thời tạo camera luôn đặt **player** ở chính giữa màn hình, **player** có thể di chuyển xung quanh bản đồ mà không bị biến mất nữa



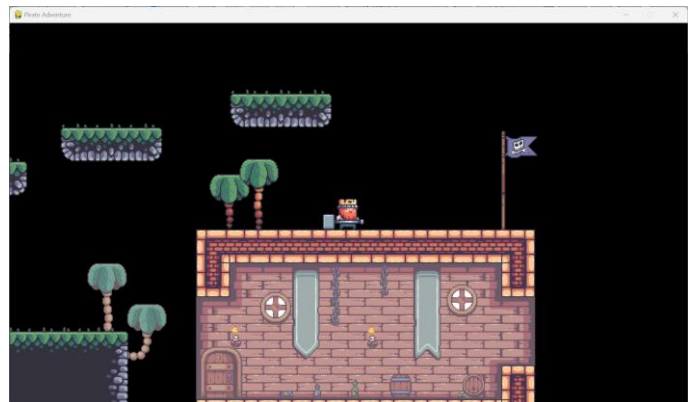
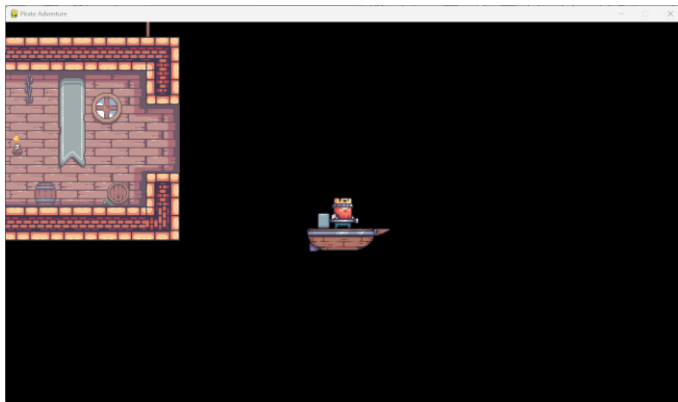
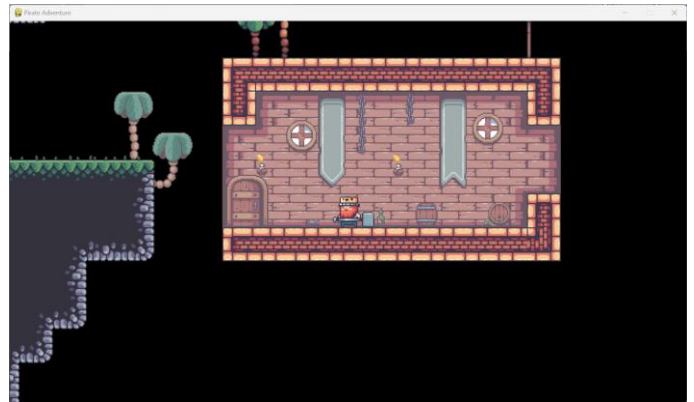
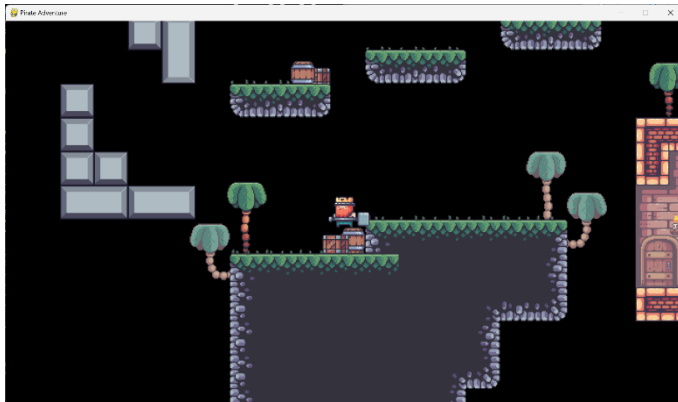
- Tạo lớp **MovingSprite** kế thừa từ lớp **Sprite** tạo ra các sprite có thể di chuyển (thuyền)
 - + Cài đặt vùng di chuyển của **MovingSprite** đi từ A đến B trong **Tiled**



- + Xử lý va chạm của **Moving Sprite** với **player**
- + Xử lý chuyển động của **player** trên **Moving Sprite** (**player** sẽ chuyển động theo **Moving sprite**)



- Tạo lớp **AnimatedSprite** kế thừa từ lớp **Sprite** tạo ra các sprite có animation, thay đổi đối tượng kế thừa của **Moving Sprite** thành **AnimatedSprite**
- Thiết lập đồ họa cho game và chia các thiết lập đồ họa thành 3 loại:
 - + **Sprite**: gồm các đối tượng đồ họa trong trò chơi với các thiết lập cơ bản
 - + **AnimatedSprite (kế thừa từ Sprite)**: gồm các đối tượng đồ họa với các thiết lập cơ bản và có animation
 - + **MovingSprite (kế thừa từ MovingSprite)**: gồm các đối tượng đồ họa với các thiết lập cơ bản, có animation và có thể di chuyển được
- Thiết lập thứ tự tạo ra đồ họa cho từng loại đồ họa khác nhau



- Thiết lập đồ họa cho **player**:
- + **idle**:



- + **fall**:



+ **attack:** tạo cơ chế tấn công cho player bằng cách nhấn phím **SPACE**



+ **jump:**



+ **run:**



c. Tuần 3:

- Tạo enemies:

+ **Tooth:** với khả năng di chuyển xung quanh các khối vật thể xác định, nếu đi hết khối vật thể, Tooth sẽ quay lại. Ngoài ra, Tooth khi bị người chơi tấn công, lần đầu tiên Tooth sẽ chạy ngược lại hướng đang di chuyển, và lần thứ hai Tooth sẽ chết. Tooth cũng sẽ tấn công và làm người chơi mất máu khi đi ngang qua người chơi



Tooth ở trạng thái chuyển động



Tooth chết khi bị tấn công lần thứ 2

+ **Shell, Pearl:** luôn duy trì vị trí cố định và không thể giết bởi người chơi. Khi nhận ra có người chơi trong phạm vi hoạt động của mình. Shell sẽ bắn ra các Pearl tiến tới người chơi, Pearl có thể bị phản lại nếu người chơi tấn công đúng lúc.



Shell ở trạng thái bình thường





Pearl



Shell ở trạng thái tấn công, và Pear sẽ được nhả ở hình số 4

- Tạo hiệu ứng va chạm giữa **Enemies** với **Player** (va chạm giữa Pearl và vật thể, va chạm giữa player và tooth, tooth bị tấn công, player bị mất máu)
- Sửa lỗi ở class **Game**: Khi ở Menu thì các vật thể trong game vẫn chuyển động gây ra lỗi cho người chơi và enemies
- Tạo class **Data** để lưu trữ số tiền thu thập được và mạng cho người chơi (số tiền ban đầu là 0, số mạng ban đầu là 3)
- Tạo class **UI** để hiển thị số mạng và số mạng lên màn hình
- Thêm các vật phẩm người chơi có thể nhặt đặt vào trong trò chơi:

+ Gold  : tăng 1 coin nhận được

+ Diamond  : tăng 15 coin nhận được

+ Skull  : tăng 1 mạng

d. Tuần 4:

3. Kết quả đạt được:

a. Các bản thiết kế

- Bản thiết kế UML:
- Bản thiết kế bản đồ:
- Bản thiết kế Menu:

b. Sản phẩm cuối cùng

4. Tài liệu tham khảo

- Code: [Creating an amazing 2D platformer in Python \[SNES inspired\]](#)
- Sử dụng Tiled: [A guide to level creation with Tiled \[Use with pygame\]](#)

5. Phụ lục: docstring