

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:

- **Lối chơi:** Tựa game "Pirate Adventure" thuộc thể loại 2D Platformer, đưa người chơi vào vai một tên cướp biển gan dạ trên hành trình tìm kiếm kho báu huyền thoại của biển cả. Trong cuộc phiêu lưu đầy kịch tính này, người chơi sẽ phải đối mặt với vô số thách thức, từ những cạm bẫy chết người đến những quái vật bí ẩn tấn công bất ngờ khi tên cướp biển đặt chân lên các hòn đảo kỳ bí. Mục tiêu chính của người chơi là thu thập càng nhiều đồng tiền vàng càng tốt, vượt qua những hòn đảo nguy hiểm và tiếp tục hành trình đầy gian nan để trở thành hải tặc vĩ đại và giàu có.
- **Bản đồ cuộc hành trình:** Tạo ra các bản đồ cho cuộc hành trình của tên cướp biển (gồm 3 bản đồ) tương ứng với các hòn đảo mà tên cướp biển sẽ đặt chân lên
- **Thiết lập đồ họa trò chơi:** các bản đồ sẽ có mức thiết lập tương tự nhau để dễ dàng thao tác và xây dựng với các layer khác nhau ứng với các chức năng khác nhau, trong mỗi layer đều có các vật thể khác nhau được đặt tên xác định

– **Xây dựng nhân vật cướp biển:** với khả năng di chuyển linh hoạt (qua trái, qua phải, nhảy) xung quanh đảo cũng như chống trả lại cái quái vật (tấn công quái vật). Tên cướp biển sẽ có 3 mạng để sinh tồn và một chiếc túi đựng những đồng vàng mình đã thu thập được trong quá trình phiêu lưu.

– **Tạo các quái vật (phe đối địch với nhân vật)**

+ **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

+ **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.

– **Cài đặt các tính năng tương tác, giao diện cho game (UI, UX)**

+ Thêm các tính năng tương tác với người chơi (Start Menu, Game Over Menu, giữa các vật thể với nhau)

+ Thêm nhạc nền cho trò chơi

+ Sound effect cho cướp biển và cho quái vật

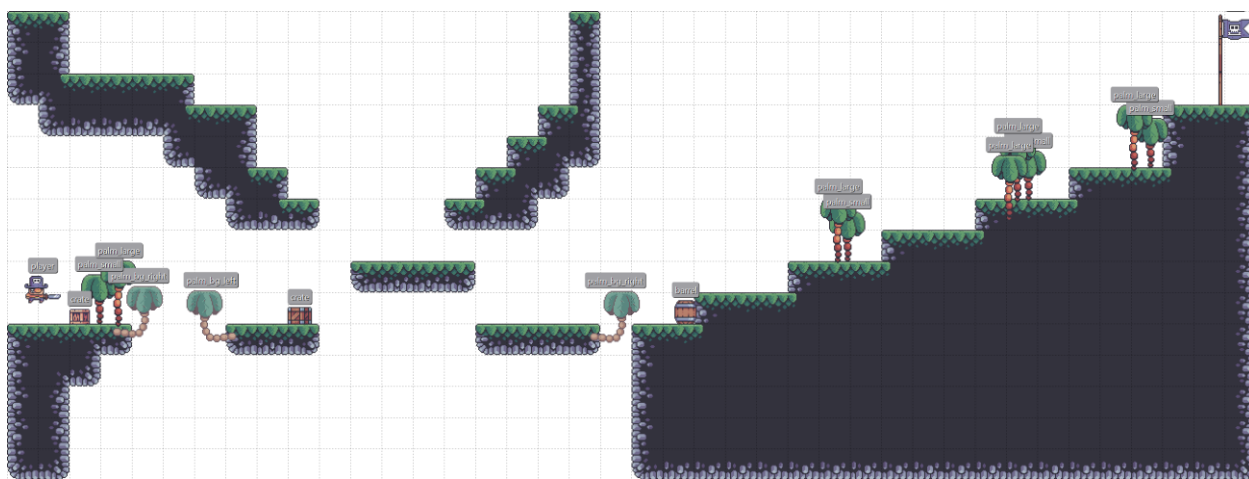
+ Cài đặt âm thanh cho trò chơi

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

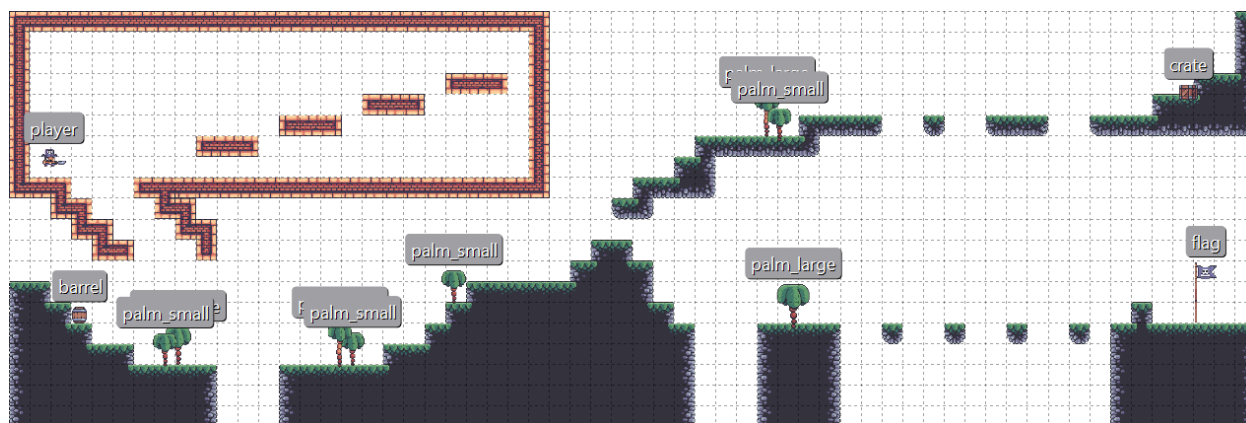
2. Quá trình thực hiện

a. Tuần 1:

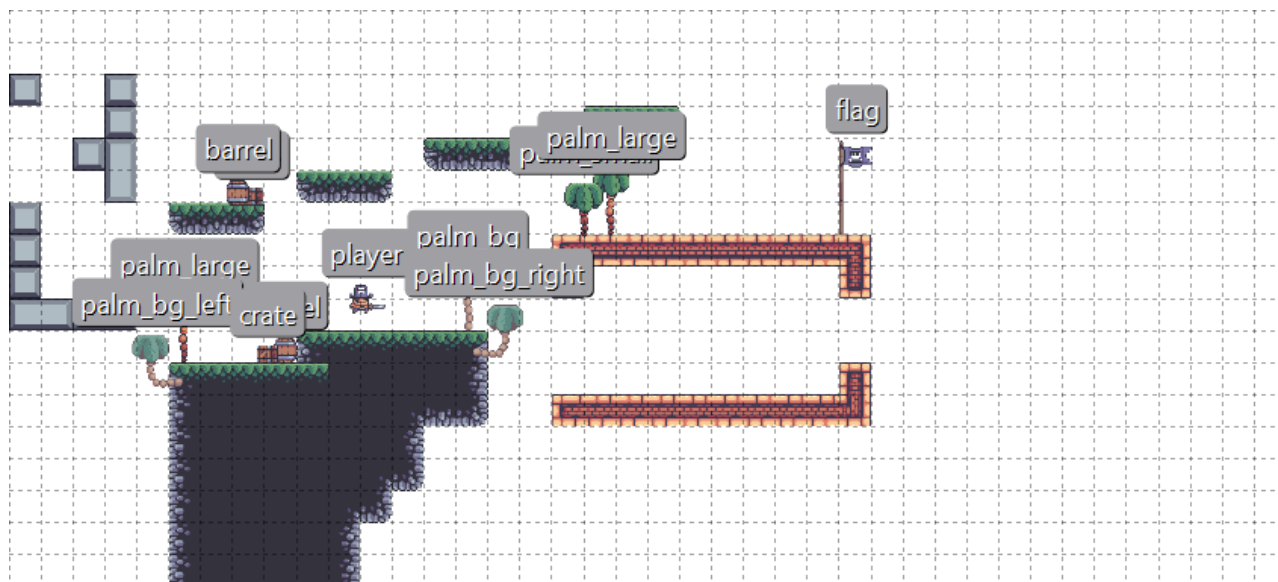
- Sử dụng asset pack: [TREASURE HUNTERS](#).
- Sử dụng phần mềm Tiled để vẽ 3 bản đồ cho trò chơi. Đây mới là phần khung của trò chơi gồm **Tile layer “Terrain”** và **Object layer “Objects”**



Bản đồ 1



Bản đồ 2



Bản đồ 3

- Làm đồ họa cho Menu bằng Photoshop và Tiled

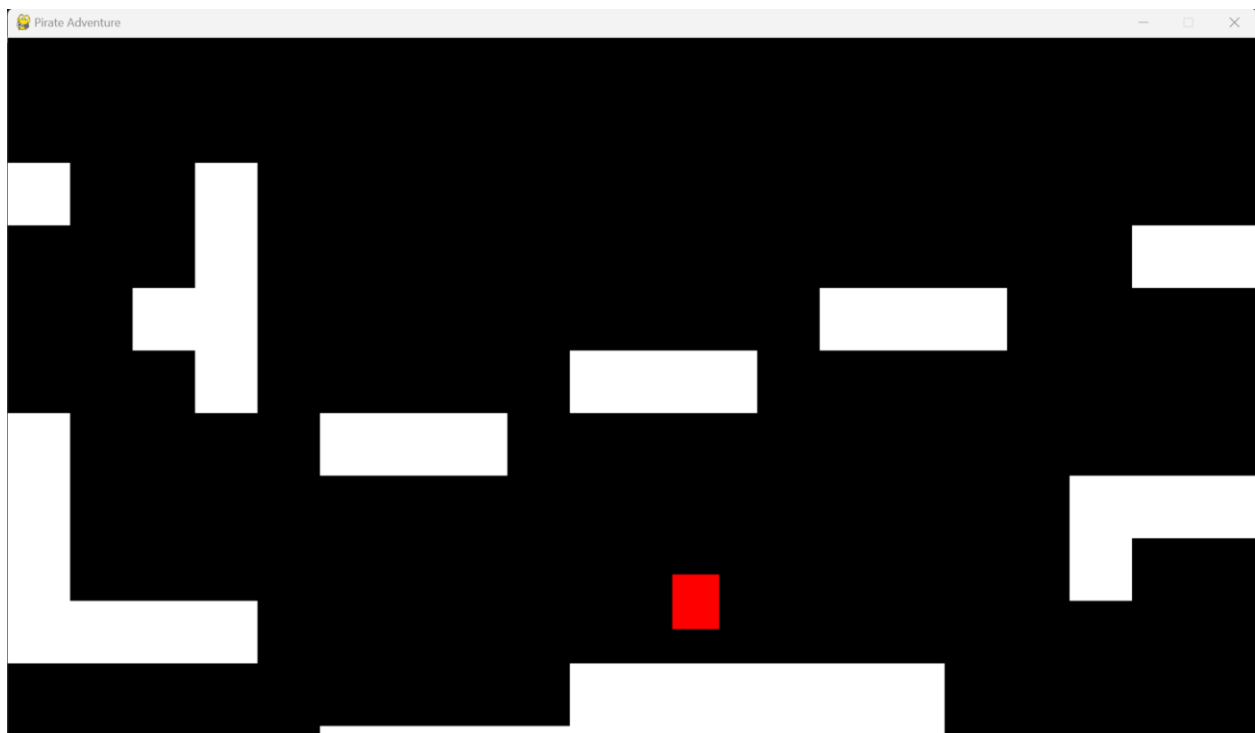


Start Menu



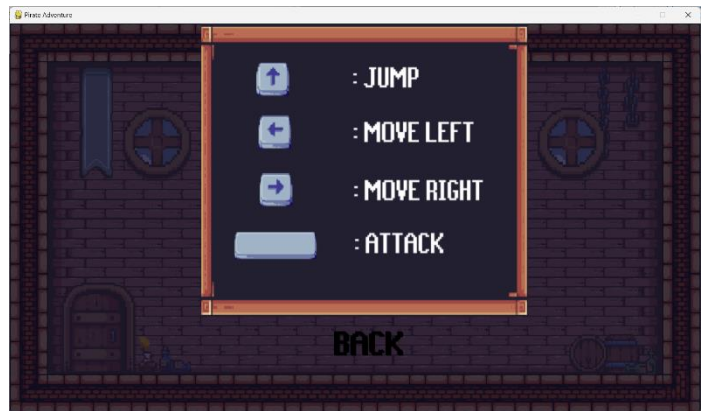
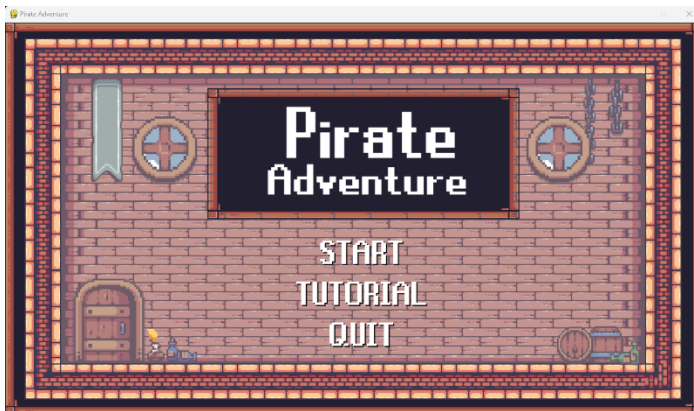
Tutorial Menu

- Code phần khung cho trò chơi “**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 nhất để chạy thử trò chơi)
 - + 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**

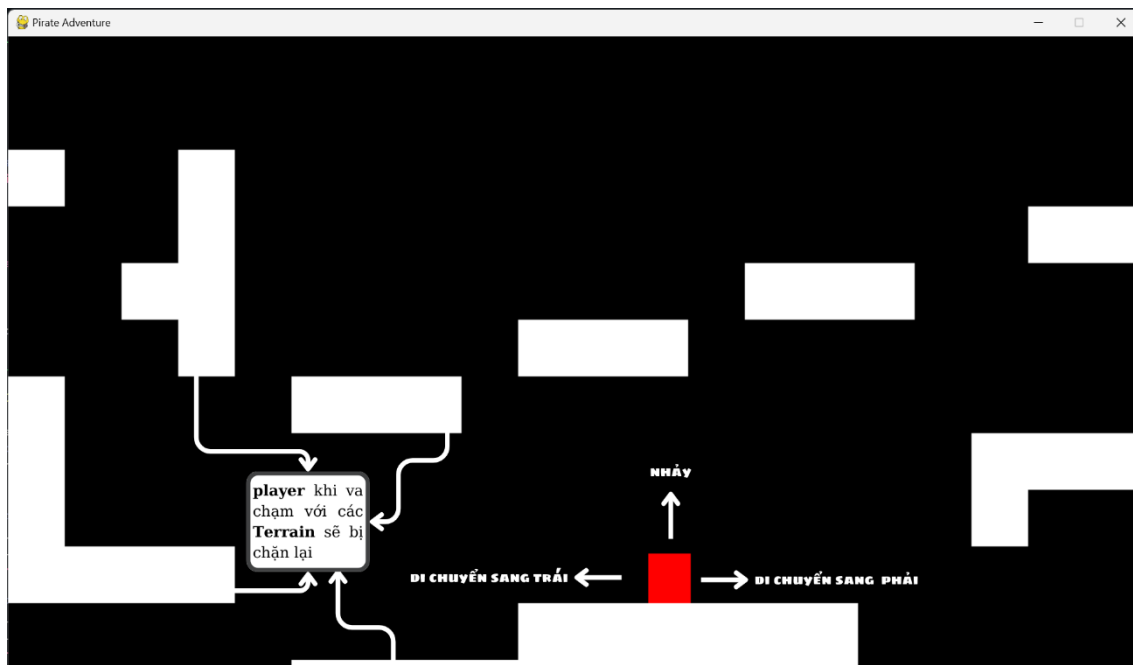


+ Tạo Menu trò chơi gồm 3 nút:

- **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 năng 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:

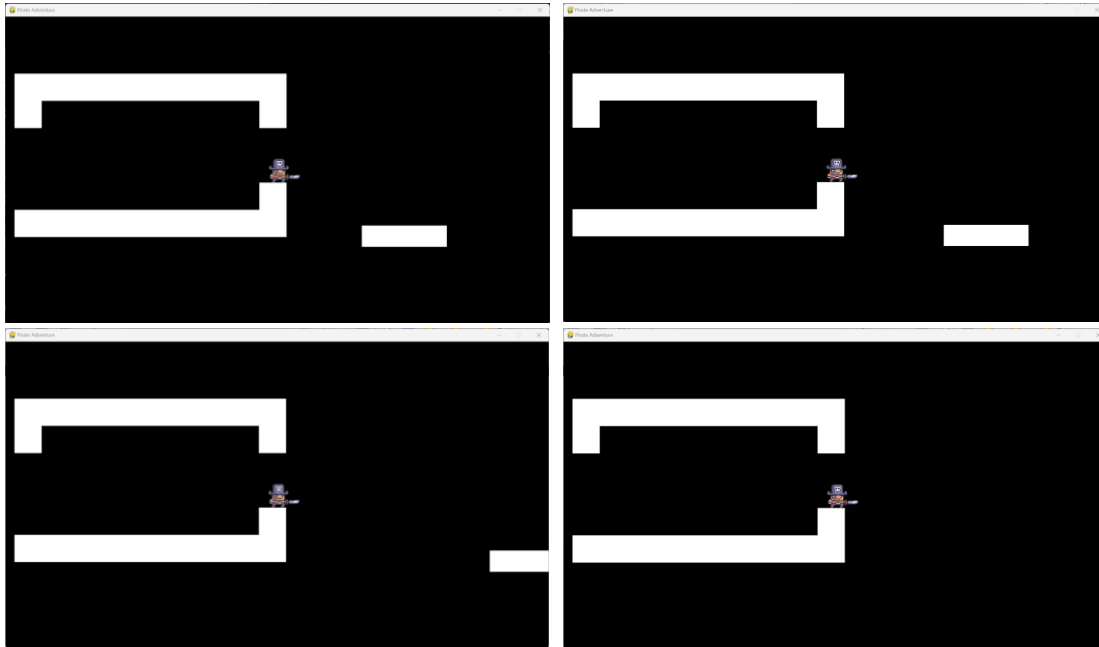
- Thêm đồ họa cho **player** (một tấm ảnh cho mọi chuyển động), đồng thời tạo camera luôn đặt **player** ở chính giữa màn hình, **player** có thể di chuyển xung quanh bản đồ mà không bị biến mất nữa



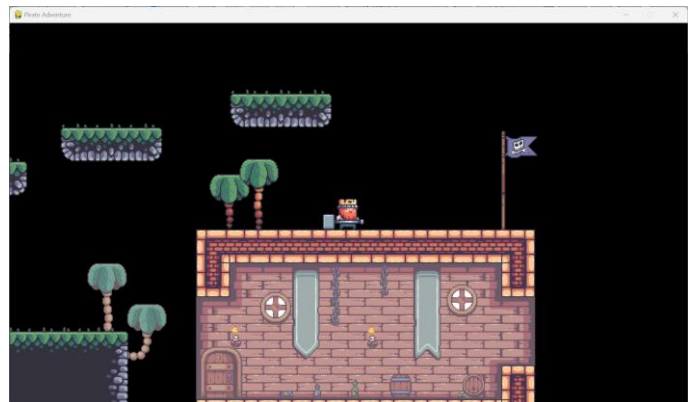
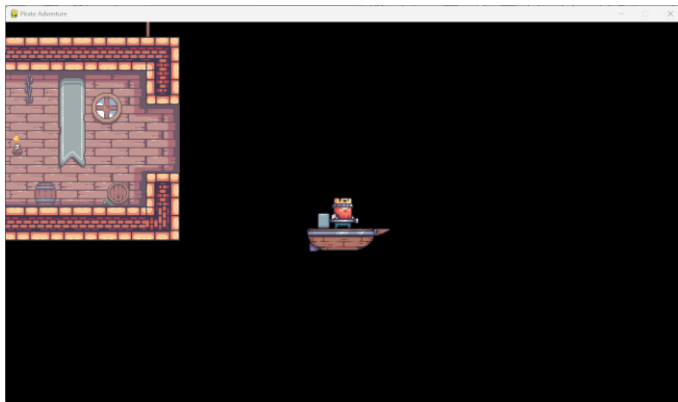
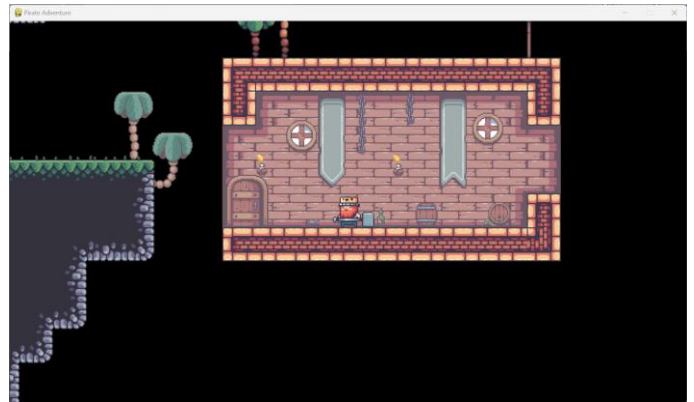
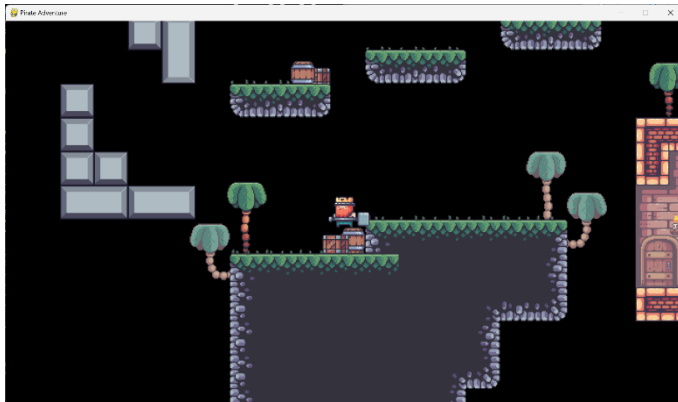
- Tạo lớp **MovingSprite** kế thừa từ lớp **Sprite** tạo ra các sprite có thể di chuyển
- + Cài đặt vùng di chuyển của **MovingSprite** đi từ A đến B trong **Tiled**



- + Xử lý va chạm giữa **Moving Sprite** với **player**
- + Xử lý chuyển động của **player** trên **Moving Sprite** (**player** sẽ chuyển động theo **Moving sprite** khi đứng trên 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
- Cài đặt thứ tự ưu tiên 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

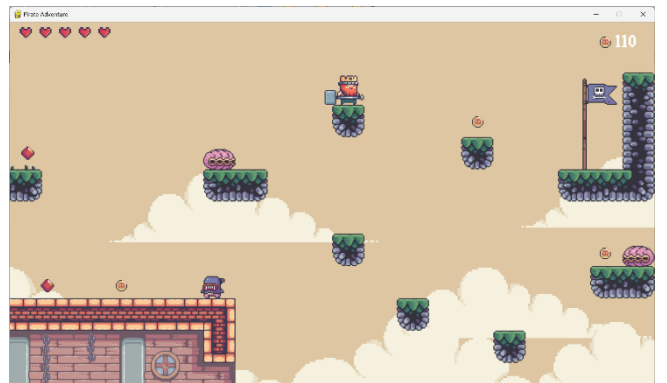
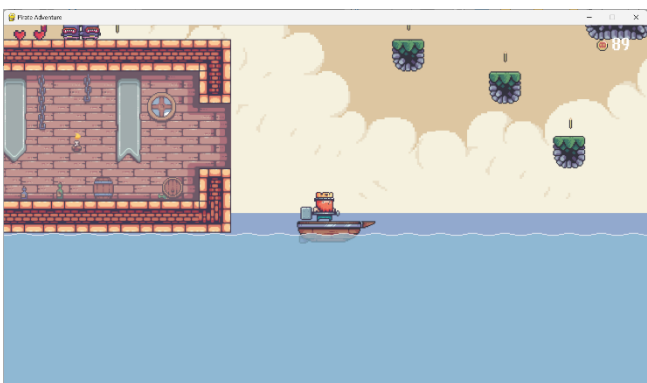
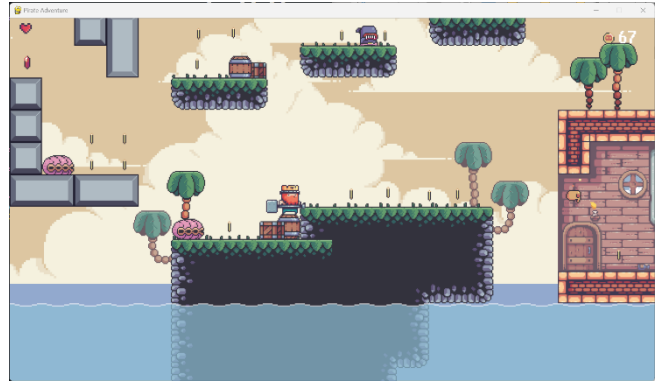
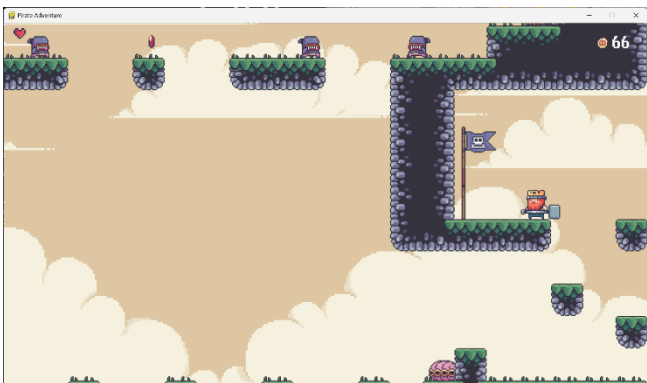
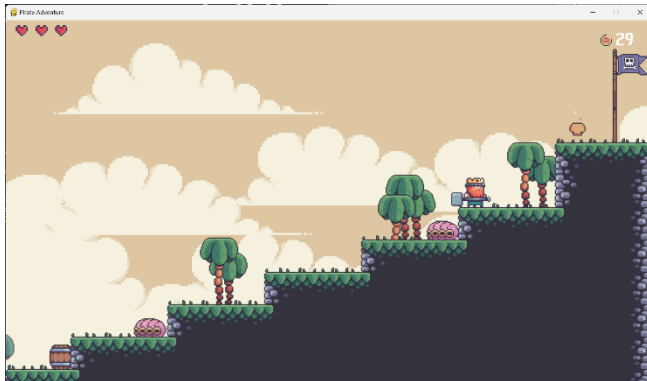


Shell ở trạng thái tấn công, và Pear sẽ được nhả ở hình số 4

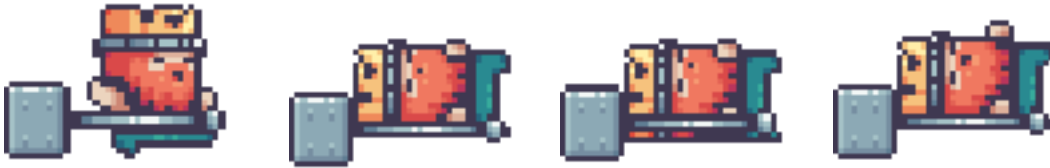
- Tạo hiệu ứng va chạm giữa **Enemies** với **Player** (va chạm giữa Pearl và vật thể, va chạm giữa player và tooth, tooth bị tấn công, player bị mất máu)
- Sửa lỗi ở class **Game**: Khi ở Menu thì các vật thể trong game vẫn chuyển động gây ra lỗi cho người chơi và enemies
- Tạo class **Data** để lưu 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:

- Tạo background cho trò chơi gồm nước, đường chân trời, mây lớn, mây nhỏ, bầu trời
- Cài đặt di chuyển giữa các bản đồ khi người chơi chạm vào cờ



- Tạo hiệu ứng chết cho **player**:



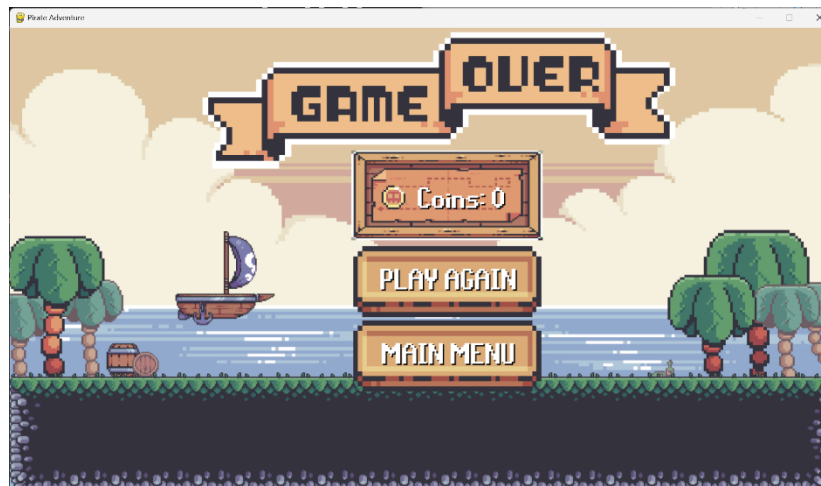
- Hiển thị số coins thu thập được và số mạng còn lên màn hình



- Thêm màn hình **Game over** khi người chơi chết, rơi xuống biển hoặc hoàn thành trò chơi (gồm hai nút bên vòm phía trên hiển thị số coin thu thập được trước khi chết):

+ **Main Menu:** quay trở lại Menu bắt đầu game

+ **Play Again:** chơi lại trò chơi từ ban đầu



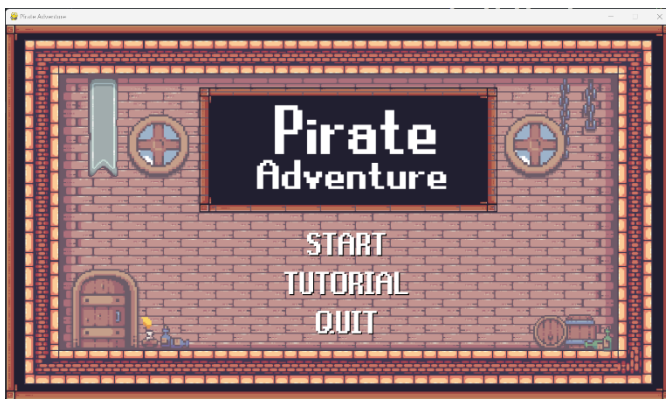
- Thiết lập âm thanh cho các hành động và chủ thể xác định
- Cài đặt nhạc cho trò chơi
- Hoàn thành trò chơi, chạy thử và kiểm tra lỗi

3. Kết quả đạt được:

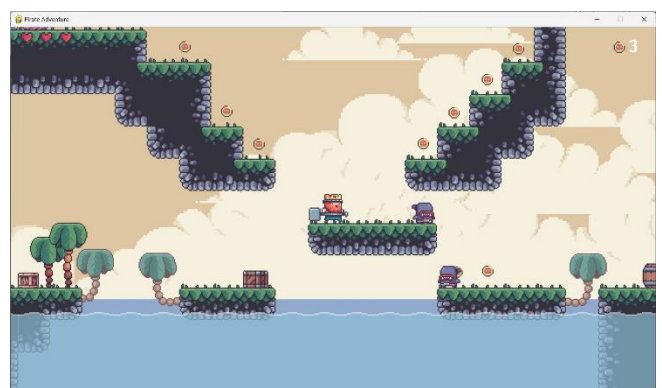
a. Các bản thiết kế: nằm trong folder “BAN_THIET_KE”

- Bản thiết kế UML:
- Bản thiết kế bản đồ:
- Bản thiết kế Start Menu:
- Bản thiết kế Game Over Menu:

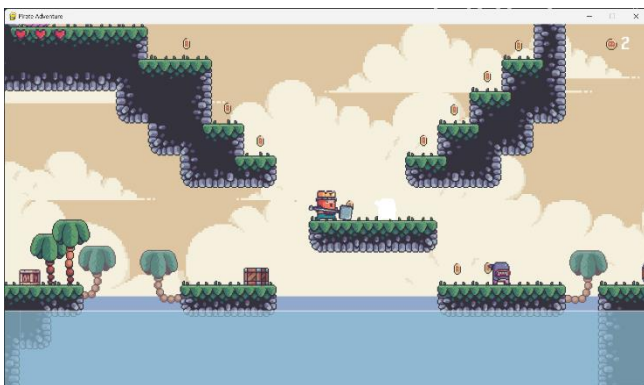
b. Sản phẩm cuối cùng:



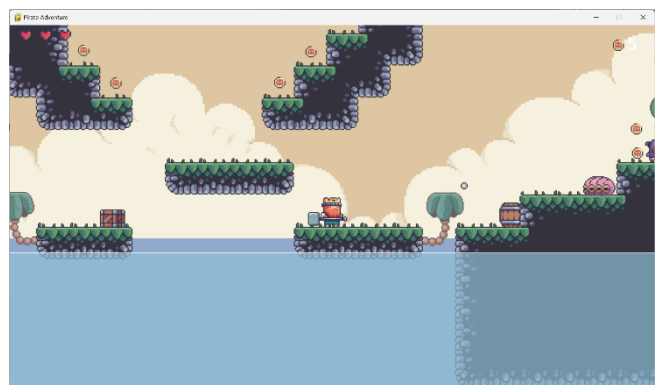
Start Menu



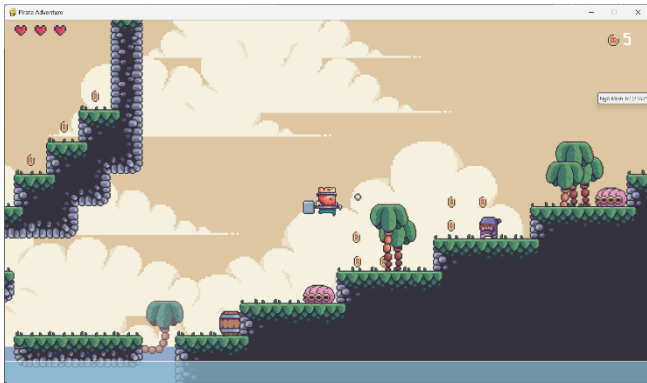
Map 1: nhật vàng



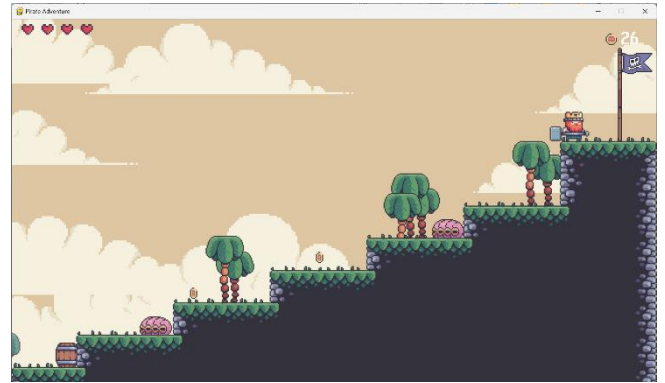
Map 1: đánh tooth



Map 1: shell tấn công



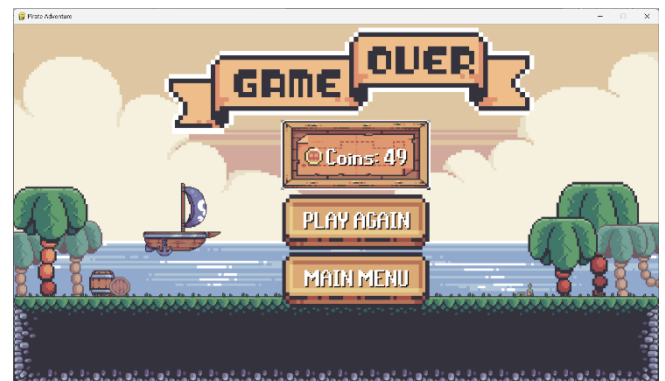
Map 1: đánh phản lại Pearl



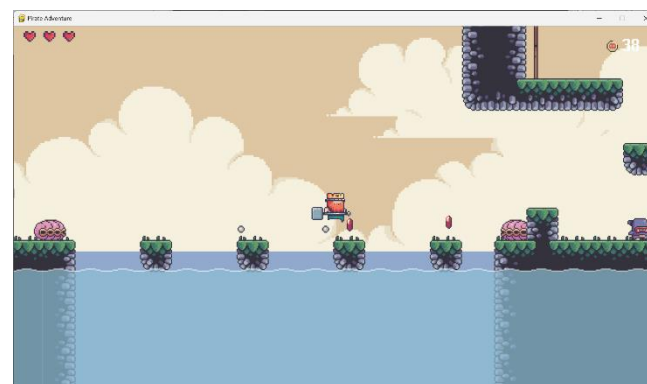
Map 1: chạm vào cờ sẽ đến map 2



Map 2: ăn đầu lâu tăng 1 mạng (trái tim)



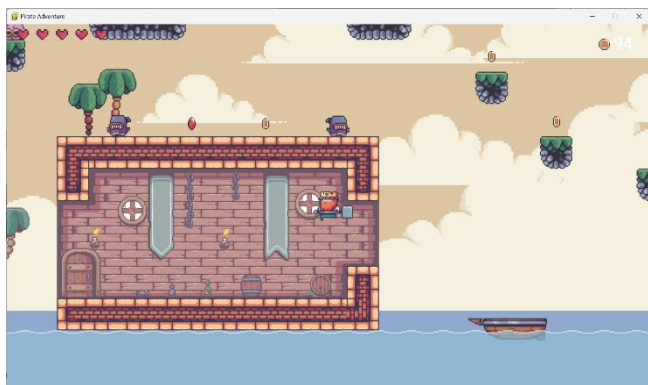
Map 2: rớt xuống nước chết, phải chơi lại từ đầu



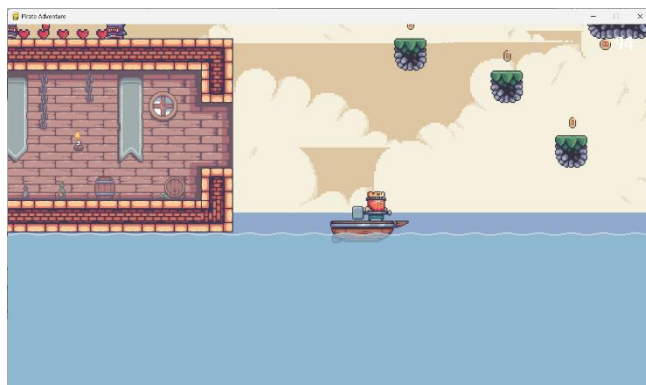
Map 2: các cạm bẫy chết người



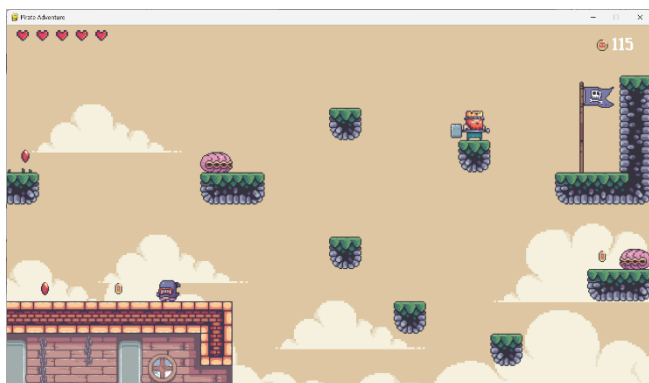
Map 2: trong lâu đài



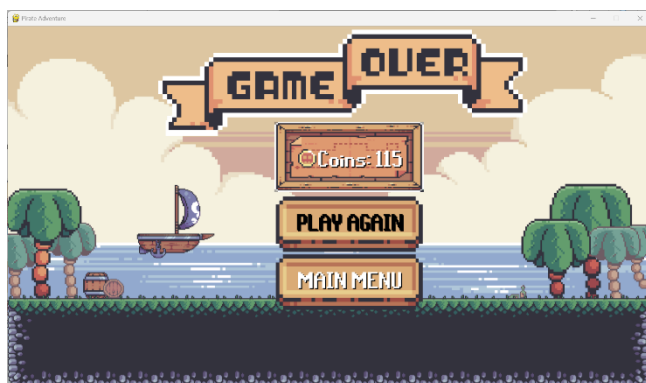
Map 3: tiếp tục phiêu lưu với map 3



Map 3: di chuyển bằng thuyền



Map 3: di chuyển đến cờ và hoàn thành trò chơi



Map 3: màn hình kết thúc trò chơi

***Đây chỉ là phần demo trò chơi bằng hình ảnh, để có trải nghiệm chân thật, thú vị hơn có thể download tại đây [github](#) và chạy file main.py (cần chạy đúng đường dẫn)**

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

<p>class Game(builtins.object)</p> <p> Methods defined here:</p> <p> <code>__init__(self)</code> Khoi tao tro choi voi cac thiet lap co ban bao gom cua so tro choi, cac bien thoi gian va cac ban do tro choi</p> <p> <code>draw_button(self, text, button, font_size)</code> Ve nut voi van ban va mau duoc chi dinh</p> <p> <code>draw_coins(self)</code> Ve so coins da thu thap duoc len man hinh</p> <p> <code>import_assets(self)</code> Import cac assets cua tro choi tu cac folder hoac tu cac folder lon chua cac folder con va trong cac folder co cac hinh anh duoc danh so thu tu de tao animation cho sprite hoac co the la ca font chu</p> <p> <code>reset_game(self)</code> Reset game</p> <p> <code>reset_to_first_map(self)</code> Dat lai tro choi ve ban do dau tien di cung voi viec reset cac thong so co ban cua tro choi (reset game())</p> <p> <code>run(self)</code> Chay tro choi</p> <p> <code>show_game_over_menu(self)</code> Hien thi Menu Game Over va xu ly cac su kien voi thao tac chuot</p> <p> <code>show_menu(self)</code> Hien thi Menu tro choi va xu ly cac su kien voi thao tac chuot</p>	<p>class Level(builtins.object)</p> <p> <code>Level(tmx_map, level_frames, audio_files, data)</code></p> <p> Khoi tao va quan ly moi truong tro choi, bao gom cac thiet lap co ban, cap nhat lien tuc cac su kien va xu ly tuong tac giua nguoi choi va tro choi.</p> <p> Methods defined here:</p> <p> <code>__init__(self, tmx_map, level_frames, audio_files, data)</code> Initialize self. See help(type(self)) for accurate signature.</p> <p> <code>attack_collision(self)</code> xu ly va cham giua nguoi choi va cac sprite co the bi tan cong</p> <p> <code>check_constraint(self)</code> Kiem tra rang buoc cua nguoi choi trong tro choi, neu nguoi choi cham vao bien thi nguoi choi se khong the di tiep, neu nguoi choi rot xuong bien thi nguoi choi se chet, neu nguoi choi cham vao co thi level se hoan thanh</p> <p> <code>create_pearl(self, pos, direction)</code> tao ra mot Pearl theo vi tri va huong di chuyen duoc chi dinh</p> <p> <code>hit_collision(self)</code> xu ly va cham giua nguoi choi va cac sprite co the gay sat thuong</p> <p> <code>item_collision(self)</code> xu ly va cham giua nguoi choi va cac Item</p> <p> <code>pearl_collision(self)</code> xu ly va cham giua Pearl va cac sprite khac</p> <p> <code>run(self, data_time)</code> Vong lap chay tro choi</p> <p> <code>setup(self, tmx_map, level_frames)</code> Thiet lap moi truong cho tro choi</p>
<p>class Player(pygame.sprite.Sprite)</p> <p> <code>Player(pos, groups, collision_sprites, frames, data)</code></p> <p> Tao lop Player ke thua tu lop Sprite cua pygame</p> <p> Day la lop tao ra nhan vat chinh trong tro choi, xu ly cac dac diem ve vi tri, hinh anh, va cham va tuong tac cua nhan vat voi moi truong tro choi.</p> <p> Method resolution order:</p> <p> Player</p> <p> pygame.sprite.Sprite</p> <p> builtins.object</p> <p> Methods defined here:</p> <p> <code>__init__(self, pos, groups, collision_sprites, frames, data)</code> Khoi tao nhan vat chinh voi vi tri va cac thuoc tinh co ban</p> <p> <code>animate(self, data_time)</code> Xu ly animation cua nhan vat</p> <p> <code>attack(self)</code> Xu ly hanh dong tan cong cua nhan vat</p> <p> <code>check_contact(self)</code> Kiem tra xem nhan vat co dang tiep xuc voi mat dat hay khong</p> <p> <code>collision(self, axis)</code> Xu ly va cham giua nhan vat va cac vat the khac trong moi truong tro choi</p> <p> <code>flicker(self)</code></p>	<p>class Sprite(pygame.sprite.Sprite)</p> <p> <code>Sprite(pos, surf=<Surface(64x64x32)>, groups=None, z=2)</code></p> <p> Khoi tao lop Sprite ke thua tu lop Sprite cua pygame</p> <p> Lop nay tao ra cac doi tuong do hoa trong tro choi nhu nhan vat, vat pham, moi truong, v.v...</p> <p> Method resolution order:</p> <p> Sprite</p> <p> pygame.sprite.Sprite</p> <p> builtins.object</p> <p> Methods defined here:</p> <p> <code>__init__(self, pos, surf=<Surface(64x64x32)>, groups=None, z=2)</code> Khoi tao sprite voi vi tri, hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh</p> <p> -----</p> <p> Methods inherited from pygame.sprite.Sprite:</p>

<pre> Tao hieu ung nhap nhay cho nhan vat khi nhan vat nhan sat thuong, trong thoi gian nay nhan vat se khong nhan sat thuong nua get_damage(self) Xu ly khi nhan vat nhan sat thuong get_state(self) Lay trang thai cua nhan vat input(self) Xu ly du lieu dau vao tu ban phim. Neu nguoi choi nhan phim mui ten trai hoac phai thi nhan vat se di chuyen theo huong tuong ung. Neu nguoi choi nhan phim mui ten len thi nhan vat se nhay move(self, data_time) Di chuyen nhan vat theo huong va toc do da duoc xac dinh platform_move(self, data_time) Di chuyen nhan vat theo platform (moving objects) update(self, data_time) Cap nhat trang thai cua nhan vat update_timer(self) Cap nhat tat ca cac timer trong nhan vat ----- Methods inherited from pygame.sprite.Sprite: </pre>	
<pre> class MovingSprite(AnimatedSprite) MovingSprite(frames, groups, start_pos, end_pos, move_dir, speed, flip=False) Khoi tao lop MovingSprite ke thua tu lop AnimatedSprite Lop nay tao ra cac doi tuong do hoa co kha nang di chuyen trong tro choi Method resolution order: MovingSprite AnimatedSprite Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, frames, groups, start_pos, end_pos, move_dir, speed, flip=False) Khoi tao AnimatedSprite voi vi tri, cac frame hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh va toc do animation check_border(self) Kiem tra xem moving object da den vi tri cuoi cung chua, neu da den thi doi huong di chuyen update(self, data_time) Cap nhat vi tri cua moving object ----- Methods inherited from AnimatedSprite: animate(self, data_time) Xu ly animation cho sprite ----- Methods inherited from pygame.sprite.Sprite: </pre>	<pre> class Item(AnimatedSprite) Item(item_type, pos, frames, groups, data) Khoi tao lop Item ke thua tu lop AnimatedSprite Lop nay tao ra cac doi tuong do hoa la Item trong tro choi Method resolution order: Item AnimatedSprite Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, item_type, pos, frames, groups, data) Khoi tao AnimatedSprite voi vi tri, cac frame hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh va toc do animation activate(self) Kich hoat item khi nguoi choi va cham voi item voi item_type la silver: tang 5 coins, diamond: tang 50 coins, skull: tang 1 mang ----- Methods inherited from AnimatedSprite: animate(self, data_time) Xu ly animation cho sprite update(self, data_time) Cap nhat animation cua sprite dam bao sprite chuyen dong lien tuc ----- Methods inherited from pygame.sprite.Sprite: </pre>

<pre> class ParticleEffectSprite(AnimatedSprite) ParticleEffectSprite(pos, frames, groups) Khoi tao lop ParticleEffectSprite ke thua tu lop AnimatedSprite Lop nay tao ra cac doi tuong do hoa la ParticleEffect trong tro choi lam nhieu vu tao hieu ung cho cac vat the trong tro choi khi va cham (cac items khi nguoi choi nhut, cac vat the khi bi ban trung) Method resolution order: ParticleEffectSprite AnimatedSprite Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, frames, groups) Khoi tao AnimatedSprite voi vi tri, cac frame hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh va toc do animation animate(self, data_time) Tao animation cho ParticleEffect ----- Methods inherited from AnimatedSprite: update(self, data_time) Cap nhat animation cua sprite dam bao sprite chuyen dong lien tục ----- Methods inherited from pygame.sprite.Sprite </pre>	<pre> class Cloud(Sprite) Cloud(pos, surf, groups, z=1) Khoi tao lop Cloud ke thua tu lop Sprite Lop nay tao ra cac doi tuong do hoa la Cloud trong tro choi Method resolution order: Cloud Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, surf, groups, z=1) Khoi tao sprite voi vi tri, hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh update(self, data_time) Cap nhat vi tri cua cloud theo thoi gian ----- Methods inherited from pygame.sprite.Sprite </pre>
<pre> class Timer(builtins.object) Timer(duration, func=None, repeat=False) class Timer dung de tao cac thoi gian cho cac su kien trong game Methods defined here: __init__(self, duration, func=None, repeat=False) Initialize self. See help(type(self)) for accurate signature. activate(self) kich hoat timer de bat dau dem thoi gian deactivate(self) ngungkiem tra thoi gian update(self) cap nhat thoi gian class CollisionSprite(pygame.sprite.Group) Khoi tao lop CollisionSprite ke thua tu lop Group cua pygame.sprite de quan ly cac sprite va cham trong tro choi Method resolution order: CollisionSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Initialize self. See help(type(self)) for accurate signature. ----- Methods inherited from pygame.sprite.AbstractGroup </pre>	<pre> class AllSprite(pygame.sprite.Group) AllSprite(width, height, clouds, horizon_line, bg_tile=None, top_limit=0) Khoi tao lop AllSprite ke thua tu lop Group cua pygame.sprite de quan ly tat ca cac sprite trong tro choi Method resolution order: AllSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self, width, height, clouds, horizon_line, bg_tile=None, top_limit=0) Initialize self. See help(type(self)) for accurate signature. camera_constraint(self) Xu ly vi tri cua camera de khong cho camera di chuyen ra khoi map create_cloud(self) Tao dam may nho draw(self, target_pos, data_time) Hien thi cac sprite trong tro choi draw_large_cloud(self, data_time) Ve dam may lon draw_sky(self) Ve bau troi ----- Methods inherited from pygame.sprite.AbstractGroup </pre>

<pre> class ToothSprite(pygame.sprite.Group) Khoi tao lop ToothSprite ke thua tu lop Sprite cua pygame.sprite de tao sprite cho doi tuong Tooth Method resolution order: ToothSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Khoi tao sprite cho doi tuong Tooth ----- Methods inherited from pygame.sprite.AbstractGroup class PearlSprite(pygame.sprite.Group) Khoi tao lop PearlSprite ke thua tu lop Group cua pygame.sprite de quan ly Pearl trong tro choi Method resolution order: PearlSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Initialize self. See help(type(self)) for accurate signature. ----- Methods inherited from pygame.sprite.AbstractGroup </pre>	<pre> class DamageSprite(pygame.sprite.Group) Khoi tao lop DamageSprite ke thua tu lop Group cua pygame.sprite de quan ly cac sprite co the gay sat thuong trong tro choi Method resolution order: DamageSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Initialize self. See help(type(self)) for accurate signature. ----- Methods inherited from pygame.sprite.AbstractGroup class ItemSprite(pygame.sprite.Group) Khoi tao lop ItemSprite ke thua tu lop Group cua pygame.sprite de quan ly cac sprite Item trong tro choi Method resolution order: ItemSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Initialize self. See help(type(self)) for accurate signature. ----- Methods inherited from pygame.sprite.AbstractGroup </pre>
<pre> class Data(builtins.object) Data(ui) Khoi tao class Data de luu tru thong tin ve so mang va so tien cua nguoi choi Trong do coins va health deu duoc dong private va duoc truy cap thong qua property de kiem soat gia tri cua chung Methods defined here: __init__(self, ui) Initialize self. See help(type(self)) for accurate signature. get_coins(self) phuong thuc getter de tra ve gia tri cua coins. khi goi Data.coins se tra ve gia tri cua coins get_health(self) phuong thuc getter de tra ve gia tri cua health. khi goi Data.health se tra ve gia tri cua health set_coins(self, value) phuong thuc setter de thiet lap gia tri cho coins. khi goi Data.coins = value se thiet lap gia tri cho coins set_health(self, value) phuong thuc setter de thiet lap gia tri cho health. khi goi Data.health = value se thiet lap gia tri cho health ----- </pre>	<pre> class UI(builtins.object) UI(font, frames) Methods defined here: __init__(self, font, frames) Khoi tao class UI de hien thi thong tin so mang va so tien cua nguoi choi create_health(self, amount) Tao so mang cho nguoi choi display_text(self) Hien thi so tien cua nguoi choi reset_coins(self) Dat lai so tien cua nguoi choi ve 0 show_coins(self, amount) Khoi tao so tien cua nguoi choi update(self, data_time) Cap nhat thong tin so mang va so tien cua nguoi choi ----- </pre>

<pre> class Heart(sprites.AnimatedSprite) Heart(pos, frames, groups) class Heart ke thua tu lop AnimatedSprite de tao ra cac doi tuong do hoa la Heart trong tro choi Method resolution order: Heart sprites.AnimatedSprite sprites.Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, frames, groups) Khoi tao AnimatedSprite voi vi tri, cac frame hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh va toc do animation animate(self, data_time) tao animation cho Heart update(self, data_time) cap nhat Heart theo thoi gian ----- </pre>	<pre> class CollisionSprite(pygame.sprite.Group) Khoi tao lop CollisionSprite ke thua tu lop Group cua pygame.sprite de quan ly cac sprite va cham trong tro choi Method resolution order: CollisionSprite pygame.sprite.Group pygame.sprite.AbstractGroup builtins.object Methods defined here: __init__(self) Initialize self. See help(type(self)) for accurate signature. ----- Methods inherited from pygame.sprite.AbstractGroup </pre>
<pre> class Tooth(pygame.sprite.Sprite) Tooth(pos, frames, groups, collision_sprites) Khoi tao lop Tooth ke thua tu lop Sprite cua pygame Lop nay tao ra cac doi tuong do hoa la Tooth gom cac thiet lap cho Tooth nhu hinh anh, vi tri, toc do, huong di chuyen, va cac dac tinh cua Tooth Method resolution order: Tooth pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, frames, groups, collision_sprites) Initialize self. See help(type(self)) for accurate signature. die(self, tooth_die_frames) cai dat hinh anh cho Tooth khi chet flicker(self) tao hieu ung nhap nhay cho Tooth khi bi nhan vat tan cong, trong thoi gian nay tooth se khong the bi tan cong reverse(self) cai dat huong di chuyen cua Tooth neu cham vao bien update(self, data_time) cap nhat Tooth theo thoi gian ----- Methods inherited from pygame.sprite.Sprite </pre>	<pre> class Pearl(pygame.sprite.Sprite) Pearl(pos, groups, surf, direction, speed) Khoi tao lop Pearl ke thua tu lop Sprite cua pygame Lop nay tao ra cac doi tuong do hoa la Pearl gom cac thiet lap cho Pearl nhu hinh anh, vi tri, toc do, huong di chuyen, va cac dac tinh cua Pearl Method resolution order: Pearl pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, groups, surf, direction, speed) Initialize self. See help(type(self)) for accurate signature. reverse(self) Kiem tra xem Pearl co dang di chuyen khong va dao huong di chuyen cua Pearl update(self, dt) Cap nhat Pearl theo thoi gian ----- Methods inherited from pygame.sprite.Sprite </pre>

class Shell(pygame.sprite.Sprite) Shell(pos, frames, groups, reverse, player, create_pearl) Khoi tao lop Shell ke thua tu lop Sprite cua pygame Lop nay tao ra cac doi tuong do hoa la Shell gom cac thiet lap cho Shell nhu hinh anh, vi tri, toc do, huong di chuyen, va cac dac tinh cua Shell Method resolution order: Shell pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, frames, groups, reverse, player, create_pearl) Initialize self. See help(type(self)) for accurate signature. reverse(self) Dao huong cua Shell state_management(self) Xu li trang thai cua Shell update(self, data_time) Cap nhat shell theo thoi gian ----- Methods inherited from pygame.sprite.Sprite	class AnimatedSprite(Sprite) AnimatedSprite(pos, frames, groups, z=2, animation_speed=6) Khoi tao lop AnimatedSprite ke thua tu lop Sprite Lop nay tao ra cac doi tuong do hoa co hieu ung chuyen dong (Animation) trong tro choi Method resolution order: AnimatedSprite Sprite pygame.sprite.Sprite builtins.object Methods defined here: __init__(self, pos, frames, groups, z=2, animation_speed=6) Khoi tao AnimatedSprite voi vi tri, cac frame hinh anh, cac nhom thuoc tinh cua sprite va thu tu duoc ve len man hinh va toc do animation animate(self, data_time) Xu ly animation cho sprite update(self, data_time) Cap nhat animation cua sprite dam bao sprite chuyen dong lien tuc ----- Methods inherited from pygame.sprite.Sprite
---	--