***pygame**

pygame.init()
pygame.display.set_mode(size)
pygame.display.set_caption(title)
pygame.time.Clock()
pygame.image.load(filename)
pygame.mixer.Sound(filename)
pygame.font.Font(filename, size)
pygame.event.get()
pygame.event.type
pygame.event.get_ticks()
pygame.event.custom_type()
pygame.time.set_timer(eventid, milliseconds)
pygame.sprite.Group()
pygame.sprite.spritecollide(sprite, group, dokill, collided=None)
pygame.Surface(size, flags=0, depth=0, masks=None)
pygame.draw.rect(surface, color, rect, width=0)
pygame.display.update()
pygame.mouse.get_pos()
pygame.mouse.get_pressed()
pygame.key.get_pressed()
pygame.key.get_focused()
pygame.time.get_ticks()
pygame.transform.rotozoom(surface, angle, scale)
pygame.transform.flip(surface, xbool, ybool)
pygame.mask.from_surface(surface)
pygame.Vector2()
pygame.sprite.Sprite.__init__(self, groups)
pygame.quit()

***Pytmx.util.load_pygame**

Load map from Tiled to game

***Os**

os.path : import join for working with file path
os : import walk.The walk function from the os module generates recursive file listings from a root directory

***random**

import choice for working with random elements

***math**

atan2 : caculate the angle of vector

degrees : conver radian to degrees

Aggregation:
Dependency: Use
Inheritance: Extends

Game

+display_surface: pygame.Surface
+clock: pygame.time.Clock
+running: bool
+menu_options: List[str]
+menu: Menu
+ui_ux: UI
+all_sprites: AllSprites
+collision_sprites: pygame.sprite.Group
+bullet_sprites: pygame.sprite.Group
+enemy_sprites: pygame.sprite.Group
+can_shoot: bool
+shoot_time: int
+gun_cooldown: int
+enemy_event: int
+spawn_positions: List[Tuple[int, int]]
+max_enemies: int
+score: int
+state: str
+guide_text: List[str]
+player_skills: PlayerSkills
+player: Player
+gun: Weapon
+bullet_surf: pygame.Surface
+enemy_frames: Dict[str, List[pygame.Surfac

+load_images() -> None
+update_skills() -> None
+input() -> None
+gun_timer() -> None
+setup() -> None
+bullet_slash_collision() -> None
+player_collision() -> None
+reset_game() -> None
+run() -> None
+spawn_enemy() -> None

UI

+display_surface: pygame.Surface
+font: pygame.font.Font
+shoot_sound: pygame.mixer.Sound
+impact_sound: pygame.mixer.Sound
+music: pygame.mixer.Sound
+game_over_sound: pygame.mixer.Sound
+slash_sound: pygame.mixer.Sound
+health_heal_sound: pygame.mixer.Sound
+fireball_use_skill_sound: pygame.mixer.Sound
+slash_use_skill_sound: pygame.mixer.Sound
+draw_sword_sound: pygame.mixer.Sound
+fire_use_skill_sound: pygame.mixer.Sound
+blood_sound: pygame.mixer.Sound

+load_sounds() -> None
+draw_health_bar(health: int) -> None
+draw_score(score: int) -> None
+draw_game_over_screen(score: int) -> None
+draw_guide(guide_text: List[str]) -> None
+draw_skills(skill_images: List[pygame.Surface],
skill_positions: List[Tuple[int, int]],
skill_cooldowns: List[int],
skill_last_used: List[int],
gun_skill2_last_used_time: int,
sword_skill2_last_used_time: int,
skill_cooldown_alpha: int,
skill2_active: bool,
player_direction: str,
all_sprites: pygame.sprite.Group,
bullet_sprites: pygame.sprite.Group) -> None

Menu

+options: List[str]
+selected_option: Optional[int]
+background: pygame.Surface
+button_image: pygame.Surface
+button_rects: List[pygame.Rect]

+draw(surface: pygame.Surface) -> None
+handle_input(event: pygame.event.Event) -> Optional[int]

***pygame.sprite.Sprite**

Use

Bullet

+image: pygame.Surface
+rect: pygame.Rect
+spawn_time: int
+lifetime: int
+direction: pygame.Vector2
+speed: int

+update(dt: float) -> None

Sprite

+image: pygame.Surface
+rect: pygame.Rect
+ground: bool

None

***pygame.sprite.Group**

Extends

Allsprites

+display_surface: pygame.Surface
+offset: pygame.Vector2

+draw(target_pos: Tuple[int, int]) ->

Extends

Enemy

+player: Player
+frames: List[pygame.Surface]
+frame_index: int
+image: pygame.Surface
+animation_speed: int
+rect: pygame.Rect
+hitbox_rect: pygame.Rect
+collision_sprites: pygame.sprite.Group
+direction: pygame.Vector2
+speed: int
+death_time: int
+death_duration: int
+hit: bool
+hit_duration: int
+hit_time: int
+invulnerable: boo

+animate(dt: float) -> None
+move(dt: float) -> None
+collision(direction: str) -> None
+destroy() -> None
+death_timer() -> None
+update(dt: float) -> None
+take_damage() -> None

Extends

PlayerSkills

+game: Game
+skill_images: List[pygame.Surface]
+skill_positions: List[Tuple[int, int]]
+skill_cooldowns: List[int]
+skill_last_used: List[int]
+gun_skill2_last_used_time: int
+sword_skill2_last_used_time: int
+skill_cooldown_alpha: int
+skill2_active: bool
+skill2_cooldown_duration: int
+slash_cooldown: int
+last_slash_time: int
+skill2_cooldown_time: int
+last_skill2_shot_time: int
+count: int
+mouse_click_cooldown: int
+last_mouse_click_time: int

+handle_skill_use(key: int) -> None
+update_skills() -> None
+perform_slash() -> None
+use_skill(skill_index: int) -> None
+handle_input() -> None
+shoot_with_default_gun() -> None
+shoot_with_skill2() -> None
+reset_skills() -> None

Use

Weapon

+player: Player
+distance: int
+player_direction: pygame.Vector2
+gun_surf: pygame.Surface
+sword_surf: pygame.Surface
+image: pygame.Surface
+rect: pygame.Rect
+weapon_type: str

+get_direction() -> None
+switch_weapon() -> None
+rotate_weapon() -> None
+update(_) -> None

Slash

+slash_surf: pygame.Surface
+image: pygame.Surface
+rect: pygame.Rect
+direction: pygame.Vector2
+speed: int
+spawn_time: int
+lifetime: int

+update(dt: float) -> None

CollisionSprite

+image: pygame.Surface
+rect: pygame.Rect

None

Player

+frames: Dict[str, List[pygame.Surface]]
+state: str
+frame_index: int
+image: pygame.Surface
+rect: pygame.Rect
+hitbox_rect: pygame.Rect
+direction: pygame.Vector2
+speed: int
+collision_sprites: pygame.sprite.Group
+health: int
+invulnerable: bool
+invulnerable_duration: int
+last_hit_time: int
+blink_duration: int
+last_blink_time: int
+visible: bool
+weapon: str

+decrease_health(amount: int) -> None
+load_images() -> None
+input() -> None
+move(dt: float) -> None
+collision(direction: str) -> None
+animate(dt: float) -> None
+update(dt: float) -> None