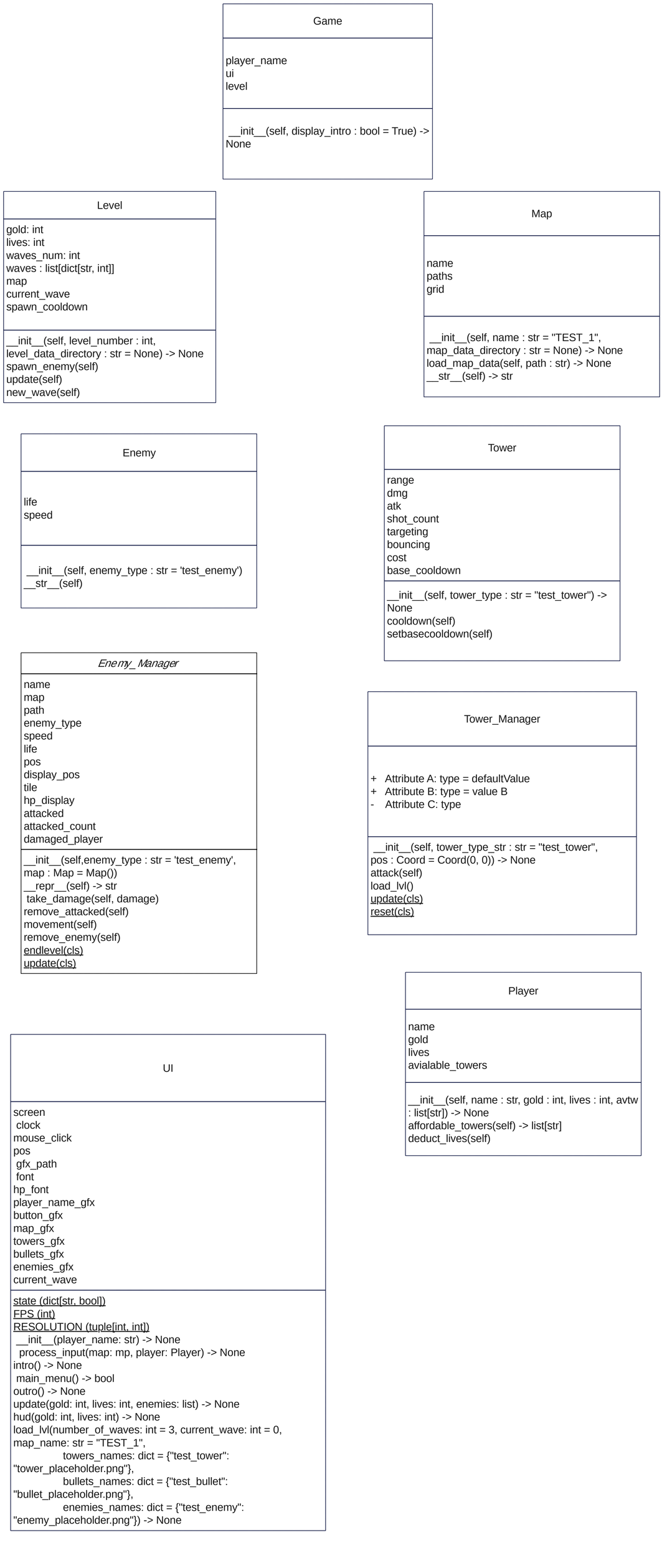
Game
player_name
ui
level
__init__(self, display_intro : bool = True) -> None
Level
gold: int
lives: int
waves_num: int
waves : list[dict[str, int]]
map
current_wave
spawn_cooldown
__init__(self, level_number : int, level_data_directory : str = None) -> None
spawn_enemy(self)
update(self)
new_wave(self)
Map
name
paths
grid
__init__(self, name : str = "TEST_1", map_data_directory : str = None) -> None
load_map_data(self, path : str) -> None
__str__(self) -> str
Enemy
life
speed
__init__(self, enemy_type : str = 'test_enemy')
__str__(self)
Tower
range
dmg
atk
shot_count
targeting
bouncing
cost
base_cooldown
__init__(self, tower_type : str = "test_tower") -> None
cooldown(self)
setbasecooldown(self)
Enemy_Manager
name
map
path
enemy_type
speed
life
pos
display_pos
tile
hp_display
attacked
attacked_count
damaged_player
__init__(self,enemy_type : str = 'test_enemy', map : Map = Map())
__repr__(self) -> str
take_damage(self, damage)
remove_attacked(self)
movement(self)
remove_enemy(self)
endlevel(cls)
update(cls)
Tower_Manager
+ Attribute A: type = defaultValue
+ Attribute B: type = value B
- Attribute C: type
__init__(self, tower_type_str : str = "test_tower", pos : Coord = Coord(0, 0)) -> None
attack(self)
load_lvl()
update(cls)
reset(cls)
Player
name
gold
lives
avialable_towers
__init__(self, name : str, gold : int, lives : int, avtw : list[str]) -> None
affordable_towers(self) -> list[str]
deduct_lives(self)
UI
screen
clock
mouse_click
pos
gfx_path
font
hp_font
player_name_gfx
button_gfx
map_gfx
towers_gfx
bullets_gfx
enemies_gfx
current_wave
state (dict[str, bool])
FPS (int)
RESOLUTION (tuple[int, int])
__init__(player_name: str) -> None
process_input(map: mp, player: Player) -> None
intro() -> None
main_menu() -> bool
outro() -> None
update(gold: int, lives: int, enemies: list) -> None
hud(gold: int, lives: int) -> None
load_lvl(number_of_waves: int = 3, current_wave: int = 0,
map_name: str = "TEST_1",
towers_names: dict = {"test_tower": "tower_placeholder.png"},
bullets_names: dict = {"test_bullet": "bullet_placeholder.png"},
enemies_names: dict = {"test_enemy": "enemy_placeholder.png"}) -> None