# Heist Simulation (Task 19)- README

## Overview

A 2D AI simulation where a Thief agent attempts to steal a gem while avoiding a patrolling and reactive Guard. The Guard uses a Finite State Machine (FSM) to switch between patrol and chase behaviors, while the Thief supports both manual control and automatic navigation using pathfinding and basic steering behaviors.

## AI Techniques Used

| Agent | Behavior Type | Techniques |
|---|---|---|
| **Guard** | FSM (Finite State Machine) | Patrol, Chase, Shoot |
| **Thief** | Behavior-Based (Manual & Auto) | Flee, Seek, Pathfinding |
| **Thief** | Planned (Future Work) | Reinforcement Learning |

## Features

- Grid-based map loaded from map.txt
- A* pathfinding algorithm for auto-navigation
- FSM logic for Guard (patrol and chase states)
- Bullet shooting and health system
- UI labels for health, FSM state, win/lose messages
- Toggle between manual and auto Thief control

## Controls

Arrow Keys: Move Thief (manual mode)

A: Toggle Auto Mode (pathfinding)

## Setup & Running the Game

Requirements:

- Python 3.9+
- Pyglet

Install via pip:

```
pip install pyglet
```

Run the Game:

```
python main.py
```

## Project Structure

- agent.py - Thief & Guard logic with steering and FSM support
- bullet.py - Bullet behavior and collision handling
- fsm.py - FSM logic for Guard (Patrol & Chase states)
- graphics.py - Color definitions
- main.py - Game loop and key handling
- map.txt - Maze layout for world generation
- pathfinding.py - A* search algorithm
- ui.py - Health bar and status labels
- world.py - Environment setup, update, rendering logic

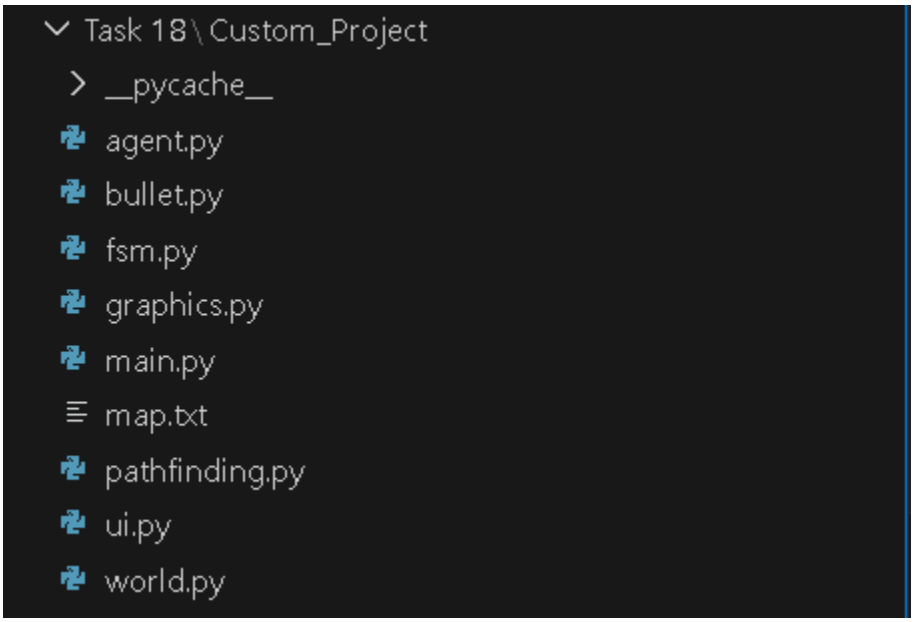


*Figure: Folder Structure*

## For Further Development of this Project

- Thief agent can learn via Reinforcement Learning (RL) to identify patterns of the guard agent and it to be act accordingly
- Adding more steering Logic to the Thief and Guard agents for better interaction
- Using a 2D image to act as the background instead of a pre coded map
- We can add more Map, with several other terrains to test out how the guard and thief agent act when given unknown terrain details to find bugs in the program

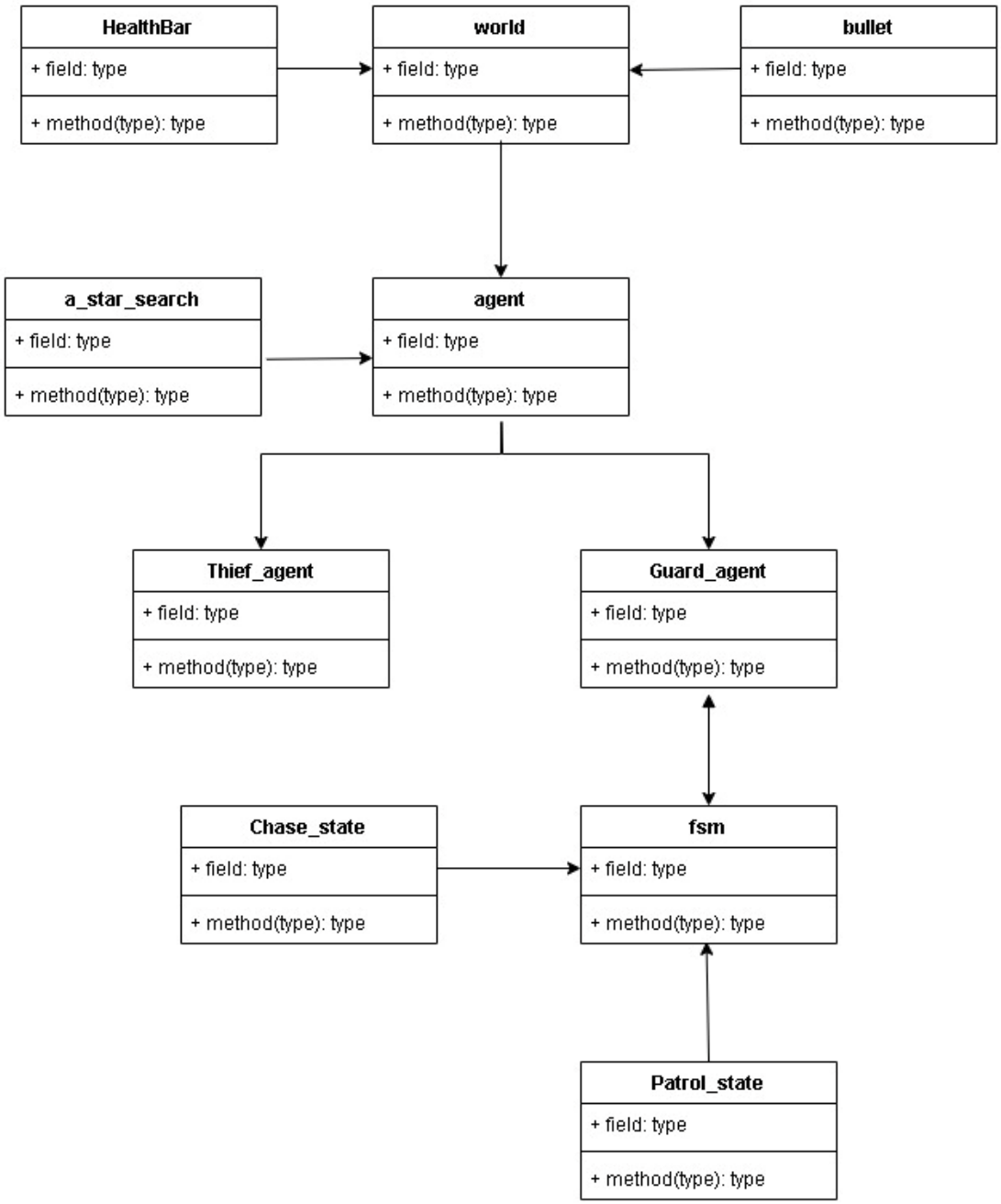


*Figure: Architecture of the System*