Group 11 Final Project

Car Racing

B123040044 侯廷翰
B123040049 劉育希
B123245016 柯伯諺

Part 2

Use **Optuna**
To search for the best
hyperparameters for training

{epsilon_decay_rate, min_epsilon, learning_rate}

⬇

Train **(15000 episodes)**

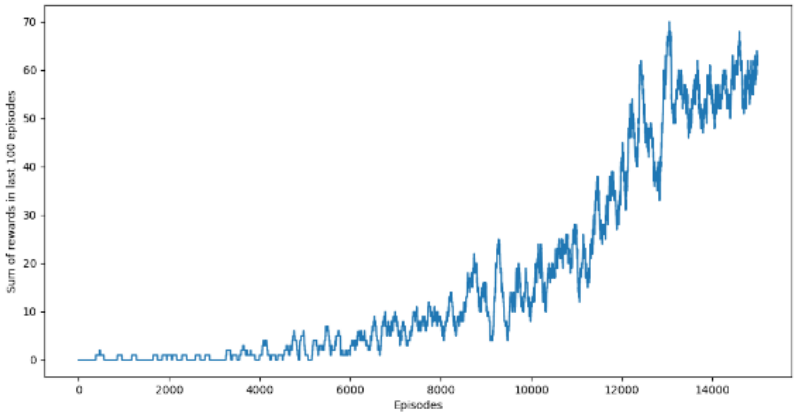
⬇

Run

Success Rate:
Around 60%

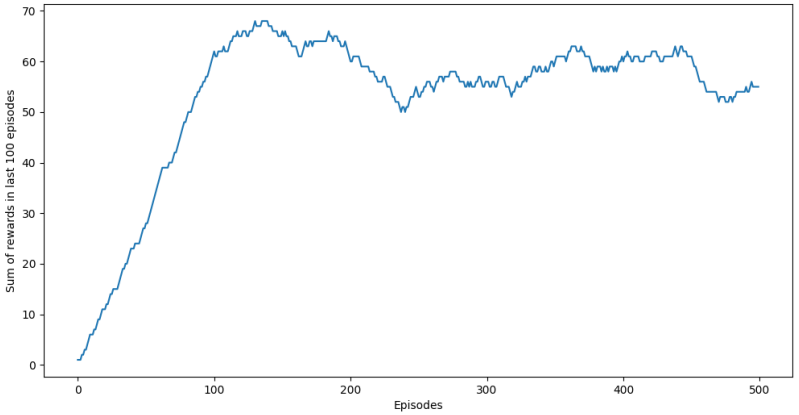
⬆

OPTUNA



◆ Training Process

(15,000 episodes)



◆ Run\Test Process

(500 episodes)

Part 3

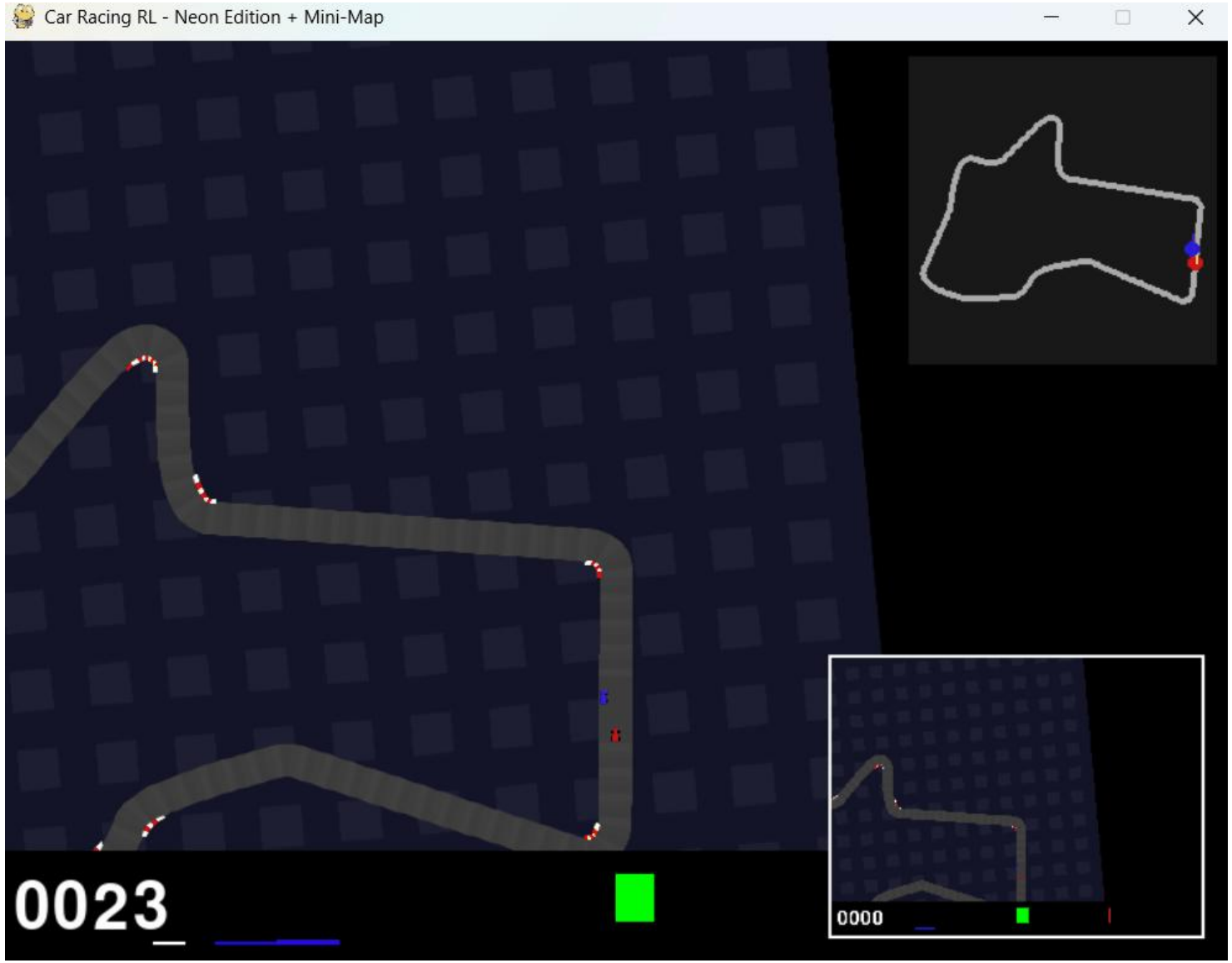
Use **Car Racing** env.

New Feature:

- Auto Smart Agent
- New Opponent
- minimap

Smart Agent:

- With **96x96** observation space
- Use **CNN** Network

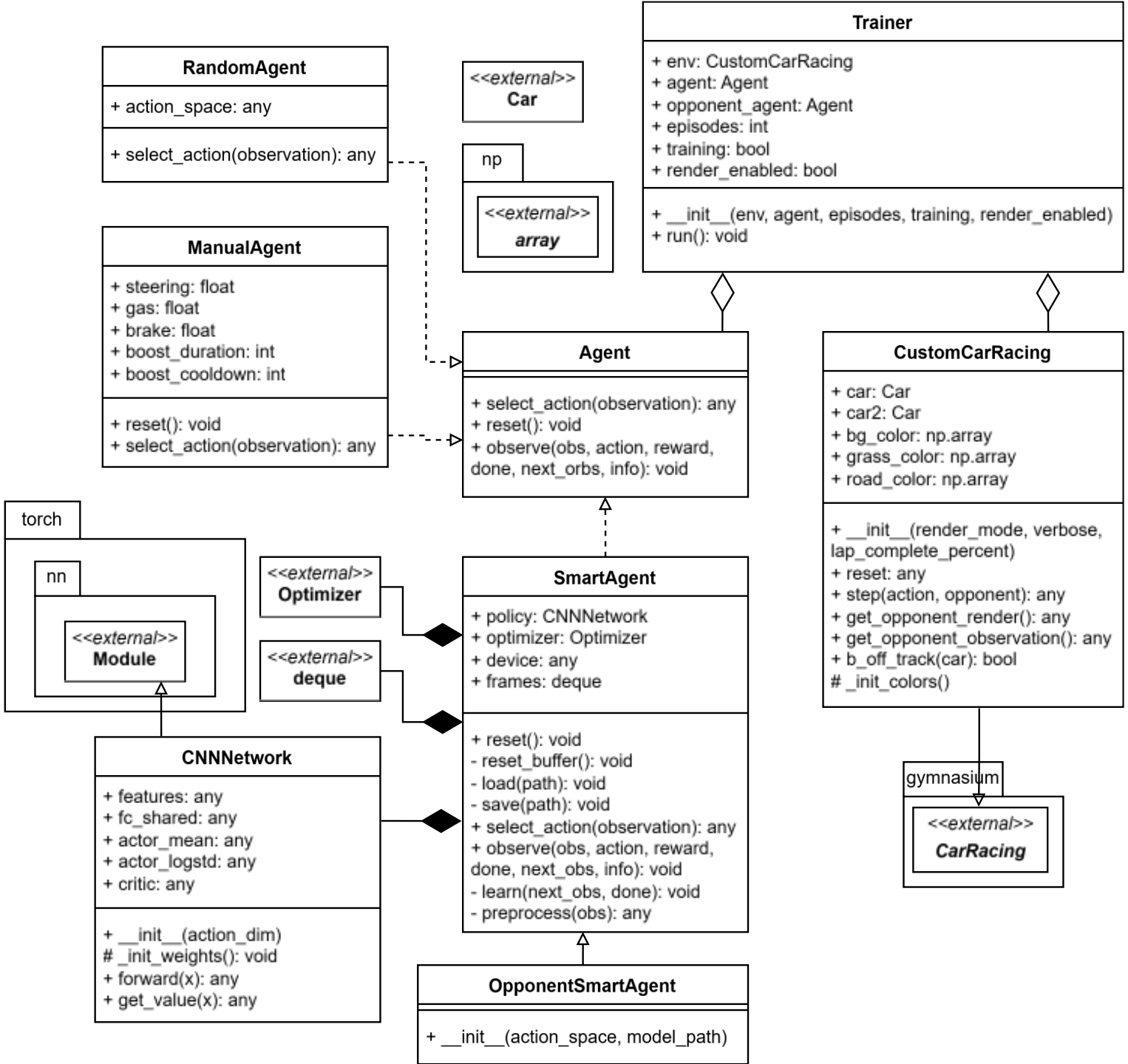


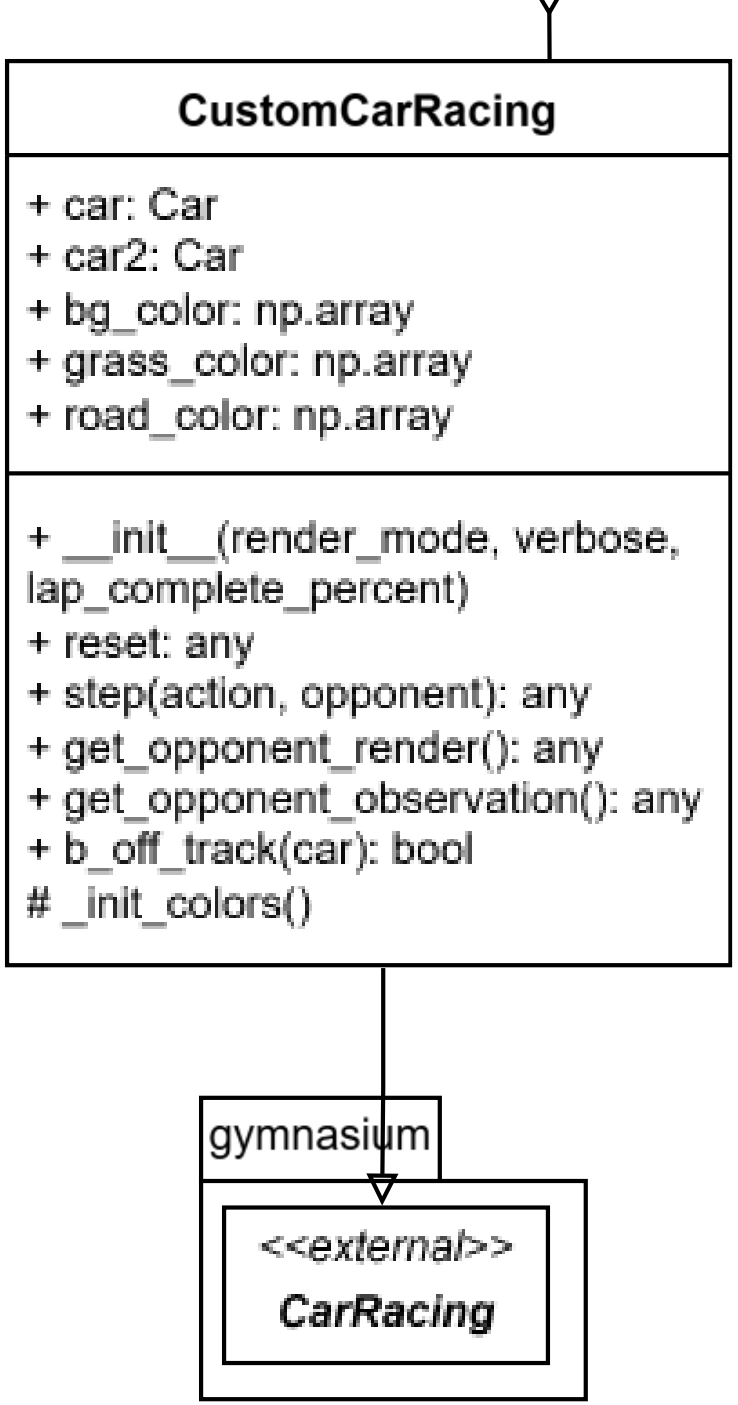
Lando! Is this the world champion hotline ?

Part 3 UML

OOP Concepts

- Inheritance
- Specialization
- Encapsulation
- Abstract
- Polymorphism
- Composition
- Aggregation





Environment

Inheritance & Specialization

CustomCarRacing ⟶ CarRacing

Function Overload

Override colors()

Agents

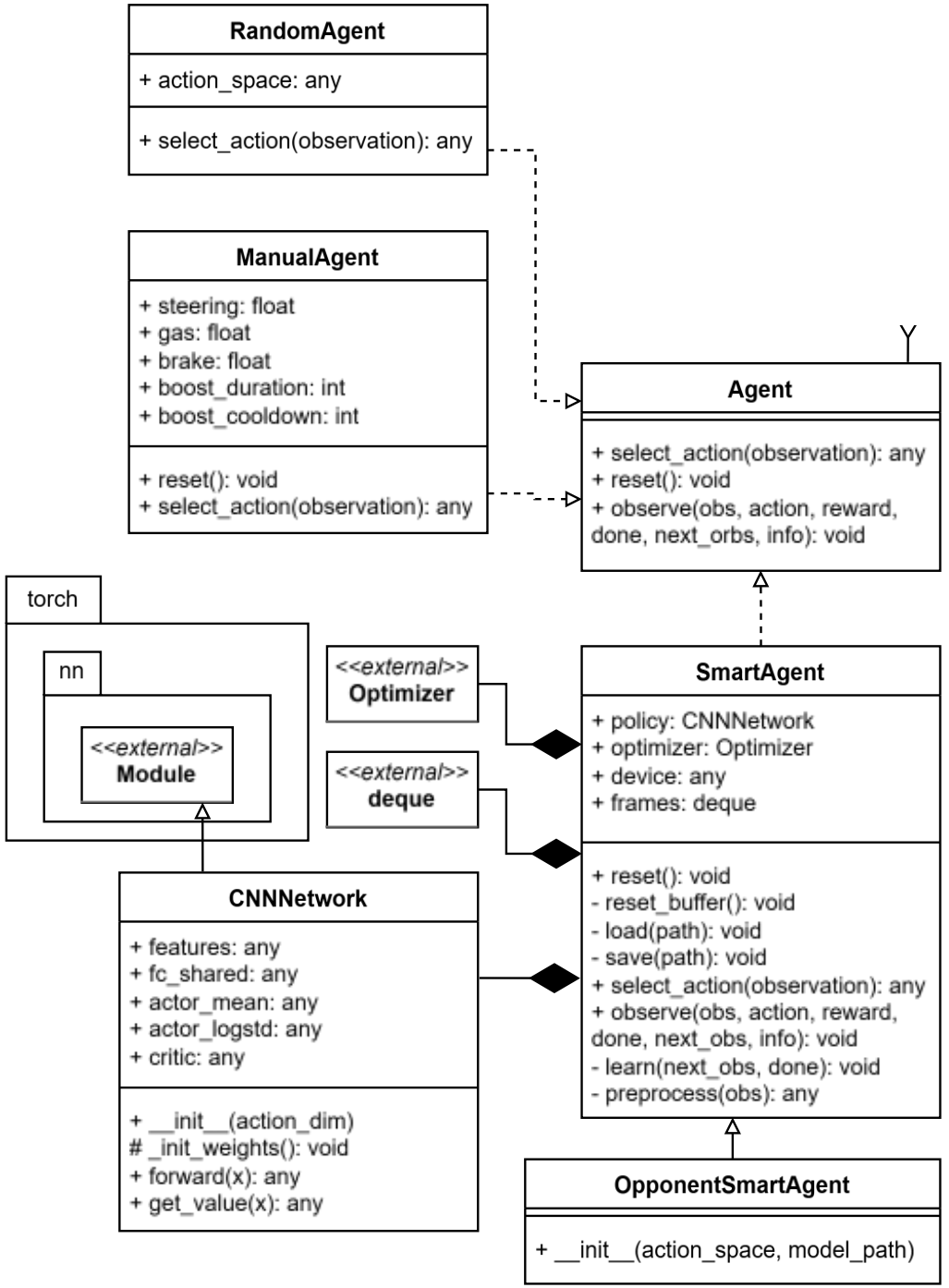
Abstract

Agent

+ select_action(observation): any
+ reset(): void
+ observe(obs, action, reward, done, next_orbs, info): void

Inheritance & Polymorphism

RandomAgent
ManualAgent
SmartAgent
→ Agent



Agents

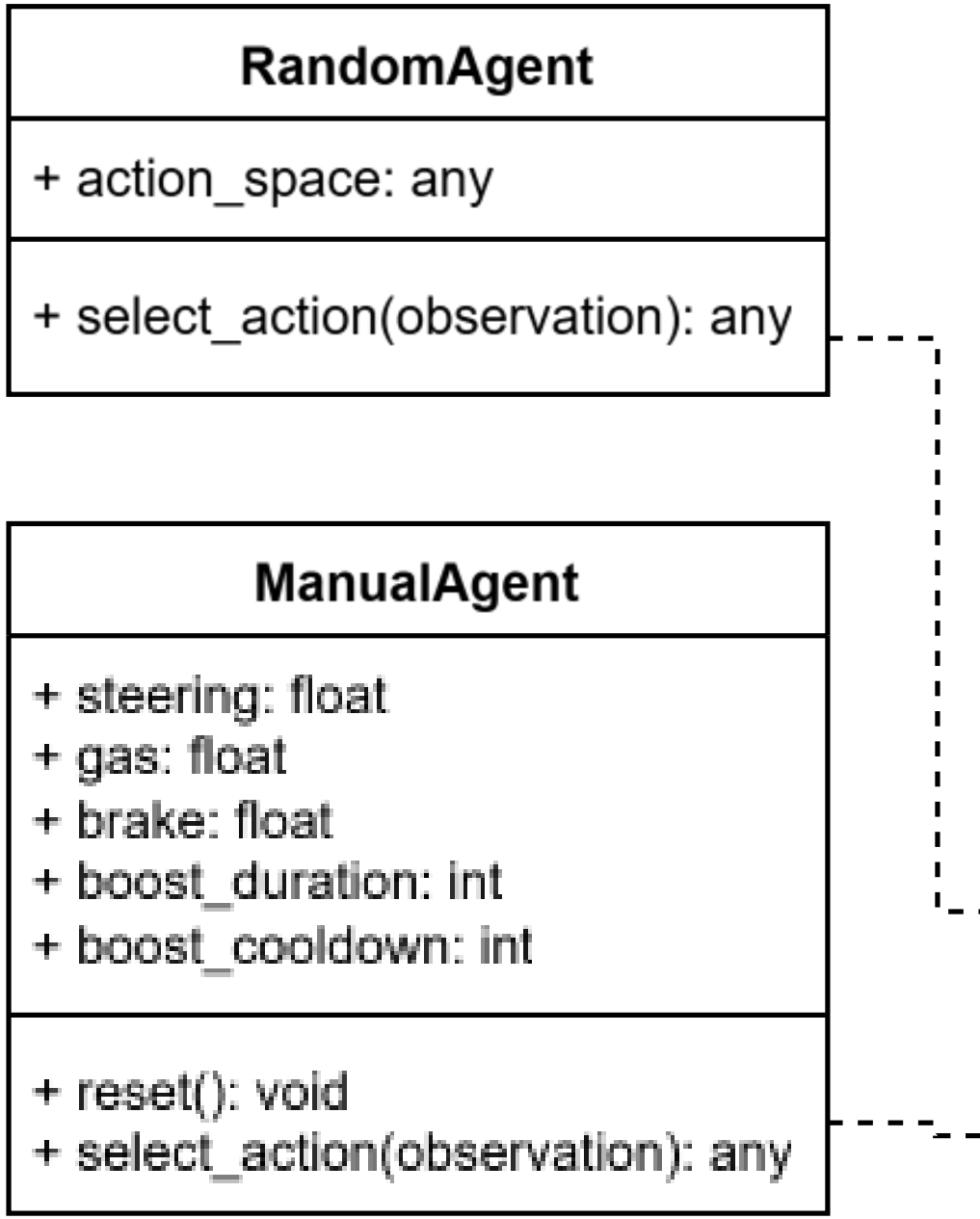
Polymorphism

Random Agent

- **select_action:**
  base on randomness
- None

Manual Agent

- **select_action:**
  base on user input
- steering
- gas
- brake
- boost_duration
- boost_cooldown



Agents

Inheritance

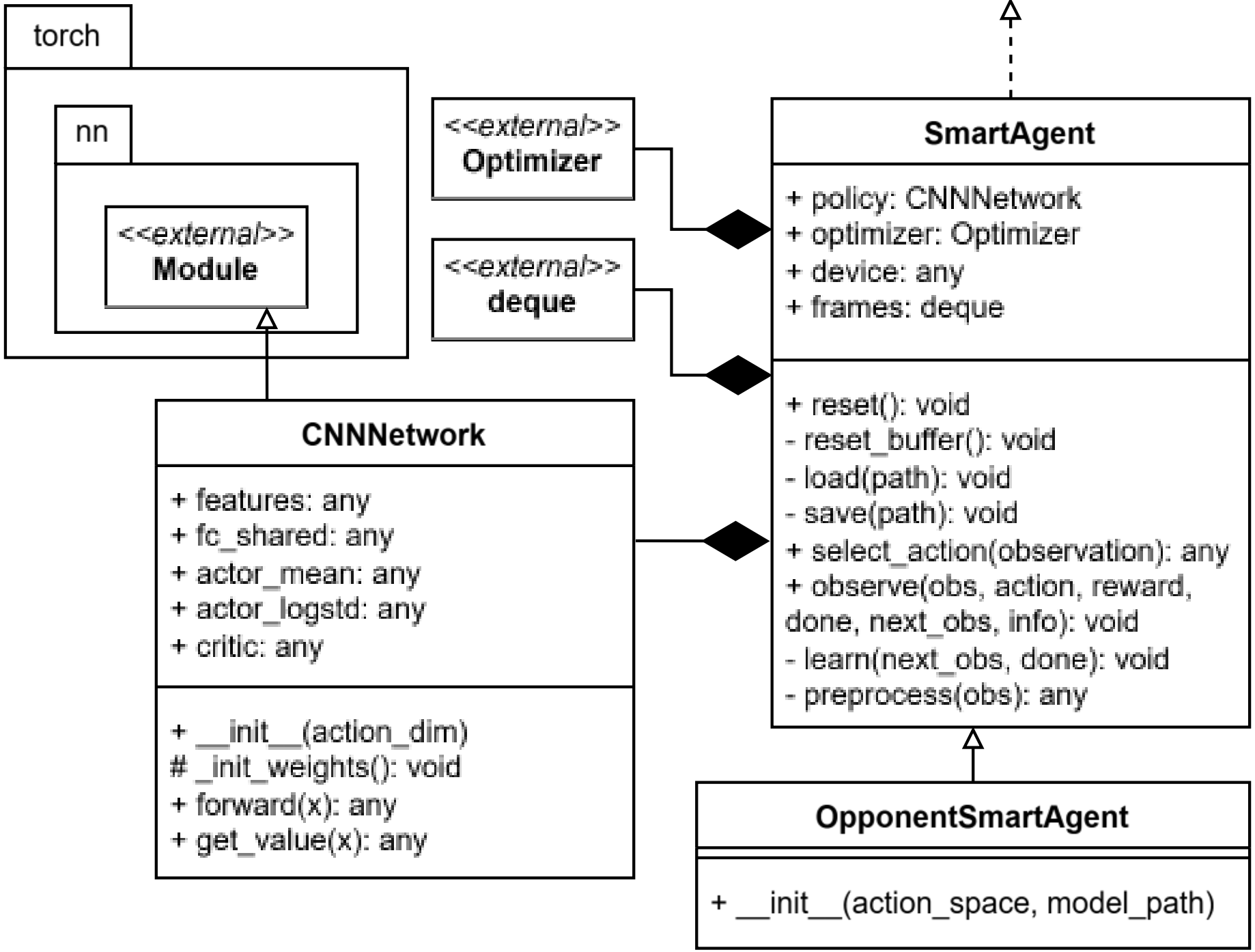
CNNNetwork
↳ nn.Module

Composition

- CNNNetwork
- Deque
- Optimizer

Encapsulation

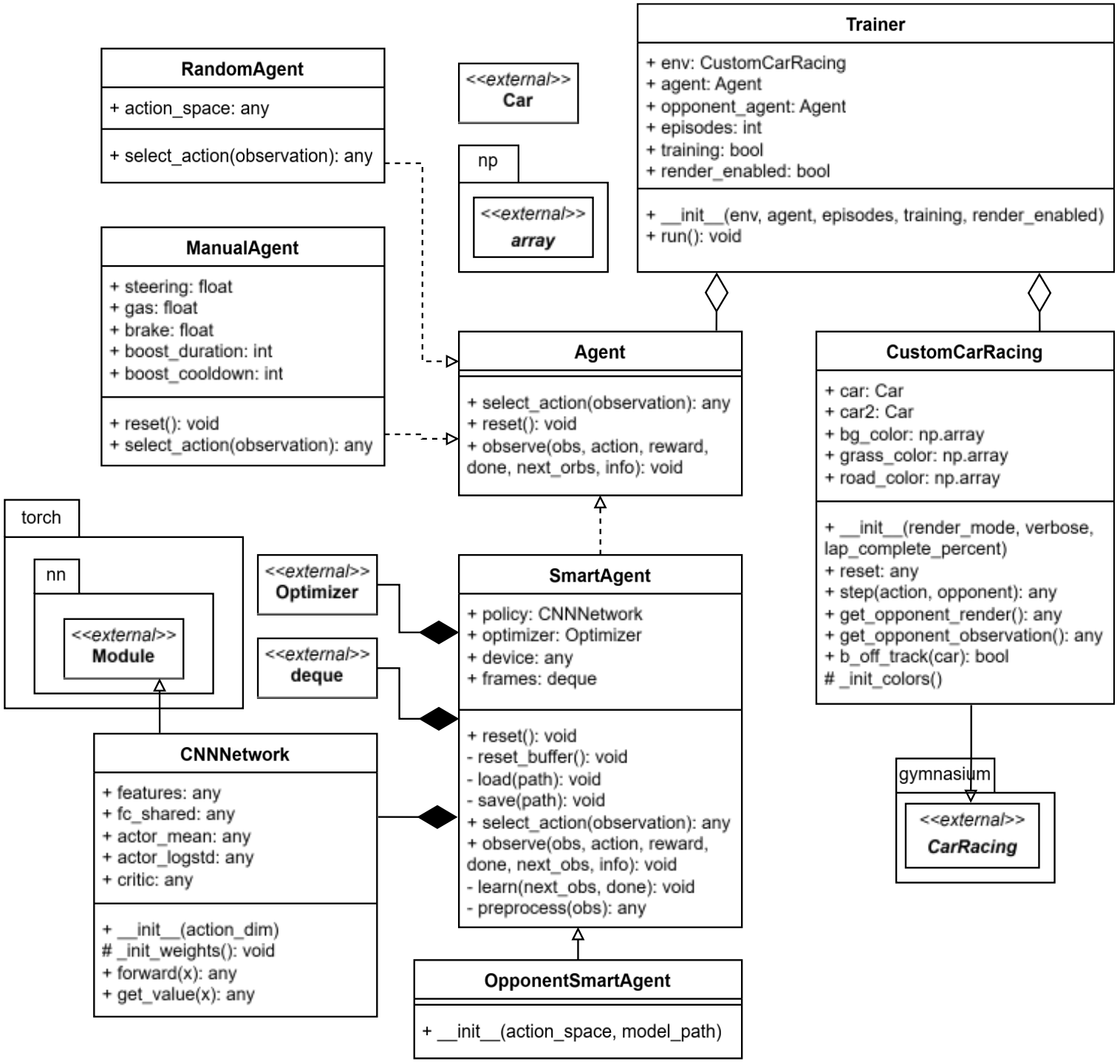
load
Save
reset_buffer
learn
preprocess(obs)



Trainer

Aggregation

- Agent
- CustomCarRacing



DEMO