



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
CARRERA DE COMPUTACIÓN

INNOVACIÓN Y EMPRENDIMIENTO

Grupo: 4

Integrantes:

Juan Escobar

Andy Suquilandi

John Laverde

Marco Andrade

Esteban Chachalo

Paralelo: C6-001

Docente: Felipe Borja

Fecha: 03/02/2025

Quito – Ecuador

2025-2026

Manual Técnico de Operación: Sistema de Aprendizaje por Refuerzo (TrustMeBro-AI)

1. Resumen Ejecutivo y Alcance

Este proyecto constituye una implementación de Inteligencia Artificial basada en Aprendizaje por Refuerzo (específicamente algoritmos Q-Learning). El sistema despliega un agente autónomo diseñado para aprender a navegar e interactuar dentro de un entorno simulado, proporcionando visualización en tiempo real del proceso de aprendizaje mediante un motor gráfico propio basado en Pygame. El desarrollo se enmarca dentro de un proyecto de investigación en Inteligencia Artificial.

2. Arquitectura del Sistema

El software se ha diseñado bajo una arquitectura modular, estableciendo una separación estricta entre la lógica de aprendizaje, el motor físico del entorno y la interfaz gráfica de usuario.

El flujo de información se estructura de la siguiente manera:

1. Inicialización: El módulo `Main.py` actúa como punto de entrada, iniciando el módulo de `Simulation`.
2. Ciclo Central (Core Loop): El módulo de simulación orquesta la interacción entre el `Game Engine` (estado actual y recompensas) y el `AI Agent` (toma de decisiones basada en Q-Learning).
3. Gestión de Datos: El agente lee y actualiza una `Q-Table` (tabla de valores Q) para optimizar su política de navegación, mientras que la interfaz carga recursos gráficos (`Assets`) para la representación visual.
4. Análisis: El sistema posee la capacidad de generar gráficos de rendimiento analítico mediante la librería `Matplotlib`.

3. Estructura de Módulos

La organización del código fuente responde a una jerarquía funcional específica:

- * `ai_agent/`: Constituye el núcleo computacional ("cerebro"). Alberga la lógica algorítmica de Q-Learning, define la política de exploración/explotación y gestiona la actualización de la tabla Q.
- * `game_engine/`: Define la física del entorno, las reglas de interacción, el sistema de colisiones y la función de recompensa.
- * `simulation/`: Funciona como el orquestador del sistema, conectando al agente con el motor del juego y administrando los ciclos de entrenamiento.
- * `interface/`: Módulo encargado de la renderización gráfica y la interfaz de usuario utilizando la biblioteca `Pygame`.
- * `assets/`: Repositorio de recursos gráficos (sprites e imágenes) requeridos para la visualización.

4. Requisitos de Instalación y Configuración

El sistema está desarrollado íntegramente en el lenguaje de programación Python.

4.1. Prerrequisitos

Es necesario disponer de un entorno Python funcional. Para la correcta operación del sistema, se deben instalar las siguientes dependencias externas:

- * pygame: Para la renderización del entorno y la interfaz.
- * matplotlib: Para la generación de gráficos analíticos de rendimiento.
- * Pillow: Para el procesamiento de imágenes.

4.2. Procedimiento de Instalación

Se debe clonar el repositorio y ejecutar la instalación de paquetes mediante el gestor pip utilizando los siguientes comandos en la terminal:

```
git clone https://github.com/ElGorel/TrustMeBro-AI.git
```

```
cd TrustMeBro-AI
```

```
pip install pygame pip install matplotlib pip install Pillow
```

5. Protocolo de Ejecución

Para iniciar la simulación y observar el comportamiento del agente en tiempo real, se debe ejecutar el script principal desde la raíz del directorio del proyecto.

Comando de ejecución:

```
python main.py
```

Resultado esperado: Al ejecutar el comando, el sistema desplegará una ventana de Pygame visualizando el entorno. Dependiendo de la configuración interna, se generarán gráficos de análisis de rendimiento mediante Matplotlib, ya sea durante el entrenamiento o al finalizar la ejecución.