

Killem

Resumen

Killem es un videojuego retro de jugabilidad sencilla y adictiva que pone a prueba tus reflejos. El objetivo consiste en aguantar el mayor tiempo posible sin morir frente a los enemigos que se aproximan ni de los disparos enemigos. Para ello, debes defenderte con tus disparos y tu escudo.

Uso

Para probar el programa, es necesario crear un entorno virtual con [Anaconda](#), aunque también puede ser creado con la propia utilidad que proporciona Python. En este tutorial se realiza con Anaconda, pero los pasos a seguir son los mismos.

Para la creación del entorno virtual ejecutar este comando:

```
conda create -n killem_env python=3.7
```

Antes de activar el entorno, es posible que sea necesario inicializar el shell:

```
conda init
```

Cierra la terminal y vuelve a abrirla para cargar la nueva configuración. Ahora, activa el entorno creado con el siguiente comando:

```
conda activate killem_env
```

A continuación, se instalaran los módulos necesarios:

```
pip install -r requirements.txt
```

Para ejecutar el programa, ejecutar el siguiente comando:

```
python main.py
```

Controles

TECLADO

- **←** : Disparo a la izquierda
- **→** : Disparo a la derecha
- **A** : Escudo a la izquierda
- **D** : Escudo a la derecha
- **ESC** : Pausar/Retoma
- **M** : Silenciar/Desilenciar
- **F3** : Mostrar/Ocultar FPS
- **F4** : Pantalla completa/Ventana

GAMEPAD (XBOX)

- **LB** : Disparo a la izquierda
- **RB** : Disparo a la derecha
- **←** o **X** : Escudo a la izquierda
- **→** o **B** : Escudo a la derecha
- **START** : Pausar/Retomar

Modos de juego

HUMANO

En este modo de juego será el propio humano el cual se enfrenta a innumerables oleadas de enemigos. Cada disparo acertado y cada disparo enemigo bloqueado suma un punto.

AGENTE

En este otro modo de juego puedes observar al agente inteligente jugar después de su largo y duro entrenamiento. Su récord personal se encuentra en cerca de 5500 puntos!

Estructura

A continuación se explica sin entrar en detalle, el árbol de directorios del código.

- `agent`: Es el directorio donde se encuentran las clases referentes al agente y al modelo de la red neuronal.
- `entities`: Aquí se definen todas las entidades y generadores de entidades.
- `game`: Directorio donde se encuentra la lógica de la partida así como botones para la interfaz grafica.
- `mixer`: Clases para el manejo de música y sonidos dinámicos.
- `model`: Es el modelo entrenado que emplea el agente para jugar al juego.
- `resources`: Todos los recursos gráficos y de sonido empleados en el juego.
- `util`: Clases y funciones auxiliares de utilidad para hacer soporte al resto del código
- `main.py`: Es el punto de partida del juego.