

Laboratorio Aprendizaje por Refuerzo

Javier Baladrón
Departamento de Ingeniería Informática
Universidad de Santiago de Chile

Instrucciones generales: Este laboratorio se realizará **en parejas**. La calificación final del laboratorio se calculará como el promedio simple de las notas de las 3 entregas. Las entregas deben realizarse a través de la Plataforma Campus Virtual.

Objetivo: El propósito de este laboratorio es aplicar diversos métodos de Aprendizaje por Refuerzo para desarrollar un agente capaz de jugar una versión simplificada del videojuego Bomberman.

En el juego Bomberman, los jugadores controlan a un personaje cuyo objetivo es encontrar la salida en un laberinto. El personaje puede moverse hacia arriba, abajo, izquierda o derecha en una cuadrícula. En el laberinto, se encuentran tres tipos de espacios: espacios de acceso libre, espacios bloqueados y espacios con cajas.

Además de los movimientos, el personaje puede colocar bombas en su posición actual. Una bomba explotará **6** unidades de tiempo después de ser colocada. La explosión abarcará el espacio donde se ubicó la bomba y los cuatro espacios adyacentes en forma ortogonal. Si la explosión alcanza una caja, esta se destruye y se convertirá en un espacio accesible. El personaje no puede pasar por los espacios ocupados por bombas (son considerados espacios bloqueados).

Para salir del laberinto, es necesario destruir una caja que tapa la puerta de salida. El personaje debe identificar bajo qué bloque se encuentra la salida, utilizar una bomba para destruirlo y luego desplazarse hasta esa posición. Sin embargo, las bombas también pueden matar al personaje. Si este se encuentra en una posición alcanzada por una explosión, el juego se pierde y termina inmediatamente.

El juego también presenta dos tipos de enemigos que siguen patrones de movimiento específicos. Un tipo de enemigo se desplaza solo horizontalmente, mientras que el otro tipo se mueve únicamente en dirección vertical. Los enemigos continuarán moviéndose en la misma dirección hasta que encuentren un obstáculo (ya sea un bloque o una bomba). En ese momento, cambiarán de dirección.

Si el personaje choca con algún enemigo, el jugador pierde y el juego concluye. Cuando un enemigo se encuentra en una posición alcanzada por una explosión, desaparecerá del juego.

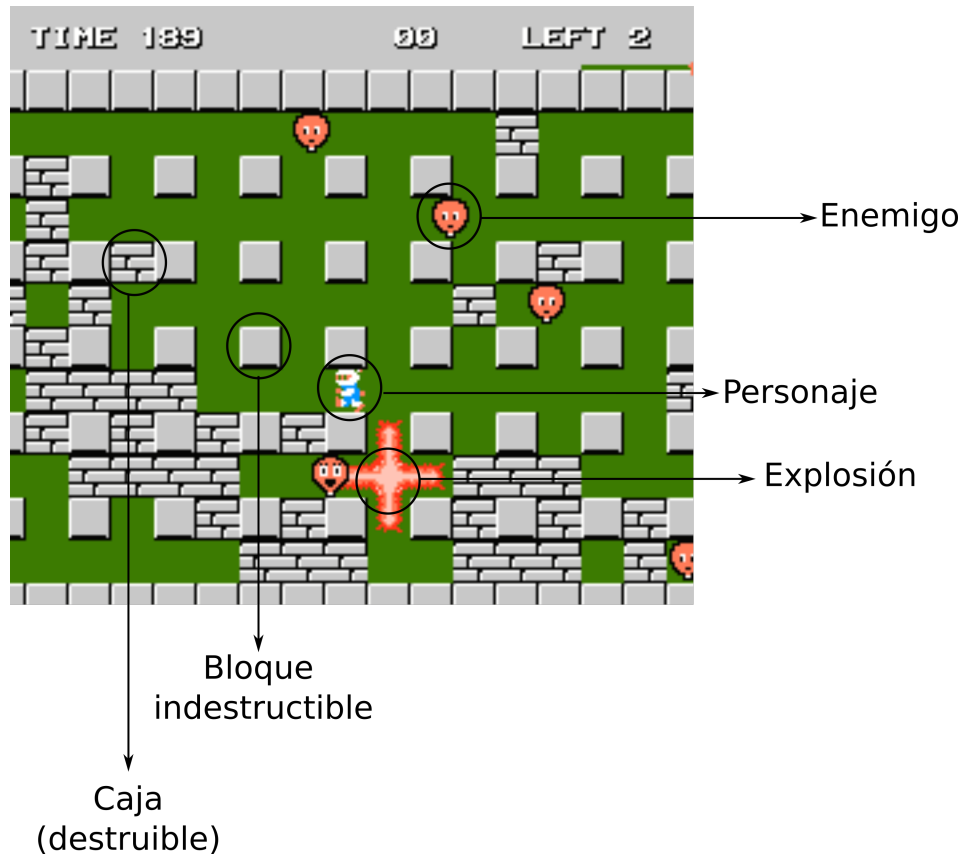


Figure 1:

Entrega 1: Implementación del ambiente - 17 Octubre

En esta entrega, se requiere implementar la versión del juego descrita en el entorno Gymnasium de Farama. Esto permitirá en las siguientes entregas entrenar a un agente utilizando Aprendizaje por Refuerzo.

La implementación debe permitir crear un nuevo entorno correspondiente a un nivel del juego. El constructor debe recibir el tamaño del nivel, la cantidad de cajas (siempre mayor que 1), la cantidad de enemigos con movimientos horizontales y la cantidad de enemigos con movimientos verticales. El entorno creado debe tener un bloque indestructible cada dos espacios, como se muestra en la imagen. Su implementación debe permitir asignar las cajas a una posición fija o como opción ubicarlas de manera aleatoria. Una caja debe ocultar la puerta.

La implementación debe incluir funciones de cambio de estado que permitan aplicar las reglas del juego. El diseño de la función de recompensa y la definición de estados debe ser propuesto por el grupo.

Además, se deben implementar dos agentes: uno que realice acciones aleatorias y otro que implemente un sistema de reglas creado por el grupo.

El 17 de octubre, deberán cargar un archivo de Jupyter Notebook que incluya la implementación del entorno y los agentes. El notebook también debe contener un análisis del rendimiento de los dos agentes implementados.

Deben ejecutar múltiples experimentos con cada agente, primero con un nivel pre-definido por ustedes y

luego con niveles aleatorios. Deben ejecutar cada agente múltiples veces, midiendo la cantidad de recompensa obtenida. Deben graficar el desempeño de cada agente, compararlos y llegar a conclusiones. El rendimiento debe calcularse como el promedio de múltiples ejecuciones con diferentes semillas aleatorias.

Entrega 2: Métodos Tabulares - 14 Noviembre

En esta etapa, se solicita la implementación de un agente que utilice el método de Monte Carlo y otro que emplee el algoritmo Q-learning. Pueden basarse en los ejercicios previos del curso para realizar esta implementación.

Deberán proporcionar un archivo Jupyter Notebook que contenga la implementación de ambos agentes y realice un análisis del rendimiento de dichos agentes. Su código debe ejecutar múltiples simulaciones con cada agente y graficar sus respectivos rendimientos. Además, se requiere ejecutar múltiples versiones de cada agente utilizando diferentes valores para la tasa de aprendizaje y el factor de descuento. Se debe mostrar el rendimiento obtenido con cada conjunto de parámetros probados. Finalmente, se espera que lleguen a conclusiones basadas en sus resultados.

Recuerden que el proceso implica:

Implementar los Agentes: Desarrollen la lógica y el código necesario para los agentes que utilicen el método de Monte Carlo y Q-learning. Asegúrense de que estos agentes interactúen adecuadamente con el entorno y sigan los algoritmos correspondientes.

Simulaciones y Evaluación: Ejecuten múltiples simulaciones con cada agente. En cada simulación, permitan que el agente interactúe con el entorno durante un número determinado de episodios y registren la recompensa acumulada en cada episodio.

Variación de Parámetros: Realicen varias versiones de cada agente, ajustando los valores de la tasa de aprendizaje y el factor de descuento. Esto les permitirá explorar cómo diferentes configuraciones afectan el rendimiento.

Análisis y Conclusiones: Utilicen los datos recopilados para graficar el rendimiento de los agentes en función del número de episodios y los parámetros elegidos. Comparando los resultados de ambas técnicas y las diferentes configuraciones de parámetros, lleguen a conclusiones sobre cuál agente y qué configuración tuvieron el mejor desempeño en el entorno del juego Bomberman.

Recuerden que la conclusión debe basarse en la interpretación de los resultados y en cómo se comportaron los agentes en el contexto del juego y las condiciones de prueba.

Entrega 3: Aprendizaje Profundo - 5 Diciembre

En esta etapa, se solicita la implementación de un agente que utilice DQN. Pueden basarse en los ejercicios previos del curso para realizar esta implementación.

De manera similar a la entrega anterior, deberán proporcionar un archivo Jupyter Notebook que contenga la implementación del agente y realice un análisis del rendimiento. Su código debe ejecutar múltiples simulaciones con el nuevo agente y graficar sus respectivos rendimientos. Además, deberá comparan su rendimiento con los obtenidos en la entrega 2. Finalmente, se espera que lleguen a conclusiones basadas en sus resultados. Deberán además realizar experimentos con niveles de mayor tamaño intentando llevar al límite de su capacidad a su implementación de DQN.