

LABORATORIO 1 - WORMS

Integrantes: Valentina Espinosa Rivera

Profesor: Roberto Gonzalez

Fecha de Entrega: 2019-10-05

Santiago de Chile

1 - 2019

ÍNDICE

Índice	1
Capítulo 1 . Introducción	2
1.1 Worms	2
1.2 Descripción del problema	2
Capítulo 2 . Definición de Objetivos	4
2.1 Objetivo General	4
2.2 Objetivos Específicos	4
2.3 Descripción de la solución	5
Capítulo 4. Conclusión	6
Capítulo 5. Bibliografía	6

CAPÍTULO 1 . INTRODUCCIÓN

1.1 WORMS

Worms consiste en un videojuego 2D originalmente lanzado a mediados de los '90. Se basa principalmente en técnicas vectoriales, es un videojuego de PC que se basa en vectores que tuvo fecha de lanzamiento en 1995 por la compañía Team17. El principal objetivo del juego es disparar y acabar con los demás enemigos evitando que sus disparos te lleguen.

1.2 DESCRIPCIÓN DEL PROBLEMA

El proyecto Consiste en crear un sistema o tablero para representar el juego Worms mediante funciones que permita generar un escenario como matriz (listas de listas) cuyas posiciones X e Y sean mayores a los enemigos que puedan existir.

La ubicación de los enemigos se dará aleatoriamente y los disparos tendrán un ángulo para evaluar el punto de caída.

Cada partida de juego es individual a partir de la generación automática de espacios, donde la aleatoriedad en la creación del espacio va asociada a la cantidad de enemigos a desplegar (Según el nivel de dificultad) y la ubicación y dirección de desplazamiento original de estos.

Los enemigos dentro del juego se desplazan libremente 1 posición por turno y tienen 1 disparo por turno. La dirección y velocidad del disparo varía solo en caso de colisiones entre disparos y jugadores o bien al recibir impactos de proyectil.

El personaje controlado por el jugador se desplaza en la dirección que uno le delimite, cuando esta se activa y luego continúa su movimiento en la misma dirección y sentido considerando los obstáculos en su paso. Lo único que puede alterar el curso de desplazamiento del jugador es disparar proyectiles o bien la colisión con un disparo, en cuyo caso el jugador es destruida.

Cada proyectil lanzado por el jugador puede reducir su velocidad si su desplazamiento va en la misma dirección que el proyectil, en cuyo caso se produce una aceleración en la dirección opuesta. Si el jugador se encuentra o alcanza el estado estacionario, entonces el jugador comienza su desplazamiento hacia la dirección opuesta hacia donde se efectúan los disparos.

El jugador puede disparar proyectiles para destruir los demás enemigos. La victoria se alcanza cuando los enemigos han sido reducidos a 0, lo que queda definido como un parámetro del juego.

Los proyectiles tienen un alcance máximo determinado por el parámetro L pudiendo alcanzar los bordes del espacio. Tras alcanzar el alcance máximo, el proyectil cae hacia $y = 0$.

El jugador puede disparar 1 proyectil por turno.

CAPÍTULO 2 . DEFINICIÓN DE OBJETIVOS

Con el problema medianamente planteado se decide realizar un objetivo general que será la meta principal del proyecto, y un listado de objetivos específicos los cuales son el medio para obtener este objetivo general.

2.1 OBJETIVO GENERAL

- Comprender lógicamente el funcionamiento del juego Worms trabajar con matrices mediante lenguaje funcional específicamente scheme y lograr hacer un programa con el cual se logre comprender el uso del paradigma funcional y la forma en la que se utilizan funciones y recursividad.

2.2 OBJETIVOS ESPECÍFICOS

- Se Diseña e implementa un TDA, en este caso corresponde al TDA de listas (listas de listas) con el cual podemos desenvolvemos de manera más eficiente con la problemática.

2.3 DESCRIPCIÓN DE LA SOLUCIÓN

Para poder concretar la solución mediante este paradigma se definirá una estructura llamada Tablero constituida por listas de de listas. además estas contienen los siguientes datos:

1. createScene: Función que retorna un espacio válido de tamaño NxM posicionando la cantidad de enemigos iniciales (parámetro E) Se debe usar recursión lineal (RL) en esta función o en alguna de las complementarias. El parámetro seed se refiere a la semilla para generar valores pseudoaleatorios (ver Nota 2). El jugador siempre parte al centro con orientación en 90°. Los parámetros D corresponde a la dificultad de la partida.

(createScene N M E D seed)

2. (checkScene scene): Función que recibe un espacio (scene) y permite verificar si éste cumple con los criterios para ser considerado como un espacio válido (Ej: Que es realmente un espacio con dimensiones válidas y enemigos y jugador válidamente ubicados dentro de este. El retorno de la función es de tipo booleano. Verdadero en caso de que se trate de un espacio válido y falso en el caso contrario.

(checkSpace space)

3. play: Función que permite especificar el curso de acción de el jugador otorgando las nuevas coordenadas de movimiento, no deben superar 5 unidades de diferencia entre el antiguo y el nuevo.

El retorno de la función es un nuevo espacio en tiempo $t+1$ con el resultado de haber cambiado la posición de el jugador. Este nuevo espacio en $t+1$ también toma en consideración la actualización del estado del juego en caso de terminar el nivel (ej: VICTORY), colisión de el jugador (DEFEAT).

7. shoot: Función que permite lanzar un proyectil. El proyectil tomará el parámetro de dirección de el jugador, mientras que la velocidad del proyectil está dada por la constante C . El retorno de la función es un nuevo espacio en tiempo $t+1$ con el resultado del desplazamiento del proyectil en ese instante de tiempo. Este nuevo espacio en $t+1$ también toma en consideración la actualización del estado del juego en caso de terminar el nivel (ej: VICTORY), colisión de el jugador (DEFEAT).

CAPÍTULO 4. CONCLUSIÓN

Mediante la realización de este laboratorio se logró elaborar una búsqueda a la problemática correspondiente mediante uso del lenguaje funcional(scheme) y a través de este se logró en parte comprender el uso de recursiones tipo lineal y cola además de su uso mediante funciones, creación de tableros mediante listas además de implementar el uso de TDA.

CAPÍTULO 5. BIBLIOGRAFÍA

Racket. (n.d.). Retrieved from <https://www.makigas.es/series/racket>

Racket Documentation. (n.d.). Retrieved from <https://docs.racket-lang.org/>

Worms (serie). (2019, August 30). Retrieved from [https://es.wikipedia.org/wiki/Worms_\(serie\)](https://es.wikipedia.org/wiki/Worms_(serie))