



APLICACIÓN DE LA TEORÍA DE AUTÓMATAS A UN VIDEOJUEGO UTILIZANDO COMO HERRAMIENTA JFLAP Y GREENFOOT

EDINSSON ADRIÁN MELO CALVO - 1151484

PRESENTADO A:

MSc. CARMEN JANETH PARADA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

TEORÍA DE LA COMPUTACIÓN

FACULTAD DE INGENIERIA

INGENIERIA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2017



RESUMEN

En el presente proyecto se muestra una de las varias aplicaciones que tiene la teoría de autómatas. En este caso, la aplicación a los videojuegos. Los estados de la máquina corresponden a los comportamientos del personaje o variables que cambian de acuerdo a varios eventos. Estos cambios están modelados por transiciones en el diagrama de estado. Las máquinas de estados ciertamente no son el medio más sofisticado para implementar inteligencia artificial en los juegos, pero el objetivo es observar la aplicabilidad que tiene la teoría de autómatas en este ámbito. Se parte de un autómata que representa el comportamiento del videojuego, utilizamos como herramienta JFLAP para su representación. Una vez obtenido el AFD se implementa internamente al videojuego, indicando el estado y la transición que se ejecuta en tiempo real.

OBJETIVOS GENERALES

- ✓ Comprender qué aplicabilidad tienen los autómatas en nuestro entorno o ámbito computacional.
- ✓ Adquirir conocimientos adicionales que nos induzcan a utilizar herramientas de apoyo, para complementar el aprendizaje en la teoría de autómatas.

OBJETIVOS ESPECÍFICOS

- ✓ Investigar sobre la aplicabilidad de la teoría de autómatas en el ámbito de los videojuegos.
- ✓ Dar uso práctico a herramientas como JFLAP y Greenfoot para modelar y comprender el funcionamiento de un autómata en un videojuego.



INTRODUCCIÓN

Greenfoot es un ambiente interactivo de desarrollo para propósitos educacionales. Permite la creación de gráficas y simulación de videojuegos. Es una herramienta fácil de utilizar, pero muy poderosa, ya que podemos crear complejos escenarios limitados por nuestra creatividad. JFLAP (del inglés, Java Formal Language and Automata Package) es un software que permite experimentar de forma gráfica con los conceptos relativos a la teoría de autómatas y lenguajes formales. Nos apoyaremos de esta herramienta para representar el autómata realizado.

La Teoría de los Lenguajes Formales tiene su origen en un campo aparentemente bastante alejado de la Informática: la Lingüística.

Los lingüistas de la llamada escuela estructuralista americana habían elaborado por los años 50 algunas ideas informales acerca de la gramática universal. Se entiende por gramática universal, una gramática que caracteriza las propiedades generales de cualquier lenguaje humano.

El primer trabajo que desarrolló teorías formales sobre gramáticas y lenguajes fue obra de Avram Noam Chomsky (1928-), quien es sin duda la figura más destacada de la lingüística moderna, tanto por desarrollar sus fundamentos matemáticos, como por sus teorías sobre el origen y la naturaleza de los lenguajes naturales, aunque éstas últimas son más discutidas (Chomsky, 1956; 1959; 1962; y 1963).

En el campo de la Informática, poco después de las primeras publicaciones de Chomsky, el concepto de Gramática Formal adquirió gran importancia para la especificación de lenguajes de programación; concretamente, se definió con sus teorías la sintaxis del lenguaje ALGOL 60 (con ligeras modificaciones sobre su versión primitiva), usándose una gramática libre de contexto. Ello condujo rápidamente al diseño riguroso de algoritmos de traducción y compilación.

Finalmente, y enlazando con el campo de la lingüística, la Teoría de Lenguajes Formales es de gran utilidad para el trabajo en otros campos de la Informática por ejemplo en Informática Teórica, Inteligencia Artificial, Procesamiento de lenguajes naturales (comprensión, generación, y traducción) y Reconocimiento del Habla.



PLANTEAMIENTO DEL PROBLEMA

A partir de un autómata finito determinista, representar un videojuego e implementar su funcionamiento utilizando como herramienta Greenfoot y JFLAP.

Sea el AFD del videojuego = $\{Q, \Sigma, F, q_0, \delta\}$

$Q = \{A, B, C, D, E, F, G, H, I\}$

$\Sigma = \{0, 1, 2 \dots 100\}$

$F = \{J\}$

$q_0 = \{A\}$

$\delta = \{(A, \text{Start}=0, A), (A, \text{Start}=1, B), (A, \text{Exit}=0, A), (A, \text{Exit}=1, J)\}$

$\delta = \{(B, \text{Lives}>1, B), (B, \text{Lives}=0, C), (B, \text{Score}<30, B), (B, \text{Score}=30, D)\}$

$\delta = \{(C, \text{Restart}=0, C), (C, \text{Restart}=1, A)\}$

$\delta = \{(D, \text{Lives}>1, D), (D, \text{HealthBoss}>1, D), (D, \text{Lives}=0, C), (D, \text{HealthBoss}=0, E)\}$

$\delta = \{(E, \text{Lives}>1, E), (E, \text{Score}<60, E), (E, \text{Lives}=0, C), (E, \text{Score}=60, F)\}$

$\delta = \{(F, \text{Lives}>1, F), (F, \text{HealthBoss}>1, F), (F, \text{Lives}=0, C), (F, \text{HealthBoss}=0, G)\}$

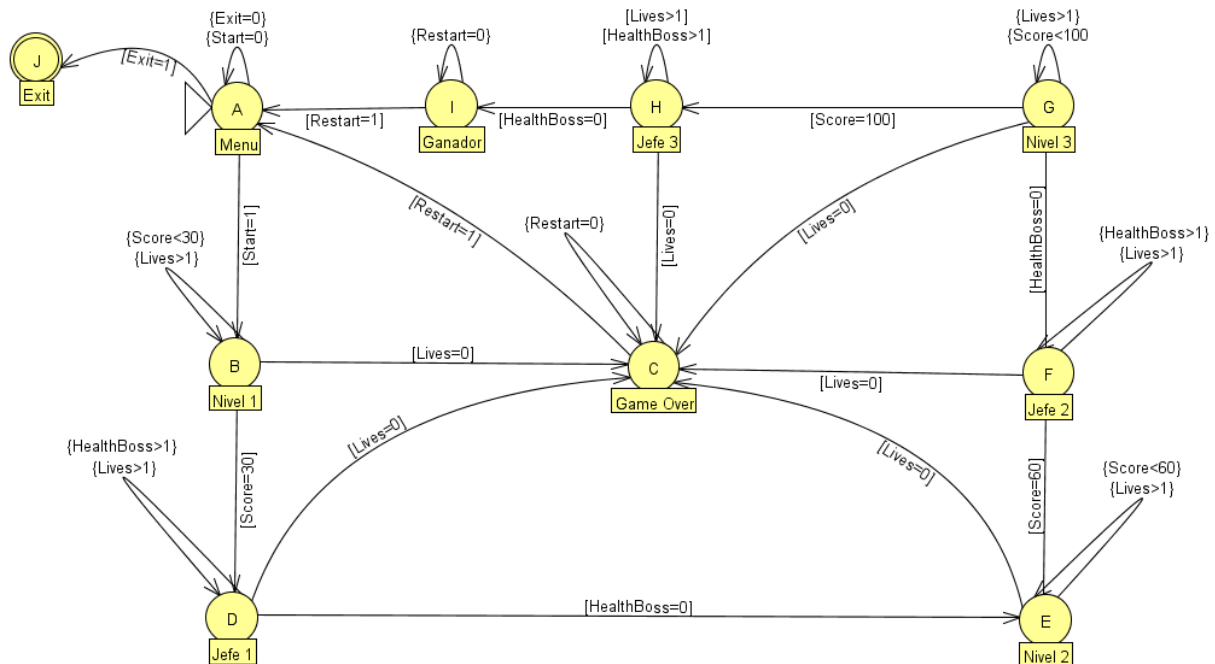
$\delta = \{(G, \text{Lives}>1, G), (G, \text{Score}<100, G), (G, \text{Lives}=0, C), (G, \text{Score}=100, H)\}$

$\delta = \{(H, \text{Lives}>1, H), (H, \text{HealthBoss}>1, H), (H, \text{Lives}=0, C), (H, \text{HealthBoss}=0, I)\}$

$\delta = \{(I, \text{Restart}=0, I), (I, \text{Restart}=1, A)\}$

REPRESENTACIÓN DEL AFD

Empezaremos con la representación del AFD utilizando JFLAP.



VARIABLES DE TRANSICIÓN

Start: Variable que representa si el botón “Start” es presionado. Puede tomar valores de 0 y 1.

- 0: Aún no presionado el botón “Start”, por lo tanto, se mantiene en el menú.
- 1: Botón presionado, por lo tanto, va al estado que representa el nivel 1.

Score: Variable acumulativa que representa los puntos que se obtienen durante el juego. Inicialmente tiene un valor de 0.

- Score (0-29): Se mantiene en el nivel 1. Cuando recibe Score (30) cambia escenario al Jefe1.
- Score (30-59): Se mantiene en el nivel 2. Cuando recibe Score (60) cambia escenario al Jefe2.
- Score (60-99): Se mantiene en el nivel 3. Cuando recibe Score (100) cambia escenario al Jefe3.

Lives: Variable que representa las vidas del jugador, inicialmente tiene un valor de 5.

- Lives (1-5): Se mantiene en el nivel actual.
- Lives (0): Se dirige al estado “Game Over”.

HealthBoss: Representa la barra de vida del jefe, inicialmente tiene el 100 de vida.

- HealthBoss (1-100): Se mantiene en el estado actual.
- HealthBoss (0): Se dirige al siguiente nivel.

Restart: Puede tomar valores de 0 y 1.

- 0: Se mantiene en el estado actual
- 1: Se va al estado “Menú”.

Exit: Puede tomar valores de 0 y 1.

- 0: Aún no presionado el botón “Exit”, por lo tanto, se mantiene en el menú.
- 1: Se va al estado terminal “Exit” y finaliza el juego.

FUNCIONAMIENTO EN GREENFOOT

Existen tres niveles, cada nivel tiene un jefe. Cuando se llega a una determinada cantidad de puntos se accede al jefe del nivel actual. Se implementó una ventana de estado, que muestra el estado actual y las transiciones que se ejecutan de acuerdo a los comportamientos en el juego.

El menú representa el estado inicial. A partir de este estado, se puede acceder a dos estados: Start (Empieza en Nivel 1) o Exit (Estado terminal, finaliza el programa).



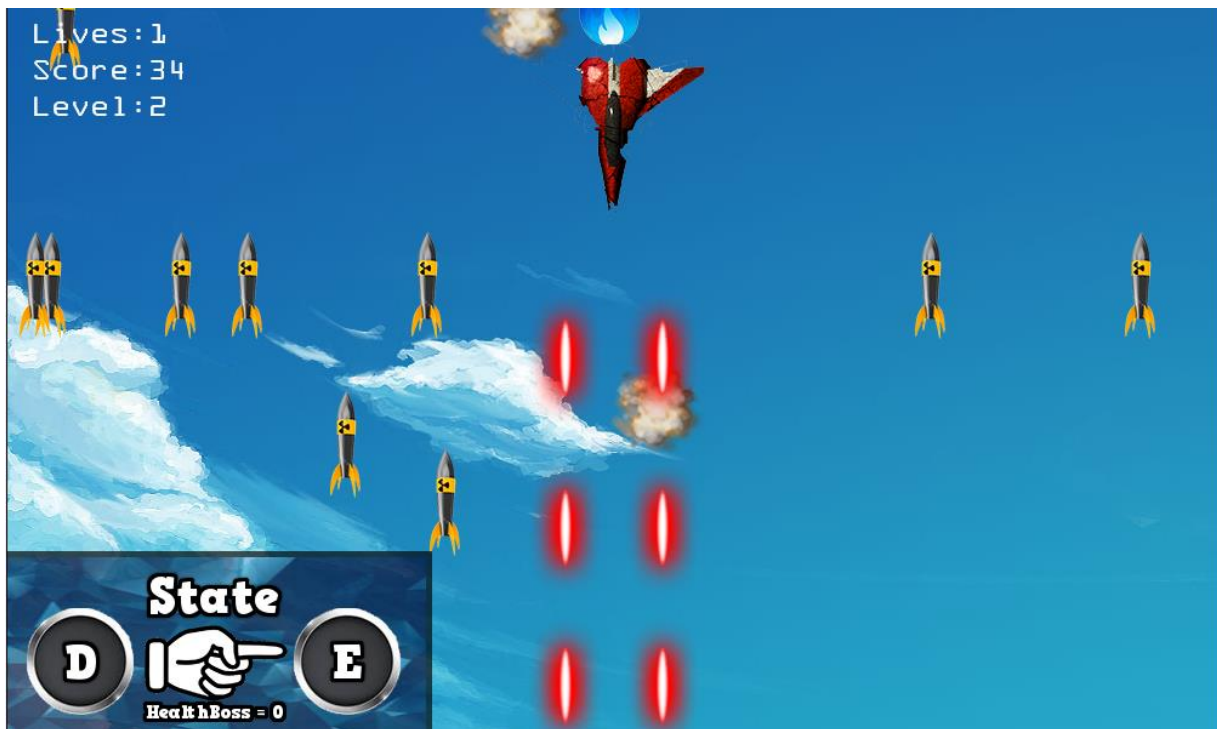
Este escenario es el estado ‘B’, representa el primer nivel. Mientras Lives>1 y Score <30 se mantiene en el nivel actual.



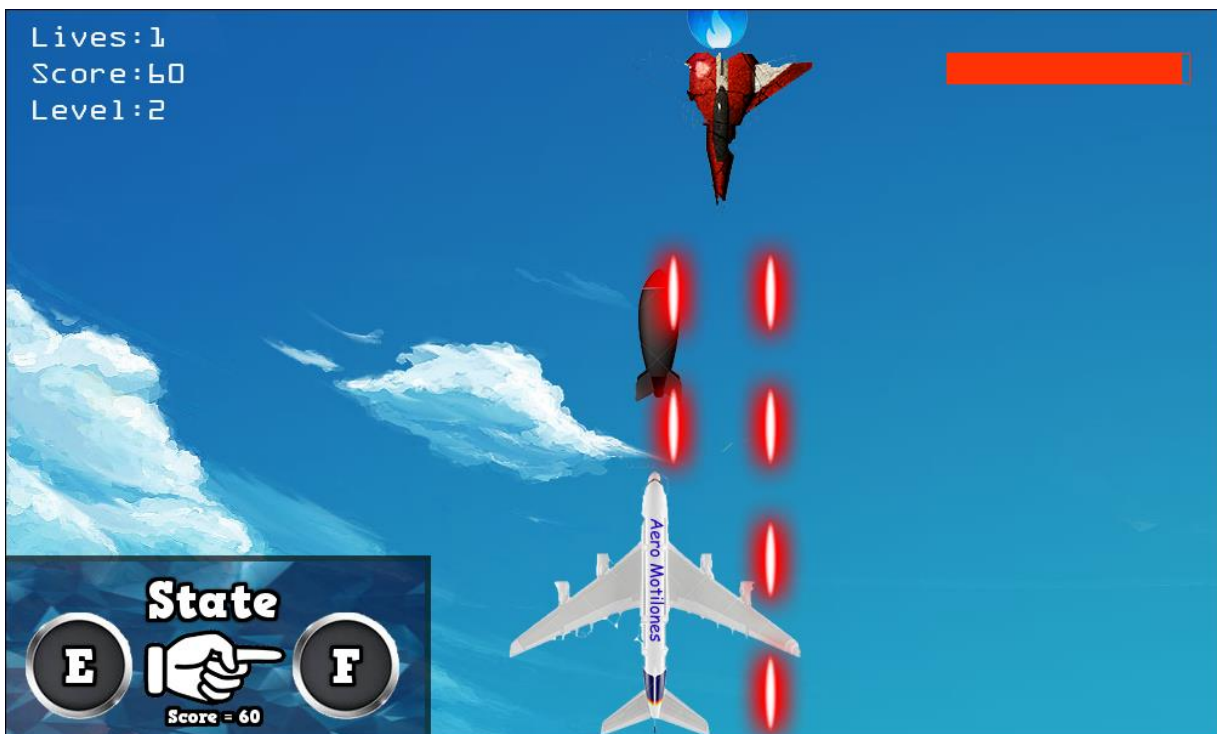
Este escenario es el estado ‘D’, representa el Jefe 1 en el juego. Es invocado cuando Score=30.



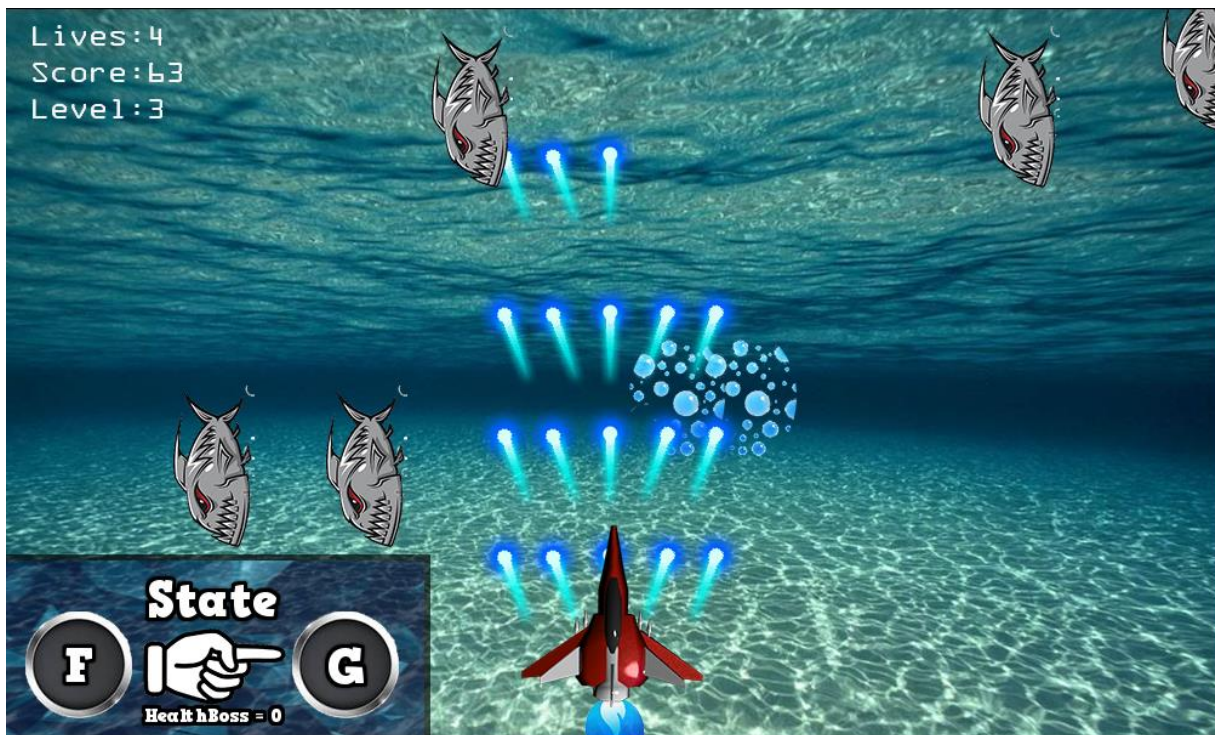
Si HealthBoss=0, pasa al siguiente nivel (en este caso, nivel 2), esto significa que el jefe del estado anterior fue derrotado.



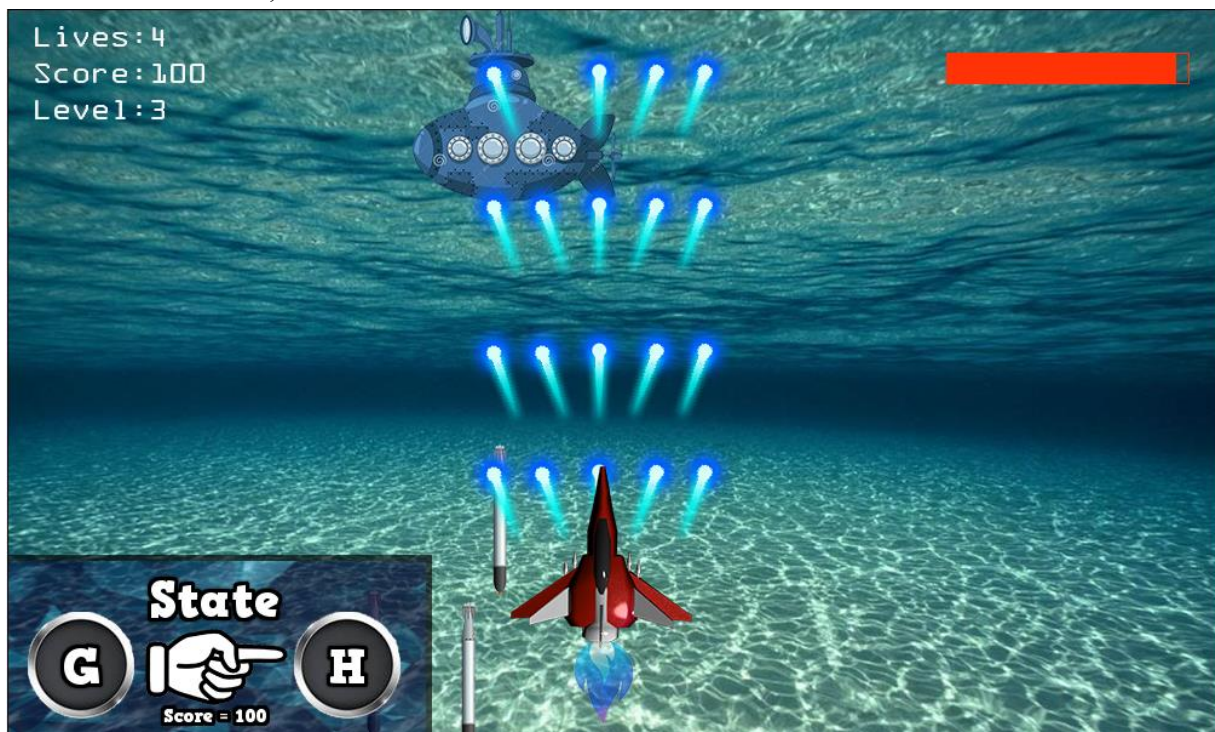
Cuando Score = 60, se hace la transición al escenario del “Jefe 2”.



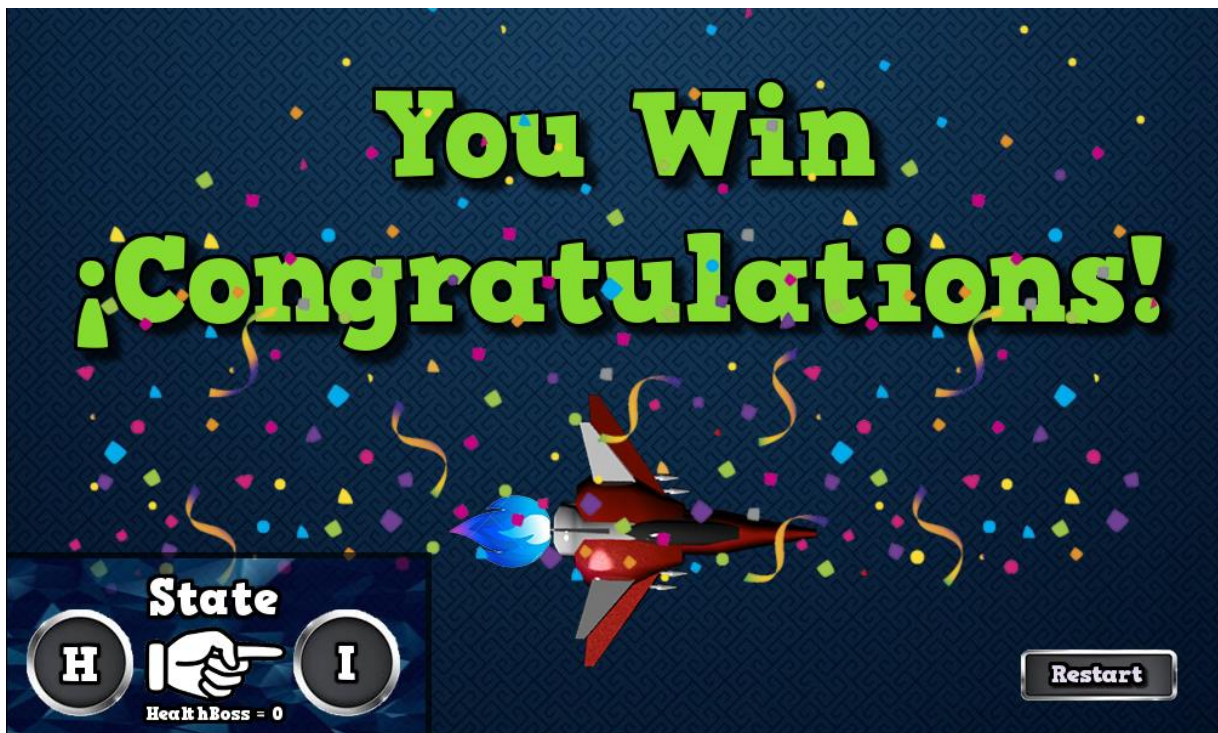
Si HealthBoss=0, pasa al siguiente nivel. Mientras Lives>1 y Score <100 se mantiene en el nivel actual (en este caso, nivel 3).



Cuando Score = 100, se hace la transición al escenario del “Jefe 3”.



Si HealthBoss=0, indica que el Jefe 3 murió, lo que nos lleva al estado “Ganador”. A partir de este estado podemos regresar al estado inicial y salir del juego con el estado terminal “Exit”.



DETALLES ADICIONALES

- ❖ Objetos como bolas de fuego azules, cuchillos de metal y anclas, presentan mayor dificultad de destruirse. Necesitan 2-5 disparos. Además, presentan un movimiento único:
 - Bolas de fuego azul: Movimiento en Zigzag.
 - Cuchillos de metal: Movimiento de persecución al objetivo.
 - Anclas: Movimiento parabólico (Ecuación cuadrática).
- ❖ A medida que los obstáculos llegan al final de la plataforma, aumentan su velocidad.
- ❖ Los jefes adquieren mayor resistencia a medida que aumenta el nivel.

BIBLIOGRAFÍA

- LENGUAJES, GRAMÁTICAS Y AUTÓMATAS. Segunda Edición, Juan Manuel Cueva Lovelle. Oviedo (España), 2001.
- APPLICATIONS OF DETERMINISTIC FINITE AUTOMATA, Eric Gribkoff, ECS 120, UC Davis, Spring 2013.
- SOFTWARE: Greenfoot, <https://www.greenfoot.org>
- SOFTWARE: JFLAP, <http://www.jflap.org>