

**Juan Diego Moron Flores, Oscar Fabian Sierra Daza, Jimmy Andrés Amezquita
Burgos, David Alejandro Alquichire Rincón**

No. de equipo de trabajo: 8

I. INTRODUCCIÓN

Se presenta el reporte técnico de la primera entrega del proyecto “Dot and Boxes” para el curso de estructuras de datos de la Universidad Nacional de Colombia, se plateara el problema, la solución que aplicaremos para resolver el mismo, se describirán los roles de los integrantes y se documentará los aspectos técnicos de este proyecto.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Se plantea el desarrollo de un videojuego llamado “Dots and Boxes” (“Puntos y cajas”), basado en un popular juego recreativo a lápiz y papel del mismo nombre; creado por el matemático francés François Édouard Anatole Lucas. El juego a desarrollar será para ser jugado por un jugador contra el computador (player vs computer). La esencia del juego consiste en dar una matriz de cuadrados con bordes transparentes, a modo de tablero, en donde los vértices de los cuadrados representan puntos que pueden ser unidos.

Los jugadores (en nuestro caso dos: player vs computer), tienen jugadas por turnos, donde una jugada consiste en unir mediante líneas verticales u horizontales dos puntos adyacentes en el tablero. Un jugador que completa el cuarto lado de un cuadrado gana cierta cantidad de puntos y le corresponde una nueva tirada. El juego termina cuando no hay más líneas que se pueden colocar y todos los cuadrados están llenos por los jugadores. El ganador del juego es el jugador con más puntos.

El objetivo principal del proyecto es el desarrollo de un oponente (bot), capaz de derrotar y dificultar las jugadas del jugador humano.

Resumiendo, las reglas del videojuego:

1. Toque en medio de dos puntos para marcar una línea (vertical u horizontal). Su objetivo es completar una caja de cuatro lados.
- 2.El jugador que completa una caja de cuatro lados tiene la oportunidad de jugar una vez más.
- 3.Completar tantas casillas como sea posible.
- 4.El jugador que complete mayor cantidad de cuadrados, gana el juego.

Según Kenneth Rosen*, la estructura de datos Árbol, se puede utilizar para analizar distintos tipos de juegos, como las damas o el ajedrez, debido a que cada jugador conoce los movimientos del otro, y por lo tanto no hay componentes de azar en el juego. En nuestro caso, en “Dots and boxes”, usaremos árboles

para implementar el videojuego, dado que el bot, es decir, el contrincante del jugador humano, debe buscar el camino más eficiente para no perder, o en otras palabras, el camino más eficiente para cerrar un cuadro. Por otra parte, deberá “engañar” al usuario para que este deje libre un solo lado por marcar de un cuadro, cosa que hará el bot en el siguiente turno. Esta búsqueda de caminos hará uso de la estructura de datos referida arriba, un árbol, ya que, partiendo de una jugada (raíz), se conocen todas las posibles próximas jugadas (hijos). El proyecto sirve como una herramienta para asentar los conocimientos impartidos en el aula de clase, y cuando de asentar conocimientos se trata, la mejor y más efectiva forma de lograrlo es a través de la práctica. Y si la práctica se hace didáctica, donde se aprende, practica y estudia, disfrutando del proceso; ¡mucho mejor!

III. USUARIOS DEL PRODUCTO DE SOFTWARE

Este proyecto al ser un video juego está destinado a todas aquellas personas que busquen un rato de entretenimiento jugando ya sea contra un contrincante humano o tomando el reto de enfrentar a la inteligencia artificial “Skyline”.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

1. Creación de usuario

- Descripción: La aplicación debe ser capaz de crear usuarios con el fin de guardar sus partidas y estadística dentro del juego,
- Acciones: El jugador inicia la aplicación, selecciona la opción crear cuenta.
- Requerimientos funcionales: La aplicación debe ser capaz de generar usuarios acordes con el registro realizado, con identificación única por jugador

2. Ingreso de usuario

- Descripción: La aplicación dará permiso de ingreso a usuarios anteriormente registrados
- Acciones: El jugador inicia la aplicación, luego ingresa su usuario y contraseña previamente registrada. El jugador selecciona la opción login

- Requerimientos funcionales: La aplicación debe permitir el ingreso de los usuarios previamente registrados, donde coincidan su usuario y contraseña

3. Menu de opciones del jugador

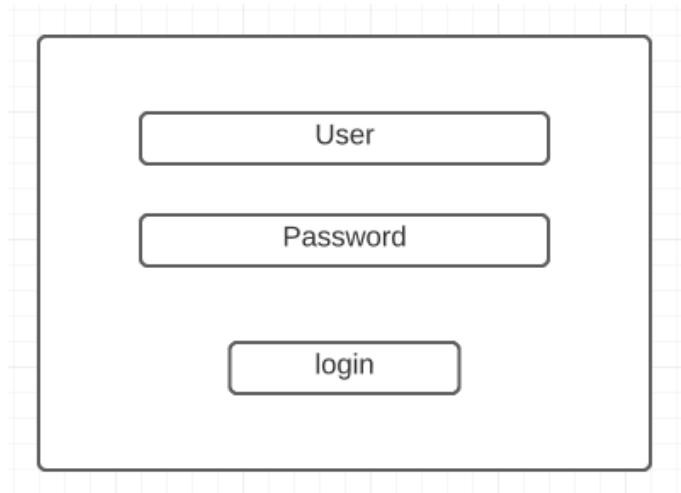
- Descripción: La aplicación después de que el usuario ingrese, mostrará las diferentes opciones de juego.
- Acciones: El usuario se loguea en la aplicación. La aplicación muestra su menú donde se presentan todas las opciones de juego
- Requerimientos funcionales: La aplicación debe permitir realizar configuraciones predefinidas para el juego. También permitirá que el jugador seleccione la modalidad de juego y debe ajustar la dificultad de la inteligencia artificial, según la opción seleccionada por el jugador.

4. Juego Dot and Boxes, jugador vs IA.

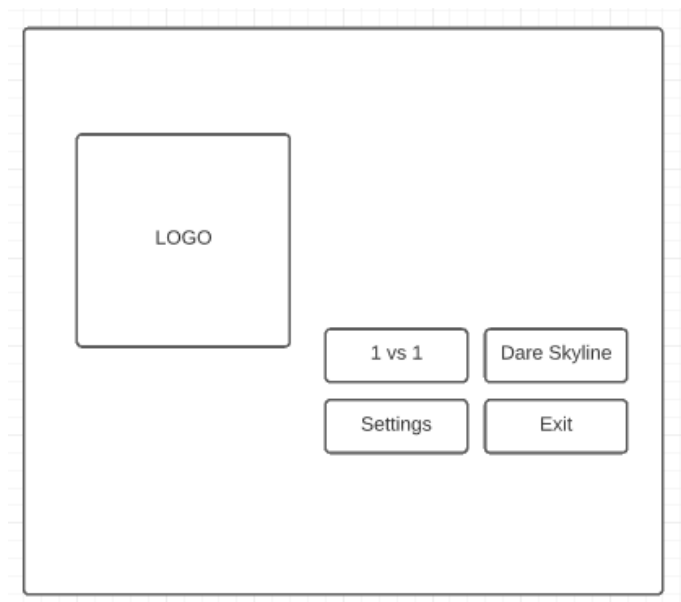
- Descripción: La aplicación permite al usuario la posibilidad de jugar contra la inteligencia artificial.
- Acciones: El usuario se loguea en la aplicación, la aplicación despliega el menú principal, el usuario selecciona la opción de juego vs inteligencia artificial y la aplicación muestra tablero de juego
- Requerimientos funcionales: La aplicación presentará una interfaz funcional acorde al juego Dot and Boxes, además que guardará las estadísticas del usuario en el juego.

5. Juego Dot and Boxes, Jugador vs Jugador

- Descripción: La aplicación permite al usuario la posibilidad de jugar contra otro jugador dentro de la aplicación
- Acciones: El usuario se loguea en la aplicación, se muestra el menú principal y se selecciona la opción de 1 vs 1. Luego se despliega el tablero de juego
- Requerimientos funcionales: La aplicación presentará una interfaz funcional acorde con el juego Dot and Boxes. Además que guardará las estadísticas del usuario en el juego.



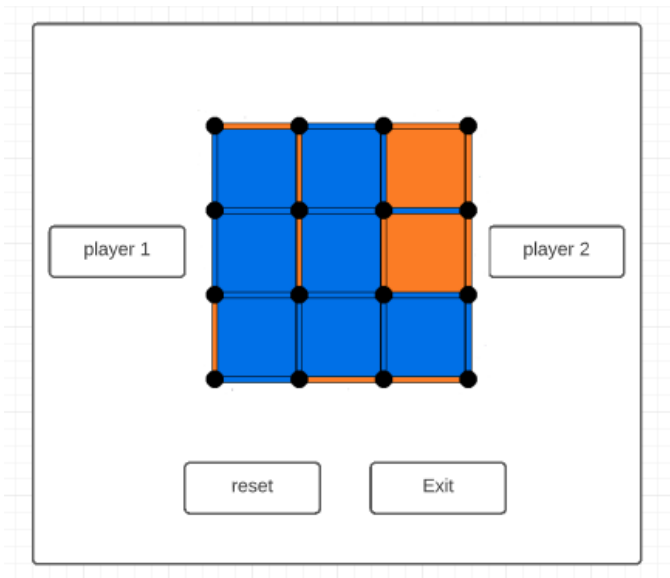
Una vez dentro de la aplicación un menú se desplegará donde el usuario será capaz de seleccionar el modo de juego así como la dificultad.



El juego debe presentarse de la siguiente forma:

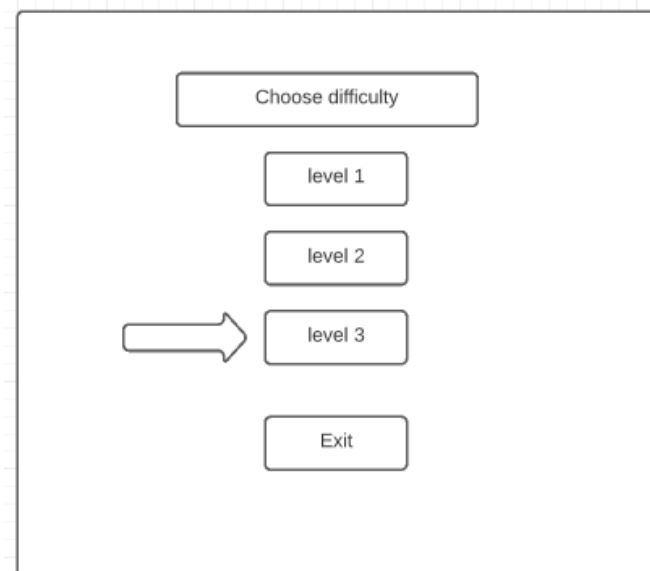
V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

La aplicación debe tener funcionalidades de login y registro de usuario.



Donde ambos usuarios pueden ver su puntuación actual, así como reiniciar el tablero actual y salir.

Se presenta además el menú de settings donde se debe seleccionar la dificultad de la inteligencia artificial que jugará contra el humano.



Para el desarrollo, se usará C++ y la librería gráfica Qt. Estará disponible en los sistemas operativos Windows y Linux. Para el control de versiones y repositorio se usará la plat

VI. PROTOTIPO DE SOFTWARE INICIAL

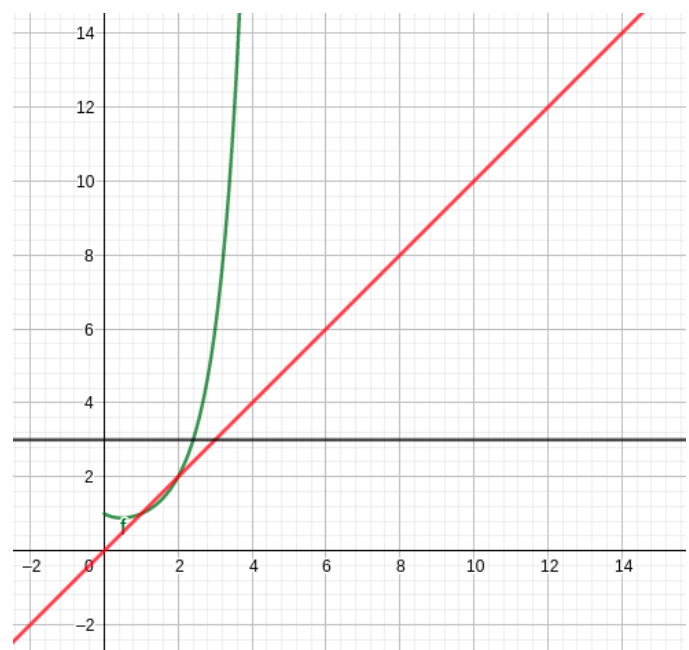
El prototipo del software inicial lo puede encontrar aquí: <https://github.com/juankipedia/DotAndBoxes>

VII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Debido a la temática del juego y a la forma en la cual será programado, este proporciona un usuario virtual (bot), inteligente en cierta forma, el cual tiene como fin ser un buen contraste. Naturalmente, tratará de ganar sobre el jugador. En consecuencia, el bot debe evaluar la cantidad de combinaciones y jugadas posibles a realizarse. Es por esto que el conjunto de posibilidades será estructurado en árboles, específicamente en árboles estructurados con listas. Se elige esta estructura de datos, debido a que para simular cualquier tipo de agrupación, ya sean variaciones, permutaciones o combinaciones, los árboles son realmente útiles. El árbol, vendrá dado por la raíz, la cual contendrá la jugada actual que se acaba de dar en el último turno, y cada uno de los hijos de esta raíz, contendrán cada posible jugada a tomar por el bot. De esta forma, seleccionará la mejor jugada. La dificultad que el “usuario virtual” le proporcionará al juego, dependerá de la cantidad de subjugadas que el mismo tendrá la capacidad de analizar. El tablero del juego será representado como una matriz de puntos es decir un arreglo bidimensional.

VIII. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

A continuación presentamos la gráfica de 3 funcionalidades básicas de nuestra inteligencia artificial, calcular el árbol de decisiones ■, buscar la mejor jugada a futuro dado un árbol de decisiones ■, obtener jugada dado un árbol de decisiones ■. Esta gráfica nos muestra el nivel de complejidad de la inteligencia artificial (eje x) vs la cantidad de operaciones realizadas (eje y).



Ahora vemos cómo se comportan en el tiempo las operaciones realizadas por la inteligencia artificial dado el nivel que se selecciona para la misma.

n	build_tree	find_future_move	get_move
1	0.000016	0.000016	0.000016
2	0.0544	0.000034	0.000016
3	0.25	0.000051	0.000016
4	1.04	0.000070	0.000016
5	3.02	0.000153	0.000016

Como podemos ver, dada la característica factorial $O(n!)$ de la función build tree, la cual es la más costosa de toda la aplicación, vemos que la cantidad de operaciones que en ella se realizan crecen considerablemente, mientras que el buscar una operación a futuro depende únicamente del nivel seleccionado, el obtener una jugada será siempre tiempo constante.

IX. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

Video demostración: https://youtu.be/Ty0qu_eN_xA

X. ROLES Y ACTIVIDADES

A continuación, se presentan los roles y actividades desarrollados por cada integrante en esta primera entrega:

Integrante	Rol	Actividades
Juan Diego Moron	IA developer, Experto algoritmos.	Desarrollo del árbol de decisiones.
Oscar Sierra	IA developer, Experto algoritmos.	Desarrollo de algoritmo de decisión
Jimmy Amezcua	Desarrollador lógica del juego	Desarrollo del tablero y lógica del juego.
David Alquichire	Experto interfaz de usuario	Desarrollado de interfaz de usuario.

XI. DIFICULTADES Y LECCIONES APRENDIDAS

Dada las características del proyecto una de las dificultades que tenemos que afrontar es cómo realizar una buena inteligencia artificial, que juegue bien y que presente un reto importante a su contrincante además de esto se necesita estudiar el juego con el fin de darle ciertos “trucos” a nuestra inteligencia artificial.

Otra dificultad importante fue el uso de librerías gráficas con el lenguaje de programación C++, dado que este no cuenta con

demasiados entornos gráficos y los que existen suelen ser bastante complejos y difíciles de usar.

XII. REFERENCIAS BIBLIOGRÁFICAS

[1] Rhodes, N. Data Structures UCSD. Coursera. Retrieved 23 November 2021, from <https://www.coursera.org/learn/data-structures>.