

# **Juego Dot and Boxes**

## **Grupo dos**

### **ID del equipo de trabajo**

#### **Autores**

**Oscar Sierra**  
**Juan Diego Moron**  
**Jimmy Amezquita**  
**David Alquichire**

*Sede Bogotá*

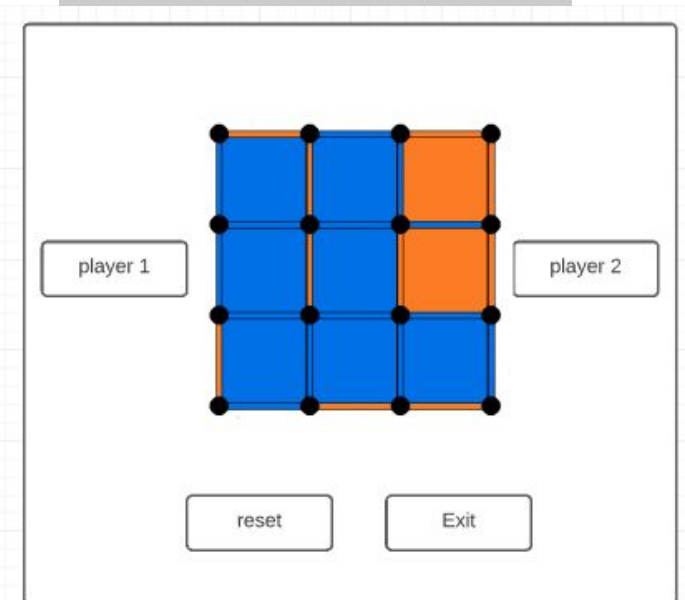
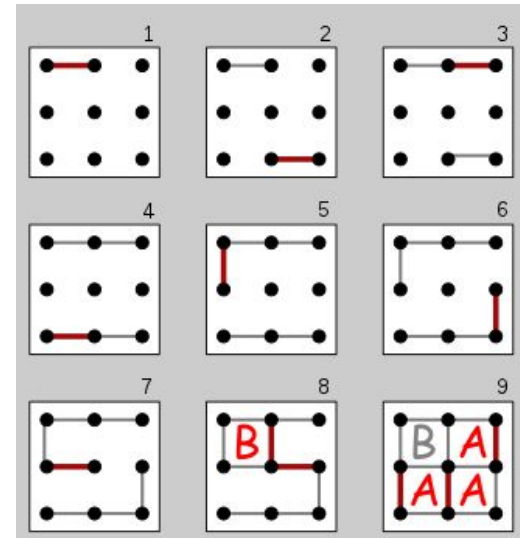


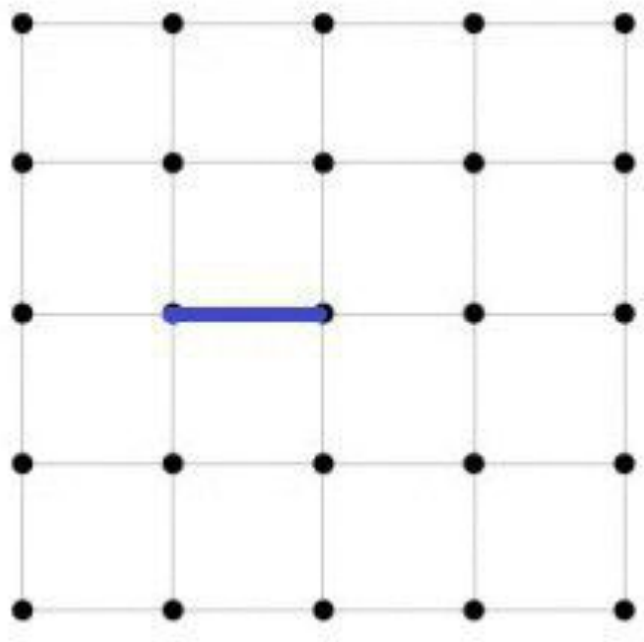
UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# Proyecto Dot and Boxes

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





**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 cuadrados, gana el juego

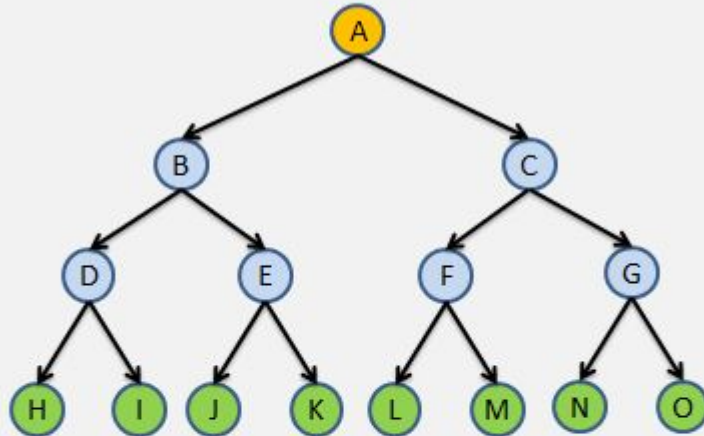


# Requerimientos funcionales

- La aplicación debe ser capaz de generar usuarios acordes con el registro realizado, con identificación única por jugador.
- La aplicación debe permitir el ingreso de los usuarios previamente registrados, donde coincidan su usuario y contraseña.
- La aplicación debe permitir realizar configuraciones predefinidas para el juego.
- La aplicación debe permitir que el jugador escoja la modalidad de juego.
- La aplicación debe ajustar la dificultad de la inteligencia artificial, según la opción seleccionada por el jugador
- La aplicación presentará una interfaz funcional acorde con el juego Dot and Boxes.
- La aplicación guardará las estadísticas del usuario en el juego.



# Uso de estructuras de datos

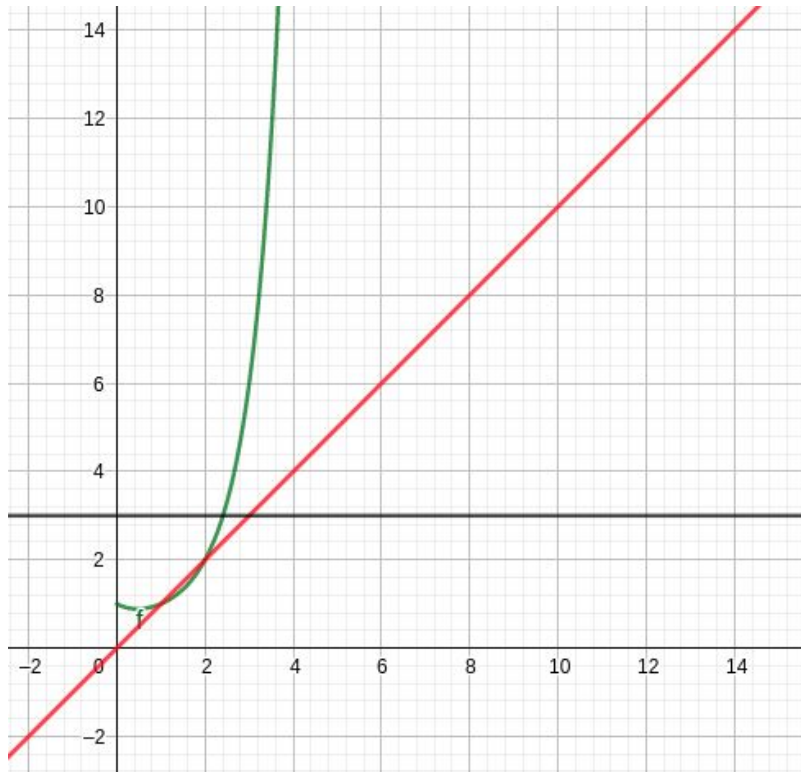


Según Kenneth Rosen\*, la estructura de datos Árbol, se puede utilizar para analizar distintos tipos de juegos, como las damas o el ajedrez, debido a que cada jugador conoce los movimientos del otro, y por lo tanto no hay componentes de azar en el juego.




El bot (“Usuario Virtual”) que se enfrenta con el jugador, 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.



# Pruebas y análisis comparativo del uso de las estructuras de datos



A continuación presentamos la gráfica de 3 funcionalidades básicas de nuestra inteligencia artificial:

-  Buscar la mejor jugada a futuro dado un árbol de decisiones
-  Calcular el árbol de decisiones
-  Obtener jugada dado un árbol de decisiones

-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.



# Lenguajes de programación y herramientas de software usados



Qt Creator es un IDE para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario con las bibliotecas Qt para C++





# Referencias

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