

LICENCIATURA EN PRODUCCIÓN DE SIMULADORES Y VIDEOJUEGOS

Explorando décadas de evolución:
La Influencia de la IA en las mecánicas de los Videojuegos y el análisis de cómo afecta a la experiencia del usuario.

Alumno: Gian Franco Tripodaro

Tutor de tesis:

Buenos Aires, 1 de octubre de 2023

Agradecimientos y dedicatorias:

Le agradezco a todos los chicos y chicas que se ofrecieron voluntariamente a ser parte de la muestra para lograr el experimento.

También le agradezco a mis compañeros de cursada, que hicieron mucho más divertido cada día de clases, y me han ayudado mucho, no solo en lo académico, sino también en lo personal, son grandes personas y les deseo todo lo mejor, ojalá seguir estando juntos en un futuro.

Le agradezco a mis padres y a mi hermano por siempre apoyarme en cada proyecto que me propongo.

Índice General

Contenido

Lista de abreviaturas de tablas y símbolos 7
Resumen 8
Abstract 9
Introducción 11
Variables Intervinientes 13
Justificación 14
Objetivo 15
Hipótesis 17
Investigación y análisis de la IA a lo largo de la historia del videojuego: 19
Definición de inteligencia artificial: 19
Aplicaciones de la inteligencia artificial en videojuegos: 19
Aprendizaje por refuerzo: 19
Basado en reglas lógicas: 20
Búsqueda de caminos o PathFinding: 20
Dificultad dinámica o adaptativa: 21
Árbol de decisión: 21
Generación procedural: 22
ChatBot o generación de texto: 23
Metodología 25
Metodología desarrollo de videojuego 25
Metodología de la investigación 25
Muestra 26
Procedimiento 27
Hardware 28
Software 28
Planteamiento del proyecto: 29
Proyecto número 1 29
Tipo de IA que se va a utilizar: 30
Condición de victoria 32
Condición de derrota 33
Proyecto número 2 34

Tipo de IA que se va a utilizar 36
Resultados y análisis de los videojuegos 43
Desafíos del desarrollo e implementación: Proyecto número 1 43
Resultados: Proyecto número 1 53
Desafíos del desarrollo e implementación: Proyecto número 2 53
Resultados: Proyecto número 2 61
Resultados y análisis de la muestra 62
Los jugadores que lograron vencer a la IA fueron 2: 62
Jugadores que no pudieron vencer a la IA: 64
Análisis global: 72
Conclusiones, recomendaciones y trabajos futuros. 75
Referencias 76
Anexo 77
Página itch.io para jugar al segundo proyecto: 77
Videos con el funcionamiento de las IA´s y Builds de los proyectos: 78
Videos de los participantes jugando: 78
Respuestas de las encuestas: 78

Índice de contenido

Proyecto 1 gameplay 30
Proyecto 1 gameplay – celular 1 31
Proyecto 1 gameplay – celular 2 31
Condición de victoria 32
Condición de derrota 33
Proyecto 2 gameplay 35
Estados de la IA 37
Variables relevantes 37
Estados de la IA y sus variables 38

Animator controller....39
Menú – modos de juego....39
Tutorial - Bienvenida....40
Modo de juego - Futpool....41
Modo de juego – 2 VS 2....42
Codigo integración Chatgpt - 1....43
Codigo integración Chatgpt - 2....44
Codigo integración Chatgpt - 3....45

Codigo integración Chatgpt - 4....46
Codigo integración Chatgpt - 5....46
Pagina OpenAI Api Keys....47
ChatBot Integrado en Unity – 1....48
ChatBot Integrado en Unity – 2....49
Error de acceso a OpenAI desde Unity....
51
Código - Buscar Pelota Comportamiento....53
Código - Mirar Pelota....54
Condición principal mirar pelota....55
Condiciones disparar al arco....55
Código – Estado patear pelota comportamiento....56
Código – Patear pelota acción....56
Código – Despejar pelota....57
Código – Ir llevándose la pelota....57
Código – StatesMachineController obtención de variables....59
Nivel de satisfacción - Preguntas del modo de juego "Player vs IA" (Con IA)....
64
Nivel de satisfacción - Preguntas del modo de juego "Freekick" (Sin IA)....65
Nivel de frustración - Preguntas del modo de juego "Freekick" (Sin IA)....66
Nivel de frustración - Preguntas del modo de juego "Player vs IA" (Con IA)....66
Nivel de desafío - Preguntas del modo de juego "Player vs IA" (Con IA)....67
Nivel de desafío - Preguntas del modo de juego "Player vs IA" (Con IA)....68
Nivel de estrategia - Preguntas del modo de juego "Freekick" (Sin IA)....69

Nivel de estrategia - Preguntas del modo de juego "Player vs IA" (Con IA)..... 69
¿Volvería a jugar? - Preguntas del modo de juego "Player vs IA" (Con IA).....70
¿Volvería a jugar? - Preguntas del modo de juego "Freekick" (Sin IA).....71
Nivel de experiencia jugando videojuegos - Preguntas globales.....72
¿Qué videojuegos jugas habitualmente? - Preguntas globales..... 73
¿Cree que la incorporación de la IA mejoró la experiencia? - Preguntas globales.....74
¿Cree que la incorporación de la IA mejoró la experiencia? - Preguntas globales.....75
Consentimiento informado..... 79

Lista de abreviaturas de tablas y símbolos

Inteligencia artificial (IA).

non-player character / Personaje no jugador (NPC).

Personaje Jugador (PJ).

Player 1 (P1)

Inteligencia artificial capaz de entender y escribir texto coherente (ChatBot)

Resumen

Esta investigación busca entender y analizar cómo la implementación de la Inteligencia artificial (IA) puede lograr mejores experiencias, ya sean más divertidas, inmersivas, desafiantes, entretenidas o interesantes en los videojuegos.
Esta respuesta se encontró a través del estudio a lo largo de la historia de la industria de videojuegos. Además del desarrollo de un prototipo propio.
El cual sería puesto a prueba por una muestra de diversos participantes.

En esta tesis se desarrollan 2 prototipos para poder llevar a cabo dicha investigación.

El primero es un juego de tipo “preguntas y respuestas” donde hay que conversar con un chatbot para obtener las respuestas necesarias.
Pero fue descartado ya que la utilización del mismo, al ser tecnología de terceros, requiere estar de acuerdo con términos y condiciones que no permitían la continuación del prototipo.

El segundo es un videojuego de fútbol, 1vs1, donde el objetivo es anotar 5 goles en el arco rival, al mismo tiempo que se defiende el arco propio.
Se desarrollaron distintos tipos de mecánicas para hacer divertida e interesante la jugabilidad.
También se desarrollaron varios modos de juego para el proyecto de fútbol, pero la investigación se centra principalmente en 3 de ellos.

El primero es el modo “Tutorial”:
Este modo se centra en explicar todos los controles y mecánicas básicas del juego. Asegurándose de que el jugador se sienta listo para disfrutar la experiencia de manera adecuada.

El segundo es el modo sin IA o modo “Penalty”:

El cual se centra en un modo de juego libre para el usuario, pero teniendo enfrente un arquero, que sus únicos movimientos son ir de un costado al otro del arco y disparar la pelota en caso de entrar en contacto con la misma.
A este personaje no se le puede considerar IA ya que sus movimientos se encuentran predeterminados, siguiendo un patrón fijo, sin importar lo que ocurra en la cancha.

Por último, el tercer modo de juego es el “Jugador vs IA” o “P1 vs CPU”:
Donde se enfrenta el jugador ante una IA que puede reaccionar a lo que está ocurriendo en cada momento de la partida para vencer al rival, siendo todo un desafío. Estando el jugador obligado a crear una estrategia para poder vencer a la máquina.
Esta IA puede moverse libremente por el escenario, yendo a buscar la pelota, regateando, defendiendo y pateando al arco.

Abstract

This research aims to understand and analyze how the implementation of Artificial Intelligence (AI) in video game mechanics can achieve more fun, immersive, challenging, entertaining, or interesting experiences. This insight was gleaned through studying the history of the video game industry, as well as the development of a prototype game. The prototype would be played by a sample of diverse participants.

This thesis involves the development of two prototypes to conduct this research.

The first is a "question and answer" type game where players interact with a chatbot to obtain the necessary answers. However, it was discarded because its use, being third-party technology, required agreeing to terms and conditions that did not allow the continuation of the prototype.

The second is a soccer video game where the objective is to score 5 goals in the opponent's goal while defending one's own goal. Various mechanics were developed to make the gameplay fun and interesting. Several game modes were also developed for the soccer project, but the research primarily focuses on three of them.

The first is the "Tutorial" mode:
This mode focuses on explaining all the controls and basic mechanics of the game, ensuring that the player feels ready to enjoy the experience properly.

The second is the mode without AI or "Penalty" mode:
This mode offers free gameplay for the user, but with a goalkeeper in front who only moves from side to side of the goal and shoots the ball if it comes into contact with it. This character cannot be considered AI since its movements are predetermined, following a fixed pattern regardless of what happens on the field.

Finally, the third game mode is "Player vs AI" or "P1 vs CPU":
Here, the player faces off against an AI that can react to what is happening at every moment of the match to defeat the opponent, posing a real challenge. The player is obliged to create a strategy to beat the machine. This AI can move freely around the field, dribble, defend, and shoot at goal.

Introducción

La industria de los videojuegos ha experimentado un crecimiento exponencial en los últimos años, y la inteligencia artificial (IA) ha emergido como un componente clave en la mejora de la experiencia de juego. La integración efectiva de la IA en la mecánica de los videojuegos plantea desafíos técnicos y de diseño que deben ser considerados. El presente estudio investigará los elementos que se deben tener en cuenta al incorporar la inteligencia artificial, para poder desarrollar un videojuego que pueda implementar una IA, con el fin de poder estudiar y analizar cómo afecta a la experiencia de los jugadores.
Ya habiendo analizado y estudiado distintos casos a lo largo de la historia.

La falta de conocimientos técnicos y experiencia necesarios para implementar eficazmente la IA en la mecánica de videojuegos puede representar un desafío significativo para los desarrolladores. Este problema se manifiesta en diversos aspectos, tales como la complejidad técnica, la asignación de recursos limitados, el equilibrio de juego y la optimización del rendimiento.

En el contexto de la incorporación de la inteligencia artificial en la mecánica de los videojuegos, Surge la pregunta:

¿Cómo afecta la implementación de la Inteligencia Artificial, en las mecánicas de los videojuegos, a la experiencia global del jugador?

La variable dependiente es:
La experiencia global del jugador, es decir,
¿Cómo la implementación de IA en las mecánicas principales afecta la experiencia total del jugador?

Para poder desarrollar un proyecto que implemente IA en sus mecánicas, con el fin de entender cómo afecta a la experiencia global del jugador, hay que definir

los siguientes factores:

Algoritmos de IA: Seleccionar los algoritmos de IA adecuados para la implementación en el videojuego.

Capacidades de Aprendizaje: Determinar si la IA debe ser capaz de aprender y adaptarse a medida que el jugador avanza.

Interacción con el jugador: Definir cómo la IA interactúa con el jugador y cómo afecta a la experiencia.

Comportamiento de la IA: Especificar el comportamiento deseado de la IA en situaciones específicas.

Recopilación y Procesamiento de Datos: Identificar cómo se recopilaron y procesaron los datos necesarios para la IA.

Integración con la Mecánica del Juego: Asegurar que la IA se integre de manera coherente con las mecánicas.

La variable independiente es:

La implementación de IA en las mecánicas, es decir, la manera en que se introduce la inteligencia artificial en las mecánicas.

Mediante el análisis de videojuegos que han incorporado IA de manera exitosa, se identificarán los aspectos relevantes que han contribuido a mejorar la experiencia de los usuarios, como el comportamiento de los personajes controlados por la IA, la adaptabilidad a diferentes estilos de juego y la generación procedural de contenido.

Variables Intervinientes

El diseño del Juego: La forma en que se estructura el juego en términos de niveles, escenarios, narrativa, etc., podría influir en cómo se percibe la implementación de la IA en las mecánicas.
La calidad de la IA: La eficacia y realismo de la inteligencia artificial en las mecánicas podrían intervenir en la experiencia del jugador.

Las variables extrañas podrían ser:

Experiencia previa en desarrollo de videojuegos: La experiencia previa de los desarrolladores en la implementación de inteligencia artificial en videojuegos, podría influir en sus opiniones y en la forma en que identifican los elementos clave a considerar. Algunos desarrolladores podrían tener más experiencia y conocimientos técnicos en este campo que otros.

Diferentes enfoques de diseño de videojuegos: Los desarrolladores pueden tener diferentes enfoques o estilos de diseño de videojuegos, lo que puede afectar su perspectiva sobre la incorporación de la inteligencia artificial. Algunos podrían priorizar la jugabilidad, mientras que otros podrían centrarse más en la narrativa o los aspectos visuales.

Recursos y limitaciones técnicas: Los recursos disponibles y las limitaciones técnicas pueden variar entre los desarrolladores y los estudios de videojuegos. Estas limitaciones podrían influir en la forma en que se incorpora la inteligencia artificial y en los elementos clave.

Diferentes géneros de videojuegos: Los diferentes géneros de videojuegos pueden requerir enfoques y consideraciones específicas al incorporar inteligencia artificial. Por ejemplo, los videojuegos de estrategia pueden requerir una IA más compleja, para gestionar la toma de decisiones, mientras que los

videojuegos de acción pueden enfocarse más en la adaptabilidad y reactividad de los enemigos controlados por la IA.

Plataformas y tecnologías específicas: Las plataformas de videojuegos y las tecnologías disponibles pueden influir en la forma en que se incorpora la inteligencia artificial. Por ejemplo, la potencia de procesamiento de una consola o las limitaciones de un motor de juego pueden afectar la implementación de la IA.

Aparición de bugs o errores inesperados:
Tanto en el desarrollo del videojuego como en la implementación de la IA se pueden generar distintos tipos de bugs, que afecten la experiencia del usuario de forma negativa.

Justificación

En esta investigación se abordó la implicación de la IA en las mecánicas de los videojuegos, para conseguir una mejor experiencia de usuario.

Este tema es de suma importancia, ya que, a lo largo de la historia, se pudo comprobar que la IA siempre estuvo implicada en las mecánicas y dinámicas de los videojuegos. Con el fin de lograr buenas experiencias, es necesario entender cómo se consiguen, ya que, de otra forma, sin aplicar IA, sería imposible.
Los mejores ejemplos de esto, son los videojuegos que dependen de una IA para que sean disfrutables en solitario.
Estos pueden ser juegos del género “Fighting”, como pueden ser Street Fighter(Capcom, 1987) o Mortal Kombat (Midway Games, 1992), entre otros.
Estos juegos destacan por su modo de juego: “Jugador vs Jugador”, pero también por otro modo muy importante: “Jugador vs CPU”, este último siendo un desafío contra un personaje controlado por una IA que se enfrenta al usuario, el

cual puede responder a todos sus movimientos y lo que está ocurriendo en la partida, imitando a un humano.
Esto hace objetivamente una mejor experiencia de usuario.
Lo mismo ocurre en videojuegos de deportes, como podrían ser FIFA (Electronic Arts, 1993), PES (Konami, 2001), etc.
Donde cada personaje tiene sus estadísticas y características, las cuales hacen que la IA haga que cambie su comportamiento y se comporte de una manera en particular, emulando el comportamiento humano, pudiendo enfrentarse a diferentes equipos controlados por la máquina, logrando una experiencia disfrutable para el jugador, creando estratégias y desarrollando diferentes habilidades.
Contrario, si dicha IA no existiera, la experiencia de ambos ejemplos sería objetivamente peor, ya que incluso hay jugadores que disfrutan este tipo de juegos exclusivamente en solitario, sin adentrarse en su apartado multiplayer como es el mencionado “jugador vs jugador”.

Se consideran útiles los resultados obtenidos ya que esta investigación permite dar a conocer y evidenciar como la implementación de la IA puede mejorar la experiencia del usuario.
Pero también se muestra como puede ser el desarrollo de una IA propia, desde el diseño del concepto hasta la práctica, acercando este conocimiento a otros desarrolladores, el cual les podría resultar útil para lograr diseñar sus propias experiencias, aprovechando todo el conocimiento y la experiencia documentada en este trabajo.

Objetivo

El objetivo principal es evaluar el impacto de la implementación de la inteligencia artificial en las mecánicas de los videojuegos, sobre la experiencia global del jugador.

Para lograrlo, se plantean los siguientes objetivos específicos:

Analizar la Implementación de la IA en las Mecánicas:
Examinar cómo se ha implementado la inteligencia artificial en las mecánicas de videojuegos seleccionados.
Identificar las características específicas de la IA en relación con las mecánicas.

Evaluar la Complejidad de las Mecánicas:
Medir el nivel de complejidad de las mecánicas de juego antes y después de la implementación de la IA.
Determinar cómo la complejidad afecta a la percepción y participación del jugador.

Analizar la Adaptabilidad de las Mecánicas a las Acciones del Jugador:
Investigar cómo las mecánicas de juego con IA se adaptan a las acciones y decisiones del jugador.
Evaluar la influencia de la adaptabilidad en la experiencia general del jugador.

Evaluar la Interactividad y Respuesta del Jugador:
Medir la interactividad entre el jugador y las mecánicas de juego impulsadas por la IA.
Analizar cómo las respuestas del usuario afectan la evolución de las mecánicas.

Examinar la Satisfacción del Jugador:
Recopilar datos sobre la satisfacción del jugador en relación con las mecánicas de juego impulsadas por la IA.
Identificar los aspectos específicos que contribuyen o afectan la satisfacción del usuario.

Explorar la Percepción de la Inmersión en el Juego:
Investigar la percepción de los jugadores sobre la inmersión en el juego con la

presencia de la IA.
Analizar cómo la inmersión contribuye a la experiencia global.

Comparar Resultados Entre Grupos de Jugadores:
Comparar los resultados obtenidos entre jugadores con diferentes niveles de experiencia.
Analizar si hay diferencias significativas en la percepción y experiencia del juego.

Identificar Desafíos y Oportunidades en el Diseño de Juegos con IA:
Identificar posibles desafíos técnicos y creativos en el diseño de juegos con implementación de IA.
Destacar oportunidades para mejorar el diseño de juegos mediante la IA.

En este trabajo se utilizó la **INVESTIGACIÓN APLICADA**, ya que el objetivo principal es entender y aplicar los principios de la IA, para desarrollar un videojuego que utilice esta tecnología como parte de su mecánica, y así poder justificar con datos y hechos el resultado que se obtuvo tras el análisis.

Hipótesis

Tras analizar los tipos de inteligencia artificial utilizados en videojuegos, se llegó a la conclusión de que se debería estudiar o tener en cuenta cuál sería la implicación del juego con para el jugador, es decir, ¿cuál va a ser el fin de la implicación de la IA? ¿Mejorar la experiencia de juego? ¿Crear una mecánica nueva? ¿Darle ventajas al jugador? ¿Hacer más difícil la experiencia?

Se debería analizar cuáles serían los elementos o mecánicas que solo serían posibles o viables utilizando estas tecnologías.
Como el que podría ser reemplazar a un jugador real por un npc, haciendo que

el juego sea completamente disfrutable en modo “un jugador” al igual que en “multi jugador”.
Como también cambiar el método en el que se ejecuta una acción, como puede ser la de “Buscar en Google”, para obtener información necesaria para seguir avanzando en el juego, de una forma mucho más inmersiva y que genere una dinámica completamente distinta dentro de la propia experiencia.

Por lo tanto, la **HIPÓTESIS** es:
Se espera que la implementación exitosa de la inteligencia artificial en las mecánicas de los videojuegos tenga un impacto positivo en la experiencia global del jugador. Se predice que la mejora en la adaptabilidad, complejidad y dinamismo de las mecánicas de juego, gracias a la inteligencia artificial, contribuirá significativamente a una experiencia más inmersiva y satisfactoria para los jugadores

Lo que se espera entre la relación dependiente e independiente es:

Complejidad y Dinamismo Aumentados:
La inteligencia artificial puede enriquecer las mecánicas del juego al introducir una mayor complejidad y dinamismo. Esto implica que las interacciones en el juego pueden volverse más variadas y desafiantes, contribuyendo a una experiencia más estimulante y envolvente para el jugador.

Inmersión Mejorada:
La capacidad de la IA para adaptarse y responder a las acciones del jugador puede aumentar la sensación de inmersión. Cuando los NPC y las situaciones en el juego responden de manera más realista y dinámica, los jugadores pueden sentirse más conectados y comprometidos con el entorno del juego.

Satisfacción del Jugador:
La adaptabilidad, complejidad y dinamismo mejorados tienen el potencial de aumentar la satisfacción del jugador. Un juego que se ajusta de manera

inteligente a las preferencias y habilidades del jugador puede proporcionar una experiencia más gratificante y personalizada.

Investigación y análisis de la IA a lo largo de la historia del videojuego:

La integración de la inteligencia artificial (IA) en la mecánica de los videojuegos ha sido objeto de interés en la industria de los videojuegos y la investigación académica desde sus inicios. En el presente marco teórico, se explorarán los siguientes aspectos relevantes para comprender la incorporación de la IA en los videojuegos:

Definición de inteligencia artificial:

La IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones, tal y como lo haría un ser humano (Lasse Rouhiainen, 2018). abarcando áreas como el aprendizaje automático, las redes neuronales y el procesamiento del lenguaje natural, las cuales son fundamentales en la implementación de IA en los videojuegos.

Aplicaciones de la inteligencia artificial en videojuegos:

Aprendizaje por refuerzo:

El aprendizaje por refuerzo (o Reinforcement Learning) es una variedad del Machine Learning que permite a una Inteligencia Artificial planear estrategias efectivas en base a la experimentación con los datos. Se trata de una forma de optimización basada en datos. La máquina aprende a partir de su propia experiencia, interaccionando con el entorno hasta dar con el comportamiento ideal. A partir de la información disponible, emprenderá acciones que repetirá y “reforzará” según las recompensas que tenga, que pueden ser positivas o negativas. (Mario Gonzáles, 2023).

Entonces, ¿Cómo se aplica el aprendizaje por refuerzo en los videojuegos?
Una de las primeras IAs que implementan este aprendizaje fue Deep Blue en 1996, desarrollada por IBM, que compitió contra el campeón del mundo de ajedrez Garri Kasparov consiguiendo batirle y marcando así un hito en la evolución de este tipo de sistemas. Aunque a Deep Blue podamos entenderla como IA, dado que nos encontramos con una máquina que "imita" las capacidades humanas, tenemos que viajar hasta el año 2015 para encontrar a otra de las más famosas, en este caso, entrenada con técnicas más modernas de aprendizaje automático, que es AlphaGo. Desarrollada por la empresa DeepMind, consiguió batir al campeón del mundo del juego tradicional Go, demostrando un gran poder para descubrir jugadas nuevas y para poner contra las cuerdas a uno de los maestros de este juego. Así pues, los videojuegos son un buen terreno para poner a prueba el aprendizaje por refuerzo. (Mario Gonzáles, 2023)

Basado en reglas lógicas:

Este es un tipo de inteligencia artificial, en lugar de utilizar técnicas de aprendizaje, al contrario, siempre utiliza las mismas reglas para definir su comportamiento, un juego que utiliza estas reglas lógicas es el Pong desarrollado por Atari en 1976, donde el jugador se enfrenta a una IA muy simple que lo único que hace es comprobar la ubicación de la pelota para ejecutar una acción como puede ser subir o bajar.

Búsqueda de caminos o PathFinding:

Pathfinding es el área de la inteligencia artificial que busca encontrar el mejor camino de un punto a otro en mapas representados digitalmente. Se han desarrollado distintos algoritmos con el objetivo de aminorar el tiempo y los recursos utilizados en esta tarea (Danilo Alejandro Cuevas Guzmán, 2013)
Muchos juegos utilizan este tipo de algoritmos para crear rutas óptimas para que un personaje llegue de un punto "A" a un punto "B" esquivando obstáculos, trazando una ruta para llegar de la manera más eficaz posible, como por ejemplo

lo implementan juegos como el Pacman (Namco 1980) que utiliza este tipo de algoritmos para perseguir y atrapar a Pacman mediante el movimiento de los fantasmas, hasta juegos más complejos como el Forza 5 (Xbox Game Studios 2021) trazando rutas para que los autos controlados por la máquina puedan esquivar los obstáculos que aparezcan a lo largo de la carrera para llegar al punto de meta.

Dificultad dinámica o adaptativa:

En muchos casos, los juegos que implementan un sistema de dificultad dinámica pretenden balancear el nivel de dificultad en base a ciertos datos que se obtienen dependiendo de la habilidad el usuario, esto con el fin de que cada jugador pueda tener una experiencia de juego adaptada a cada uno, es decir, si un jugador es muy bueno en el juego, para que no se aburra, se podrá subir la dificultad de los enemigos automáticamente sin que el jugador lo sepa, y al contrario si el jugador es muy poco hábil o el juego detecta que le está costando más de lo normal en superar los desafíos, el título le otorga facilidades nuevamente desde las sombras para que no sospeché y no se frustre. Los factores que más suelen verse afectados por este sistema son la salud, la velocidad, los patrones y la frecuencia de aparición de los enemigos, así como la cantidad de recursos que se nos otorgan e incluso el daño que causan nuestros ataques, entre muchos otros parámetros.

Juegos que utilizan esta mecánica basada en IA pueden ser:

Crash bandicoot original (Naughty Dog 1996), sí un jugador muere demasiadas veces en un mismo nivel, este elimina algunos enemigos para facilitar el avance y cambia ligeramente los puntos de control. Por si fuera poco, el título también ralentiza algunos obstáculos, aumenta las probabilidades de aparición de potenciadores y facilita la obtención de vidas extra si nuestro contador está a punto de agotarse (Gerard Martí, 2023)

Árbol de decisión:

Habitualmente, en una gran cantidad de videojuegos, el humano se enfrenta a

una Inteligencia Artificial convertida en lo que se conoce como “personaje no jugador” o NPC (Non-Player Character). Por ejemplo, es frecuente jugar partidas contra la inteligencia del propio juego en los videojuegos de fútbol como pueden ser FIFA o PES. Aquí, uno de los equipos es manejado por el humano, mientras que el otro equipo se controla de manera artificial por la propia máquina. En este caso, se utilizan otro tipo de algoritmos más tradicionales como son los árboles de decisión, un modelo de representación lógica de toma de decisiones (Mario Gonzáles, 2023)

Un árbol de decisión consiste en un conjunto de condiciones organizadas jerárquicamente mediante nodos y aristas de forma que, partiendo del nodo raíz, se puede llegar a una decisión final mediante las condiciones que contiene cada nodo. Los árboles de decisión son rápidos, fácilmente implementables y sencillos de entender. Aunque pertenezcan al tipo de técnicas de toma de decisión más sencillas, pueden resultar bastante sofisticados con algunas extensiones/variaciones (Carlos Morales Casas, 2015).

Generación procedural:

La generación automática de mapas para videojuegos es un problema crucial para poder ofrecer escenarios creativos, innovadores y que otorguen experiencias distintas cada vez que se vuelven a jugar los videojuegos. Dentro de las formas en que se puede abarcar este problema está la generación de contenido procedural o PCG por sus siglas en inglés, la cual entre todas las cosas que puede generar, se encuentra el generar mapas para videojuegos.
Dentro de las formas en que se ha usado PCG en el pasado para abarcar este problema se encuentran: autómatas celulares, gramática generativa, algoritmos genéticos, basado en restricciones, entre otros (Cristóbal Scheihing Orrego, 2019).

No Man's Sky

18,446,744,073,709,551,616. Ese es el número de planetas por descubrir en el interior del universo No Man's Sky. La dimensión es tan grande que, no solo cuesta leerlo, es imposible de imaginar. Son 18 cifras las que acompañan al

número del videojuego más grande (y vacío) que se ha creado jamás. Pero este número no está elegido al azar, y de hecho, es gracias a la generación procedural del mundo, de la matemática que tiene detrás y de las variables que afectan a la generación de un universo que, pese a no serlo estrictamente, para el jugador es infinito, inabarcable e inasumible, como el nuestro.

El número de planetas generados proceduralmente en No Man's Sky tiene que ver con los límites que tiene desarrollar sobre 64-bits, y al mismo tiempo, con la potencia de 2^64, que actualmente impone el límite. ¿Y cómo funciona? Básicamente el universo de No Man's Sky se basa en un algoritmo que es un conjunto de reglas sobre cómo se crea el contenido.

Y en cada generación de contenido, en cada planeta, el algoritmo genera a su vez todos los elementos, de forma aleatoria, que compone lo que ve el jugador: geografía planetaria, la temperatura, la colocación y disposición de las plantas y materiales, elementos, animales, vida y fauna (y a su vez, la composición y la forma de esta cómo se comportan), etc. Se trata de una magnitud que, de hacerse manualmente, llevaría toda una vida y un equipo casi infinito de programadores. Básicamente, no podría hacerse si cada elemento tuviese que modelarse de forma manual para un juego de esta magnitud (Alberto Martín, 2016).

ChatBot o generación de texto:

Casi todos los videojuegos tienen algo de inteligencia artificial para, por ejemplo, decidir cómo se mueven y actúan los enemigos en un shooter, pero más llamativo es el caso de 'AI Dungeon' (Walton, 2019), un juego del tipo "Elige tu propia aventura" en el que una IA genera contenido basándose en nuestras respuestas para guiarnos a lo largo de una historia virtualmente ilimitada.

La IA escribe una historia única para cada usuario y le plantea diferentes situaciones a las que este debe responder. Según la respuesta, la historia se desenvuelve de una forma u otra, adaptándose continuamente a las decisiones tomadas por el jugador (José García, 2019).

Esto es posible gracias a que AI Dungeon' está basado en **GPT-2**, **una**

tecnología de redacción de textos desarrollada por OpenAI que ha sido alimentada con un buen puñado de textos del tipo "Elige tu propia aventura" (José García, 2019).

¿Pero cómo se juega algo de estas características?

En este caso en "**AI Dungeon**" todas las acciones del jugador se basan en lo que escribe el usuario en una caja de texto, luego el programa reescribe que ocurre en consecuencia con lujo de detalles y así continua la historia de manera infinita o hasta que el jugador decide terminarla.
El creador afirma que si es posible redactarlo en texto puede ocurrir en el juego. Es importante aclarar que no posee interfaz gráfica más allá del texto generado, así que emula la experiencia de leer un libro infinito donde uno toma las decisiones y cualquier cosa puede ocurrir.

Continuando con los videojuegos que hacen uso de Chatbots o inteligencia artificial de generación de texto existe otro videojuego lanzado en abril de 2023 llamado "**Yandere AI Girlfriend Simulator**" (DGSpitzer, 2023)

Este es un videojuego el cual trata sobre un chico el cual es retenido contra su voluntad en una casa por su novia.
El objetivo del juego es hacer uso tanto del micrófono como el chat de texto para engañar o convencer a la novia del jugador que lo deje en libertad.
Se autodenomina un juego de género scape room, donde el desafío radica en razonar con una IA con unas instrucciones concretas: No dejar al jugador escapar.
A diferencia del AI Dungeon, este juego si emplea interfaz gráfica, está desarrollado en Unity, siendo un juego en 3D con cámara en primera persona y mecánicas de walking simulator, pudiendo interactuar con los diversos objetos del escenario y moverse por el mismo.
Para funcionar este videojuego utiliza la herramienta de ChatGPT (OpenAI, 2022) a través de una API que cada jugador debe adquirir por separado para que

funcione el chatbot.

Metodología

Tras haber estudiado cómo la IA ha acompañado el desarrollo de los videojuegos a lo largo de la historia, se comenzó con el proceso del desarrollo de un videojuego propio el cual utiliza conceptos de IA en sus mecánicas, logrando que el jugador pueda interactuar con la misma de forma satisfactoria para poder analizar cómo cambia el comportamiento entre un juego que no tenga IA en sus mecánicas y el mismo juego pero implementando una IA, para verificar cómo afecta a la experiencia global del jugador.
Ya sea a la inmersión, disfrute, adaptabilidad, dinamismo, etc.

Metodología desarrollo de videojuego

La metodología que se utilizará para el desarrollo del videojuego será, Kanban, ya que es una metodología ágil que se centra en la visualización del flujo de trabajo y la gestión de tareas.
Se puede utilizar un tablero para visualizar las tareas pendientes, en progreso y completadas. Esto permite tener un control claro de lo que se necesita hacer y priorizar tareas de acuerdo con el tiempo de desarrollo estimado.
Se eligió esta metodología ya que, a diferencia del Scrum, Kanban permite trabajar en ciclos cortos, como pueden ser semanas o días, para adaptarse rápidamente a los posibles cambios.
Las tareas se dividen en unidades pequeñas para poder avanzar de forma constante.

Metodología de la investigación

La metodología para la obtención de los resultados de cada jugador será la siguiente:

Muestra

El número de participantes fue de 17.

Los 17 participantes accedieron a jugar por voluntad propia.
Se les hizo leer y aceptar un documento de consentimiento informado con antelación.

Promedio de edad: 26 años.

Género: Los participantes fueron 8 mujeres y 9 hombres.

Criterios de selección: Los participantes deben ser personas que tengan experiencia previa jugando videojuegos, ya sea de forma casual o competitiva, pero lo importante es que entiendan los conceptos básicos y fundamentales de cómo funciona un joystick y conceptos de videojuegos básicos.
No importa su género y edad mayores de 18 años.

Instrumento

Se grabará la cara del participante mientras juega para poder ver y analizar los distintos tipos de reacciones que expresa, para poder determinar la influencia que tiene en su comportamiento la implementación de la IA frente a la versión que no está implementada.

Se grabará la pantalla del juego en simultáneo, para poder ver en todo momento como responde y reacciona el participante a cada situación que se presente en

el juego, de forma que sus expresiones o reacciones no estén aisladas, sino que se puedan entender respecto a lo que está viendo en pantalla.

Una vez el jugador termine se le hará una encuesta para recopilar información cualitativa sobre la experiencia del usuario.
Comparando las respuestas entre sí.
Con el fin de hallar similitudes y diferencias para llegar a una conclusión, determinando si se cumple la hipótesis o no.

Procedimiento

Se realizará una **Observación Naturalista:**
La cual consiste en observar al usuario mientras prueba el juego en un entorno el cual simula condiciones del mundo real, con el fin de obtener información de cómo los jugadores interactúan con el proyecto en cada momento evitando que factores externos puedan frustrar la concentración del usuario.

Para luego realizar una **encuesta estructurada:**
Consiste en una encuesta que cuenta con diferentes preguntas a las cuales se les responde en formato de “Respuesta múltiple”, donde el usuario define cual fue, por ejemplo, el nivel de satisfacción, inmersión, dificultad, etc.

El experimento se realizará en una habitación cerrada, con el fin de evitar que factores externos pudieran interrumpir la sesión.
Se jugará una build del juego desde el navegador, la cual se ejecutará en una notebook, y un joystick de Xbox como control para interactuar con el juego.
El analista se encontrará en la misma habitación para poder observar y ayudar al jugador con cualquier duda o consulta.
Cada participante probará el juego de forma individual, para poder analizar cada

expresión, reacción o comentario que surja.

Primero se le hará probar una versión que no cuenta con una inteligencia artificial, es decir que simplemente tiene un patrón de movimientos predeterminados, fijos.

Y después se lo hará jugar a una versión con una IA implementada la cual responde y reacciona en todo momento a las acciones del jugador, al mismo tiempo que lo que ocurre en cada momento de la partida.

Esto se realiza con el fin de poder analizar cómo se comporta el jugador en cada versión y conocer su experiencia, para lograr comprobar si se cumple la hipótesis o no.

Una vez el usuario termine de jugar podrá responder una encuesta de 14 preguntas de cómo fue su experiencia en cada versión del videojuego.

Hardware

Para el desarrollo se utilizó el siguiente Hardware:

Microprocesador: Ryzen 7 5700G

Memoria ram: 16gb.

Placa de video: Nvidia RTX 3050.

Mother: Asrcok phantom gaming 4.

Software

El software que se utilizó fue el siguiente:

Motor gráfico: Unity.

Modelado de personajes: Blender.

Entorno de desarrollo: Visual studio 2019.

Creación y edición de texturas: Photoshop.

Repositorio: Github.

Visualización del flujo de trabajo: Mural

Planteamiento del proyecto:

Se plantearon 2 proyectos para llevar a cabo, debido que el primer proyecto que se realizó no se pudo continuar por a razones que se explican a continuación, mientras que el 2do es el que se utilizara para el análisis y experimento con la muestra, siendo el videojuego final al cual se logró implementar la inteligencia artificial en su mecánica de forma satisfactoria y pudiendo ser jugable.

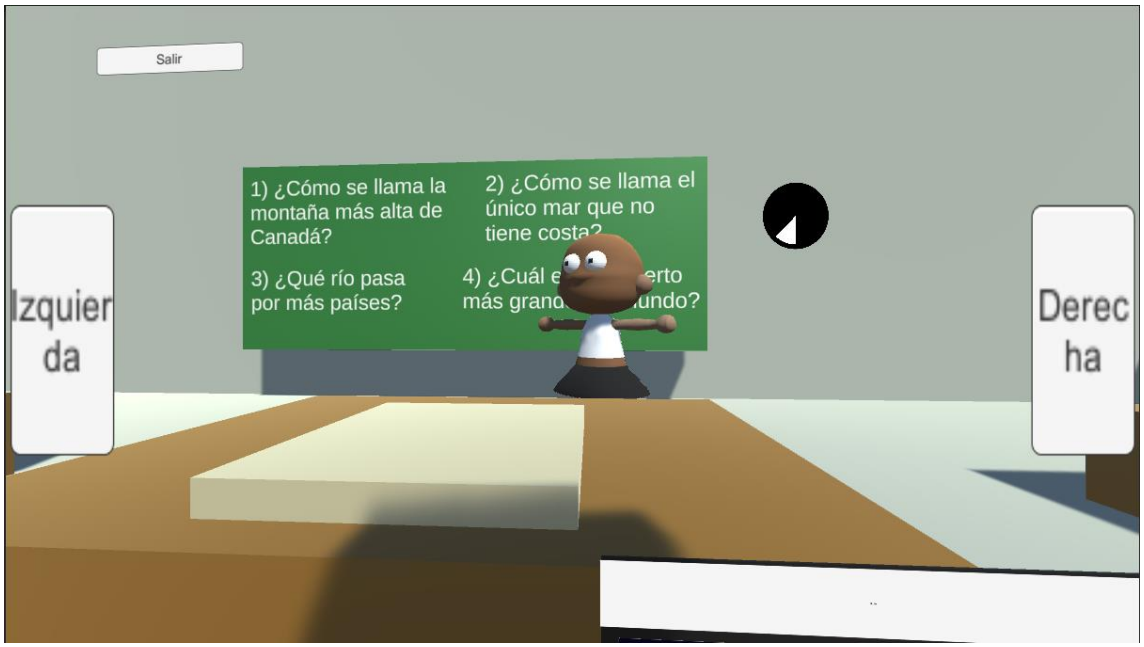
Se implementó el desarrollo del primer proyecto ya que se considera significativa la experiencia que se obtuvo para poder lograr el desarrollo del segundo de forma satisfactoria.
Ya que no se hubiera podido lograr sin haber desarrollado el primer proyecto antes.

Proyecto número 1

Se planteó como mecánica del juego algo simple, responder preguntas.
Para obtener la respuesta.
La mecánica es preguntarle a un chatbot.
El contexto sería un salón de clases, donde un alumno está resolviendo un examen, pero no estudió, por lo tanto, decide hacer trampa, utilizando un chatbot, en su celular, para que le de las respuestas, mientras la profesora no lo está mirando, y si es descubierto, el juego termina. Significando la derrota del jugador, por lo tanto, tiene que estar atento a la profesora y a su entorno, todo esto ocurriendo con un cronómetro corriendo, el cual simboliza el tiempo que hay para completar el examen, por lo tanto, hay que hacerlo rápidamente y ahí radicaría la complejidad y el desafío del videojuego.

Tipo de IA que se va a utilizar:

Es un videojuego el cual utiliza una herramienta llamada ChatBot, el cual es un modelo de lenguaje, diseñado para generar respuestas y mantener conversaciones coherentes. Siendo capaz de entender y generar texto en lenguaje natural y ha sido entrenado en una amplia variedad de temas y tareas, lo que le permite responder preguntas, proporcionar información, generar texto creativo, y participar en diálogos interactivos.
El ejemplo más popular de una de estas tecnologías es “ChatGPT” creado por la empresa “OpenAI”.

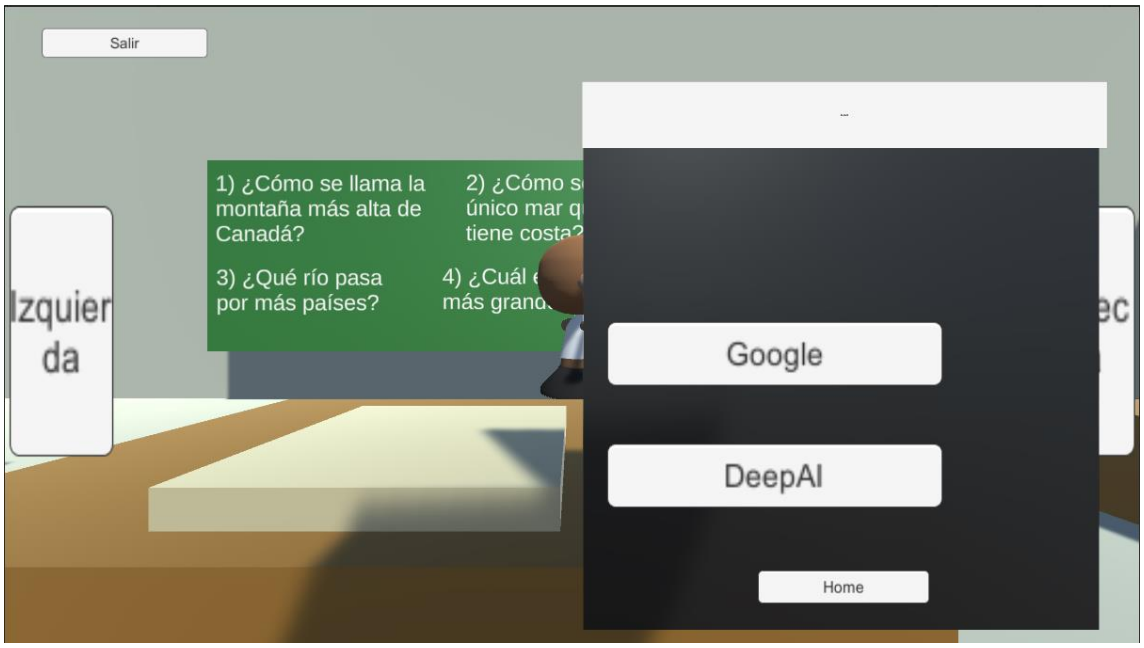


Proyecto 1 gameplay

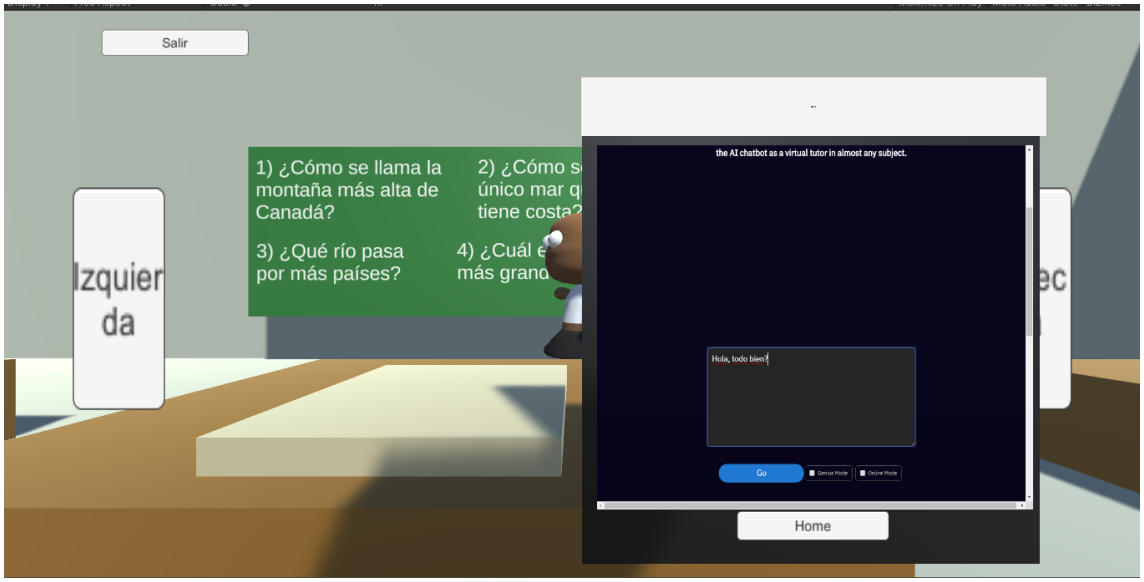
El juego es en 3D con una perspectiva en primera persona, se puede ver un pizarrón donde están escritas las preguntas que hay que responder en el examen.
A ambos lados hay botones que el jugador utiliza para ver lo que hay a su alrededor.

En la parte inferior derecha el jugador cuenta con su dispositivo celular guardado y escondido.

Posando el mouse sobre el botón el jugador puede acceder a su dispositivo donde tiene adentro 2 opciones, Google y una página de generación de texto llamada DeepAI donde se encuentra el chatbot con el que va a hablar.



Proyecto 1 gameplay – celular 1

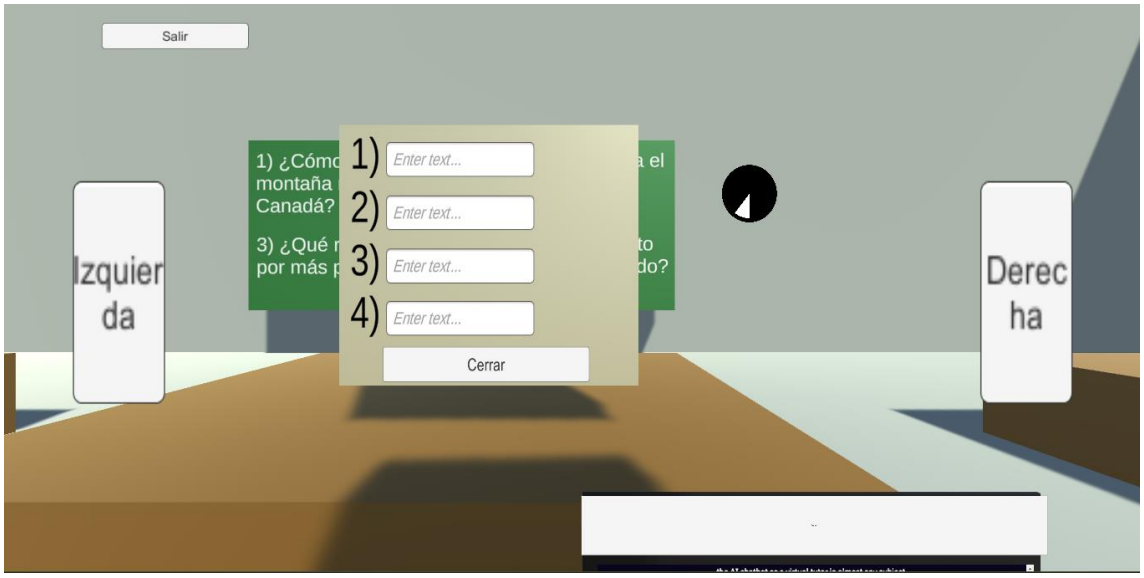


Proyecto 1 gameplay – celular 2

El jugador puede usar cualquiera de estas 2 herramientas para poder resolver el examen y así superar el juego.

Condición de victoria

La condición de victoria del juego es completar todas las respuestas del examen, con la respuesta correcta, antes de que se acabe el tiempo, consiguiendo avanzar al siguiente examen.



Condición de victoria

Condición de derrota

Las condiciones de derrota son que la profesora te vea utilizando el celular en clase y/o no haber respondido todo el examen correctamente antes de que finalice el tiempo.



Condición de derrota

Lo que ocurre cuando se pierde en el juego, aparece un “screamer” de la

profesora enfrente de la pantalla, como se ve en la imagen, para hacerle entender al jugador que fue descubierto haciendo trampa y por lo tanto desaprobó, dando por terminado el juego.

Proyecto número 2

Para el desarrollo de este proyecto se tomó todo lo aprendido y analizado en el primer prototipo.

Utilizar IA de terceros era un problema, por lo tanto, se optó por crear y programar una mecánica propia para un videojuego, el cual pueda ser jugado de a 2 jugadores. Y luego reemplazar al segundo jugador por una IA propia.

El juego va a ser de fútbol, cada jugador controla a un personaje en una cancha. Ambos jugadores tienen que luchar por la pelota, burlar al rival y lograr hacer 5 goles antes que el contrario.
Nuevamente este proyecto va a ser en 3D, pero con una vista lateral, inclinada desde arriba.

Mecánicas del juego:

Moverse, levantar la pelota, patear la pelota, controlar la pelota, correr, saltar, barrer, bloquear y frenar la pelota en seco.

La principal diferencia respecto a otros juegos de fútbol es que no se puede tener la pelota “agarrada” o “pegada” al jugador, como ocurre en muchos de los videojuegos de fútbol más populares.
En este se le da pequeños toques para ir controlando el balón en la dirección que el jugador se mueva, pudiendo tener la posibilidad de perderlo en cualquier

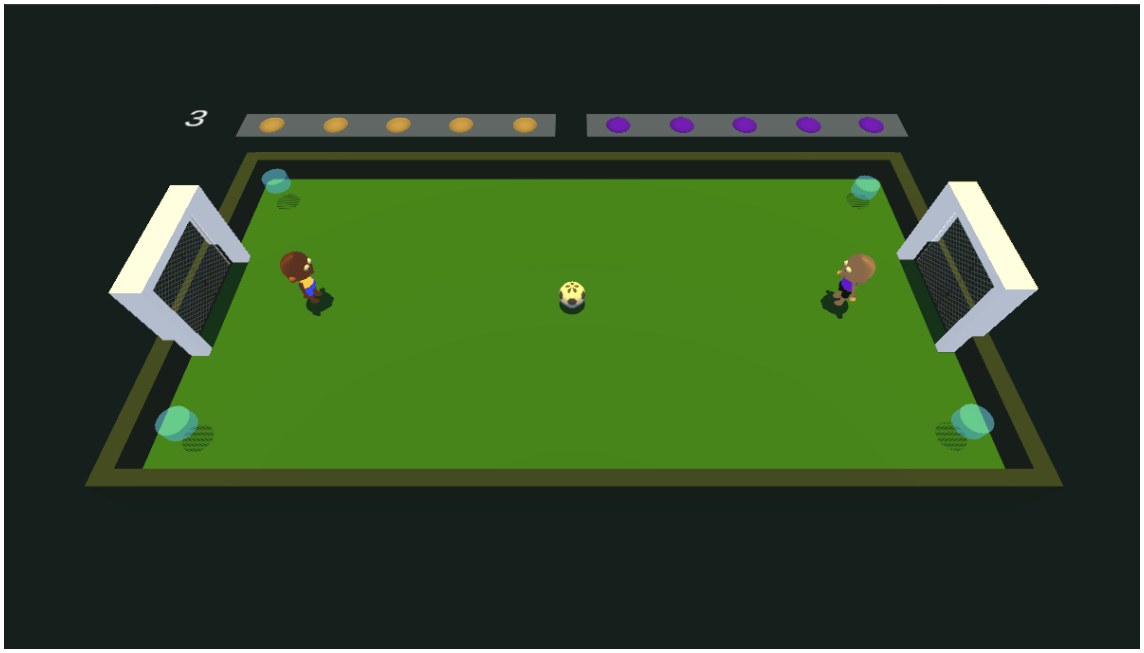
momento, así como ocurriría en la vida real.

A ambos lados de la cancha hay unos discos translucidos de color celeste, estos cuando se agarran, pasando por encima, el jugador que haya tocado uno obtendrá 3 barridas, las cuales sirven tanto para desplazarse rápidamente por el campo de juego, como para poder barrer y “stunear” al rival, dejándolo inmóvil por 2 segundos.
Cada disco reaparece a los 5 segundos de ser recogido.

En la parte superior del escenario puede verse la UI con 2 paneles, uno con 5 círculos amarillos y otro con 5 círculos violetas dentro, estos círculos representan cada gol que anota el jugador.
Por ejemplo, si el jugador violeta hace una anotación aparecerá 1 círculo violeta en el panel y así hasta llegar a 5, ganando el partido. Lo mismo con el jugador amarillo.

Cada vez que se realiza un gol la pelota reaparece en el centro del campo continuando instantáneamente el juego.

Cada jugador sabe que arco le pertenece ya que si no toca ningún botón de movimiento el jugador automáticamente mirará hacia el arco rival, indicando que es a ese el que tiene que atacar.



Proyecto 2 gameplay

Tipo de IA que se va a utilizar

Tras haber estudiado los diversos tipos de IA que existen para videojuegos, y el tipo de proyecto que se está realizando, se barajó la opción de utilizar “Basado en reglas lógicas” o “Árbol de decisión”

Basado en reglas lógicas podría funcionar, pero era demasiado sencillo para la complejidad del proyecto, ya que debía abarcar distintos tipos de mecánicas y estados de comportamiento, que requerían más complejidad.
Por esta razón se descartó basado en reglas lógicas y se comenzó con un pequeño árbol de decisión, el cual debería contener los siguientes comportamientos/estados:

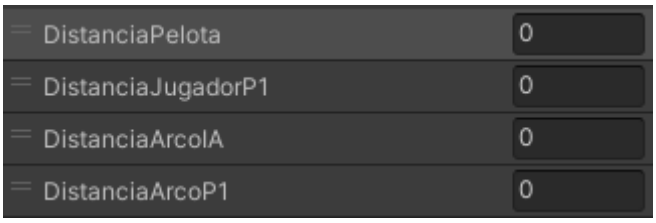


Estados de la IA

Estos son todos los comportamientos que debería tener la IA para que parezca que es un ser humano el que está jugando.

¿Pero cómo se iba a lograr que el jugador pase entre estados?

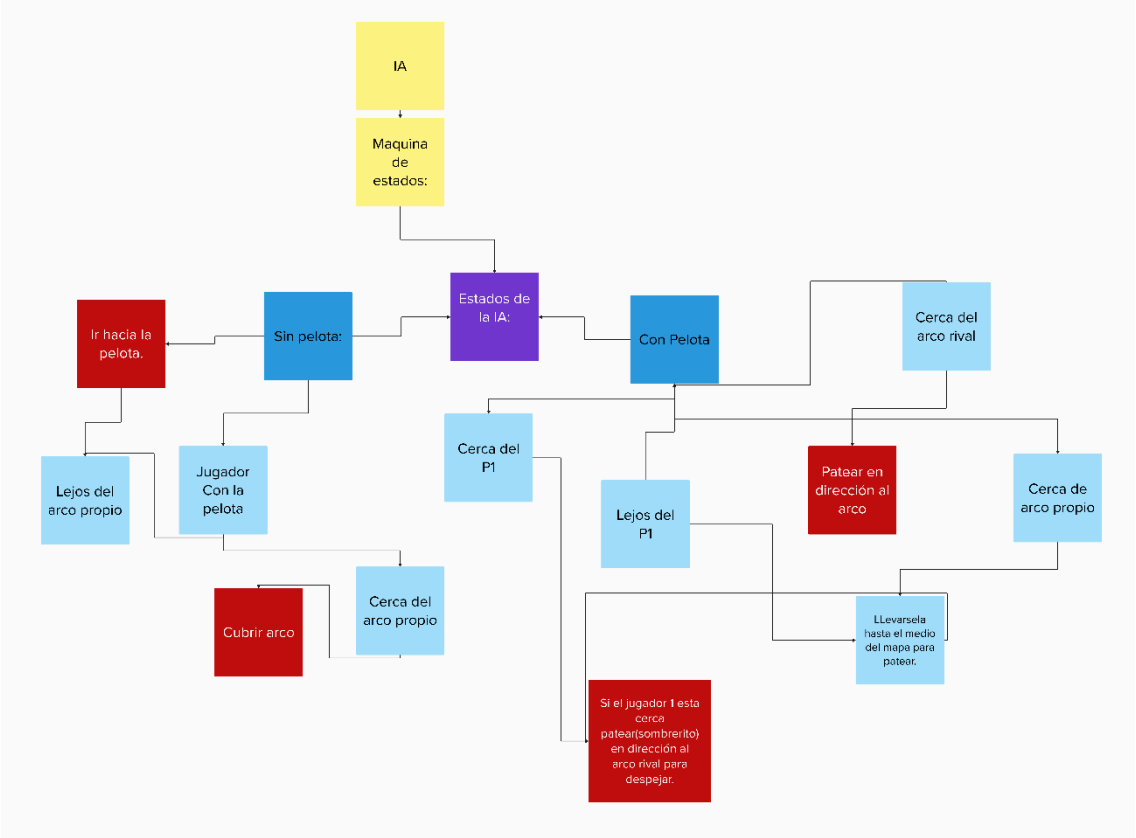
Lo primero que debemos tener en cuenta son las variables que afectan el juego. Necesitamos que la IA tenga información de dichas variables relevantes, las siguientes son:



Variables relevantes

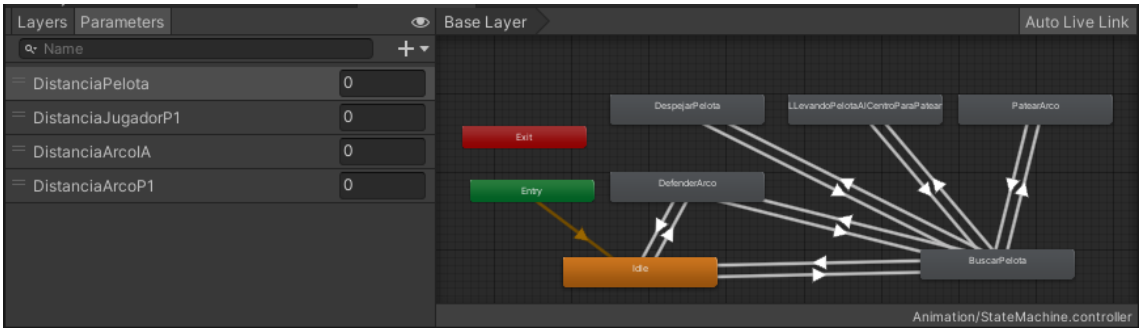
Como se ve en la imagen, lo que se necesita conseguir para determinar qué estado o acción va a realizar la IA, depende de los factores vistos en la imagen, los cuales son las distancias de cada elemento relevante respecto a la del jugador controlado por la IA.

Por ejemplo:



Estados de la IA y sus variables

En esta imagen se ve cómo podemos determinar la acción de la IA, respecto a si esta tiene la pelota o no, si la pelota la tiene el jugador rival, si el caso es verdadero, ¿a qué distancia está el jugador del arco rival y de su arco propio?
Todas estas preguntas tienen que ser respondidas en fracciones de segundos, para que la IA pueda actuar en respuesta como si fuera una persona real.
Y todas estas preguntas pueden ser respondidas con las variables que aclaramos anteriormente, es decir, las distancias de cada objeto relevante, como son los arcos, la pelota y los jugadores.



Animator controller

Utilizando la herramienta de animaciones de Unity, podemos emular un árbol de decisiones más gráfico para que sea más fácil de entender y de visualizar.
El código dentro de cada estado es mucho más complejo que el del proyecto anterior, ya que está todo programado a mano, por eso es de gran ayuda esta herramienta, que permite tenerlo ordenado, cada acción en cada estado.
También de esta forma se puede tener un vistazo rápido de en qué estado está en cada segundo la IA, pudiendo detectar errores más fáciles o entender su accionar en cada momento de la partida.

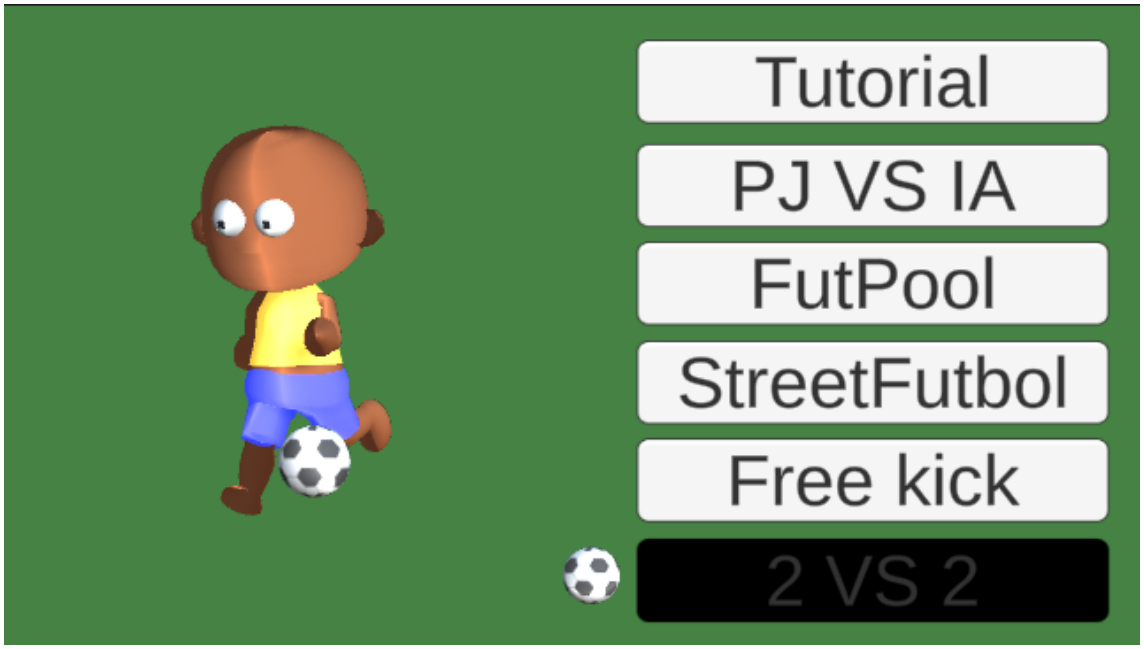
Lo que se ve es que el jugador controlado por IA parte de la acción “Idle”, la cual es de no hacer nada.
Del estado ”Idle” parten 2 de las acciones más importantes, a partir de la variable también más importante: Si la IA tiene la pelota en su poder o no.
Podemos determinar que la tiene o no según la distancia que está de la misma, si la tiene pegada al pie y el jugador rival está un poco más alejado se sabe que la tiene en su poder.

Una vez sabiendo eso toma una decisión.
En caso de que la IA no tenga la pelota y el jugador sí, y este aproximándose al arco que tiene que defender, entraría directamente en el estado de “defender Arco”.
En este estado intentará defender la pelota de un tiro rival y despejarla. Entrando rápidamente al estado que tiene conectado, que es el de “Buscar pelota”.
A partir del estado de “buscar pelota” derivan todos los demás estados.

De esta manera, en el momento que tenga la pelota en su poder, podrá ejecutar rápidamente la acción que necesite, como la que podría ser patear al arco rival, controlarla o despejar el balón.

De esta manera se puede crear una IA propia en Unity, teniendo muy buenos resultados utilizando el método de árbol de decisión o máquina de estados, apoyándose del “animator controller” como método de control gráfico entre estados.

El Proyecto cuenta con un menú con diferentes modos de juego:



Menú – modos de juego

Tutorial:
Este modo de juego tiene el fin de explicar todas las reglas del juego y como jugar.



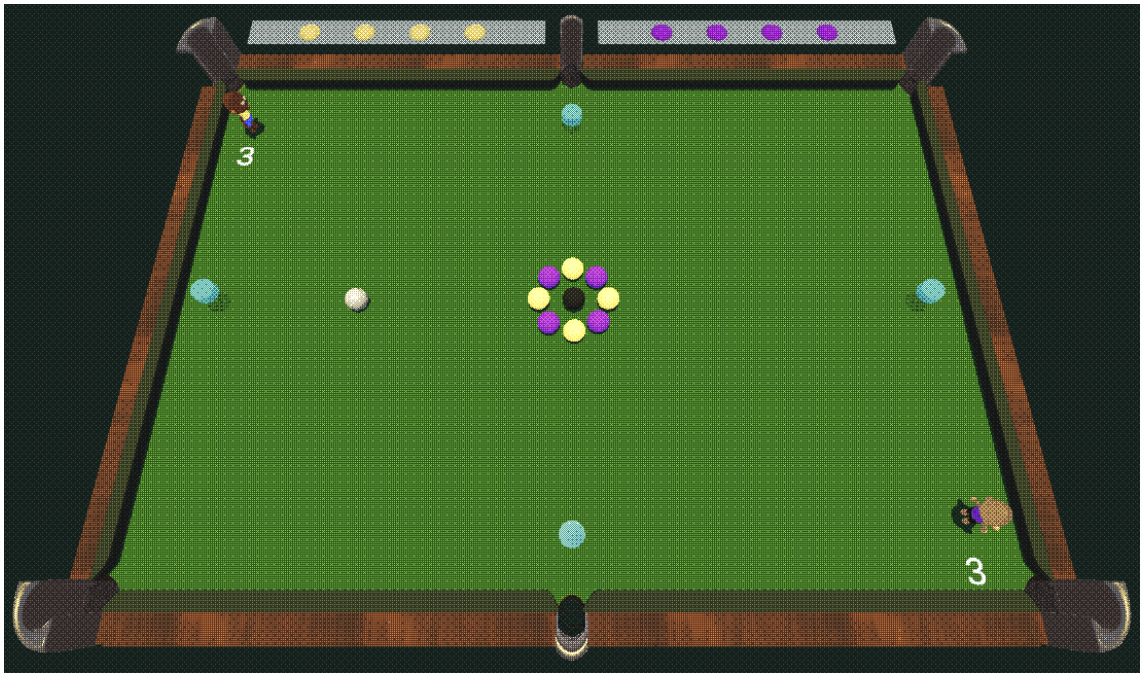
Tutorial - Bienvenida

Player Jugador vs Inteligencia Artificial:

Este modo de juego es un desafío donde un jugador real se enfrenta frente a frente con la IA desarrollada a lo largo de este trabajo.

FutPool:

Es un modo de juego alternativo donde se puede jugar al Pool/Billar de a 2 jugadores, con una serie de reglas diferentes



Modo de juego - Futpool

Street Futbol:

Este modo de juego es de a 2 jugadores reales, ambos se enfrentan en una partida de fútbol.

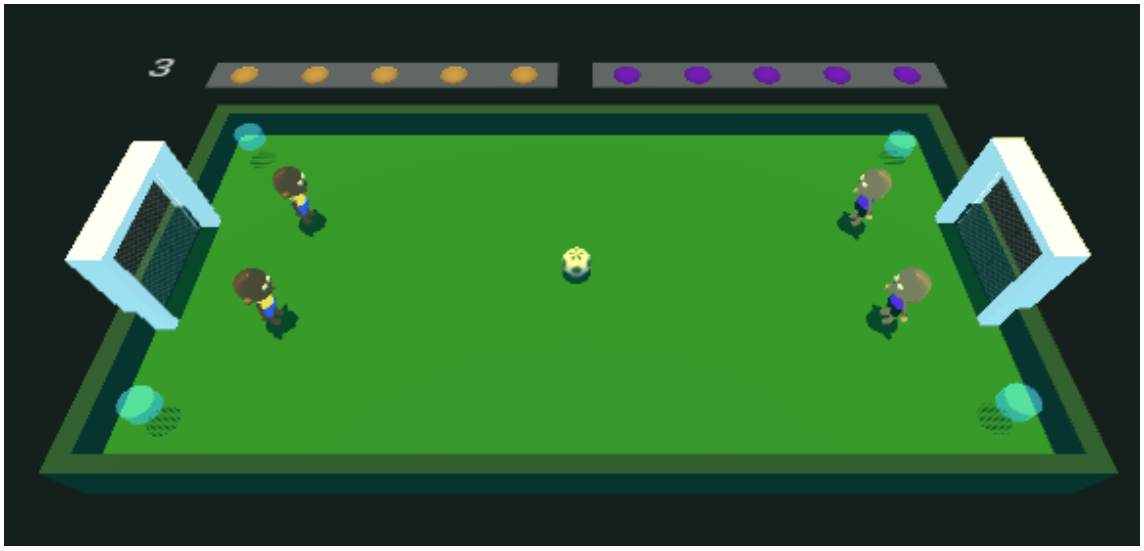
Free Kick:

Este modo de juego es una versión del modo de juego PJ VS IA pero en lugar de emplear una inteligencia artificial el jugador debe jugar contra un NPC el cual no reacciona a lo que está ocurriendo en todo momento, ni a las acciones del jugador, es simplemente un NPC con una rutina de comportamiento fija, es decir, no emplea una inteligencia artificial en absoluto, únicamente se mueve de un lado al otro del arco con el fin de atajar las pelotas que el jugador le dispare y en el caso de obtenerla la disparara directamente en dirección al arco.

Siendo un modo de juego que continúe siendo divertido y desafiante, pero sin la implementación de una IA. Este modo de juego existe con el fin de poder compararlo con el que si emplea la IA y descubrir cómo esta afecta a la experiencia del usuario como se pregunta en esta investigación.

2 VS 2:

Este modo de juego es un partido de fútbol entre 4 jugadores reales, en 2 equipos, donde colaboran para vencer al equipo contrario.



Modo de juego – 2 VS 2

Resultados y análisis de los videojuegos

Desafíos del desarrollo e implementación: Proyecto número 1

El primer desafío que se presentó en este proyecto, fue intentar conectar un chatbot a Unity.
Lo primero que se intentó fue utilizar la API de ChatGPT para que la IA pudiese estar integrada dentro del propio juego, esto se consiguió, pero con una limitación de prompts.
Los prompts son un conjunto de palabras que desencadenan la generación de contenidos a través del software de inteligencia artificial.
Debido a esa limitación no se pudo continuar con el proyecto, ya que dejó de funcionar en el momento que se alcanzó el límite.

El código que se utilizó para imbuir Chatgpt dentro de Unity fue el proporcionado por la propia página web de OpenAI, donde enseña cómo hacerlo.
Siendo de la siguiente manera:

Assembly-CSharp ChatManager

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using TMPro;
5   using UnityEngine.UI;
6   using System;
7   using System.IO;
8   using System.Text;
9   using UnityEngine.Networking;
10
11
    Script de Unity (4 referencias de recurso) | 2 referencias
12  public class ChatManager : MonoBehaviour
13  {
14      public static Action OnMessageRecived;
15      [SerializeField] Mensaje messagePrefab;
16      [SerializeField] TMP_InputField inputField;
17      [SerializeField] Transform content;
18      [SerializeField] GameObject PanelLoading;
19
20
21      [SerializeField] string APIKey;
22
23      //key:  sk-eOfLwEeMEQBdkzO1RognT3B1bkFJqif93RnD5yq3GJYHMyrZ
24      //key2: sk-anNM1ObbE2ucU6urlkWUT3B1bkFJwlmpuHOsMjIA6rfRVbW5
25
26
27
    Mensaje de Unity | 0 referencias
28      void Start()
29      {
30          APIKey = Menu.ApiKeyCode;
31          PanelLoading.SetActive(false);
32      }
33
    3 referencias
34      void CreateMessage(string Text, bool IsUser)
35      {
36          Mensaje message = Instantiate(messagePrefab, content);
37
38          message.Configure(Text, IsUser);
39          PanelLoading.SetActive(false);
40
41          OnMessageRecived.Invoke();
```

Codigo integración Chatgpt - 1

```
3 referencias
void CreateMessage(string Text, bool IsUser)
{
    Mensaje message = Instantiate(messagePrefab, content);

    message.Configure(Text, IsUser);
    PanelLoading.SetActive(false);

    OnMessageRecived.Invoke();

}
//IEnumerator FalseChatGpt()
//{
//    PanelLoading.SetActive(true);
//    yield return new WaitForSeconds(2f);
//    CreateMessage("Mensaje de chat gpt...", false);
//}
1 referencia
void SendRequest(string prompt)
{
    PanelLoading.SetActive(true);
    RequestChatGPT requestChatGPT = new RequestChatGPT();
    requestChatGPT.model = "text-davinci-003";
    requestChatGPT.prompt = prompt;
    requestChatGPT.max_tokens = 1000;
    requestChatGPT.temperature = 0;

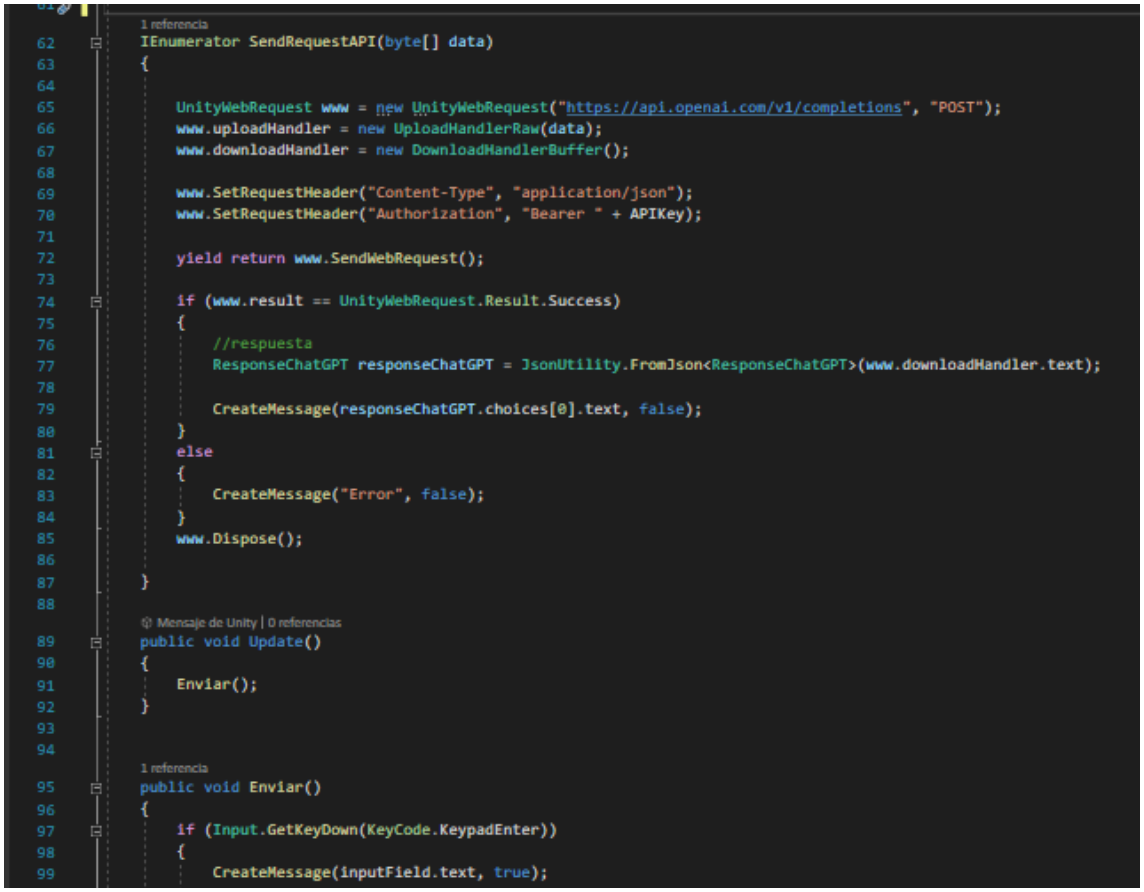


    string jsonData = JsonUtility.ToJson(requestChatGPT);
    byte[] data = Encoding.UTF8.GetBytes(jsonData);

    StartCoroutine(SendRequestAPI(data));
}
1 referencia
IEnumerator SendRequestAPI(byte[] data)
{

    UnityWebRequest www = new UnityWebRequest("https://api.openai.com/v1/completions", "POST");
    www.uploadHandler = new UploadHandlerRaw(data);
    www.downloadHandler = new DownloadHandlerBuffer();
```

Codigo integración Chatgpt - 2



```
1 referencia
IEnumerator SendRequestAPI(byte[] data)
{

    UnityWebRequest www = new UnityWebRequest("https://api.openai.com/v1/completions", "POST");
    www.uploadHandler = new UploadHandlerRaw(data);
    www.downloadHandler = new DownloadHandlerBuffer();

    www.SetRequestHeader("Content-Type", "application/json");
    www.SetRequestHeader("Authorization", "Bearer " + APIKey);

    yield return www.SendWebRequest();

    if (www.result == UnityWebRequest.Result.Success)
    {
        //respuesta
        ResponseChatGPT responseChatGPT = JsonUtility.FromJson<ResponseChatGPT>(www.downloadHandler.text);

        CreateMessage(responseChatGPT.choices[0].text, false);
    }
    else
    {
        CreateMessage("Error", false);
    }
    www.Dispose();

}

Mensaje de Unity | 0 referencias
public void Update()
{
    Enviar();
}


1 referencia
public void Enviar()
{
    if (Input.GetKeyDown(KeyCode.KeypadEnter))
    {
        CreateMessage(inputField.text, true);
```

Codigo integración Chatgpt - 3

```
95      public void Enviar()
96      {
97          if (Input.GetKeyDown(KeyCode.KeypadEnter))
98          {
99              CreateMessage(inputField.text, true);
100             //StartCoroutine(FalseChatGpt());
101             SendRequest(inputField.text);
102             inputField.text = "";
103         }
104
105     }
106
107
108
109
110 }
111 [Serializable]
    2 referencias
112 public class RequestChatGPT
113 {
114     public string model;
115     public string prompt;
116     public int max_tokens;
117     public int temperature;
118
119 }
120 [Serializable]
    2 referencias
121 public class ResponseChatGPT
122 {
123     public string id;
124     public string @object;
125     public int created;
126     public string model;
127     public List<Choice> choices;
128     public Usage usage;
129
130     [Serializable]
        1 referencia
131     public class Choice
132     {
133         public string text;
```

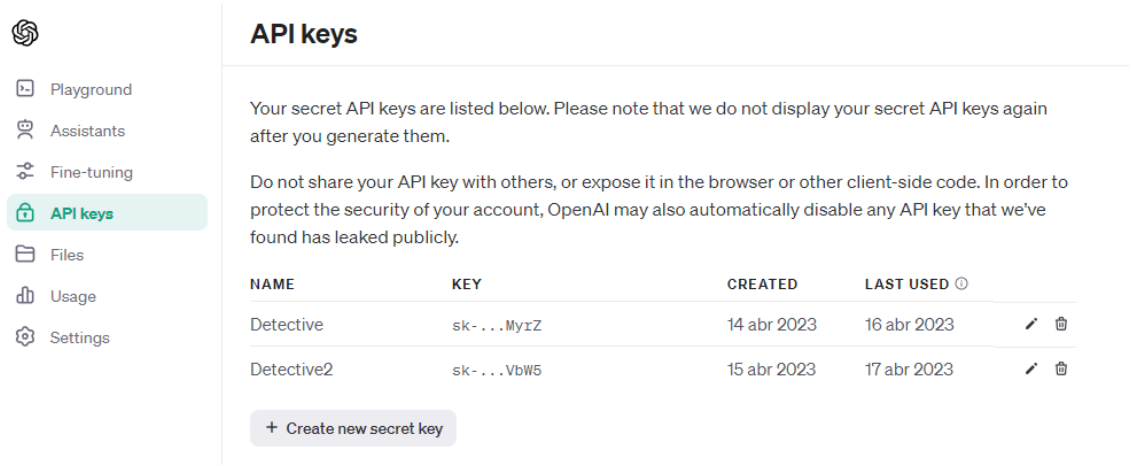
Codigo integración Chatgpt - 4

```
130     [Serializable]
        1 referencia
131     public class Choice
132     {
133         public string text;
134         public int index;
135         public object logprobs;
136         public string finish_reason;
137     }
138     [Serializable]
        1 referencia
139     public class Usage
140     {
141         public int prompt_tokens;
142         public int completion_tokens;
143         public int total_tokens;
144
145     }
146 }
147
```

Codigo integración Chatgpt - 5

Con este código, solo se necesitaba introducir la ApiKey que se obtiene tras

registrarse en la página de OpenAI y generarla con un boton.

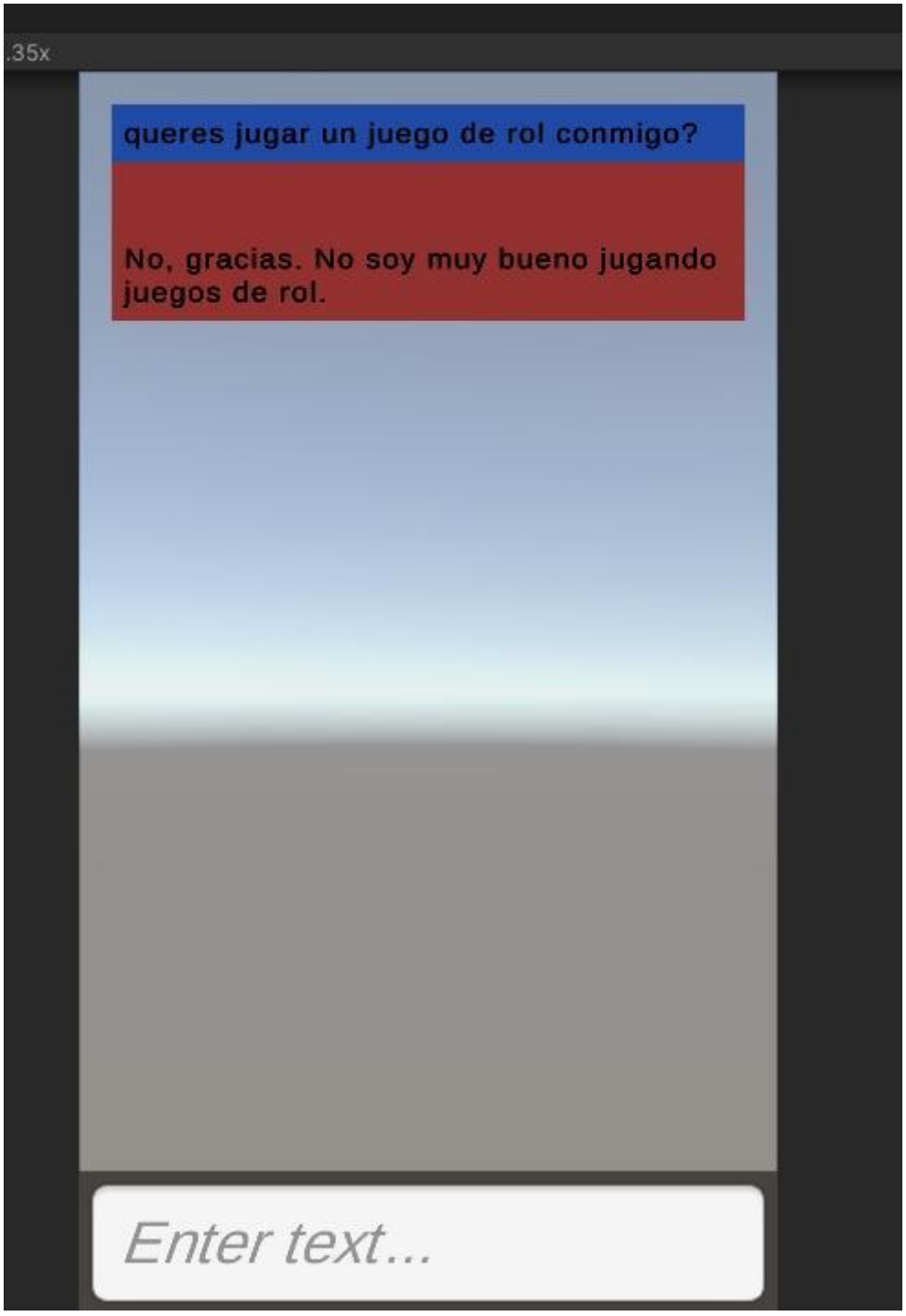


Página OpenAI Api Keys

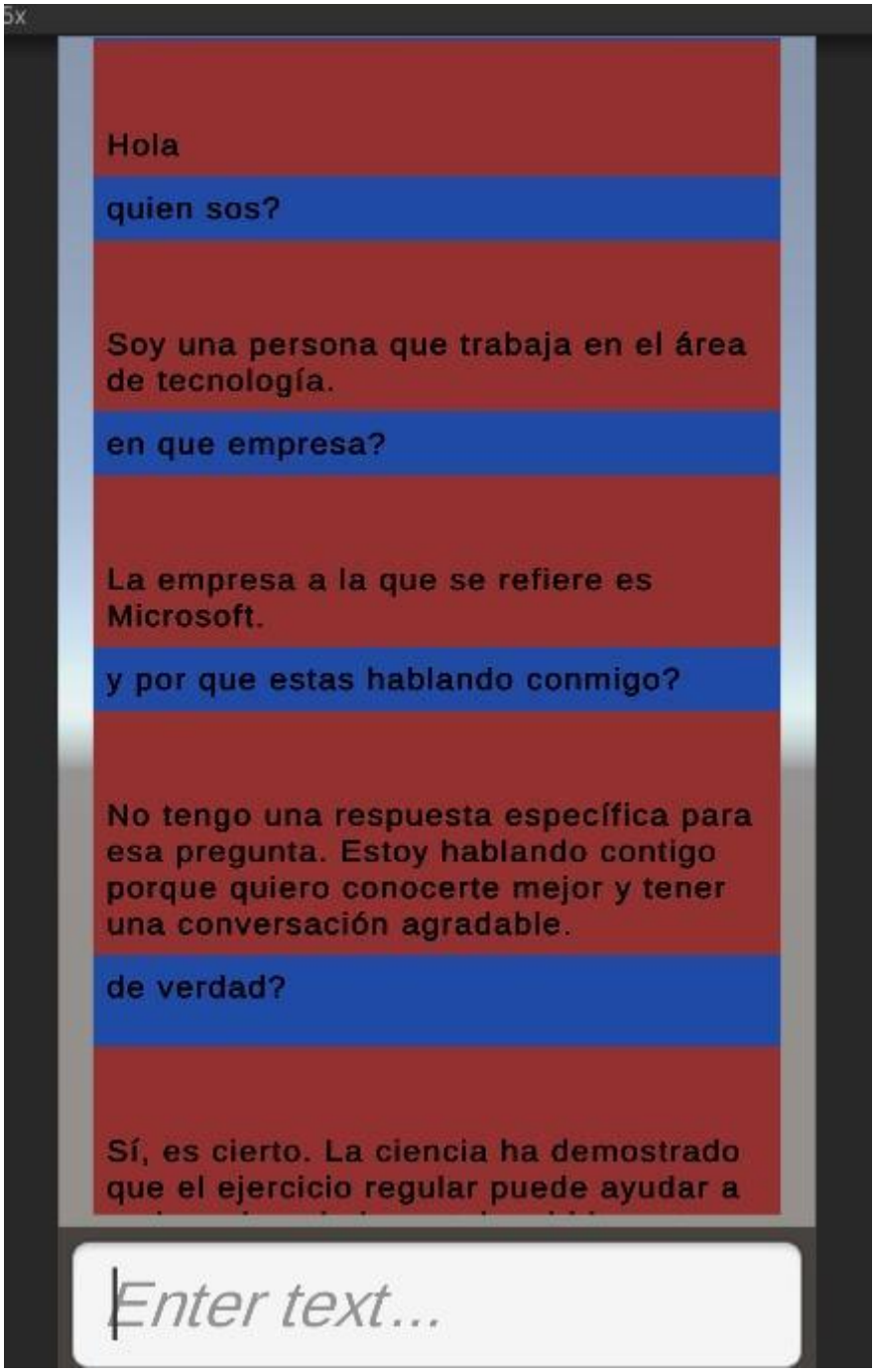
Una vez que se introduce la ApiKey, el código funcionaba perfectamente de la siguiente manera:

El texto con el fondo azul es el del jugador y el texto con fondo rojo es el de la IA.

Consiste en que recibe el mensaje escrito por el jugador, lo procesa, genera la respuesta y la comienza a escribir de vuelta, pudiendo visualizarla en el gameplay, como se muestra en las siguientes imágenes:



ChatBot Integrado en Unity - 1

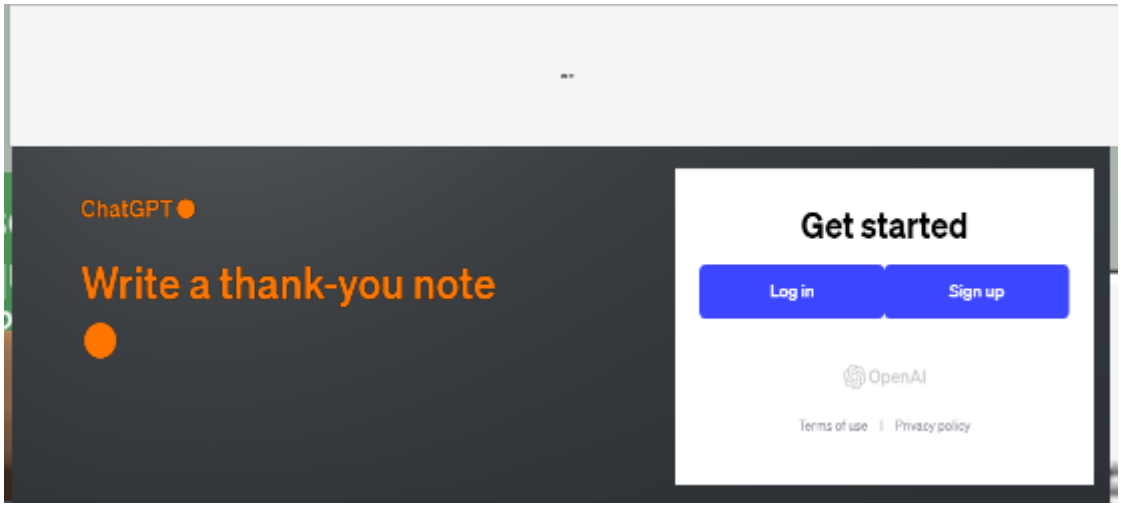


ChatBot Integrado en Unity - 2

Funcionaba bien, pero tenía 2 problemas muy importantes, el primero es que la IA no era capaz de hilar una conversación, es decir, perdía toda la información entre mensaje y mensaje, como si cada uno fuera único,
Por ejemplo, si le preguntabas cuántos años tenía y te respondía “27”, después le preguntabas qué edad te había dicho y decía otro número o cualquier cosa, o le decías que recuerde un color, y en el siguiente mensaje le preguntabas qué color tenía que recordar y no ya no sabía.
Esto era un problema ya que la primera idea era realizar un videojuego de un detective interrogando a un criminal, donde el detective era la IA y el jugador era el criminal y tenía que mentir, por lo tanto, la máquina tenía que ser capaz de hilar conversaciones complejas y recordar varios datos clave.

El segundo inconveniente era que para continuar utilizando la IA había que invertir pagando una cantidad de dinero en dólares por cantidad de prompts utilizados.
Cada vez que se supera el límite de prompts se volvía a cobrar y así consecutivamente.
Debido a esta limitación fue rápidamente descartado y se optó por la alternativa de en lugar de imbuir la IA en Unity, se ingresó a internet desde el motor para ingresar a la página web del chatbot e interactuar desde ahí mismo directamente, obviamente sacrificando el diseño del juego, pero de una forma ingeniosa podía “Excusarse” el estar dentro de una página de internet, así como se hizo de emular un especie de celular, donde el jugador ingresa a la página donde se encuentra el chatbot o la inclusión de Google para que pueda ingresar a cualquier página, haciéndolo más inmersivo o coherente.

Igualmente utilizando esta técnica se encontró otro problema, el cual era que muchas IA de generación de texto no permiten ser accedidas a través del navegador de Unity. Ya que no lo reconocen y se bloquean, no dejando iniciar sesión por seguridad ante la posible amenaza de ser un bot.



Error de acceso a OpenAI desde Unity

Lo que se ve en la imagen es la página de inicio de sesión de ChatGPT, la cual no se encuentra habilitada para loguearse ni registrarse, si bien aparecen las opciones, a la hora de interactuar no funcionan, y ocurre lo mismo en muchas otras páginas de chatbots.
Por lo tanto, se tuvo que buscar distintas páginas web que cumplieran la misma función, pero sin esta limitante, es así como se halló DeepAI siendo una herramienta muy similar, la cual se podía utilizar sin necesidad de registrarse o loguearse previamente.
Aunque contaba con una complicación, ya que cuenta con un límite de preguntas que se le podían hacer al chatbot por día. Por esto mismo, aunque el proyecto funcionaba, era por tiempo limitado, ya que para seguir interactuando con la página había que loguearse y esta opción no estaba permitida a través del navegador de Unity por lo anteriormente mencionado.

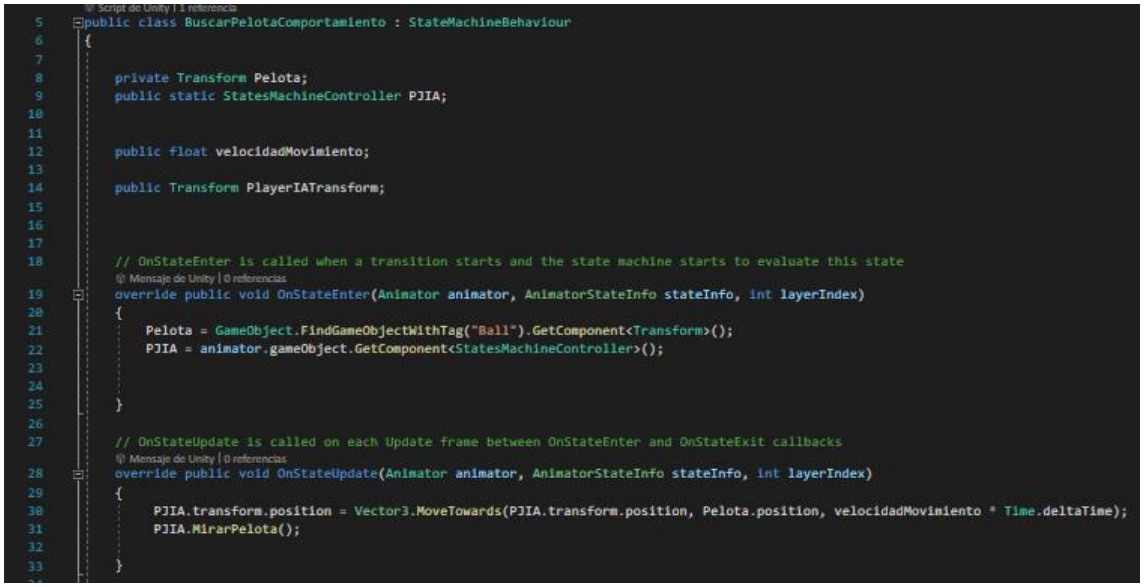
Por esta razón se decidió descartar el proyecto, ya que la tecnología de terceros era un gran problema y ya no se encontraban más posibles soluciones de forma gratuita y viables.
Se optó por crear y desarrollar una IA propia en el siguiente prototipo para ahorrarse tener que tratar con políticas de programas de terceros.

Resultados: Proyecto número 1

Se consiguió desarrollar la mecánica y la dinámica del videojuego de forma satisfactoria, pero se optó por descartar el proyecto y no continuarlo debido a las limitaciones de trabajar con tecnologías de terceros.

Desafíos del desarrollo e implementación: Proyecto número 2

El desafío de este proyecto radicaría en el desarrollo de una IA propia, teniendo que planear cómo será el tipo de método que se iba a utilizar, se eligió el método árbol de decisión debido a que se encontró una manera de visualizar el flujo de cada estado de forma visual, utilizando el animator controller de Unity.
También se tuvo que pensar y programar cada estado y como iba a ser la lógica de pasaje entre estos estados.
La forma que se eligió para pasar entre estados era según cada variable previamente mencionadas, habiendo condiciones que según la proximidad para con el Jugador controlado por la IA iban a desencadenar en la entrada de un estado u otro.
En cada estado debía haber una serie de acciones para que funcione.
Por ejemplo, en el estado de buscar la pelota funciona de la siguiente manera:



```
Script de Unity | 1 referencia
public class BuscarPelotaComportamiento : StateMachineBehaviour
{

    private Transform Pelota;
    public static StatesMachineController PJIA;


    public float velocidadMovimiento;

    public Transform PlayerIATransform;



    // OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
    Mensaje de Unity | 0 referencias
    override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        Pelota = GameObject.FindGameObjectWithTag("Ball").GetComponent<Transform>();
        PJIA = animator.gameObject.GetComponent<StatesMachineController>();


    }

    // OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
    Mensaje de Unity | 0 referencias
    override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        PJIA.transform.position = Vector3.MoveTowards(PJIA.transform.position, Pelota.position, velocidadMovimiento * Time.deltaTime);
        PJIA.MirarPelota();

    }
```

Código - Buscar Pelota Comportamiento

Lo que ocurre en este script, el cual es el que ejecuta el jugador controlado por la IA cuando está en estado “BuscarPelota”, es:
Obtiene la información del Transmform del Gameobject de la pelota y del jugador IA, además de su script StateMachineController, el cual se encarga de controlar distintas variables fundamentales para el pasaje entre estados y el correcto funcionamiento de cada uno.
Ya con estas variables declaradas en el update, se ejecuta una vez por frame que el jugador IA siga a la pelota, mientras que entra también al método llamado “MirarPelota”, el cual se encarga de que el jugador rote en dirección al balón, como si lo estuviera mirando y por ejemplo no corra de espaldas o de lado, viéndose confuso y sin sentido.

El método MirarPelota hace lo siguiente:

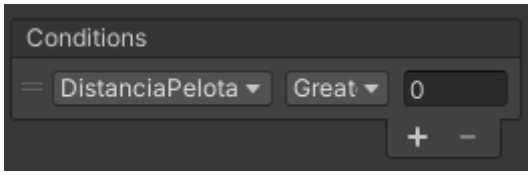
```
1 referencia
public void MirarPelota()
{
    if (Player2.tocandoBolaBlancaP2 && pelota.transform.position.x < 1)
    {
        gameObject.transform.LookAt(arcoP1.position);
    }
    if (pelota.transform.position.y > 2 )
    {
        gameObject.transform.LookAt(BuscarPelotaComportamiento.PJIA.transform.position);
    }
    else
    {
        gameObject.transform.LookAt(pelota.transform.position);
    }
}
```

Código – Mirar pelota

Hay 2 condiciones, una que entra cuando el player2, es decir, el que está siendo controlado por la IA, está en contacto con la pelota y la distancia de la pelota respecto al mismo jugador es menor que 1, en el eje x, la IA apuntará hacia el arco rival, mirándolo.
Esta condición es esencial para que en el momento que la IA dispara la pelota, sea en dirección al arco rival y no a cualquier lado.

La segunda condición es necesaria, ya que cuando la pelota alcanza cierta altura, el modelo de la IA seguía siguiendo la pelota con la mirada, causando que se ponga de espaldas al piso y se mueva de forma extraña.
Con esa condición que dice que “cuando la distancia de la pelota sea mayor a 2 en el eje y”, el jugador comenzará a mirar en la dirección en la que se está moviendo, arreglando ese problema.
El “else” sirve para que nuevamente prosiga a mirar la pelota en el instante en el que la distancia vuelva a ser menor que 2 en el eje vertical.

La condición principal para poder acceder a este estado es la siguiente:

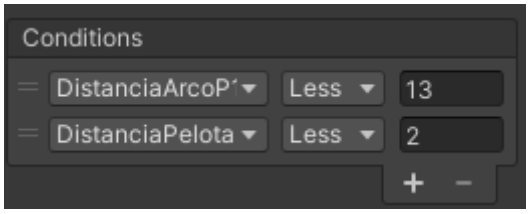


Condición principal mirar pelota

Esto significa que la IA estará en el estado de BuscarPelota todo el tiempo que la distancia de la pelota respecto al jugador sea mayor a 0, determinando que la busque todo el tiempo que no la tenga en su poder, y también la esté mirando.

Por esta razón este es el evento que hace de nexo entre todas las demás acciones, exceptuando la de defender el arco, ya que en este evento no busca a la pelota, sino que a su propio arco para defenderlo.

El evento de disparar al arco funciona de la siguiente manera:



Condiciones disparar al arco

Cuando la distancia del jugador AI sea menor a 13 y esté a una distancia menor que 2 entrará al estado “PatearArco”.
El cual funciona de la siguiente manera:

```
Script de Unity | 0 referencias
public class PatearPelotaComportamiento : StateMachineBehaviour
{
    public StatesMachineController accionesIA;


    // OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
    Mensaje de Unity | 0 referencias
    override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        Debug.Log("PATEA");

        //accionesIA.ChutarIA();
        accionesIA.PatearPelota();

    }

    // OnStateMove is called right after Animator.OnAnimatorMove()
    Mensaje de Unity | 0 referencias
    override public void OnStateMove(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        accionesIA = animator.gameObject.GetComponent<StatesMachineController>();
    }

}
```

Código – Estado patear pelota comportamiento

Lo que ocurre en este script es que en el update entra directamente al evento llamado “PatearPelota”, dentro del script “StatesMachineController”, el cual hace lo siguiente:

```
1 referencia
public void PatearPelota()
{
    //Debug.Log("No entra del todo");
    if (Player2.tocandoBolaBlancaP2)
    {

        transform.LookAt(arcoP1.position);
        Rbball.AddForce(transform.forward.normalized * 2f, ForceMode.Impulse);
    }

}
```

Código – Patear pelota acción

Entra a la condición en el momento que el player2 esté tocando la pelota y ejecuta nuevamente que apunte hacia el arco y le añade una fuerza al Rigidbody de la pelota, concretando una patada al arco.

Las demás acciones que puede realizar la IA son las siguientes:

despejar el balón y llevar la pelota hasta el arco rival.

Despejar el balón:

```
1 referencia
public void DespejarPelota()
{
    if (Player2.tocandoBolaBlancaP2 && distanciaArcoIA < 4)
    {
        //despejar pelota
        Rbball.AddForce(transform.transform.up * 0.2f, ForceMode.VelocityChange);
        Rbball.AddForce(/*Player2.movimiento*/ arcoP1.transform.position * 0.2f, ForceMode.Impulse);
    }
}
```

Código - Despejar Pelota

Despejar la pelota ocurre cuando el player2 esta tocando la pelota y cuando la distancia de su propio arco es menor a 4.
Lo que hace es añadirle una fuerza en vertical según hacia donde esta mirando la IA y de la misma manera, pero en horizontal, consiguiendo el efecto de que ambas fuerzas se combinan y pueda patear la pelota en dirección contraria a su arco para poder despejar la pelota, pero no lo suficientemente alejado para poder seguir controlándola más adelante.

llevar la pelota hasta el arco rival:

```
1 referencia
public void IrLlevandoseLaPelota()
{

    if (Player2.tocandoBolaBlancaP2 && contLlevarPelota > 2)
    {

        Rbball.isKinematic = true;
        Rbball.isKinematic = false;
        Rbball.AddForce(arcoP1.transform.position * 80f, ForceMode.Force);
        TutorialController.contLlevarseLaBola++;
        contLlevarPelota = 0;
    }

}
```

Código – Ir llevándose la pelota

Llevar la pelota hasta el arco rival entra cuando el player está tocando la pelota y un timer llamado “contLlevarPelota” es mayor a 2.
Esto se hizo con el objetivo de que la IA pueda ir dándole pequeños toques a la pelota cada 2 segundos, para poder ir llevándola hacia donde tiene que disparar.
Lo que ocurre dentro de la condición es lo siguiente:
El balón se hace kinematic y deja de serlo en el mismo segundo, con el fin de poder estabilizar la pelota con cada toque, logrando un efecto de control absoluto muy satisfactorio y realista.
Se le añade una fuerza necesaria para que se vea como un pequeño toque y que siga en su control, alejándola lo justo para avanzar.
Luego reinicia el contador para que se repita el ciclo de volver a entrar a la condición, en cuanto pasen 2 segundos para poder volver a golpear la pelota.

La siguiente imagen muestra cómo se obtuvo las distancias de todas las variables que son fundamentales para el desarrollo de esta IA, la misma se hizo dentro del script “StatesMachineController”.

```
{
    public Transform jugadorP1;
    public Transform pelota;
    public Transform arcoP1;
    public Transform arcoIA;


    public float distanciaJugadorP1;
    public float distanciaPelota;
    public float distanciaArcoP1;
    public float distanciaArcoIA;


    public Vector3 puntoInicial;

    public Animator StateMachine;

    public Rigidbody Rbball;

    public float contLlevarPelota;

    //private Player2 PlayerIA;


    Mensaje de Unity | 0 referencias
    private void Start()
    {
        //PlayerIA = gameObject.GetComponent<Player2>();
        //animator.GetComponent<Animator>();
        puntoInicial = transform.position;

        contLlevarPelota = 0;
    }


    Mensaje de Unity | 0 referencias
    private void Update()
    {
        distanciaJugadorP1 = Vector3.Distance(transform.position, jugadorP1.position);
        distanciaPelota = Vector3.Distance(transform.position, pelota.position);
        distanciaArcoIA = Vector3.Distance(transform.position, arcoIA.position);
        distanciaArcoP1 = Vector3.Distance(transform.position, arcoP1.position);

        StateMachine.SetFloat("DistanciaJugadorP1", distanciaJugadorP1);
        StateMachine.SetFloat("DistanciaPelota", distanciaPelota);
        StateMachine.SetFloat("DistanciaArcoIA", distanciaArcoIA);
        StateMachine.SetFloat("DistanciaArcoP1", distanciaArcoP1);
```

Codigo – StatesMachineController obtención de variables

Primero se declaran todas las variables.
Hacerlas todas públicas definitivamente no es la manera más óptima de programar, pero como el juego es muy pequeño y no requiere de tantos recursos, se puede contar con esa libertad, agilizando el desarrollo.

Después se declaran variables de tipo Floats, en las cuales almacenar las distancias y tenerlas fácilmente accesibles para consultar en todo momento.

Finalmente se ejecuta en el update que mediante una variable de tipo vector3 se almacene las distancias, comparando la posición tanto del objeto como del jugador2, obteniendo por resultado la distancia en la que se encuentran, igualándolas a las variables previamente mencionadas, y debajo de todo se

establece cada una de estas variables para que respondan a través del Animator controller.

De esta forma ya se tienen todos los comportamientos mínimos y necesarios para poder jugar contra una IA que emula el comportamiento humano.

Resultados: Proyecto número 2

Gracias al desarrollo e implementación de mecánicas propias, se logró crear una inteligencia artificial de tipo “Árbol de decisión”, a través de condiciones lógicas de programación y el “Animator Controller” de Unity para poder visualizar y navegar entre los diferentes estados creados.

El objetivo del juego era poder reemplazar al segundo jugador con una IA para que el primer jugador pueda jugar en forma “Single Player” y siga siendo una experiencia disfrutable, que de otra forma, en solitario sería completamente imposible o no tendría gracia, ya que ahora se puede competir contra la máquina, existiendo la posibilidad de perder y ganar, teniendo que aprender y mejorar en el uso de las mecánicas y dinámicas, desarrollando distintas estrategias tanto para defender tu portería, como de conseguir llegar al arco rival con la pelota para poder anotar, siendo todo un desafío, divertido e interesante para el jugador.

Resultados y análisis de la muestra

Luego de haber hecho el experimento con los 17 participantes se pudo obtener información valiosa para lograr determinar si la hipótesis:
“Se espera que la implementación exitosa de la inteligencia artificial en las mecánicas de los videojuegos tenga un impacto positivo en la experiencia global del jugador. Se predice que la mejora en la adaptabilidad, complejidad y dinamismo de las mecánicas de juego, gracias a la inteligencia artificial, contribuirá significativamente a una experiencia más inmersiva y satisfactoria para los jugadores”. Se cumple o no.

Para poder analizar cada aspecto por separado y de forma ordenada, primero se segmenta entre los **jugadores que lograron vencer a la IA** y **quienes no pudieron vencerla**.
Por último se analizarán las respuestas y el gameplay global de los jugadores para poder determinar un resultado final.

Los jugadores que lograron vencer a la IA fueron 2:

El jugador número 4 y el número 13,

Ambos calificaron el nivel de satisfacción como “muy alto” en el modo de juego “Freekick” el cual no tiene implementado IA, sin embargo marcaron el nivel de desafío y el de frustración muy bajo y por eso se llega a la conclusión de que simplemente les resultó un modo “entrenamiento” para practicar a jugar antes de enfrentar el verdadero desafío que es la IA, la cual fue catalogada por ambos como muy desafiante y también muy satisfactorio.
El nivel de frustración subió un poco en ambos casos pero se mantiene muy por debajo de un nivel preocupante, además ambos marcaron que volverían a jugar

contra la IA, mientras que el anterior modo, el jugador número 4 marcó que no lo volvería a jugar, se puede entender que marcó esta opción ya que le resultó muy sencillo, no siendo un reto significativo, al igual que el jugador 13, sin embargo el si marcó que lo volvería a jugar.
También se puede ver que ambos marcaron que necesitaron un nivel de estrategia muy bajo en el modo sin IA, y ambos marcaron que necesitaron un nivel muy alto en el modo de juego con IA.
Volviendo al hecho de que ese si les resultó un verdadero desafío.
Otra cosa que estos jugadores tenían en común era su alto nivel de experiencia jugando videojuegos, pudieron entender rápidamente las mecánicas y manejarse con soltura durante toda la experiencia.
Al jugador número 13 le llevó varios intentos, pero él mismo insistió en querer seguir intentando hasta ganarle a la IA, volviendo a hacer hincapié en que el desafío, la satisfacción y la estrategia necesaria resultaron altos, y aunque la frustración haya subido no fue lo suficiente como para hacer al jugador rendirse.
El jugador número 4 pudo vencer a la IA al primer intento, aunque con un resultado muy ajustado, estando a un punto cada uno de la victoria, dejando el nivel de desafío, estrategia y satisfacción en lo más alto, ya que mientras más ajustado es el resultado, más satisfacción se obtiene al final.
Por esta misma razón se puede ver que el nivel de satisfacción del modo sin IA es más bajo y en ambos casos se marcó el nivel de desafío con el mínimo, debido a que no lo consideraron un verdadero reto.

Finalmente ambos coincidieron en que la incorporación de la IA mejoró la experiencia del juego.

Se los pudo ver en la grabación y al momento de jugar con mucho interés y ganas. Sin embargo, por la personalidad de cada uno, el jugador 4 estaba más tranquilo, mientras que el 13 estaba más ansioso.
Ambos completaron el tutorial y ganaron el modo Freekick con facilidad, no mostraron mayor entusiasmo, más bien estaban entendiendo y experimentando

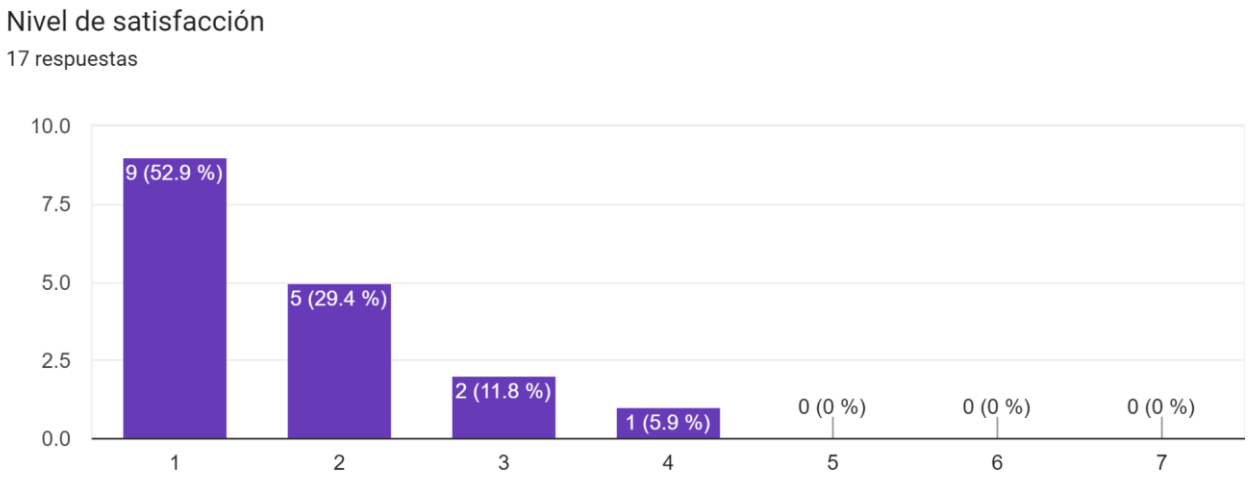
con la mecánica.
Finalmente el jugador número 4 comenzó a enfrentarse a la IA, identificó que era un gran desafío pero se vio capaz, asique se decidió en enfrentarla un total de 9 veces hasta que lo consiguió.
Se pudo ver que ambos se divirtieron mucho, pero debido al gran desafió y el ambiente que se generó con el jugador número 4, intentando vencer a la IA una y otra vez, lejos de frustrarse y querer abandonar, lo hizo motivarse más y sentirse desafiado hasta conseguirlo, llegando a la conclusión de que lo disfrutó mucho más que el jugador número 13, que lo vencio al primer intento, al no haber tenido que intentarlo más veces no fue consciente del verdadero desafío que se acababa de haber enfrentado y superado.

Jugadores que no pudieron vencer a la IA:

Estos fueron la gran mayoría, de hecho, solamente 2 participantes le ganaron, así que este enfoque debe ser más general ya que son los 15 jugadores restantes.

Todos se clasificaron con menor nivel de experiencia en videojuegos que los participantes que ganaron, exceptuando el jugador número 9.
Se podría decir que vencer a la IA es un desafío difícil para jugadores no experimentados y también para jugadores con experiencia, sin embargo el jugador número 9 solamente jugó 2 partidas, en la primera se tomó por sorpresa el rápido ataque de la IA, al igual que le pasó a la mayoría de los jugadores, sin embargo en su segundo intento demostró bastante más control y tranquilidad, perdiendo por un resultado un poco más ajustado, se cree que si hubiera continuado jugando unas partidas más la hubiera podido superar fácilmente, ya que en ocasiones dice que se había olvidado con qué botón se hacía la habilidad de tirarse a barrer.

A pesar de no haber vencido a la IA la gran mayoría de los participantes votaron de forma muy satisfactoria la sensación que tuvieron.



Nivel de satisfacción - Preguntas del modo de juego "Player vs IA" (Con IA)

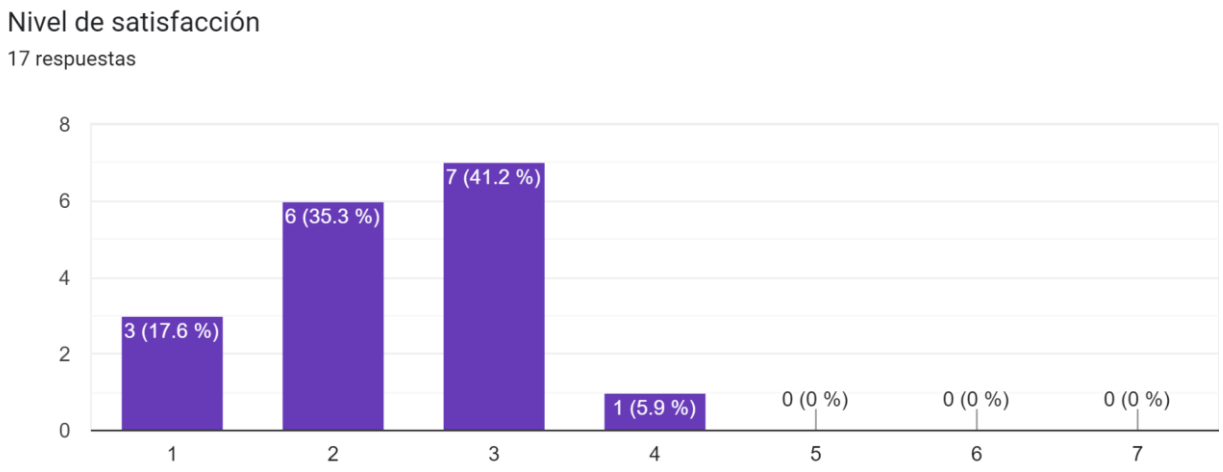
podemos ver que únicamente una persona puso el nivel de satisfacción en 4 significando “Normal”, ni satisfactoria ni insatisfactorio, esa persona fue el jugador número 14, quien podemos ver que clasificó el nivel de frustración en 3, es decir un poco alto, y el nivel de desafío al máximo, sin embargo marcó en nivel de experiencia en videojuegos como el segundo más bajo.

Llegando a la conclusión de que el desafío era demasiado difícil para su habilidad, provocando que no pudiera disfrutar la experiencia, ya que además indicó que no lo volvería a jugar. Siendo la única persona quien no volvería a jugar vs la IA.

Sin embargo y a pesar de que el nivel de desafío y estrategia que se debía emplear fue clasificado como mucho más alto que el modo de juego sin IA, también ese modo fue clasificado como menos satisfactorio que el que era vs la IA, dejando en evidencia que ante un mayor desafío hay mayor satisfacción al completar el objetivo.

Se vió en varios de los videos a jugadores festejando goles que le hacían a la

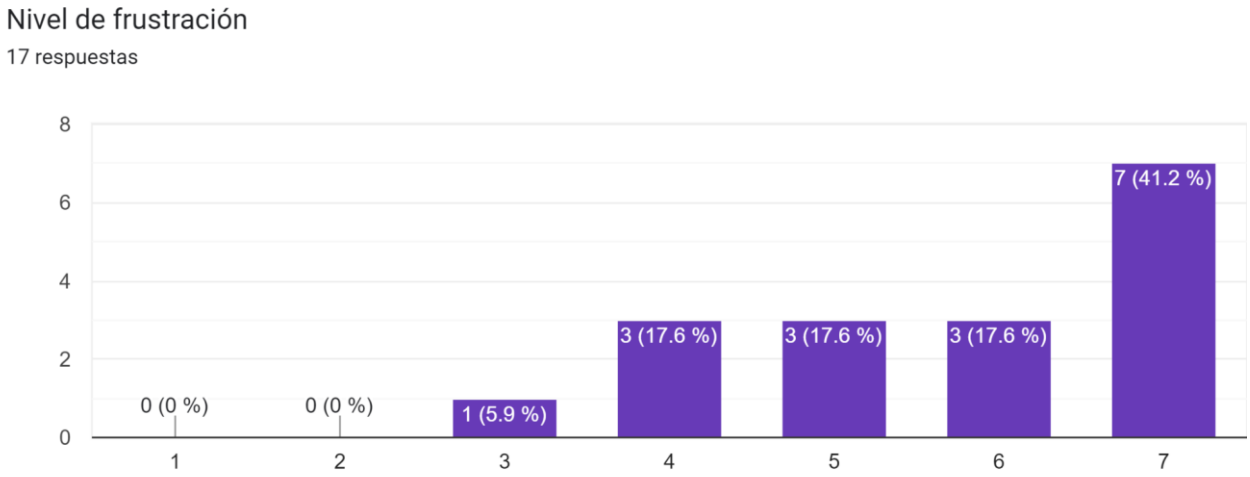
IA, mientras que con el primer modo sin esta, no ocurría tanto.



Nivel de satisfacción - Preguntas del modo de juego "Freekick" (Sin IA)

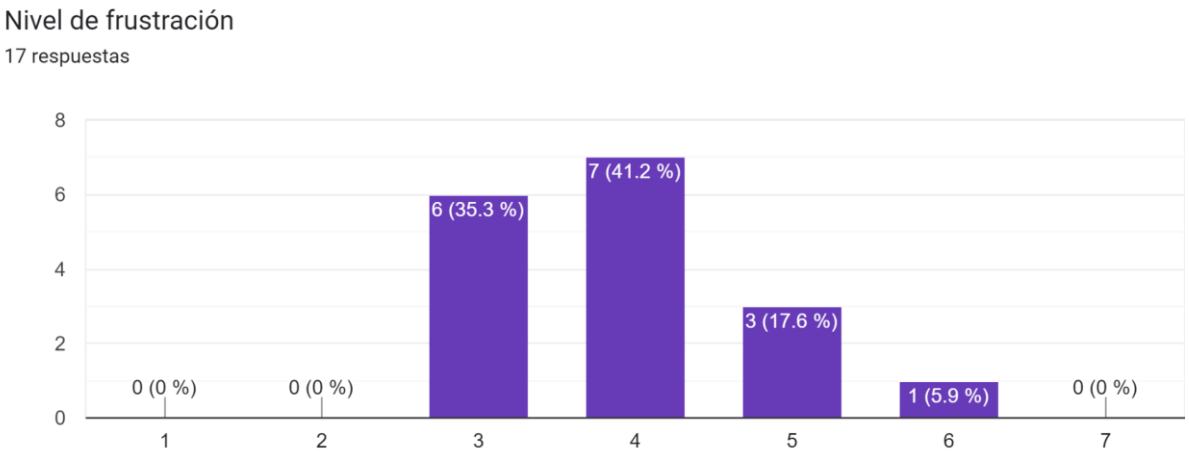
En este gráfico se puede ver como la gran mayoría votó el modo de juego sin IA como algo satisfactorio, en la casilla número 3.
Dejando en evidencia nuevamente que la inclusión de la IA mejoró mucho la experiencia de los jugadores, agregándoles un mayor desafío y necesidad de emplear estrategia y adaptarse al rival en cuestión, cosa que sin la IA se ve que no ocurre.
Solo una persona lo votó como normal, y fue nuevamente el jugador número 14, quien se recuerda que era la persona con menor experiencia jugando videojuegos.

También se ve que el nivel de frustración es muy bajo, al igual que el nivel de desafío en el modo de juego sin IA.
Mientras que en el modo de juego con IA es considerablemente más alto, sin llegar a los rangos más altos.



Nivel de frustración - Preguntas del modo de juego "Freekick" (Sin IA)

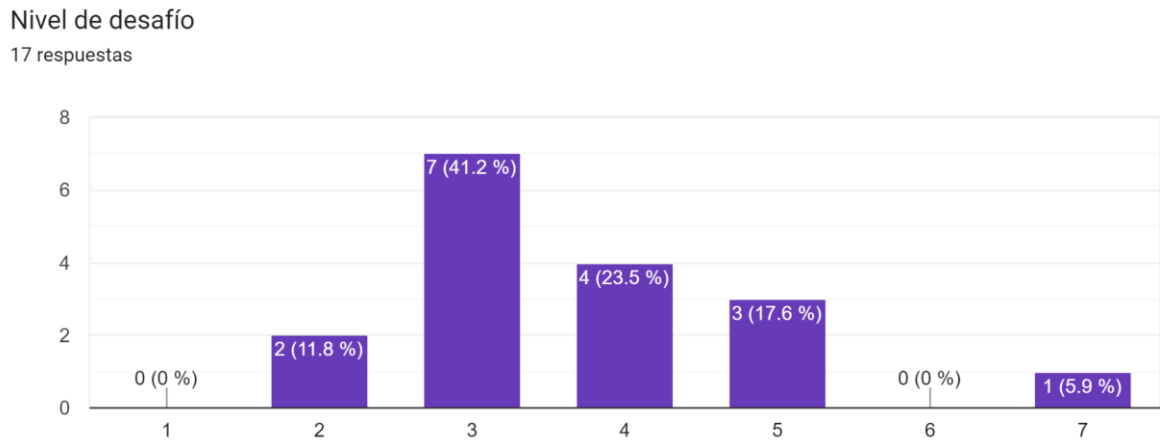
Se ve que el nivel de frustración, la mayoría votó lo mínimo posible, siendo algo no tan bueno, como se podría llegar a pensar, ya que para poder lograr un verdadero estado de “Flow” el videojuego debe tener al jugador inmerso en la experiencia, frente a un desafío que no le resulte demasiado simple o poco desafiante como se ve que ocurre en este caso.



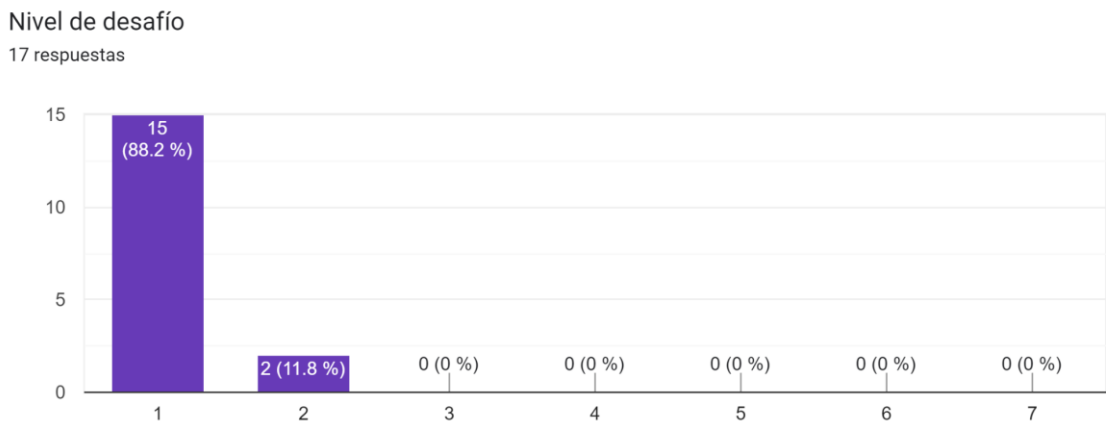
Nivel de frustración - Preguntas del modo de juego "Player vs IA" (Con IA)

Para continuar con el punto del nivel de frustración es importante que no sea alto pero tampoco lo mínimo posible, se puede ver que en el caso del modo de juego vs la IA la frustración aumenta, siendo la mayoría votando entre “Normal” y “algo frustrante”. solamente una persona lo voto con casi nada frustrante y fue el jugador número 8, quien se puede ver que tiene un nivel de experiencia alto jugando videojuegos, también se lo vió en el video desempeñándose bastante bien, se le ofreció volver a intentarlo pero no lo hizo, sin embargo por cómo jugó se cree que lo hubiera podido superar con un par de intentos más, ya que sólo intentó una vez.

Para finalizar con el tema de la frustración, volvemos nuevamente al nivel de desafío de ambos modos de juego para compararlos, ya que ambas respuestas tienen una gran relación y puede justificarse con ellas el estado de flow.



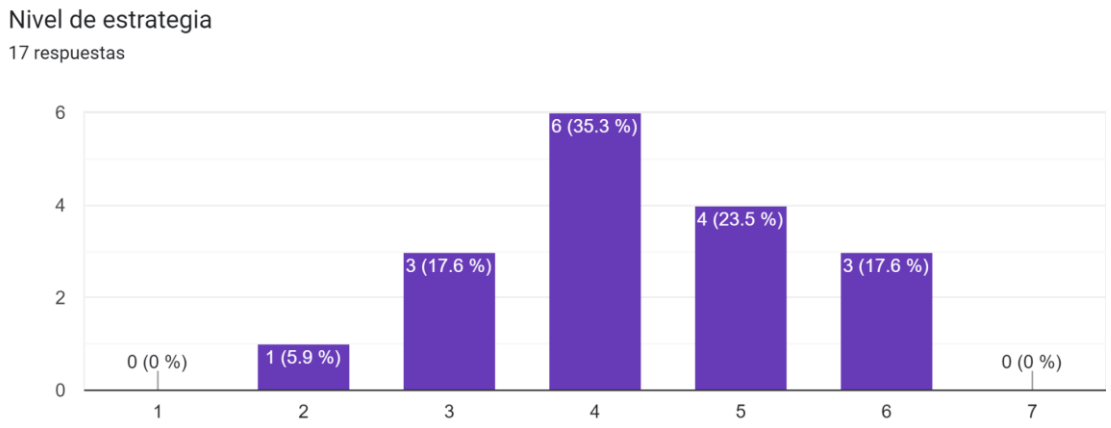
Nivel de desafío - Preguntas del modo de juego "Freekick" (Sin IA)



Nivel de desafío - Preguntas del modo de juego "Player vs IA" (Con IA)

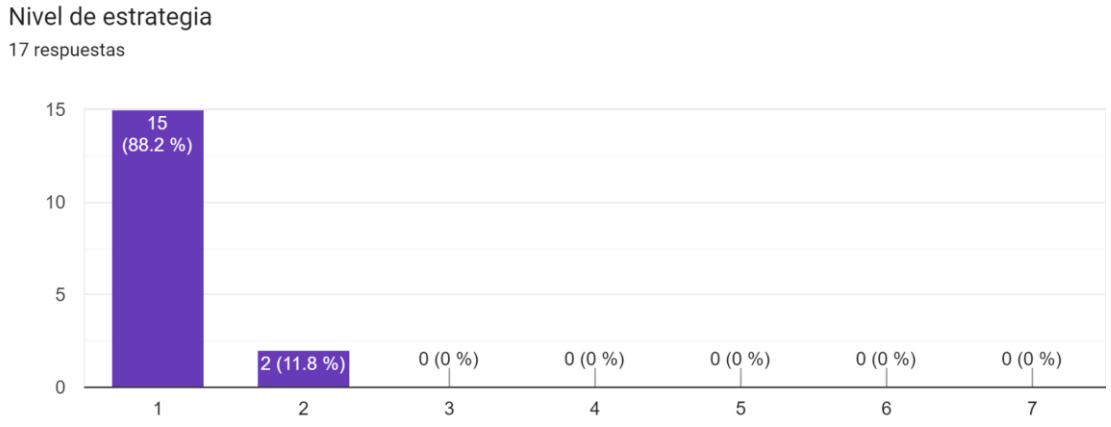
Comparando ambas respuestas, queda en evidencia que el nivel de desafío fue clasificado como lo más alto posible, cumpliendo otra vez con la hipótesis de que su inclusión mejoró completamente la experiencia de los jugadores, respecto a inmersión, desafío y satisfacción, logrando que los jugadores puedan entrar en un estado de flow genuino contra la IA, mientras que en el modo sin IA simplemente estaban jugando o experimentando con la mecánica, no siendo un verdadero desafío, por esa razón, la frustración era mínima en el primer caso, y justamente por eso no es necesariamente bueno en todos los casos tener al jugador sin absolutamente nada de frustración, hubiera sido completamente fatal para el experimento que se hubiera catalogado al modo de juego vs la IA como extremadamente frustrante o muy frustrante, ya que hubiera arruinado la experiencia de los jugadores completamente y no se cumpliria la hipotesis de su mejora.

Para finalizar con este tema repasamos las respuestas del nivel de estrategia empleada para terminar de comprobar el punto en cuestión, respecto al estado de flow.



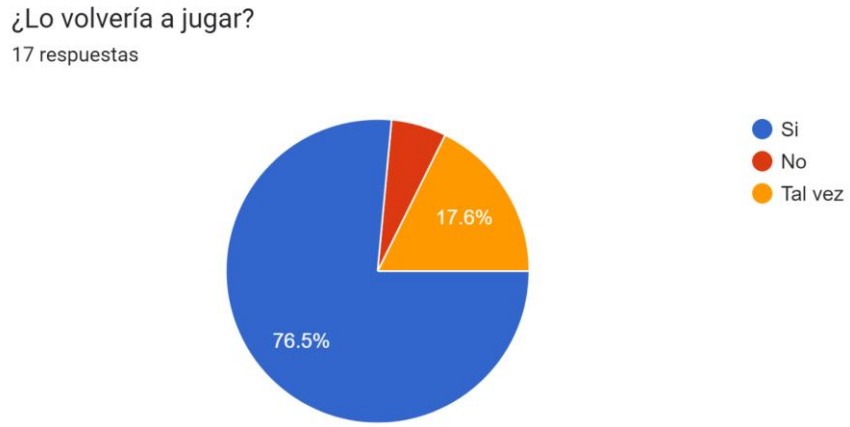
Nivel de estrategia - Preguntas del modo de juego "Freekick" (Sin IA)

Puede verse a simple vista como el nivel de estrategia empleada es mayormente “Normal”, es decir, no se necesita crear una estrategia para ganar el primer modo de juego, sin embargo, tampoco es un modo que se gana solo, pero queda en evidencia que al no haber una IA que desafíe al jugador no le es necesario emplear un esfuerzo para vencer dicho modo.
Mientras que en el modo vs la IA el nivel de desafío fue contestado como máximo por todos los jugadores, menos uno, pero que también lo catalogó como el segundo más alto.



Nivel de estrategia - Preguntas del modo de juego "Player vs IA" (Con IA)

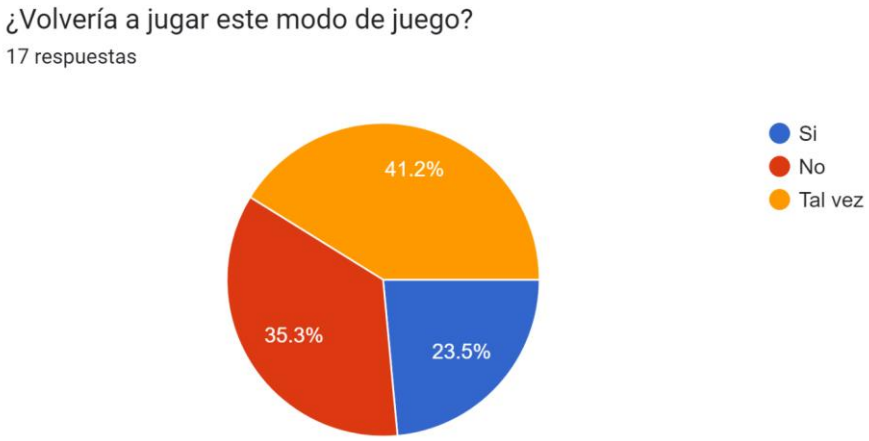
Sin embargo, a pesar de que solo 2 jugadores pudieron vence a la IA,



¿Volvería a jugar? - Preguntas del modo de juego "Player vs IA" (Con IA)

El 76.5% volvería a jugarle y el 17.6% respondieron “tal vez” y solo una persona respondió que no, la cual fue mencionada anteriormente.

A diferencia del modo de juego sin IA que solamente el 23.5% de los participantes lo volverían a jugar con seguridad, mientras que el 36.3% asegura que no lo volvería a jugar y el 41.2% de los jugadores no están seguros.



¿Volvería a jugar? - Preguntas del modo de juego "Freekick" (Sin IA)

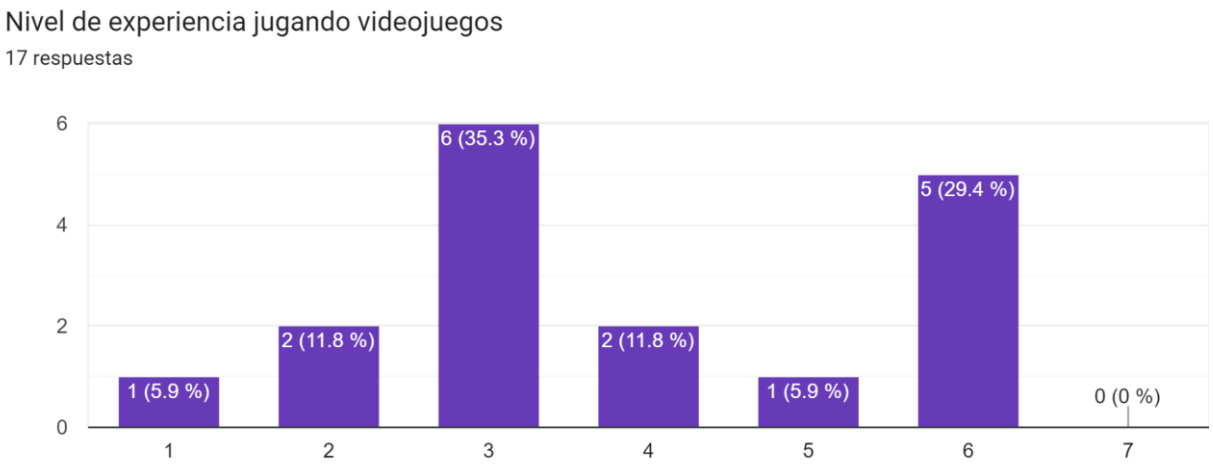
Con estas respuestas nuevamente se hace hincapié en que la inclusión de la IA fue una gran mejora en la experiencia de los jugadores, todos exceptuando sólo uno, queriendo volver a enfrentarse a esta, para así poder vencerla de una vez, siendo un desafío difícil pero alcanzable y divertido.

Análisis global:

Después de analizar puntualmente las respuestas de los participantes, respecto a los modos de juego que experimentaron, también se les hizo unas preguntas globales, propias de cada jugador, que se consideró relevantes para el estudio, como las que son:
El nivel de experiencia jugando videojuegos, ¿que videojuegos juega habitualmente? y ¿cree que la incorporación de la IA mejoró la experiencia?

Respecto a la experiencia de los participantes en videojuegos no fue una sorpresa ver que quienes más experiencia tenian son quienes pudieron vencer a la IA, como se dijo anteriormente, sin embargo el global tenían una experiencia media, con la mayoría, habiendo 9 participantes votando entre más de lo normal.

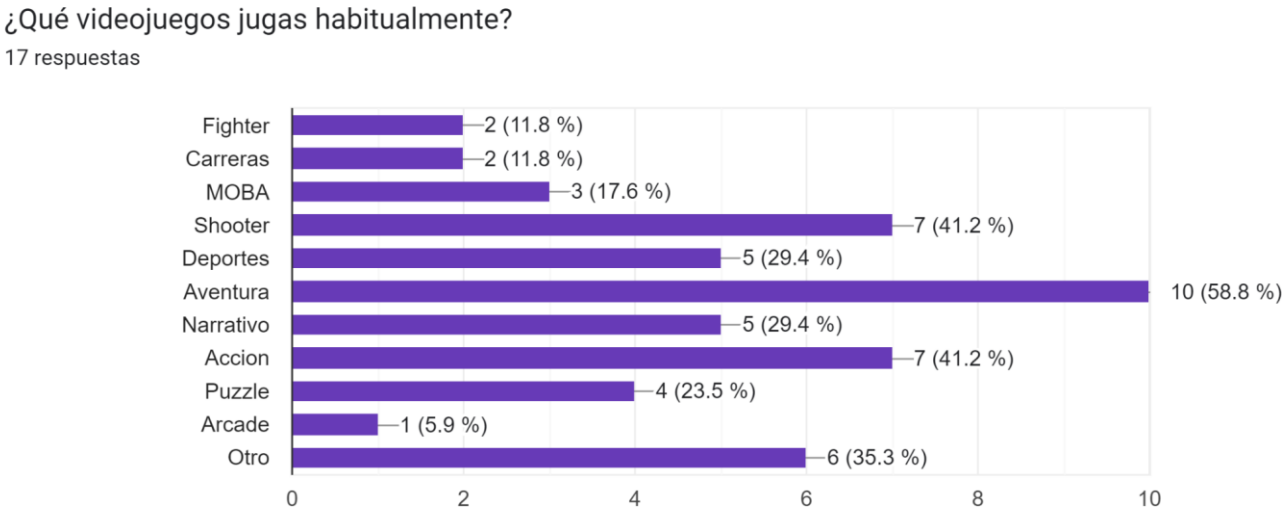
pero también hubo una gran cantidad de participantes que votaron que tenían muy poca experiencia jugando videojuegos, aunque ninguno tenía una nula experiencia, ya que era necesario para los criterios de la muestra que tengan algo de experiencia, como se ve en el siguiente cuadro.



Nivel de experiencia jugando videojuegos - Preguntas globales

Se podría decir que fue una muestra bastante equilibrada respecto a nivel de experiencia en videojuegos, y eso dio a lugar a múltiples situaciones, pero siempre llegando a la misma conclusión previamente estipulada.

También se les preguntó qué tipo de juegos jugaban habitualmente, para saber si estarían acostumbrados a un juego del estilo o su target era otro.
Como se ve en la imagen, todos jugaban distintos tipos de videojuegos.
La gran mayoría tiene experiencia en juegos de aventura, shooters, acción y otros.
Siendo la minoría quienes jugaban juegos de tipo fighter, arcade o Moba.



¿Qué videojuegos jugas habitualmente? - Preguntas globales

En este caso es muy difícil catalogar el tipo de videojuego que se desarrolló, ya que a pesar de tener motivo de deportes, siendo un videojuego de Fútbol 1 vs 1, poco se parece a un verdadero partido de fútbol o a un juego de fútbol tradicional, como el que puede ser el Fifa, Pes, etc.
Según sus tecnicas y dinamicas con la pelota y el rival, se llegó a la conclusión de que es un tipo de juego que podría ser catalogado más fácilmente como de tipo “Arcade”, ya que no sigue ninguna regla lógica del fútbol, ni pretende simular, siendo puramente jugar con el movimiento del personaje, sus habilidades y la pelota.
En esta encuesta vemos que justamente el tipo de videojuego menos jugado por los participantes, son los arcade, dejando en evidencia, porque les costó a algunos superar los desafíos y hacerse al movimiento de la pelota, mientras otros lo entendieron bastante más rápido.

Para finalizar se les preguntó a los participantes si creen que la incorporación de la IA mejoró a la experiencia, y todos exceptuando 3 personas respondieron que SÍ



¿Cree que la incorporación de la IA mejoró la experiencia? - Preguntas globales

Se puede ver que el 82.4% de los participantes respondieron de forma afirmativa, mientras que solo el 17.6% de ellos respondieron Tal vez, y ninguno respondió de forma negativa, siendo un muy buen resultado para validar la hipótesis planteada en esta investigación.

Conclusiones, recomendaciones y trabajos futuros.

Desde la primera vez que se implementó IA en los videojuegos fue para poder lograr objetivos que de otra manera no se podrían lograr, ese objetivo siempre fue de la mano con mejorar la experiencia del usuario.
como por ejemplo, pudiendo convertir el “Pong” en un juego no necesariamente para 2 jugadores.

Con toda la información obtenida sobre IA en los videojuegos, se logró

desarrollar 2 proyectos a lo largo de este trabajo, el primero fue descartado, pero se tomó todo lo aprendido para desarrollar el segundo, el cual fue óptimo para el estudio junto a la muestra de 17 jugadores y así poder llegar a una conclusión respecto a la Hipótesis planteada.

La hipótesis planteada dice:

“Se espera que la implementación exitosa de la inteligencia artificial en las mecánicas de los videojuegos tenga un impacto positivo en la experiencia global del jugador. Se predice que la mejora en la adaptabilidad, complejidad y dinamismo de las mecánicas de juego, gracias a la inteligencia artificial, contribuirá significativamente a una experiencia más inmersiva y satisfactoria para los jugadores”

Luego de haber analizado los resultados de las encuestas podemos llegar a la conclusión de que la hipótesis se cumple, ya que todos los jugadores disfrutaron más el modo de juego que tenía la inteligencia artificial incorporada, y esto es así debido a que empleaba un nivel de desafío y uso de la estrategia, obligando al usuario a mejorar su adaptabilidad logrando una mejor inmersión y satisfacción cuando conseguía el objetivo, como el que puede ser anotar un gol, aunque también como consecuencia subía la frustración cuando este no se alcanzaba, aunque nunca subía demasiado como para querer abandonar el juego, estando en un equilibrio denominado “FLOW” donde el jugador tiene que estar concentrado para ganar, siendo exigido, pero tampoco a un nivel tan alto como para verse completamente superado y perder las ganas de seguir. Esto solo pudo ser posible añadiendo la inteligencia artificial, en caso de jugar en solitario.

Referencias

Alberto, M. (2016). El universo infinito de No Man’s Sky no es más que un reflejo del nuestro. *Hipertextual*, 1, 1-1.

Cuevas, D. (2013). Análisis de Algoritmos Pathfinding. *Pontificia Universidad Católica De Valparaíso Facultad De Ingeniería Escuela de Ingeniería Informática,* 4, 4-4.

García, J. (2019). 'AI Dungeon' es un juego tipo "Elige tu propia aventura" basado en IA al que puedes jugar desde el móvil. *Xataka*, 1, 1-1.

Gonzáles, M. (2023). Cómo se aplica la Inteligencia Artificial en los videojuegos. *Instituto de Ingeniería del Conocimiento*, 1, 1-1.

Martí, G. (2023). 15 juegos que usan la dificultad dinámica o adaptativa. *VANDAL*, 1, 1-1.

Morales, C. (2015). Utilización De Máquinas De Estados Y Árboles De Decisión Para El Videojuego Starcraft. *Universidad Carlos III de Madrid Escuela Politécnica Superior*, 43, 43-43.

Orrego, C. (2019). Diseño De Algoritmo De Generación Procedural Enfocado A Videojuegos. *Universidad Técnica Federico Santa María Departamento De Informática Santiago – Chile*, 3, 3-3.

Rouhiainen, L. (2018). Inteligencia Artificial 101 cosas que debes saber hoy sobre nuestro futuro. *Barcelona:* Alienta.

Anexo

Página itch.io para jugar al segundo proyecto:

https://giantripo.itch.io/streetfutbol

Contraseña para ingresar: Futbol123

Videos con el funcionamiento de las IA´s y Builds de los proyectos:

https://drive.google.com/drive/folders/1-m7dG336JVPnOx0E2ZqyRq8V3gyL3neG?usp=sharing

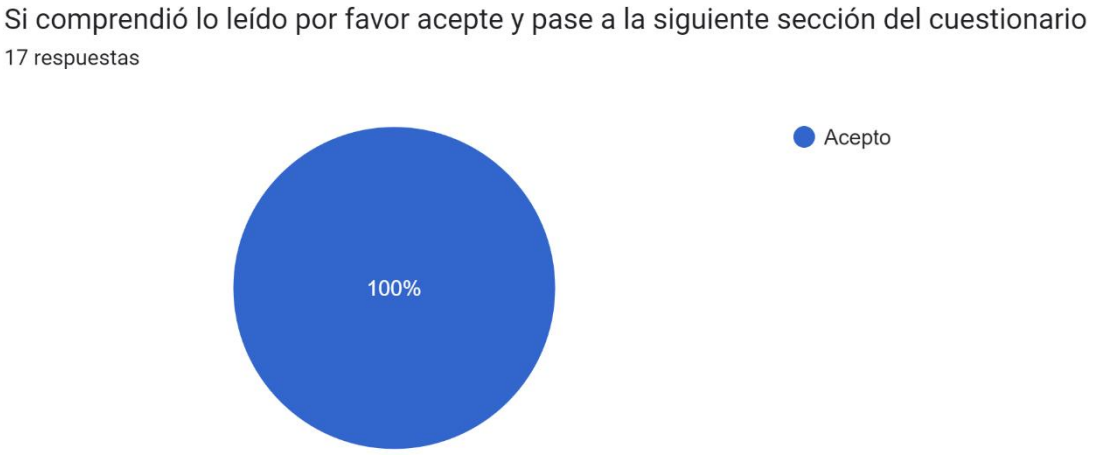
Videos de los participantes jugando:

https://drive.google.com/drive/folders/1FOJCzyMaUF5CDqNlalmUY-y1WPP6tsyd?usp=sharing

Respuestas de las encuestas:

Las respuestas de la encuesta se leen de la siguiente manera:

1 = Muy alto
2 = Bastante alto
3 = Algo alto
4 = Normal - Medio
5 = Algo bajo
6 = Bastante bajo
7 = Muy bajo



Consentimiento informado

Los 17 participantes leyeron y aceptaron el siguiente consentimiento informado:

Consentimiento Informado

POR FAVOR, LEA CON ATENCIÓN LO QUE ENCONTRARÁ A CONTINUACIÓN. ACEPTE EL CONSENTIMIENTO SÓLO SI ESTÁ DE ACUERDO CON SU PARTICIPACIÓN Y SI COMPRENDE CLARAMENTE SUS DERECHOS. SU ACEPTACIÓN ES UN REQUERIMIENTO NECESARIO PARA EL ESTUDIO. SI DESEA UNA COPIA DE ESTE FORMULARIO DE CONSENTIMIENTO INFORMADO, PUEDE PEDIRLA.

Esta investigación se desarrolla con fines académicos, a cargo de la Dra. Prof. Mariana Psyrdellis, bajo el aval de la Universidad Abierta Interamericana y a los fines de la materia de Metodología de Investigación, carrera de Licenciatura en Producción de Simuladores y Videojuegos.

La política del departamento universitario es que todos los participantes son voluntarios y usted tiene el derecho de abandonar en cualquier momento que lo desee, sin ningún perjuicio, si no está de acuerdo con continuar. Sus respuestas son confidenciales y secretas. La utilización de la información será de modo general, y no se va a identificar de manera individual a los participantes. Este

mismo estudio será anónimo, y no implica ningún tipo de devolución a los participantes sobre los resultados.

Descripción del estudio:

Este estudio consta de un cuestionario para clasificar la experiencia vivida tras haber jugado al videojuego "StreetFutbol" de Gian Franco Tripodaro

Usted podría sentir que sus respuestas han sido poco satisfactorias, sin embargo, tenga en cuenta que ninguna de las pruebas contempla respuestas correctas o incorrectas. Con su colaboración, estará ayudando al avance y desarrollo del conocimiento, junto con contribuir a la formación profesional de los alumnos.

Confidencialidad:

Su nombre no aparecerá en ninguno de nuestros archivos, de manera que no podrán identificarse sus respuestas particulares. Recuerde que no se está buscando conocer las respuestas de una persona en particular, sino las de grupos de personas en general.

Cualquier duda respecto de sus derechos como participante en este estudio pueden ser referidas al Comité de Ética de la Universidad Abierta Interamericana, quien avaló la realización de esta investigación.

Abandono del estudio:

Usted puede decidir abandonar el presente estudio en cualquier momento que lo desee, y es libre de hacerlo sin ningún problema.

Participante	Consentimiento informado	Nivel de satisfacción(SIN IA)	Nivel de desafío (SIN IA)	Nivel de frustración(SIN IA)	Nivel de estrategia(SIN IA)	¿Volvería a jugar este modo de juego?(SIN IA)	Nivel de satisfacción(CON IA)	Nivel de desafío(CON IA)	Nivel de frustración(CON IA)	Nivel de estrategia(CON IA)	¿Lo volvería a jugar?(CON IA)	Nivel de experiencia jugando videojuegos	¿Qué videojuegos jugas habitualmente?	¿Cree que la incorporación de la IA mejoró la experiencia?

Jugador 1	Acepto	3	3	6	6	Tal vez	2	1	5	1	Si	3	Carreras, Aventura, Narrativo, Puzzle	Si
Jugador 2	Acepto	2	3	5	5	Si	2	1	3	1	Si	5	Narrativo, Otro	Si
Jugador 3	Acepto	1	2	7	4	Si	1	1	5	1	Si	3	Aventura, Puzzle, Otro	Si
Jugador 4	Acepto	1	7	7	6	No	1	2	5	2	Si	2	Carreras, Shooter, Deportes, Aventura, Accion	Si
Jugador 5	Acepto	3	3	5	3	No	2	1	4	1	Si	4	MOBA, Aventura, Puzzle	Si
Jugador 6	Acepto	1	4	4	2	Tal vez	2	1	4	1	Tal vez	6	Puzzle, Arcade	Si
Jugador 7	Acepto	2	3	7	6	Tal vez	1	1	4	1	Si	3	Fighter, Shooter, Deportes, Accion	Si
Jugador 8	Acepto	3	5	7	5	No	1	1	6	1	Si	3	Shooter, Aventura	Si
Jugador 9	Acepto	3	5	7	5	No	1	1	4	2	Si	2	Fighter, Shooter, Aventura, Accion	Si

Jugador 10	Acepto	2	4	7	4	No	1	1	4	1	Si	3	Shooter, Deportes, Aventura, Narrativo, Accion	Si
Jugador 11	Acepto	3	3	6	4	Si	1	1	4	1	Si	4	MOBA, Narrativo, Otro	Si
Jugador 12	Acepto	2	3	3	4	No	3	1	3	1	Tal vez	6	Deportes	Tal vez
Jugador 13	Acepto	2	4	7	5	Si	1	1	3	1	Si	1	MOBA, Shooter, Aventura, Accion, Otro	Si
Jugador 14	Acepto	4	4	4	4	Tal vez	4	1	3	1	No	6	Aventura, Narrativo, Accion, Otro	Tal vez
Jugador 15	Acepto	3	3	4	3	Tal vez	3	1	3	1	Si	6	Aventura, Accion	Si
Jugador 16	Acepto	3	5	6	4	Tal vez	1	2	4	1	Si	3	Shooter, Deportes	Si
Jugador 17	Acepto	2	2	5	3	Tal vez	2	1	3	1	Tal vez	6	Otro	Tal vez