



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

TEMA:

**“TÉCNICAS DE APRENDIZAJE AUTOMÁTICO, APLICADAS EN UN
PROTOTIPO DE VIDEOJUEGO”**

Proyecto de investigación presentado previo a la obtención del Título de Ingeniero en
Informática y Sistemas Computacionales

AUTOR:

Mora Parra Alex Iván

TUTOR:

Ing. Mg. Segundo Corrales

LATACUNGA – ECUADOR

Marzo 2022



Ingeniería
Informática Y Sistemas
Computacionales

DECLARACIÓN DE AUTORÍA

Yo Mora Parra Alex Iván, declaro ser autor del presente proyecto de investigación: **“TÉCNICAS DE APRENDIZAJE AUTOMÁTICO, APLICADAS EN UN PROTOTIPO DE VIDEOJUEGO”**, siendo el Ing. Mg. Segundo Humberto Corrales Beltrán tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Mora Parra Alex Ivan

C.I. 180489963-9



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título:

“Técnicas de aprendizaje automático, aplicadas en un prototipo de videojuego”, de Mora Parra Alex Ivan, de la carrera de Ingeniería en informática y sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, marzo 2022



El Tutor

Mg. Segundo Humberto Corrales Beltrán

C.I: 050240928-7



APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de Ciencias de la Ingeniería y Aplicadas; por cuanto, el postulante: Mora Parra Alex Iván con el título de Proyecto de titulación: “Técnicas de aprendizaje automático, aplicadas en un prototipo de videojuego” han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, marzo 2022

Para constancia firman:

Lector 1 (presidente)
Nombre: Mg. Rene Quisaguano
CC: 172189518-1

Lector 2
Nombre: Dr. José Cadena
CC: 050155279-8

Lector 3
Nombre: Mg. Edwin Quinatoa
CC: 050256337-2



AGRADECIMIENTO.

Agradezco eternamente a mi padre y madre por apoyarme en todo este proceso y por brindarme las herramientas necesarias para elaborar esta propuesta y a mi hermana por sus consejos y ayuda.

También agradezco a la Universidad Técnica de Cotopaxi junto a todos los docentes que me han brindado sus conocimientos dentro de mi formación académica y a mis amigos que colaboraron plenamente en este proceso.

Alex



DEDICATORIA

Dedico este proyecto principalmente a mi madre, por tener paciencia en todas las metas que me he propuesto y por brindarme su amor incondicional a pesar de las dificultades, a mi padre ya que gracias a él y sus incontables esfuerzos este proyecto no sería posible.

Alex



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

Título: “Técnicas de aprendizaje automático, aplicadas en un prototipo de videojuego”

Autor: Mora Parra Alex Iván

RESUMEN

El machine learning en la actualidad ha tomado gran relevancia en diversos aspectos de la sociedad, esta se ve envuelta en ámbitos que van de lo más habitual como el rastreo de preferencias en redes sociales hasta actividades más complejas como la conducción automática de vehículos, debido a esta versatilidad este tipo de tecnologías se pueden adaptar a software con propósitos de entretenimiento como es el caso de los videojuegos, los mismo que a su vez han sufrido un incremento exponencial en cuanto a desarrollo y ventas en la última década.

En consecuencia, el presente proyecto de investigación consiste en detallar la incorporación y evaluación de algoritmos de machine learning dentro de un prototipo de videojuego resaltando los diversos procedimientos referentes al desarrollo de estos en base a metodologías ágiles y en técnicas de aprendizaje automatizado incorporados en este tipo de software.

El propósito de incorporar machine learning dentro de un prototipo de videojuego es desarrollar un agente inteligente que esté a la altura del jugador, el cual sea capaz de procesar toda la información recolectada en base a las acciones llevadas a cabo por el jugador y responder en tiempo real a estas, con el fin de representar un comportamiento que refleje acciones llevadas a cabo por un ente pensante y así generar una interacción evolutiva con el usuario.

Palabras Clave: Machine Learning, Videojuegos, Inteligencia Artificial.



ABSTRACT

Machine learning has currently taken great relevance in various aspects of society, this is involved in areas such as the most common as tracking preferences in social networks to more complex activities such as automated driving of vehicles, given this versatility this type of technology can be adapted to software for entertainment purposes as in the case of video games, which in turn have suffered an exponential increase in terms of development and sales in the last decade.

Consequently, this research project consists of detailing the incorporation and evaluation of machine learning algorithms within a video game prototype, highlighting the various procedures related to the development of these based on agile methodologies and automated learning techniques incorporated in this type of software.

The purpose of incorporating machine learning within a video game prototype is to develop an intelligent agent that is at the level of the player, which is able to process all the information collected based on the actions carried out by the player and respond in real time to these, in order to represent a behavior that reflects actions carried out by a thinking entity and thus generate an evolutionary interaction with the user.

Keywords: Machine Learning, Video games, Artificial Intelligence

AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que:

La traducción del resumen al idioma Inglés del artículo científico cuyo título versa: **“TÉCNICAS DE APRENDIZAJE AUTOMÁTICO, APLICADAS EN UN PROTOTIPO DE VIDEOJUEGO.”** presentado por: **Mora Parra Alex Iván**, estudiante de la Carrera de **Ingeniería en Informática y Sistemas Computacionales** perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas** lo realizó bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al **petionario** hacer uso del presente aval para los fines académicos legales.

Latacunga, 21 marzo del 2022

Atentamente,



CENTRO
DE IDIOMAS

Mg. Marco Paúl Beltrán Semblantes

DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0502666514



ÍNDICE

PORTADA	i
DECLARACIÓN DE AUTORÍA	ii
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN	iii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN	iv
AGRADECIMIENTO	v
DEDICATORIA	vi
RESUMEN	vii
ABSTRACT	viii
AVAL DE TRADUCCIÓN	ix
1. INFORMACIÓN GENERAL	xv
2. INTRODUCCIÓN	1
2.1. EL PROBLEMA	1
2.1.1. Situación Problemática	1
2.1.2. Formulación del Problema	2
2.2. OBJETO Y CAMPO DE ACCIÓN	2
2.2.1. Objeto de estudio	2
2.2.2. Campo de acción	2
2.3. BENEFICIARIOS	2
2.4. JUSTIFICACIÓN	2
2.5. HIPÓTESIS	3
2.6. OBJETIVOS	3
2.6.1. General	3
2.6.2. Específicos	4
2.7. SISTEMAS DE TAREAS	4
2.8. FUNDAMENTACIÓN TEÓRICA	5
2.8.1. Inteligencia Artificial	5
2.8.2. Videojuegos	9
1.8.3. Herramientas de Desarrollo	12
1.8.4. Herramientas de Modelado	12



1.8.5.	Lenguajes de Desarrollo	13
3.	MATERIALES Y MÉTODOS.....	15
3.1.	Tipos de Investigación	15
3.1.1.	Investigación Documental	15
3.1.2.	Investigación Cuantitativa	15
3.1.3.	Investigación Cualitativa	15
3.2.	Técnicas e Instrumentos.....	15
3.3.	Metodologías para el desarrollo de Videojuegos.....	16
3.3.1.	Metodología SUM para el desarrollo de videojuegos	16
3.3.2.	Desarrollo de la Inteligencia Artificial.	41
4.	ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	48
4.1.	Resultados de la elaboración del Prototipo.....	48
4.2.	Resultados del Entrenamiento del Agente Inteligente.	49
4.2.1.	Treinta minutos de tiempo de ejecución del entrenamiento.	49
4.2.2.	Una Hora de tiempo de ejecución del entrenamiento.....	51
4.2.3.	Cuatro horas de ejecución del Entrenamiento.	52
4.3.	Resultados del Cuestionario referente al comportamiento del agente inteligente.	54
5.	CONCLUSIONES Y RECOMENDACIONES.	59
5.1.	Conclusiones.....	59
5.2.	Recomendaciones.	60
6.	BIBLIOGRAFÍA.....	61
	ANEXOS	63
	Anexo A. Presupuesto del Prototipo (Puntos de Función)	64
	Anexo B. Tabla comparativa de Metodologías Ágiles.....	69
	Anexo C. Cuestionario referente al comportamiento del agente Inteligente.....	70



ÍNDICE DE FIGURAS.

Figura 1: Cronograma de desarrollo del videojuego.....	18
Figura 2: Cronograma de desarrollo del videojuego.....	18
Figura 3: Actividades llevadas a cabo por el Cliente del proyecto.	19
Figura 4: Actividades llevadas a cabo por el Productor Interno del proyecto.	19
Figura 5: Estados del Videojuego.	22
Figura 6: Primera interfaz del videojuego.....	22
Figura 7: Menú de Mejoras y Estado del Jugador.....	23
Figura 8: Menú del Jugador (Indicador de segundo personaje).....	23
Figura 9: Diagrama de estados del progreso planteado dentro del videojuego.....	25
Figura 10: Concept Art del personaje Dax.	25
Figura 11: Diseño final del Personaje Dax.	26
Figura 12: Concept Art del personaje Def	27
Figura 13: Diseño final del personaje Def.	27
Figura 14: Concept Art del Enemigo01.	28
Figura 15: Diseño Final del Enemigo01.	28
Figura 16: Concept Art del jefe Final (The Boss).....	29
Figura 17: Diseño Final del jefe (The Boss).....	29
Figura 18: Tileset empleado en el desarrollo de los entornos dentro del videojuego.....	32
Figura 19: Tileset empleado en el desarrollo de los entornos dentro del videojuego.....	33
Figura 20: Mapeado de los diversos personajes del videojuego.....	33
Figura 21: Configuración de la interacción con el ambiente del personaje Dax.....	34
Figura 22: Animación del Personaje Dax.	34
Figura 23: Configuración de estados referentes a la animación realizada.	35
Figura 24: Estado del motor de colisiones del personaje Dax.	36
Figura 25: Límites generados para el segundo nivel del videojuego.	37
Figura 26: Gestión del armamento disponible para el jugador.	38
Figura 27: Interfaz de mejora y estado del jugador.....	39
Figura 28: Muestra de Texto al interactuar con objetos del mapa.....	40
Figura 29: Navegación entre Carpetas del Proyecto dentro del CMD.....	42



Figura 30: Creación del ambiente virtual basado en Python.....	42
Figura 31: Activación del entorno virtual.	42
Figura 32: Instalación del Paquete MLagents de unity.	43
Figura 33: Instalación del paquete de PYtorch.	43
Figura 34: Fragmento del código empleado en el componente OnEpisodeBegin().	44
Figura 35: Fragmento del código empleado en el componente CollectObservations().	44
Figura 36: Fragmento del código empleado en el componente OnActionsRecived().	44
Figura 37: Fragmento del código empleado en el componente Heuristic().	44
Figura 38: Fragmento del código empleado en el componente OnTrigger2D().	45
Figura 39: Activación del Ambiente previo a la ejecución del entrenamiento.	46
Figura 40: Escenario de entrenamiento del Agente.	46
Figura 41: Ejecución del comando para el entrenamiento del agente.....	46
Figura 42: Puesta en marcha del algoritmo de Aprendizaje destinado al Agente.....	47
Figura 43: Parámetros establecidos para el entrenamiento.	47
Figura 44: Resultado de PYtorch primera sesión de entrenamiento.	51
Figura 45: Gráfico indicativo de disminución de pérdida de datos.....	52
Figura 46: Índice de perdida de datos.	53
Figura 47: Resultado Primera Pregunta Cuestionario Testers.	54
Figura 48: Resultado segunda Pregunta Cuestionario Testers.	55
Figura 49: Resultado tercera Pregunta Cuestionario Testers.	55
Figura 50: Resultado cuarta Pregunta Cuestionario Testers.	56
Figura 51: Resultado quinta Pregunta Cuestionario Testers.	56
Figura 52: Resultado sexta Pregunta Cuestionario Testers.....	57
Figura 53: Resultado séptima Pregunta Cuestionario Testers.....	58
Figura 54: Octava Pregunta Cuestionario Testers.....	58



ÍNDICE DE TABLAS.

Tabla 1: Beneficiarios.....	2
Tabla 2: Sistemas de tareas en base a los objetivos planteados.	4
Tabla 3: Clasificación de videojuegos según la ESRB.	11
Tabla 4: Técnicas e Instrumentos	15
Tabla 5: Especificaciones del Videojuego.....	20
Tabla 6: Presupuesto planificado para el desarrollo del videojuego.	30
Tabla 7: Estado general del Proyecto.	31
Tabla 8: Estado de la Primera Interacción ejecutada.....	32
Tabla 9: Estado de la Segunda Interacción ejecutada.	35
Tabla 10: Estado de la Tercera Interacción ejecutada.	37
Tabla 11: Estado de la Cuarta Interacción ejecutada.....	39
Tabla 12: Estado de la Quinta Interacción ejecutada.	40
Tabla 13: Ficha de Observación (Tiempo transcurrido 30m)	50
Tabla 14: Ficha de Observación (1 hora)	51
Tabla 15: Ficha de Observación Cuarta hora.	52
Tabla 16: Cálculo de puntos de función.....	65
Tabla 17: Valores del factor de Ajuste.	65
Tabla 18: Características generales del Sistema (GSC's).	66
Tabla 19: Tabla comparativa Metodologías Ágiles.....	69

1. INFORMACIÓN GENERAL

Título:

Técnicas de aprendizaje automático, aplicadas en prototipo de videojuego.

Fecha de inicio:

25 - octubre - 2021

Fecha de finalización:

25 – febrero - 2021

Lugar de ejecución:

Tungurahua/Santiago de Píllaro/Píllaro/Carlos Contreras y Orquídeas

Facultad que auspicia

Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia:

Ingeniería en Informática y Sistemas Computacionales

Proyecto de investigación vinculado:

Proyecto Formativo: Modelamiento de algoritmos para sistemas de información

Equipo de Trabajo:

Tutor de investigación: Ing. Segundo Corrales.

Investigador: Alex Ivan Mora Parra

Área de Conocimiento:

06 Información y Comunicación (TIC) / 061 Información y Comunicación (TIC) / 0613 software y desarrollo y análisis de aplicativos.

Línea de investigación:

Ciencias Informáticas para la modelación de Sistemas de Información a través del desarrollo de software.

Sub líneas de investigación de la Carrera:

Gráficos y Visualización, Introducción a la Interacción Humano- Computador

2. INTRODUCCIÓN

2.1. EL PROBLEMA

2.1.1. Situación Problemática

En la actualidad el ‘Machine learning’ o aprendizaje automático ha tomado gran relevancia en la sociedad actual, ya que esta se ha visto involucrada en diversos aspectos que van de lo más mundano como el rastreo de patrones de comportamiento y preferencias en redes sociales hasta actividades más complejas como la conducción automática de vehículos o el tratamiento de voz en aplicaciones de traducción de idiomas en tiempo real [1].

Esta rama de la inteligencia Artificial al ser tan versátil se ha visto inmersa en distintos tipos de software, tanto en programas que se encuentran en un constante manejo de datos como en aplicaciones destinadas netamente al entretenimiento, como es el caso de los videojuegos, los mismos que han tomado gran relevancia a nivel mundial y en el cual cada día es más común encontrar gente que se decante por este hobby, por ende el desarrollo de este tipo de software ha ido en incremento y consigo la necesidad de profesionales dedicados a estas actividades se ha convertido en prioridad dentro de esta industria que se ha vuelto una de las más lucrativas de los últimos años, además es una de las pioneras en la implementación de tecnologías en auge ya sea el uso de realidad aumentada, inteligencia artificial o simplemente motores gráficos cuyo fin es retratar texturas que asemejen la realidad [2].

Lamentablemente dentro de Sudamérica existen pocas instituciones dedicadas a la enseñanza de los procedimientos necesarios para el desarrollo de videojuegos, esto ha provocado que no se formen profesionales dedicados netamente a estos procedimientos y por ende no se puedan ubicar en esta amplia industria, que siempre está en constante cambio y se adapta a las tendencias tecnológicas actuales como es el caso del aprendizaje automático.

Dentro de Ecuador el desarrollo de videojuegos podría entrar en el campo de lo inexistente o escaso, por no decir nulo, existen algunas Universidades que han considerado incluir el desarrollo de videojuegos como cátedra optativa en las mallas de sus carreras, pero no como una carrera completa, a pesar de que dentro del país posea diversos profesionales especializados en el desarrollo de software y diseño gráfico, ya que en sí un videojuego es la combinación de estos.

2.1.2. Formulación del Problema

¿Es posible integrar el aprendizaje automatizado dentro de un prototipo de videojuego?

2.2. OBJETO Y CAMPO DE ACCIÓN

2.2.1. Objeto de estudio

Inteligencia Artificial aplicada a un prototipo de videojuego.

2.2.2. Campo de acción

120000 matemáticas / 1203 Ciencia de los ordenadores / 1203.04 Inteligencia Artificial

2.3. BENEFICIARIOS

Tabla 1: Beneficiarios.

Beneficiarios Directos	Beneficiarios Indirectos
<ul style="list-style-type: none">● Profesionales Dedicados al desarrollo de software.● Estudiantes de carreras relacionadas al desarrollo de software	<ul style="list-style-type: none">● Video jugadores.

2.4. JUSTIFICACIÓN

La presente propuesta se enfoca en diversos aspectos relacionados al desarrollo de software de entretenimiento (videojuegos), partiendo de socializar los procedimientos y conocimientos necesarios para el desarrollo y puesta en marcha este tipo de software, todos estos conocimientos que fueron adquiridos en el transcurso de estudio en la Carrera de Ingeniería Informática y Sistemas Computacionales.

Otro factor de enfoque es crear interés en el desarrollo de videojuegos, ya que esta rama no es muy explotada en Ecuador, a pesar de que representa una gran fuente de ingresos a nivel mundial y crea muchas fuentes de empleo debido a que se requiere un gran número de profesionales dedicados a diversas ramas no solo a lo que se refiere a desarrollo de software.

Profesionales dedicados a la creación de software dentro de nuestro país no optan por el desarrollo de videojuegos, y los pocos que lo hacen lo toman como pasatiempo, sin buscar un enfoque comercial, por estas razones esta industria es prácticamente inexistente, y esto parte principalmente de diversos factores en los que se puede incluir la falta de ofertas académicas enfocadas en el desarrollo de este tipo de software, aunque existen materias optativas dentro de carreras de desarrollo, pero estas no llegan a abarcar ni los conceptos fundamentales.

Generalmente los videojuegos no se contemplan como un sistema o software convencional, ya que estos poseen sus propias funcionalidades o mecánicas que no necesariamente se encuentran definidas, o se basan en las necesidades de un usuario específico, por lo que este tipo de software incluso posee mecánicas de desarrollo que se alejan en gran medida a los explorados dentro el desarrollo de sistemas más convencionales, como por ejemplo un sistema contable o el desarrollo de un sistema basado en la web. Gracias a esta diversidad la inclusión de herramientas como la inteligencia artificial resultan un proceso que se adapta en gran medida, la misma que incluso puede llegar a facilitar el desarrollo y aceptación del producto. [12]

Principalmente el presente proyecto de investigación busca explorar la aplicación de técnicas de aprendizaje automatizado en un entorno de entretenimiento como lo son los videojuegos con el fin de brindar al usuario una experiencia más evolutiva y compleja en relación a las capacidades del mismo.

2.5. HIPÓTESIS

La implementación de algoritmos de machine learning dentro de videojuegos facilita el desarrollo de agentes inteligentes y crea interacciones más dinámicas con el jugador.

2.6. OBJETIVOS

2.6.1. General

Aplicar técnicas de inteligencia Artificial enfocadas en el aprendizaje automatizado dentro del desarrollo de un prototipo de videojuego empleando el motor gráfico Unity.

2.6.2. Específicos

- Consultar de fuentes bibliográficas todo lo relevante acerca de aprendizaje automatizado aplicado al desarrollo de videojuego en base a investigaciones de fuentes primarias de información.
- Analizar diversas metodologías de desarrollo ágil mediante su comparativa para determinar cuál de estas es la idónea para el desarrollo de videojuegos.
- Evaluar la eficacia del algoritmo de aprendizaje automatizado incorporado dentro del prototipo de videojuego a través de instrumentos de recolección de datos.

2.7. SISTEMAS DE TAREAS

Tabla 2: Sistemas de tareas en base a los objetivos planteados.

OBJETIVO ESPECÍFICO	ACTIVIDAD	RESULTADO	TECNICAS E INSTRUMENTOS
Consultar de fuentes bibliográficas todo lo relevante acerca de aprendizaje automatizado aplicado al desarrollo de videojuego en base a investigaciones de fuentes primarias de información.	Identificar antecedentes investigativos. Comprar propuestas de diversos autores	Marco teórico	Introducción. Problemática Fichas bibliográficas

<p>Analizar diversas metodologías de desarrollo ágil mediante su comparativa para determinar cuál de estas es la idónea para el desarrollo de videojuegos.</p>	<p>Tablas comparativas Diseño del videojuego Desarrollo de los módulos videojuego Pruebas de funcionamiento</p>	<p>Pantalla de menú. Gameplay (Tutorial y juego en sí). Reporte de testers</p>	<ul style="list-style-type: none"> ● Checklist. ● Técnicas de diseño 2d ● Resultados de los testers
<p>Evaluar la eficacia del algoritmo de aprendizaje automatizado incorporado dentro del prototipo de videojuego a través de instrumentos de recolección de datos.</p>	<p>Cuestionarios sobre el funcionamiento del videojuego Comprobación del correcto funcionamiento del videojuego</p>	<p>Pruebas de testing</p>	<ul style="list-style-type: none"> ● Checklist ● Entrevistas ● observación

2.8. FUNDAMENTACIÓN TEÓRICA

2.8.1. Inteligencia Artificial

2.8.1.1. Antecedentes

Los primeros trabajos que pueden ya considerarse como el embrión de la Inteligencia Artificial moderna aparecen en la década de los 40, aunque no sería hasta 1950 cuando realmente estos estudios y propuestas consiguen una verdadera repercusión gracias al artículo “Computing Machinery and Intelligence”, escrito por Alan Turing, uno de los padres de la IA, y publicado en el volumen 59 de la revista “Mind”. En este artículo se profundiza sobre la posibilidad de que una máquina pueda imitar el comportamiento de la mente humana. Además, se propone el archiconocido Test de Turing, orientado a demostrar si una determinada máquina es inteligente o no. Ese artículo es, probablemente, el catalizador que aglutina el conocimiento acumulado con anterioridad en otras disciplinas como la lógica y la algorítmica, llevándolo al nacimiento

de una nueva ciencia. Ciencia que no sería oficialmente bautizada hasta 1956, durante la conferencia de Dartmouth. [3]

Indiscutiblemente, Alan Turing fue uno de los mayores contribuyentes al consenso teórico post 1950, a tal punto, que muchos lo consideran el padre de la Inteligencia Artificial. Este matemático-filósofo aportó dos trabajos seminales, la máquina de Turing y el Test de Turing (1936 y 1950, respectivamente), que pueden considerarse hitos fundacionales de la Ingeniería Informática. En efecto, el primero define el carácter algorítmico de ‘computación’, mientras que el segundo precisa de manera sistemática un método para acopiar evidencia de vida mental inteligente en computadores programados. [4]

Por lo expresado se puede determinar que el nacimiento de la inteligencia artificial como concepto parte de la idea de Alan Turing y su razonamiento sobre a qué se puede considerar una máquina capaz de simular el comportamiento humano y que implica que estas tengan la capacidad de razonar y adoptar estrategias para la resolución de un determinado problema.

2.8.1.2. Definición

Por extraño que pueda parecer, lo cierto es que no hay un consenso entre científicos e ingenieros sobre lo que es la Inteligencia Artificial, y mucho menos se ha llegado a una definición exacta y concisa que nos permita dirimir qué programas son o no inteligentes. El problema es que ni siquiera tenemos la certeza de que seamos capaces de definir qué es la inteligencia (no artificial). [5]

La ciencia llamada inteligencia artificial (IA, 1956), se ha dado por objetivo el estudio y el análisis del comportamiento humano. De esta manera, las aplicaciones de la IA se sitúan principalmente en la simulación de actividades intelectuales del hombre. Es decir, imitar por medio de máquinas, normalmente electrónicas, tantas actividades mentales como sea posible, y quizás llegar a mejorar las capacidades humanas en estos aspectos. [6]

De acuerdo a lo expresado anteriormente se concluye que el concepto empleado para la definición de Inteligencia artificial varía dependiendo del campo de aplicación de esta, pero en general se puede definir como una exploración del comportamiento humano y cómo se puede replicar estos dentro de una máquina.

2.8.1.3. Campos de aplicación

Debido a la naturaleza de este tipo de algoritmos los campos de aplicación varían dependiendo de del propósito, por lo general se abarcan cinco áreas en las que se pueden englobar la aplicación de la Inteligencia Artificial, las mismas que se detallan a continuación:

Tratamiento de lenguajes naturales: en este campo se pueden englobar aplicaciones que realicen traducciones entre idiomas, interfaces hombre-máquina que permitan interrogar una base de datos o dar órdenes a un sistema operativo, etc., de manera que la comunicación sea más amigable con el usuario.

Sistemas expertos: En esta área están englobados aquellos sistemas donde la experiencia de personal cualificado se incorpora a dichos sistemas para corregir deducciones más cercanas a la realidad.

Robótica: Navegación de robots móviles, control de brazos de robots, ensamblaje de piezas, etc.

Problemas de percepción: visión y habla. Reconocimiento de objetos y del habla, detección de defectos en piezas por medio de visión, apoyo en diagnósticos médicos, etc.

Aprendizaje: Modelización de conductas para su posterior implementación en computadoras.
[1]

Como se menciona las áreas de aplicación permiten determinar qué tipo de algoritmo se puede emplear para solucionar un problema en concreto o simplemente en que área se pretende realizar la exploración de resultados al aplicar técnicas de Inteligencia Artificial.

2.8.1.4. Inteligencia Artificial aplicada a videojuegos

2.8.1.4.1. Redes Neurales

El concepto de una red neuronal en informática parte de la comparativa entre las propias neuronas, las cuales trabajan en base al transporte y procesamiento de información (Sinapsis). Para el caso de una Red Neural en informática se describe una red neural como “Redes Neuronales, buscan imitar a la neurona en términos computacionales. Se hace esto ya que se

considera que la neurona es la base de la capacidad de aprendizaje. No obstante, la emulación de la neurona no es perfecta, a la fecha se investigan algunas cualidades importantes como la Computación Distribuida para tolerar Ruido (Distorsión) en la entrada y para el aprendizaje. Cada conexión Neuronal computacional posee un peso numérico que limita la fuerza con la que el impulso sale de la neurona, de no ser así la información crecería demasiado con resultados imprevisibles. Las Redes Neuronales reconocen solo dos estados: 0 (No hay impulso) y 1 (Si lo hay), esto las asemeja a las compuertas lógicas de cualquier tipo: AND, XOR, OR” [1]

2.8.1.4.2. Machine learning

El Machine Learning o aprendizaje automático consta de diversos métodos científicos cuyo propósito involucra la utilización de diversos ordenadores u otros dispositivos con la capacidad computacional necesaria para llevar a cabo diversas operaciones con el fin de extraer diversos patrones relacionados a datos obtenidos previamente para posteriormente predecir comportamientos y en base a esto tomar una serie de decisiones para mejorar en gran medida los resultados esperados en la tarea asignada. [7]

2.8.1.4.3. ML Agents

El kit de herramientas de agentes de aprendizaje automático de Unity (ML-Agents) es un proyecto de código abierto que permite que los juegos y las simulaciones sirvan como entornos para entrenar agentes inteligentes. Proporcionamos implementaciones (basadas en PYtorch) de algoritmos de última generación para permitir a los desarrolladores de juegos y aficionados entrenar fácilmente a agentes inteligentes para juegos 2D, 3D y VR/AR. Los investigadores también pueden usar la API de Python fácil de usar proporcionada para capacitar a los agentes mediante el aprendizaje por refuerzo, el aprendizaje por imitación, la neuro evolución o cualquier otro método. Estos agentes capacitados se pueden usar para múltiples propósitos, incluido el control del comportamiento de los NPC (en una variedad de entornos, como multiagente y adversario), pruebas automatizadas de compilaciones de juegos y evaluación de diferentes decisiones de diseño de juegos antes del lanzamiento. El ML-Agents Toolkit es mutuamente beneficioso tanto para los desarrolladores de juegos como para los investigadores de IA, ya que proporciona una plataforma central donde los avances en IA pueden evaluarse en los entornos ricos de Unity y luego ponerse a disposición de las comunidades más amplias de investigación y desarrollo de juegos. [8]

2.8.2. Videojuegos

Existen diversas conceptualizaciones para determinar que abarca el término videojuego, y esta varía dependiendo del punto de vista del análisis o el campo de estudio, por ejemplo Caillois [9] determina que juego abarca “una actividad que es esencialmente libre/voluntaria, separada en el tiempo y el espacio, incierta e improductiva que se rige por las reglas de la fantasía”, partiendo de esto se puede decir que un videojuego representa una actividad que se desarrolla en base al uso de medios electrónicos, con el fin de llevar a cabo una actividad que no necesariamente implica mecánicas que reflejen comportamientos reales, en la mayoría de los casos estos se derivan de conceptos apegados a la ficción.

1.8.2.1. Tipos de videojuegos

Generalmente los videojuegos se pueden agrupar en distintos tipos que engloban características únicas lo que permite una clasificación que se basa en factores como gameplay y temática, por lo general se pueden agrupar en grupos como los mencionados a continuación: [10]

- **Juegos de Lucha:** Los cuales se pueden considerar los más conocidos, se basa en combates entre por lo general dos personajes, este tipo de juego abarca por lo general contenido más violento.
- **Beat'em up/Shoot'em up:** famosos por las arcadas su gameplay se basa en el concepto de uno contra todos, por lo general se plantea la idea del personaje invencible con una misión, el beat'em up no emplea armas de fuego, a diferencia que los shoot'em up que sus mecánicas de juego se basan en el uso de armas.
- **Plataforma:** Este tipo de videojuegos se caracterizan por presentar al video jugador una serie de obstáculos y un objetivo que debe alcanzar empleando el entorno para seguir avanzando hasta su cometido, un ejemplo muy famoso es Mario Bros.
- **Simuladores:** uno de los géneros más populares en la actualidad, aunque su premisa sea muy sencilla el atractivo radica en el poder que se le otorga al usuario para poder crear o llevar una actividad simulando aspectos de la vida real, que abarcan profesiones, historia, relaciones o creación de civilizaciones, The Sims es un gran exponente en este tipo de videojuegos.







- **Deportivos:** videojuegos enfocados en representar las mecánicas y conceptos de deportes, ya sea fútbol, básquet, golf, etc., por lo general este tipo de videojuegos abarcan grandes masas por sus escasas de violencia y su grado de entretenimiento.
- **Estrategia:** juegos que imponen retos a los usuarios y en los cual la necesidad de plantear diversas estrategias para resolver el problema.
- **Aventura:** por lo general son videojuegos de mundo abierto donde el usuario explora los alrededores en busca de secretos o simplemente para cumplir su cometido.
- **Rol:** juegos que se basan en manipular las estadísticas y características del personaje diseñado o elegido por el usuario, por lo general abarcan mundos de fantasía.
- **BattleRoyal:** Es un género relativamente nuevo impulsado por el juego online, donde una serie de jugadores alrededor del mundo se enfrentan en un mapa y cuyo objetivo es ser el último superviviente, por lo general se enfrentan entre 50 y 100 jugadores simultáneos.
- **Shooters:** este tipo abarcan la mayoría de juegos de guerra o juegos cuya mecánica se basa en el uso de armas de juego.

1.8.2.2. Clasificación de los videojuegos

Todo videojuego debe apearse al modelo de clasificación del *Entertainment Software Rating Board*, el cual es un sistema de clasificación de contenido de videojuegos establecido en el año 1994, el cual sirve de guía sobre el contenido presentado dentro de un videojuego, este sistema está enfocado a los padres y consumidores para determinar qué temáticas abordan determinados videojuegos.

Existen 6 clasificaciones que abarcan parámetros de edad y tipo de contenido incluido dentro de un videojuego. [11]

Tabla 3: Clasificación de videojuegos según la ESRB.

Símbolo	Nombre	Descripción
	Todos	El contenido por lo general es apto para todas las edades. Puede que contenga una cantidad mínima de violencia.
	Todos +10	El contenido por lo general es apto para personas de 10 años o más. Puede que contenga más violencia de caricatura, de fantasía o ligera, lenguaje moderado o temas mínimamente provocativos
	Adolescentes	El contenido por lo general es apto para personas de 13 años o más. Puede que contenga violencia, temas insinuantes, humor grosero, mínima cantidad de sangre, apuestas simuladas o uso poco frecuente de lenguaje fuerte
	Maduro +17	El contenido por lo general es apto para personas de 17 años o más. Puede que contenga violencia intensa, derramamiento de sangre, contenido sexual o lenguaje fuerte.
	Adultos Únicamente +18	El contenido es apto solo para adultos de 18 años o más. Puede que incluya escenas prolongadas de violencia intensa, contenido sexual gráfico o apuestas con moneda real.
	Clasificación Pendiente	No se asignado una calificación final de ESRB. Solo aparece en la publicidad, marketing y materiales promocionales relacionados a un juego físico (por ejemplo, el que viene cerrado en caja) que se espera que tenga una clasificación de ESRB y debe ser sustituida por una clasificación de ESRB y debe ser sustituida por una clasificación de juego una vez que se haya asignado.

Debido a la naturaleza del proyecto la clasificación entraría en un segundo plano ya que el enfoque del proyecto no tiene contemplado la comercialización a gran escala.

1.8.3. Herramientas de Desarrollo

1.8.3.1. Unity

Unity es una herramienta de desarrollo de videojuegos creada por la empresa Unity Technologies. Unity es una herramienta que no engloba únicamente motores para el renderizado de imágenes, de físicas de 2D/3D, de audio, de animaciones y otros motores, sino que engloba además herramientas de *Networking* para multijugador, herramientas de navegación NavMesh para Inteligencia Artificial o soporte de Realidad Virtual. [12]

Principalmente en motor gráfico de Unity trabaja con el lenguaje de programación C# el cual se basa en el paradigma orientado a objetos, creado por Microsoft y cuya estructura adopta conceptos de lenguajes como C, C ++, Java y Visual Basic.

1.8.3.2. Unreal Engine

Unreal Engine, es uno de los motores de juego más populares y usados del momento, perteneciente a la compañía Epic Games. Su funcionamiento se basa en código C ++ y su primera versión se creó en 1998, aunque hasta 2015 no estuvo disponible de forma gratuita y pública.

Unreal Engine es un entorno de desarrollo que incluye todas las herramientas necesarias para construir un juego o simulación, como editor de vídeo, estudio de sonido, código o renderización de animaciones, entre otras características. [13]

1.8.4. Herramientas de Modelado

1.8.4.1. Adobe Illustrator

Adobe Illustrator, es una de las plataformas más utilizadas por los diseñadores al momento de crear sus piezas gráficas. Además, es catalogado uno de los programas que ha permanecido con más tiempo vigente, ya que lleva más de 20 años en el rubro de diseño y arte digital. Está dentro Adobe Systems, y viene a ser uno de los pioneros en el diseño vectorial. Su plataforma cuenta con diferentes herramientas para elaborar distintas piezas gráficas en una mesa de trabajo. [14]

1.8.4.2. Adobe Animate

Adobe Animate (previamente conocido como Adobe Flash Professional, Macromedia Flash, y FutureSplash Animator) es uno de los programas más conocidos de la casa Adobe, así como sus hermanos Adobe Illustrator y Adobe Photoshop. Se trata de una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código a través de un lenguaje de scripting llamado ActionScript. Animate es una investigación de animación que trabaja sobre "fotogramas" y está destinada a la producción y entrega de contenido interactivo para diferentes audiencias de todo el planeta sin importar un mínimo la plataforma. Es en nuestros días desarrollado y comercializado por Adobe Systems Incorporated y es parte integrante de la familia Adobe Creative Suite [15]

1.8.4.3. Adobe Photoshop

Photoshop es un programa de edición de imágenes desarrollado por Adobe Systems Incorporated, que fue creado en 1987 por un estudiante de la universidad de Michigan llamado Thomas Knoll, bajo el nombre de Display. Photoshop sirve para editar o crear imágenes con el uso de varias herramientas como pinceles, lápices, rellenos, formas, textos, relieves o efectos, entre otros. El programa soporta toda clase de formatos de imagen como JPG, GIF, PNG, PDF, TIFF... etc. Para el diseñador gráfico, es muy útil para crear logotipos, flyers, tarjetas y otros. Para el ilustrador, sirve para crear dibujos digitales sin necesidad de utilizar una imagen previa. [16]

1.8.4.4. Aseprite

Aseprite es un programa enfocado en la creación de imágenes 2d basados en el estilo artísticos de PixelArt, estilo retro enfocado en las eras de 8 bits y 16 bits. Aparte posee herramientas de animación basadas en la animación por cuadros. Generalmente este programa se utiliza en el ámbito de la creación de videojuegos. [17]

1.8.5. Lenguajes de Desarrollo

1.8.5.1. C#

C # (pronunciado "See Sharp") es un lenguaje de programación moderno, orientado a objetos y con seguridad de tipos. C # permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en el ecosistema .NET. C # Tiene sus raíces en la familia de lenguajes C y será inmediatamente familiar para los programadores de C, C ++, Java y JavaScript.

C # es un lenguaje de programación orientado a objetos y orientado a componentes. C # proporciona construcciones de lenguaje para admitir directamente estos conceptos, lo que convierte a C # en un lenguaje natural en el que crear y utilizar componentes de software. Desde su origen, C # ha agregado funciones para admitir nuevas cargas de trabajo y prácticas emergentes de diseño de software. [18]

1.8.5.2. C++

C++ es un lenguaje de programación que proviene de la extensión del lenguaje C para que pudiese manipular objetos. Fue diseñado a mediados de los años 80 por el danés Bjarne Stroustrup. Su intención fue la de extender el lenguaje de programación C (con mucho éxito en ese momento) para que tuviese los mecanismos necesarios para manipular objetos. Por lo tanto, C++ contiene los paradigmas de la programación estructurada y orientada a objetos, por lo que se le conoce como un lenguaje de programación multiparadigma.

Las principales ventajas de programar en C++ son:

Alto rendimiento: Es una de sus principales características, el alto rendimiento que ofrece. Esto es debido a que puede hacer llamadas directas al sistema operativo, es un lenguaje compilado para cada plataforma, posee gran variedad de parámetros de optimización y se integra de forma directa con el lenguaje ensamblador.

Lenguaje actualizado: A pesar de que ya tiene muchos años, el lenguaje se ha ido actualizando, permitiendo crear, relacionar y operar con datos complejos y ha implementado múltiples patrones de diseño.

Multiplataforma

Extendido: C y C++ están muy extendidos. Casi cualquier programa o sistema están escritos o tienen alguna parte escrita en estos lenguajes (desde un navegador web hasta el propio sistema operativo).

Las principales desventajas de C ++ es que se trata de un lenguaje muy amplio (con muchos años y muchas líneas de código), tiene que tener una compilación por plataforma y su depuración se complica debido a los errores que surgen. Además, el manejo de librerías es más complicado que otros lenguajes como Java o .Net y su curva de aprendizaje muy alta. [19]

3. MATERIALES Y MÉTODOS

3.1. TIPOS DE INVESTIGACIÓN

3.1.1. Investigación Documental

Dado a que una de los principales objetivos del proyecto es servir de guía para que profesionales o estudiantes opten por el desarrollo de videojuegos, la recolección de material bibliográfico es una parte esencial para cumplir con esta expectativa, además al ser un tema poco explorado dentro de la región el material de apoyo es escaso y por ende difícil de encontrar.

3.1.2. Investigación Cuantitativa

Se opta por este tipo de investigación debido a las diversas herramientas que se pretende emplear dentro del proyecto, esto hace que sea necesario diversos instrumentos que arrojen datos referentes a la aceptación del producto, rendimiento de las propias herramientas, eficacia de los algoritmos planteados para el aprendizaje automatizado que se pretende emplear, entre otros datos que se pueden recolectar mediante la observación de resultados.

3.1.3. Investigación Cualitativa

Debido a la naturaleza del proyecto y dado a que la mayoría de videojuegos basan sus criterios de evaluación en base a trama, diseño y aceptación del público, este tipo de investigación cumpliría el rol de determinar la parte de satisfacción del cliente y de esta se pueden obtener datos que aporten para futuras entregas o para mejorar el producto desarrollado.

3.2. TÉCNICAS E INSTRUMENTOS

Tabla 4: Técnicas e Instrumentos

No.	TÉCNICAS	INSTRUMENTOS
1	Revisión Bibliográfica	Fichas bibliográficas.
2	Observación	Ficha de observación
3	Entrevistas	Cuestionario

3.3. METODOLOGÍAS PARA EL DESARROLLO DE VIDEOJUEGOS

El desarrollo de videojuegos al ser muy versátil y complejo no necesariamente sigue una metodología preestablecida, dado a que abarca diversas áreas de desarrollo, por lo que tratar de englobar una única metodología es casi imposible. A pesar de esta situación en los últimos años se ha propuesto una metodología que adopta técnicas de desarrollo ágil en base a la experiencia de las empresas dedicadas al desarrollo de videojuegos.

3.3.1. Metodología SUM para el desarrollo de videojuegos

Es una de las pocas metodologías enfocadas netamente en el desarrollo de videojuegos, esta se desprende de SCRUM y adopta diversos puntos como grupos de desarrollo pequeños y etapas pequeñas de desarrollo, el principal objetivo de esta metodología es generar videojuegos de calidad en base al tiempo y costo, tomando en cuenta la mejora continua para el mejoramiento de la eficacia y eficiencia gestionando de manera óptima los recursos y riesgos del proyecto con el fin de alcanzar una alta productividad del equipo.

La elección de esta metodología recae en el análisis realizado en base a una comparativa realizada entre las metodologías SCRUM, XP, y SUM (Anexo B).

Este proceso se lleva a cabo dentro de 5 fases las mismas que son:

- Concepto.
- Planificación.
- Elaboración.
- Beta.
- Cierre.
- Evaluación de riesgos.

3.3.1.1. Fase de conceptualización

Dentro de la fase de conceptualización se plantea todo lo referente a la idea del proyecto, esto incluye diversos aspectos, en los que consta la narrativa del juego, aspectos sobre las mecánicas, posibles alcances, en esta etapa es posible encontrar conceptos tempranos tanto de personajes como de escenarios.

La fase de conceptualización dentro del desarrollo de un videojuego es el pilar en donde se detalla a ser posible todo lo referente a la idea planteada, es por esto que la duración de esta fase no se la puede definir dentro de un marco de tiempo, debido a que las ideas no se pueden planificar o programar, por lo general esta fase puede llegar a desarrollarse en intervalos de tiempo no especificados, que van desde los 3 meses hasta los 2 años.

En cuanto al concepto planteado para el presente proyecto, se limitó a la creación de un prototipo de un videojuego que se apegue al género RPG de acción, en el cual se ponga en práctica el uso de tecnologías de aprendizaje automático por parte de un agente inteligente, este rol se lo asignó a los enemigos, con el fin de generar una interacción Humano- Computador evolutiva.

Basados en el concepto general planteado, es necesario profundizar en la narrativa que va a poseer el prototipo, misma que servirá para detallar las diversas mecánicas asociadas a la jugabilidad e interacciones por parte de los usuarios. Dado a que el tiempo asignado para la presente investigación es reducido el concepto y la idea se han adaptado para presentar una experiencia tradicional en relación a los videojuegos 2D de 16 bits cuya jugabilidad está enfocado a ser dinámica y fácil de entender.

3.3.1.2. Fase de planificación

3.3.1.2.1. Planificación Administrativa

Dentro de la fase de planificación se establecen diversos hitos que se deben cumplir con las siguientes fases del desarrollo, dentro de la planificación administrativa se incluyen los siguientes aspectos:

3.3.1.2.1.2. Grupos de desarrollo

Los grupos de desarrollo abarcan distintos departamentos que se encargan de acciones concretas dentro del proyecto. La división de estos grupos varía dependiendo del número de personas, pero generalmente mantienen la siguiente estructura:

- **Cliente:** Representa los intereses de todos aquellos que se ven materialmente afectados por los resultados del proyecto.



Figura 3: Actividades llevadas a cabo por el Cliente del proyecto.

- **Productor Interno:** Encargado de guiar el desarrollo, de promover buenas prácticas y de interactuar con los clientes.



Figura 4: Actividades llevadas a cabo por el Productor Interno del proyecto.

- **Equipo de desarrollo:** Representa a todos los miembros encargados de construir el videojuego. Generalmente está conformado por los siguientes miembros:
 - **Diseñador de juego:** Diseña el gameplay, historia, ambientación, personajes, niveles y todos los elementos que hacen a la experiencia del jugador.
 - **Programador:** Diseña, implementa y verifica el software que compone al juego.
 - **Artista Sonoro:** Encargado de crear todos los efectos de sonido y música del juego.
 - **Artista gráfico:** Crea todo el contenido gráfico del juego, incluyendo el arte de concepto, arte 2d, modelos 3d, animaciones y texturas.

3.3.1.2.2. Especificación del Videojuego

Tabla 5: Especificaciones del Videojuego.

Especificación del Videojuego	
Campo	Concepto
Concepto	
Título	Dax & Def vs the IA
Diseñador	Alex Iván Mora Parra
Género	RPJ de acción
Plataforma	PC
Versión	Prototipo funcional
Sinopsis de Jugabilidad y Contenido	Dos hermanos buenos para nada se ven involucrados en la guerra entre humanos y máquinas controladas por una poderosa Inteligencia Artificial que ha dejado una sociedad sin tecnología ni héroes.
Licencia	Idea Original
Mecánica	<ul style="list-style-type: none"> ● Combate Melé ● Sistemas de mejora de equipo (Armas) ● Sistemas de obtención de experiencia ● Jugabilidad basada en el uso de dos personajes con sus propias mecánicas. ● Exploración de Mapas semi abiertos. ● Sistema monetario. ● Interacción con NPC.
Tecnología	<ul style="list-style-type: none"> ● Hardware empleado en el desarrollo. <ul style="list-style-type: none"> ○ Tarjeta NVIDIA GTX 1050 ○ Procesador Intel Core i7 8th Generación. ○ Memoria: 12 GB de RAM. ● Software empleado en el desarrollo. <ul style="list-style-type: none"> ○ Windows 10 pro. ○ Untiy versión 2020.3.13f1 ○ Ml agents 2.2.1

	<ul style="list-style-type: none"> ○ Visual Studio 2019. ○ Aseprite v1.1.1. ○ Adobe Illustrator 2021 ○ Adobe Photoshop 2021 ○ Adobe animate 2021. ○ DirectX versión 11 ○ Python 3.9.4 ○ GitHub ○ Git
Público	Jóvenes (Aún no clasificado)
Control de versiones	
El control de versiones se lleva a cabo mediante GitHub en el cual se ha destinado una rama principal que se actualiza después de comprobar la funcionalidad de una característica o función.	
Visión general del Juego	
El juego está basado en una jugabilidad evolutiva de aprendizaje, en base a las acciones realizadas por el usuario, al ser un prototipo este alcance será limitado por el tiempo de desarrollo y las herramientas necesarias para llevar a cabo la meta.	
Mecánica del Juego	
Cámara	Cámara enfocada en la parte superior del escenario, seguimiento del jugador, no modificable por el usuario
Periféricos	Teclado, mouse
Controles	<ul style="list-style-type: none"> ● Teclas WASD para el movimiento ● Tecla Espacio para Atacar ● Puntero del mouse para interactuar con el menú
Puntaje	<ul style="list-style-type: none"> ● Los puntajes variarán dependiendo del enemigo, esto se traduce en el incremento de la experiencia. ● La cantidad de monedas recolectadas al igual que la experiencia varía.
Guardar/Cargar	<ul style="list-style-type: none"> ● Guardado el estado entre pantallas.

- Guardado manual no contemplado.
- Progreso se borra en cuanto el usuario pierde.

Estados del Juego

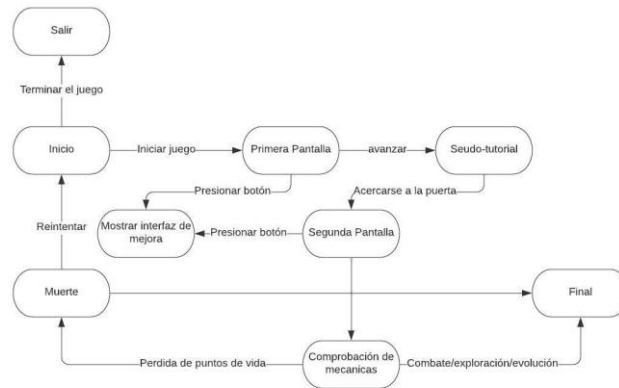


Figura 5: Estados del Videojuego.

Interfaces

Nombre de la Pantalla	Pantalla de Inicio
Descripción	Es la primera interacción con el ambiente del juego, sirve de puente para la interacción con el prototipo de las mecánicas.
Estado del juego	Inicio del juego.
Imagen	

Figura 6: Primera interfaz del videojuego.

Nombre de la Pantalla	Menú de estado y mejora del personaje
Descripción	Interfaz dedicada a mostrar el estado del personaje junto a las opciones de mejora de armas y cambio de personaje.
Estado del juego	<ul style="list-style-type: none"> • Juego en ejecución • Disponible mientras se juega

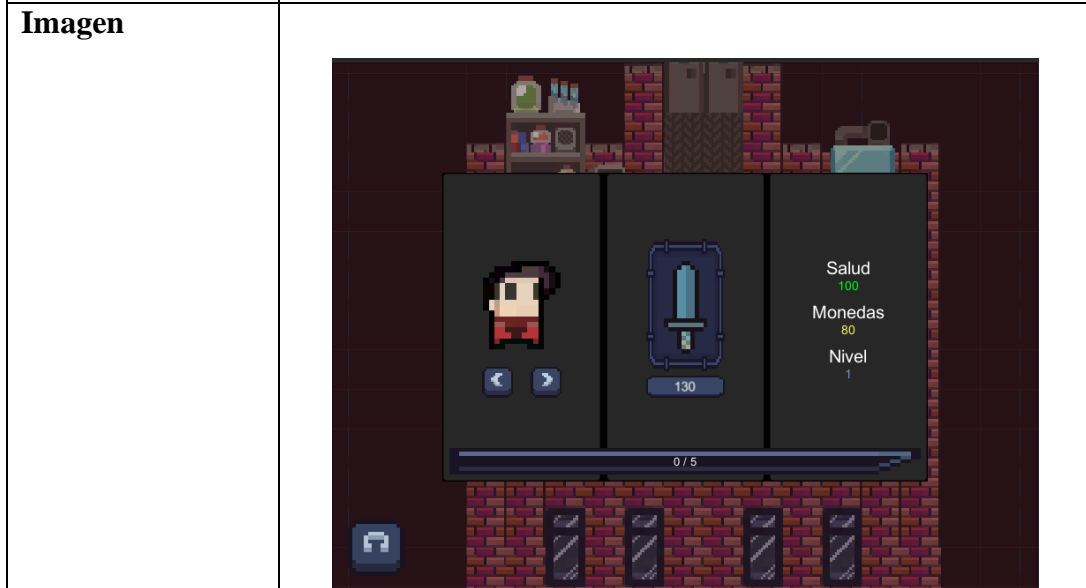


Figura 7: Menú de Mejoras y Estado del Jugador.



Figura 8: Menú del Jugador (Indicador de segundo personaje)

Niveles	
Título del Nivel	Nivel 1 de prueba de prototipo
Encuentro	Al iniciar el juego, se puede regresar en cualquier momento a este nivel.
Descripción	Primer nivel del juego, pseudo tutorial.
Objetivos	Servir de puente entre el prototipo de funcionamiento, sin objetivos en específico.
Progreso	Acercarse a la puerta para pasar al siguiente nivel.
Enemigos	Ninguno
Ítems	<ul style="list-style-type: none"> ● Hamburguesa para recuperar la vida.
Personajes	NPC de prueba de dialogo.
Título del Nivel	Nivel 2 de prueba de prototipo.
Encuentro	Disponible al pasar del primer nivel de prueba
Descripción	Nivel enfocado en mostrar las características planteadas en la conceptualización del videojuego.
Objetivos	Verificar funcionalidades del prototipo.
Progreso	<ul style="list-style-type: none"> ● Explorar todo el mapa ● Interactuar con los enemigos. ● Interactuar con el entorno.
Enemigos	<ul style="list-style-type: none"> ● Ninguno
Ítems	<ul style="list-style-type: none"> ● Cofres. ● Cajas
Personajes	<ul style="list-style-type: none"> ● Jugador ● Enemigos
Progreso del Juego	

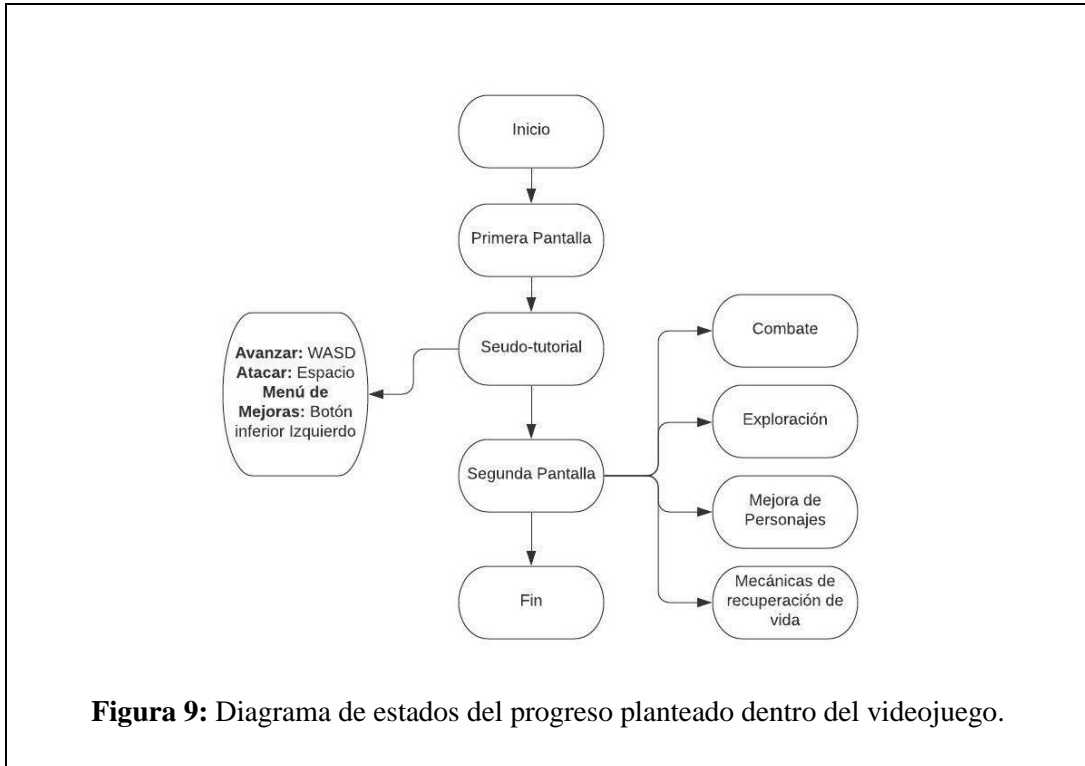


Figura 9: Diagrama de estados del progreso planteado dentro del videojuego.

Personajes	
Nombre del Personaje	Dax
Descripción	El hermano mayor, el músculo del equipo un muchacho sin ninguna habilidad en especial a excepción por su gran conocimiento en cómics y cine. De niño aprendió junto a su hermana el mundo del combate con espadas, pero lo dejó a las 5 clases al igual que su hermana.
Imagen	<p>Concept Art Totalmente Right? Miso Pixel Art? 16bits o 8bits? Rojo Marca distintiva Enojado KO</p>

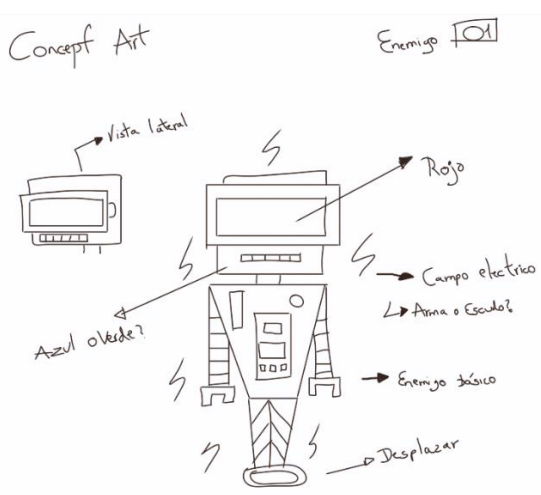

Figura 10: Concept Art del personaje Dax.

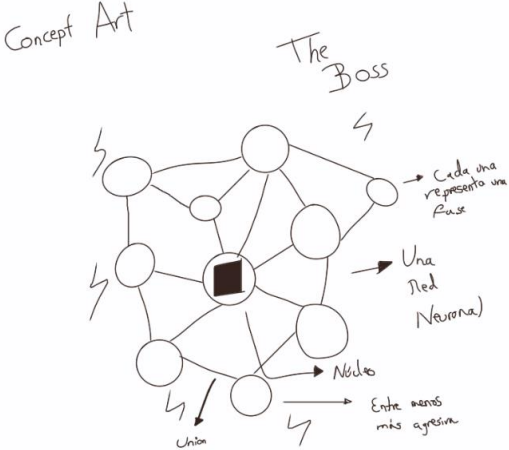
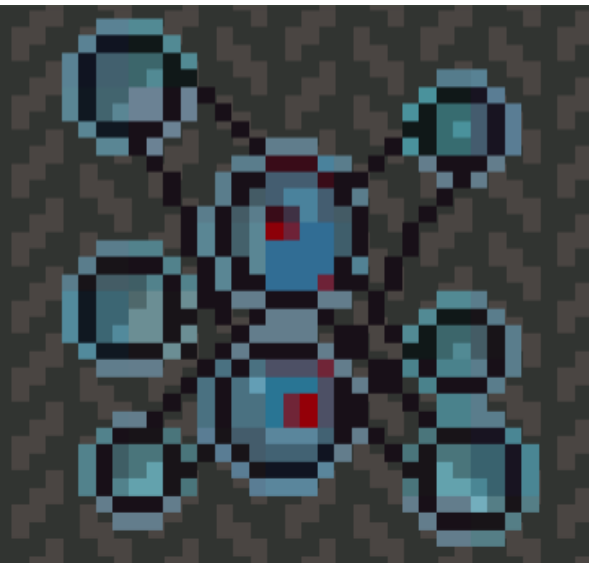


Figura 11: Diseño final del Personaje Dax.

Concepto	El personaje en apariencia refleja un diseño común sin ningún elemento que resalte, dado a que ese es el propósito, se trata de un personaje común y corriente que por coincidencias de la vida se ve envuelto en una guerra.
Encuentro	Está disponible al principio del juego y en el desarrollo del mismo.
Habilidades	<ul style="list-style-type: none"> ● Combate con espadas ● Destrucción de obstáculos.
Armas	<ul style="list-style-type: none"> ● Dagas. ● Espadas cortas. ● Claymore. ● Espadas de todo tipo.
Ítems	Ninguno
Nombre del Personaje	Def
Descripción	Hermana menor de Dax, el cerebro del equipo, pero igual de fuerte que su hermano, una muchacha con conocimientos básicos de programación y una gran fascinación por la tecnología.

<p>Imagen</p>	<div data-bbox="667 280 1248 750" data-label="Image"> </div> <p>Figura 12: Concept Art del personaje Def</p> <div data-bbox="742 869 1171 1346" data-label="Image"> </div> <p>Figura 13: Diseño final del personaje Def.</p>
<p>Concepto</p>	<p>El personaje en apariencia refleja un diseño de una joven desinteresada en llamar la atención del resto, una mujer de apariencia común y corriente.</p>
<p>Encuentro</p>	<p>Disponible al principio del juego y en el desarrollo del mismo</p>
<p>Habilidades</p>	<ul style="list-style-type: none"> ● Combate con espadas. ● Uso de mecanismos. ● Uso de Vehículos.
<p>Armas</p>	<ul style="list-style-type: none"> ● Dagas. ● Espadas cortas.

	<ul style="list-style-type: none"> • Claymore. <p>Espadas de todo tipo.</p>
Ítems	Ninguno
Enemigos	
Nombre	Enemigo 01
Descripción	La creación de una malévola Inteligencia Artificial, de reacción rápida y ataque considerable.
Encuentro	A partir de la segunda pantalla.
Imagen	<div style="text-align: center;">  <p>Figura 14: Concept Art del Enemigo01.</p> </div> <div style="text-align: center; margin-top: 20px;">  <p>Figura 15: Diseño Final del Enemigo01.</p> </div>
Habilidades	<ul style="list-style-type: none"> • Ataque.

	<ul style="list-style-type: none"> • Persecución del jugador.
Armas	<ul style="list-style-type: none"> • Campo magnético que provoca daño.
Ítems	Ninguno
Nombre	The Boss
Descripción	La Inteligencia Artificial que sumió a la humanidad a una era sin tecnología y sin sus héroes.
Encuentro	Jefe final del videojuego.
Imagen	<div style="text-align: center;">  <p>Figura 16: Concept Art del jefe Final (The Boss)</p> </div> <div style="text-align: center; margin-top: 20px;">  <p>Figura 17: Diseño Final del jefe (The Boss)</p> </div>

Habilidades	<ul style="list-style-type: none"> ● Desprender sus nodos para ataques independientes. ● Crear enemigos. ● Teletransportarse por el escenario. 																																																										
Armas	Rayos láser																																																										
Ítems	Ninguno																																																										
Detalles de Producción																																																											
Fecha de Inicio	25 - octubre - 2021																																																										
Fecha de terminación	18 – febrero - 2021																																																										
Presupuesto	<p>El presupuesto realizado para la generación del prototipo se obtuvo mediante la técnica de estimación de puntos de función (ver Anexo A)</p> <p style="text-align: center;">Tabla 6: Presupuesto planificado para el desarrollo del videojuego.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="3">Recursos</th> <th colspan="4">PRESUPUESTO PARA LA ELABORACIÓN</th> </tr> <tr> <th>Cantidad</th> <th>Unidad</th> <th>V. Unitario</th> <th>Valor Total</th> </tr> <tr> <th></th> <th></th> <th>\$</th> <th>\$</th> </tr> </thead> <tbody> <tr> <td>Lenguaje de Programación C#</td> <td style="text-align: center;">1</td> <td>Programa</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Motor Gráfico Unity</td> <td style="text-align: center;">1</td> <td>Programa</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <td>Creative Cloud</td> <td style="text-align: center;">1</td> <td>Licencia</td> <td style="text-align: center;">19.99</td> <td style="text-align: center;">239.88</td> </tr> <tr> <td>Desarrollo del prototipo</td> <td style="text-align: center;">1</td> <td>Programa</td> <td style="text-align: center;">4.341,83</td> <td style="text-align: center;">26051</td> </tr> <tr> <td>Equipo de Desarrollo</td> <td style="text-align: center;">1</td> <td>Equipo</td> <td style="text-align: center;">1700</td> <td style="text-align: center;">1700</td> </tr> <tr> <td>Servicio de Internet</td> <td style="text-align: center;">1</td> <td>Servicio</td> <td style="text-align: center;">27</td> <td style="text-align: center;">324</td> </tr> <tr> <td colspan="4" style="text-align: right;">Sub Total</td> <td style="text-align: center;">28.314,88</td> </tr> <tr> <td colspan="4" style="text-align: right;">10%</td> <td style="text-align: center;">2.831,488</td> </tr> <tr> <td colspan="4" style="text-align: right;">TOTAL</td> <td style="text-align: center;">31.146,368</td> </tr> </tbody> </table>	Recursos	PRESUPUESTO PARA LA ELABORACIÓN				Cantidad	Unidad	V. Unitario	Valor Total			\$	\$	Lenguaje de Programación C#	1	Programa	0	0	Motor Gráfico Unity	1	Programa	0	0	Creative Cloud	1	Licencia	19.99	239.88	Desarrollo del prototipo	1	Programa	4.341,83	26051	Equipo de Desarrollo	1	Equipo	1700	1700	Servicio de Internet	1	Servicio	27	324	Sub Total				28.314,88	10%				2.831,488	TOTAL				31.146,368
Recursos	PRESUPUESTO PARA LA ELABORACIÓN																																																										
	Cantidad		Unidad	V. Unitario	Valor Total																																																						
			\$	\$																																																							
Lenguaje de Programación C#	1	Programa	0	0																																																							
Motor Gráfico Unity	1	Programa	0	0																																																							
Creative Cloud	1	Licencia	19.99	239.88																																																							
Desarrollo del prototipo	1	Programa	4.341,83	26051																																																							
Equipo de Desarrollo	1	Equipo	1700	1700																																																							
Servicio de Internet	1	Servicio	27	324																																																							
Sub Total				28.314,88																																																							
10%				2.831,488																																																							
TOTAL				31.146,368																																																							

3.3.1.2.2.1. Características Funcionales y Características no funcionales.

3.3.1.2.2.1.1. Características funcionales.

- El juego debe poseer un sistema escalable de armas y experiencia.
- Deben existir dos personajes controlados por el usuario capaces de realizar acciones específicas.

- Los enemigos deben ser controlados por el computador y su comportamiento debe ser basado en Machine learning.
- El prototipo debe poseer un nivel en que se refleje la mayoría de las mecánicas detalladas.
- El juego debe ser desarrollado para ejecutarse en PC.
- El juego debe poseer mecánicas de juego sencillas y de fácil entendimiento.

3.3.1.2.2.1.2. Características no funcionales.

- El juego debe ser desarrollado en Unity.
- La estética del juego debe ser basada en Pixel Art.
- El juego debe acogerse al género de RPG de acción.
- El juego debe ser apto para audiencias que superan los 10 años de edad.

3.3.1.2.2.2.Hitos a cumplir / Interacciones por cada mecánica de juego.

Tabla 7: Estado general del Proyecto.

Estado General	
Campo	Descripción
Detalles del Proyecto	
Título	Dax & Def vs the IA
Diseñador	Alex Iván Mora Parra
Género	RPG de acción
Plataforma	PC
Fecha de inicio	25 - octubre - 2021
Fecha de Término	25 – febrero - 2021
Planeado	100%
En desarrollo	50%
Sin Planear	50%
Terminado	40%

Tabla 8: Estado de la Primera Interacción ejecutada.

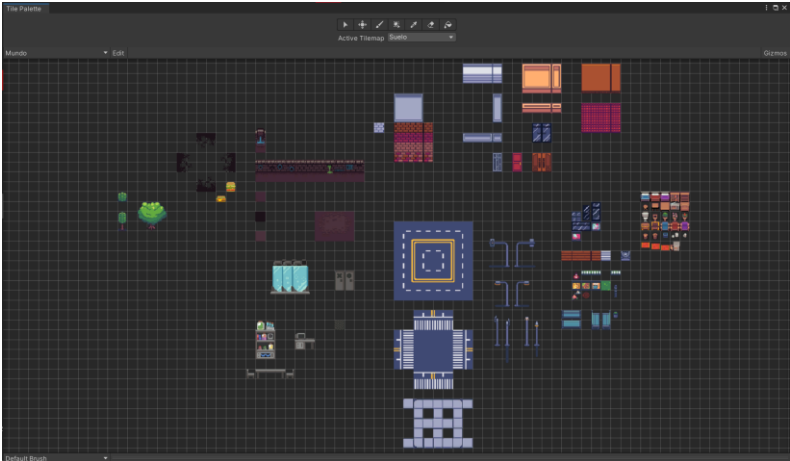
Hitos/Interacciones	
Campo	Descripción
#	I01
Inicio	30/11/2021
Días	10
Fin	13/12/2021
Meta	Mapeado y movimiento de personajes
%	100%
FEATURE LOG	
Nombre	Desarrollo del Tilemap
Estado	100%
Días	5 días
Comentarios	<p>Tilemap hace referencia a la técnica de colocar diversos Sprite(imágenes) más conocidos como el tileset de forma que estos al unirse en una cuadrícula den como resultado objetos que reunidos forman el escenario o niveles.</p>  <p>The image shows a screenshot of a tilemap editor interface. The main area is a dark grid with various colorful tiles and objects placed on it. The tiles include different colors like blue, red, orange, and green. There are also some larger, more complex objects like a blue square with a yellow border and a green object. The interface has a top menu bar with options like 'Mundo', 'Edit', and 'Active Tilemap'. There are also some tool icons and a 'Default Brush' label at the bottom left.</p>

Figura 18: Tileset empleado en el desarrollo de los entornos dentro del videojuego.

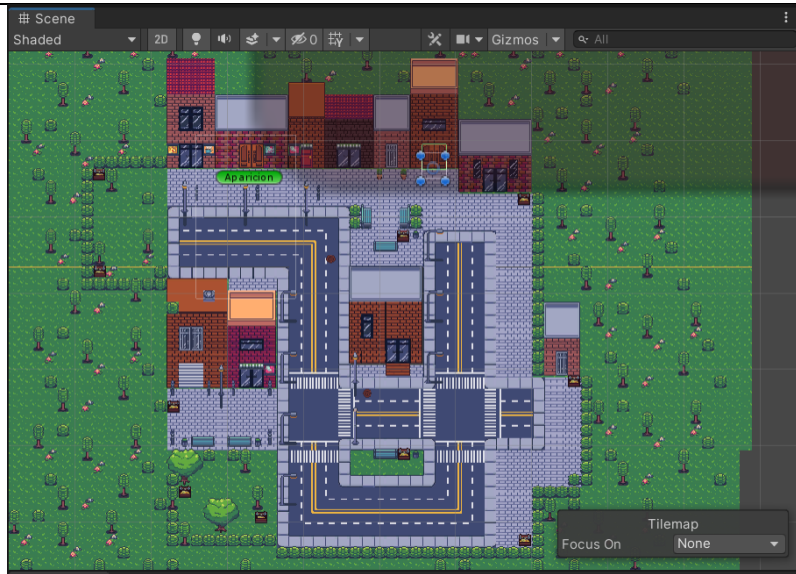


Figura 19: Tileset empleado en el desarrollo de los entornos dentro del videojuego.

Nombre Mapeado de personajes

Estado 100%

Días 2 días

Comentarios El mapeado de los personajes hace referencia al diseño e incorporación de los personajes que interactúan con el escenario.

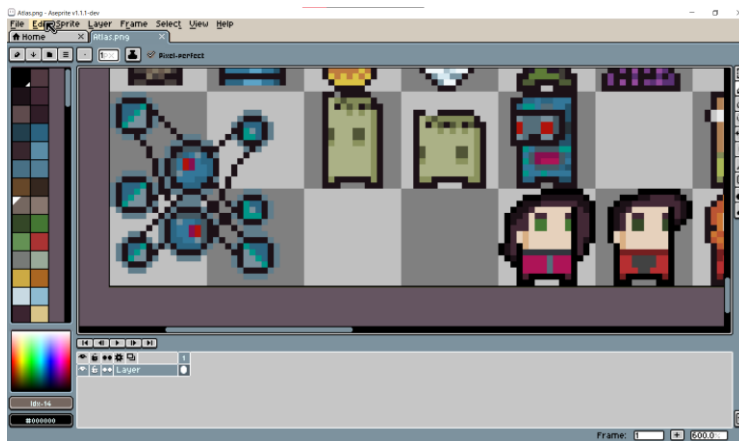
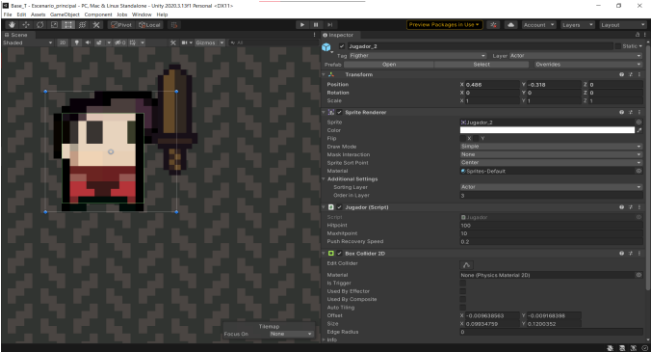
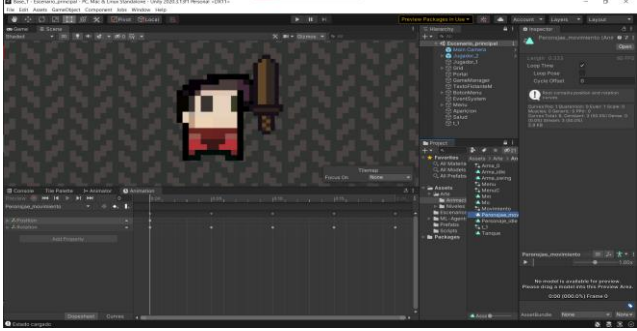


Figura 20: Mapeado de los diversos personajes del videojuego.

	 <p>Figura 21: Configuración de la interacción con el ambiente del personaje Dax.</p>
Nombre	Desarrollo del motor de movimiento
Estado	100%
Días	2 días
Comentarios	El motor de movimiento es el resultado de la codificación basado en coordenadas en las que intervienen velocidad, movimiento y detección de colisiones y que se actualizan a medida que el usuario interactúa con el personaje.
Nombre	Animaciones relacionadas al movimiento del personaje
Estado	90%
Días	1 día
Comentarios	<p>Dentro de Unity existe la posibilidad de llevar a cabo la animación de personajes en menor medida a comparación de Adobe Animate, pero para fines de la investigación y del prototipo, se optó por utilizar el animador Unity.</p>  <p>Figura 22: Animación del Personaje Dax.</p>

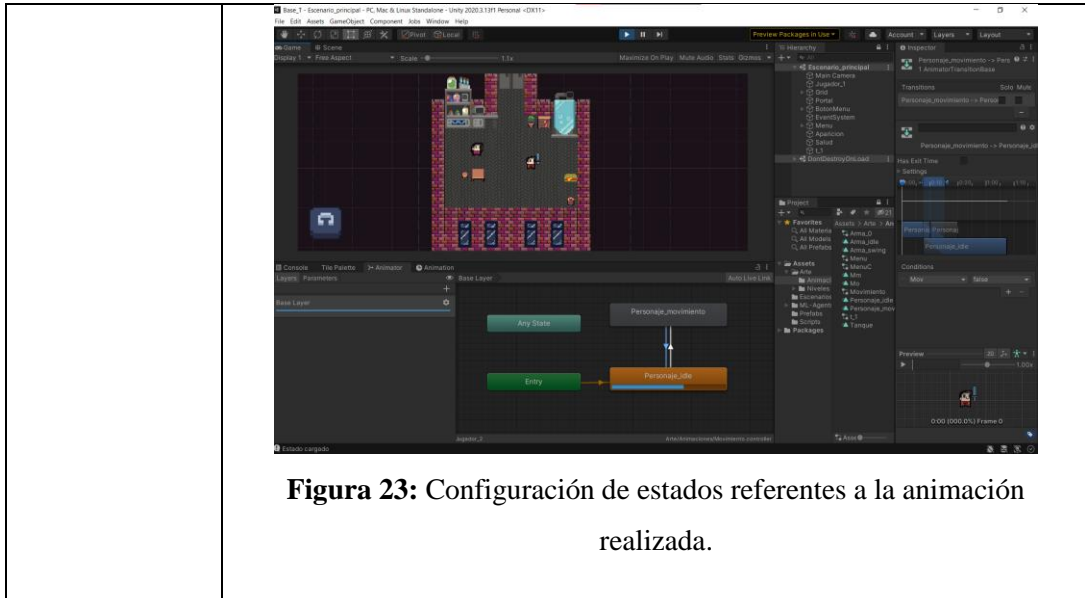


Tabla 9: Estado de la Segunda Interacción ejecutada.

Hitos/Interacciones	
Campo	Descripción
#	I02
Inicio	13/12/2021
Días	12
Fin	28/12/2021
Meta	Interacción con el entorno / motor de colisiones.
%	80%
FEATURE LOG	
Nombre	Desarrollo de colisiones con el ambiente.
Estado	80%
Días	4 días
Comentarios	La mecánica de colisiones se basa en el uso de cuadrículas que al chocar o interponerse con un objeto da como resultado una interacción ya sea esta de bloqueo o desenlace de una determinada acción. Generalmente para este apartado se hace uso de un BoxCollider2D el cual es un componente que se agrega a los elementos con los que se desea realizar una interacción y que va acompañado junto a la codificación de estados del mismo.

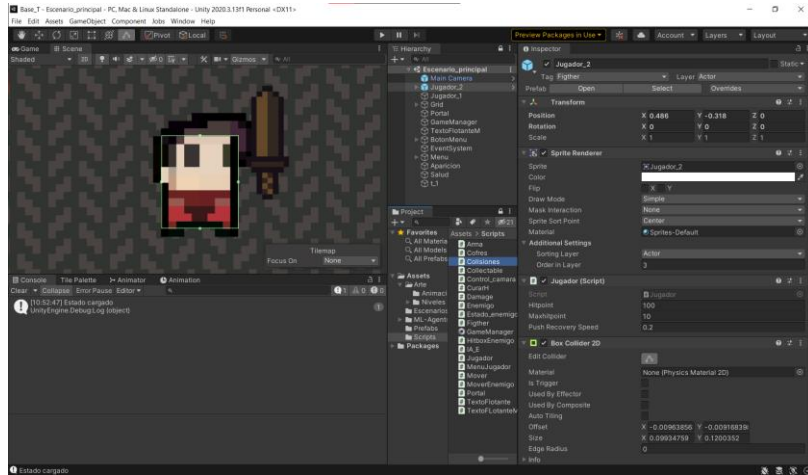


Figura 24: Estado del motor de colisiones del personaje Dax.

Nombre	Desarrollo de mecánicas de interacción con el mapa.
Estado	90%
Días	6 días
Comentario	<p>Interacciones con el mapa hace referencia a todos los objetos con los cuales el usuario interactúa de formas más activa y estos son parte de activa de la jugabilidad, es decir en este caso sería los siguientes objetos:</p> <ul style="list-style-type: none"> ● Cajas. ● Cofres. ● Restauradores de vida. ● Enemigos. <p>Para cada uno de estos objetos es necesario diversas codificaciones con las cuales limitar las acciones que se llevan a cabo al interactuar con estos objetos.</p>
Nombre	Limitación del mapeado
Estado	100%
Días	2 días
Comentarios	La limitación del mapa tiene como objetivo garantizar que el usuario no exceda el mapeado de determinado nivel o traspase estructuras u objetos, en palabras sencillas se trata de levantar

muros invisibles con los cuales el usuario simplemente choque y no le permitan seguir adelante.

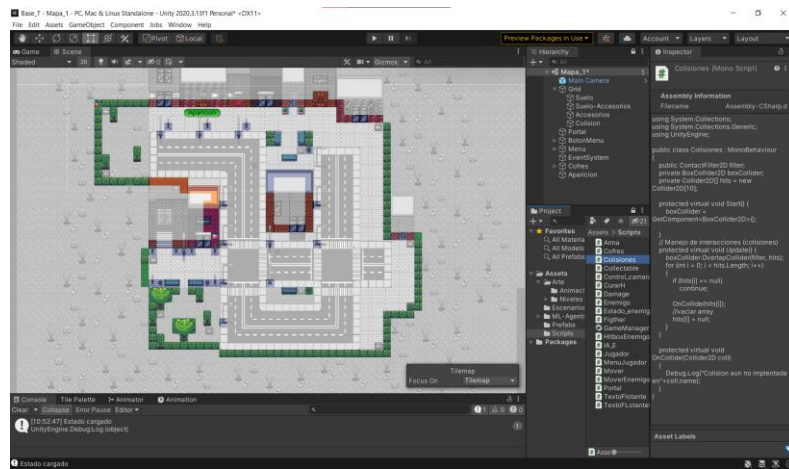


Figura 25: Límites generados para el segundo nivel del videojuego.

Tabla 10: Estado de la Tercera Interacción ejecutada.

Hitos/Interacciones	
Campo	Descripción
#	I03
Inicio	30/12/2021
Días	2 días
Fin	31/12/2021
Meta	Sistema de combate.
%	100%
FEATURE LOG	
Nombre	Gestión de armas
Estado	100%
Días	1
Comentarios	El sistema de gestión de armas, se enfoca en el manejo y evolución de todo el equipamiento del personaje, es decir el conjunto de armas que este tendrá a disposición junto a las características que éstas poseerán.

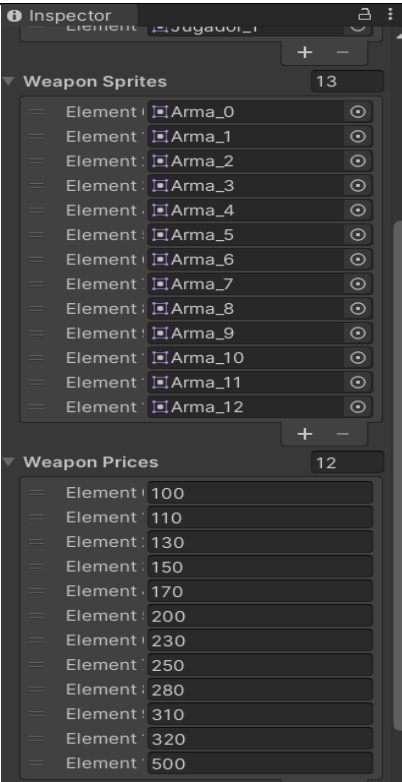
	 <p>The image shows a game development inspector window. It has two main sections: 'Weapon Sprites' and 'Weapon Prices'. The 'Weapon Sprites' section contains 13 items, each labeled 'Element' followed by 'Arma_0' through 'Arma_12'. Each item has a small square icon and a circular handle. The 'Weapon Prices' section contains 12 items, each labeled 'Element' followed by a numerical value: 100, 110, 130, 150, 170, 200, 230, 250, 280, 310, 320, and 500. Each price item also has a small square icon and a circular handle.</p>
	<p>Figura 26: Gestión del armamento disponible para el jugador.</p>
	<p>Esto incluye a su vez la codificación necesaria para que los valores de ataque y empuje mejoren a la par que el arma cambia de aspecto y precio de adquisición.</p>
Nombre	Desarrollo del combate en relación de objetos y el entorno
Estado	100%
Días	1 día
Comentarios	<p>El desarrollo del combate en relación de objetos y el entorno es netamente la codificación de estados al momento chocar el arma con algún objeto contemplado dentro del mapa, los objetos que de momento poseen interacción con el arma son los siguientes:</p> <ul style="list-style-type: none"> ● Objetos destruibles (cajas). ● Enemigos.

Tabla 11: Estado de la Cuarta Interacción ejecutada.

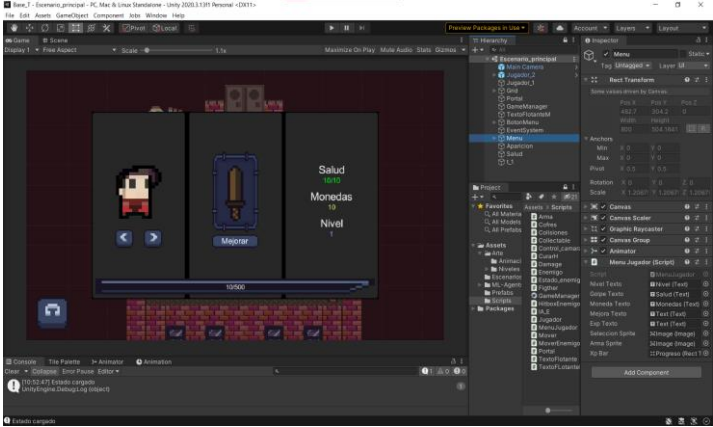
Hitos/Interacciones	
Campo	Descripción
Iteración #	I04
Inicio	01/01/2022
Días	6 días
Fin	07/01/2022
Meta	Texto y GUI
%	100%
FEATURE LOG	
Nombre	Desarrollo de la Interfaz
Estado	90%
Días	2 días
Comentarios	<p>Este apartado consta de la creación de la parte visual con la cual el usuario interactuara a lo largo del juego, dentro de este consta el menú de cambio de personajes, mejora de armas y estado del personaje.</p> 
Nombre	Muestreo del estado del personaje
Estado	100%
Días	3 días

Figura 27: Interfaz de mejora y estado del jugador.


Comentarios	El muestreo del estado del personaje consta de una serie de codificaciones que se encarga de recolectar datos necesarios para su posterior muestreo dentro de un determinado apartado en la interfaz, estos datos se encuentran en un estado de constante actualización basado en las acciones tomadas por el usuario y acciones resultantes de la interacción con el ambiente.
Nombre	Gestor de Texto
Estado	100%
Días	1 día
Comentarios	<p>El desarrollo del gestor de texto se deriva del motor de colisiones y se desenvuelve de igual forma que esté, el texto es codificado de cierta forma que este pueda ser empleado en cualquier ámbito dentro del juego, ya sea interacciones con objetos, muestreo de mensajes emergentes, identificación de acciones (daño realizado o daño recibido, recuperación de vida, etc.).</p> <div style="text-align: center;">  </div> <p>Figura 28: Muestra de Texto al interactuar con objetos del mapa.</p>

Tabla 12: Estado de la Quinta Interacción ejecutada.

Hitos/Interacciones	
Campo	Descripción
#	I05
Inicio	07/01/2022
Días	8
Fin	19/01/2022
Meta	Programación e integración de la IA del agente.
%	92%
FEATURE LOG	

Nombre	Levantamiento del ambiente
Estado	100%
Días	1 día
Comentarios	Esta sección será explicada con mayor detalle a continuación.
Nombre	Programación del agente
Estado	100%
Días	2 días
Comentarios	Esta sección será explicada con mayor detalle a continuación.
Nombre	Entrenamiento del agente
Estado	80%
Días	3 días
Comentarios	Esta sección será explicada con mayor detalle a continuación.

3.3.2. Desarrollo de la Inteligencia Artificial.

Dentro de la investigación para llevar a cabo la incorporación de un agente inteligente surgieron diversas herramientas que están enfocadas a la creación de ambientes inteligentes dentro de un videojuego, para la presente investigación se hace uso de MAgents como base de desarrollo y entrenamiento del agente, para este rol se ha determinado que la fuerza enemiga dentro del videojuego sea capaz de reaccionar de manera dinámica a las acciones llevadas a cabo por el usuario.

Con el fin de explicar con mayor profundidad el proceso de creación del agente junto con su entrenamiento para su posterior integración dentro del proyecto, se ha propuesto separar esta sección de la parte de hitos e Interacciones, dado a que el principal objetivo de esta investigación se basa en la Inteligencia Artificial incorporada dentro de un prototipo funcional.

3.3.2.1. Levantamiento del Ambiente de aprendizaje.

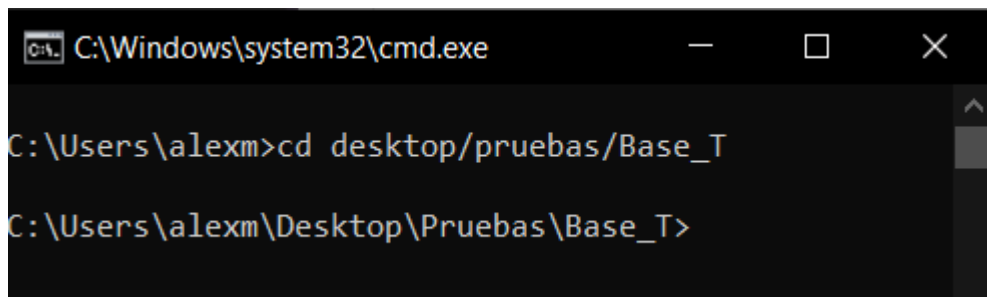
Esta es la fase preliminar encargada de levantar las herramientas necesarias para generar un ambiente de trabajo que permita llevar a cabo la codificación destinada a los enemigos presentes dentro del videojuego.

Dentro de este proceso se hace uso de las siguientes herramientas:

- Python versión 3.9.4.

- MLagents versión 2.0.1.
- PYtorch versión 1.10.1
- Git

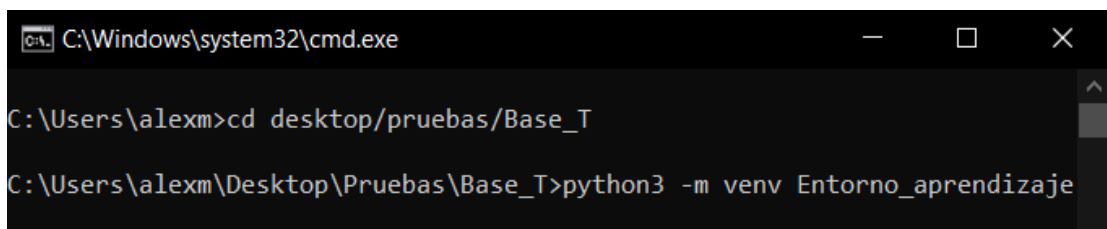
Como primer paso para el levantamiento de ambiente es necesario tener de antemano instalado una versión de Python ya sea 3.6.1 o superior y clonar el repositorio de MLagents de GitHub, una vez cumplido estos requisitos se hace uso de la consola de Windows con el fin de crear un entorno virtual basado en Python, para ello es necesario ubicarse en la carpeta en la cual se encuentran el proyecto en desarrollo junto a la carpeta donde se encuentra el repositorio clonado del proyecto de MLagents.



```
C:\Windows\system32\cmd.exe
C:\Users\alexm>cd desktop/pruebas/Base_T
C:\Users\alexm\Desktop\Pruebas\Base_T>
```

Figura 29: Navegación entre Carpetas del Proyecto dentro del CMD.

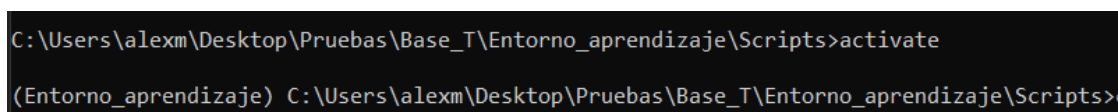
Seguido de esto, se procede a generar el entorno virtual de Python necesario para instalar los paquetes y herramientas para llevar a cabo el entrenamiento del agente.



```
C:\Windows\system32\cmd.exe
C:\Users\alexm>cd desktop/pruebas/Base_T
C:\Users\alexm\Desktop\Pruebas\Base_T>python3 -m venv Entorno_aprendizaje
```

Figura 30: Creación del ambiente virtual basado en Python.

Una vez creado el entorno virtual se debe activar el mismo, con el fin de instalar dentro de este los paquetes necesarios para el entrenamiento del agente.



```
C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje\Scripts>activate
(Entorno_aprendizaje) C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje\Scripts>
```

Figura 31: Activación del entorno virtual.

Para la instalación de los paquetes es necesario utilizar el administrador de paquetes de Python para descargar e instalar dentro del ambiente los paquetes de MLagents y PYtorch.

```
(Entorno_aprendizaje) C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje>pip install mlagents_
```

Figura 32: Instalación del Paquete MLagents de unity.

```
(Entorno_aprendizaje) C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje>pip3 install torch==1.10.1+cu102 torchvision==0.11.2+cu102 torchaudio==0.10.1+cu102 -f https://download.pytorch.org/whl/cu102/torch_stable.html
```

Figura 33: Instalación del paquete de PYtorch.

Con esto el ambiente virtual necesario para el entrenamiento estaría listo para ejecutar los algoritmos de aprendizaje propios del paquete de MLagents.

3.3.2.2. Programación del agente.

Para que se puede ejecutar los algoritmos de aprendizaje de MLagents es necesario preparar el ambiente en el cual se establecen las directrices de comportamiento del agente, para lo cual son necesarios diversos archivos de codificación en los que se debe parametrizar las siguientes directrices:

- Comportamiento de la IA
- Estado del agente
- Movimiento del Agente.

3.3.2.2.1. Comportamiento de la IA.

La codificación de la IA encargada del agente se basa en diversos componentes que permiten determinar diversos estados en los que se encuentran los actores y escenario en donde se ejecutan los entrenamientos, a continuación, se detallan dichos componentes:

- **OnEpisodeBegin()**: Este componente se encarga de inicializar las variables necesarias para la ejecución del algoritmo de aprendizaje, es decir se programa los diversos estados como son el caso de la posición del objetivo con relación al agente en entrenamiento y las acciones a tomar en cuanto el agente cumpla con su objetivo.

```

public override void OnEpisodeBegin()
{
    float[] possiblePosicionx = new float[] {0.5f,1.5f };
    int randomIndex = Random.Range(0, possiblePosicionx.Length);
    float posicionX = possiblePosicionx[randomIndex];

    jugador.transform.position = new Vector3(posicionX, -.5f,0);
    caja.transform.position = (randomIndex == 0) ? new Vector3(possiblePosicionx[1], -.5f, 0) : new Vector3(possiblePosicionx[0], -.5f, 0);
    transform.position = new Vector3(1, 0, 0);
}

```

Figura 34: Fragmento del código empleado en el componente OnEpisodeBegin().

- **CollectObservations():** Este componente se encarga de instaurar los sensores del agente necesarios para que este sea consciente del ambiente que lo rodea, este componente almacena las diversas interacciones del agente con el resto de objetos codificados para interactuar.

```

public override void CollectObservations(VectorSensor sensor)
{
    sensor.AddObservation(transform.position);
    sensor.AddObservation(jugador.position);
}

```

Figura 35: Fragmento del código empleado en el componente CollectObservations().

- **OnActionsRecived():** En esencia este componente es un comparador de acciones basado en las decisiones llevadas a cabo por el agente.

```

public override void OnActionReceived(ActionBuffers actions)
{
    estado_Enemigo.horizontal = actions.ContinuousActions[0];
}

```

Figura 36: Fragmento del código empleado en el componente OnActionsRecived().

- **Heuristic():** Este componente es una forma alternativa de llevar a cabo el entrenamiento, es decir si decidimos instaurar un aprendizaje por imitación este componente permitirá tomar control del agente, para indicar de forma manual el objetivo del mismo.

```

public override void Heuristic(in ActionBuffers actionsOut)
{
    ActionSegment<float> continuousActions = actionsOut.ContinuousActions;
    continuousActions[0] = Input.GetAxisRaw("Horizontal");
}

```

Figura 37: Fragmento del código empleado en el componente Heuristic().

● **OnTrigger2D():** Finalmente este componente es el encargado de otorgar las recompensas al momento de interactuar con los objetos, aquí también se imponen las penalizaciones por comportamientos erróneos.

```
private void OnTrigger2D(Collider2D collider2D)
{
    if (collider2D.gameObject.name.Equals("Jugador_2"))
    {
        Debug.Log("Jugador Encontrado");
        SetReward(+20f);
        EndEpisode();
    }
    if (collider2D.gameObject.name.Equals("Colision"))
    {
        Debug.Log("Caja Encontrada");
        SetReward(-20f);
        EndEpisode();
    }
}
```

Figura 38: Fragmento del código empleado en el componente OnTrigger2D().

En base a lo descrito posteriormente el comportamiento de la IA tiene como objetivo generar un comportamiento de búsqueda y ataque por parte del agente, sin importar que el jugador se encuentre en constante movimiento, además el algoritmo tiene la posibilidad de crecimiento exponencial lo que se traduce en una IA más potente en relación al tiempo de entrenamiento de la misma.

3.3.2.3. Entrenamiento del Agente.

El proceso de entrenamiento del agente es relativamente sencillo y consta de tres simples pasos, los mismos que se detallan a continuación:

● **Activación del ambiente virtual:** El primer paso derivado de la previa creación del ambiente virtual basado en Python, para la activación del ambiente simplemente se debe acceder a la consola de Windows y navegar hasta la carpeta del proyecto donde se encuentra a su vez la carpeta del ambiente virtual, la cual contiene diversas carpetas y archivos; para proceder con la activación es necesario ubicarse en la carpeta scripts y ejecutar el archivo de nombre “activate”.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\alexm>cd desktop/pruebas/base_t/Entorno_aprendizaje/scripts
C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje\Scripts>activate
(Entorno_aprendizaje) C:\Users\alexm\Desktop\Pruebas\Base_T\Entorno_aprendizaje\Scripts>
```

Figura 39: Activación del Ambiente previo a la ejecución del entrenamiento.

- **Ejecución del comando de entrenamiento:** Una vez activado el entorno virtual, se procede a navegar a la carpeta raíz del proyecto en donde se ejecutará el comando referente a la ejecución del entrenamiento.

Cabe recalcar que previamente es necesario tener preparado el entorno de aprendizaje dentro del editor junto con la configuración adecuada de la IA perteneciente al agente.

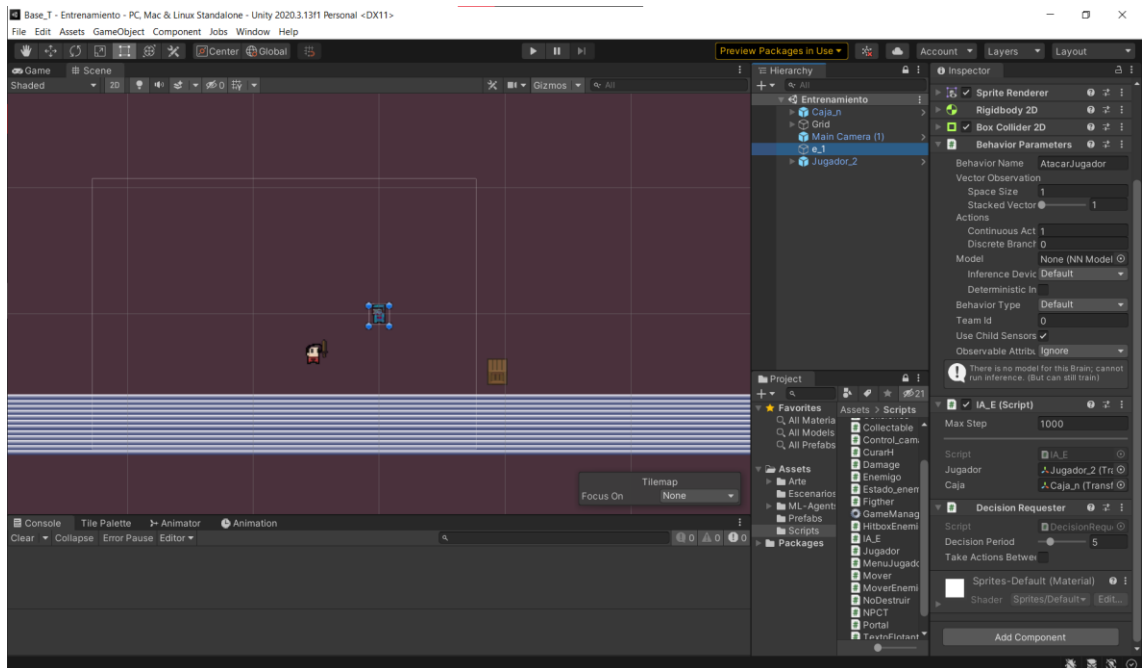


Figura 40: Escenario de entrenamiento del Agente.

```
(Entorno_aprendizaje) C:\Users\alexm\Desktop\Pruebas\Base_T>mlagents-learn --run-id Test0015
```

Figura 41: Ejecución del comando para el entrenamiento del agente.

- **Puesta en marcha:** Una vez ejecutado el comando que habilita el entrenamiento del Agente es necesario comprobar todos los parámetros de comportamiento establecidos, los

mismos que se detallan en la consola, para posteriormente ejecutar el juego lo que permitirá el inicio del entrenamiento.

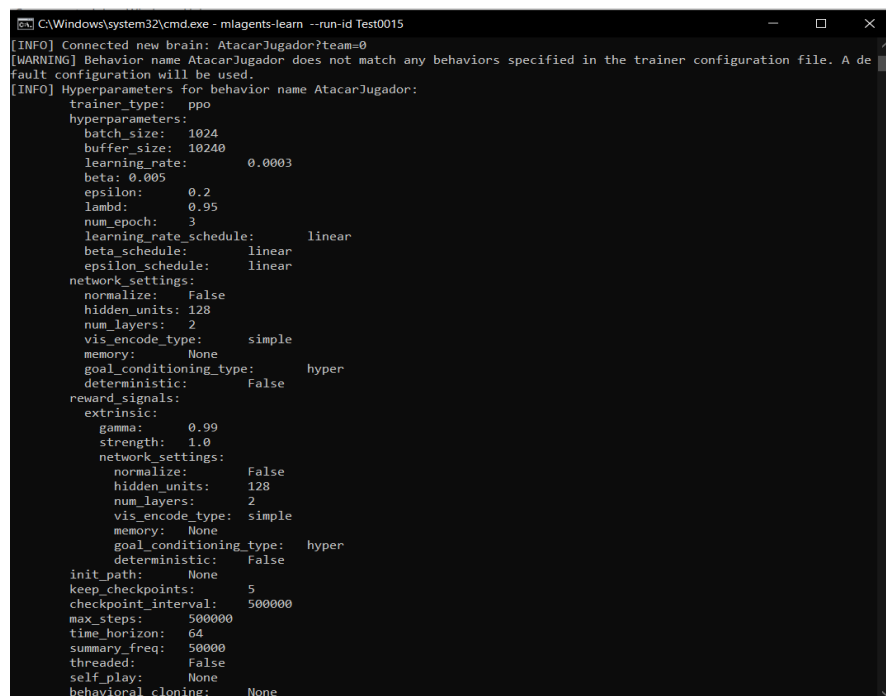


```
C:\Windows\system32\cmd.exe - mlagents-learn --run-id Test0015
g: Distutils was imported before Setuptools. This usage is discouraged and may exhibit undesirable behaviors or errors.
Please use Setuptools' objects directly or at least import Setuptools first.
warnings.warn(

Version information:
ml-agents: 0.28.0,
ml-agents-envs: 0.28.0,
Communicator API: 1.5.0,
PyTorch: 1.10.1+cu102

c:\users\alex\desktop\pruebas\base_t\entorno_aprendizaje\lib\site-packages\setuptools\distutils_patch.py:25: UserWarning
g: Distutils was imported before Setuptools. This usage is discouraged and may exhibit undesirable behaviors or errors.
Please use Setuptools' objects directly or at least import Setuptools first.
warnings.warn(
[INFO] Listening on port 5004. Start training by pressing the Play button in the Unity Editor.
```

Figura 42: Puesta en marcha del algoritmo de Aprendizaje destinado al Agente.



```
C:\Windows\system32\cmd.exe - mlagents-learn --run-id Test0015
[INFO] Connected new brain: AtacarJugador?team=0
[WARNING] Behavior name AtacarJugador does not match any behaviors specified in the trainer configuration file. A de
fault configuration will be used.
[INFO] Hyperparameters for behavior name AtacarJugador:
  trainer_type: ppo
  hyperparameters:
    batch_size: 1024
    buffer_size: 10240
    learning_rate: 0.0003
    beta: 0.005
    epsilon: 0.2
    lambda: 0.95
    num_epoch: 3
    learning_rate_schedule: linear
    beta_schedule: linear
    epsilon_schedule: linear
  network_settings:
    normalize: False
    hidden_units: 128
    num_layers: 2
    vis_encode_type: simple
  memory: None
  goal_conditioning_type: hyper
  deterministic: False
  reward_signals:
    extrinsic:
      gamma: 0.99
      strength: 1.0
  network_settings:
    normalize: False
    hidden_units: 128
    num_layers: 2
    vis_encode_type: simple
  memory: None
  goal_conditioning_type: hyper
  deterministic: False
  init_path: None
  keep_checkpoints: 5
  checkpoint_interval: 500000
  max_steps: 500000
  time_horizon: 64
  summary_freq: 50000
  threaded: False
  self_play: None
  behavioral_cloning: None
```

Figura 43: Parámetros establecidos para el entrenamiento.

El entrenamiento puede poseer diversos intervalos de tiempo y esto afecta directamente a la capacidad de reacción y comportamiento del agente inteligente, estos resultados se muestran en la siguiente sección junto a diversas observaciones.

4. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.

En base a lo detallado anteriormente, se puede apreciar diversos resultados derivados del desarrollo del prototipo, los mismo que se profundizan a continuación.

4.1. RESULTADOS DE LA ELABORACIÓN DEL PROTOTIPO.

Dado a que el desarrollo de videojuegos no se apega a una metodología de desarrollo única o en una previamente establecida, el optar por la metodología experimental SUM ha dado como resultado el cumplimiento de lo propuesto en el presente proyecto.

La idea propuesta para el desarrollo del prototipo es la base fundamental en cuanto a esta metodología se refiere y la misma provee los criterios de evaluación que son contemplados para determinar la calidad del producto y determinar los posibles cambios o mejoras a futuro.

El videojuego resultante de esta metodología cumple con los criterios básicos de aceptación del producto basados en el concepto de prototipo funcional de un videojuego y de acuerdo con la idea planteada por el mismo se puede detallar el cumplimiento de los mismos, pese a no existir un modelo formal de evaluación de videojuegos se generalizó los siguientes aspectos de evaluación basándose en el análisis de obras similares tanto en diseño como jugabilidad.

- **Gráficos:** Según la concepción de la idea se buscaba optar por el diseño de un videojuego basado en el estilo de arte Pixel Art, el mismo que se ve reflejado en todo aspecto del videojuego. El propósito de la idea a su vez buscó emular la experiencia de los juegos arcade de 16bits.
- **Diseño de Personajes:** El diseño de los personajes se ajusta al arte propuesto por la idea, pese a ser un prototipo se buscó profundizar en el aspecto gráfico de los personajes y equipamiento con el fin de mostrar una jugabilidad llamativa para el usuario.
- **Diseño de Ambientes:** El Diseño del ambiente del videojuego al igual que el diseño de los personajes busca encajar en la idea planteada y esto a criterio personal se llevó a cabo de manera exitosa y sin ningún contratiempo.
- **Jugabilidad:** El resultado de la jugabilidad es satisfactorio dado a que esta se apega a los requisitos del género de RPG con el añadido del combate activo y evolutivo.
- **Gameplay:** Dentro del gameplay se puede recalcar diversos aspectos: ya que este proyecto se basa en un prototipo la cantidad de niveles desarrollados no son los suficientes

para demostrar el alcance de la jugabilidad, esto se debe a que el tiempo de desarrollo del producto final era limitado y la experiencia previa en el desarrollo de este tipo de software era mínima. Por ende, las mecánicas de juego se ven limitadas al ser un prototipo enfocado al uso de Agentes Inteligentes. A pesar de estos contratiempos el resultado de la jugabilidad se puede apreciar en la sencillez de interacción con el ambiente propuesto.

- **Narrativa:** Pese a que la idea del desarrollo del videojuego se encuentra detallada el prototipo no presenta narrativa ninguna activa, pero esta se encuentra contemplada en futuras versiones del videojuego.
- **Sonido:** Al tratarse de un prototipo no se vio en la necesidad de incorporar efectos sonoros dado a que el enfoque de la investigación no es el videojuego en sí, sino el apartado del machine learning incorporado dentro del mismo.
- **IA:** Dado a que este es el enfoque principal del proyecto de investigación este apartado será detallado más adelante.

4.2. RESULTADOS DEL ENTRENAMIENTO DEL AGENTE INTELIGENTE.

Este apartado es el enfoque principal del proyecto de investigación y para poder detallar el estado del entrenamiento de la IA es necesario especificar las directrices en las que se basa la presente evaluación del algoritmo de aprendizaje:

- **Tiempo de aprendizaje:** El principal factor relacionado al entrenamiento y de este depende en gran medida el comportamiento del algoritmo planteado.
- **Acciones tomadas en relación al tiempo:** Representa el número de pasos ejecutados para llevar a cabo la tarea propuesta.
- **Tipo de modelo utilizado:** Dado a que se emplea un algoritmo prefabricado los modelos a elegir son limitados a dos: aprendizaje por imitación y aprendizaje automatizado.

4.2.1. Treinta minutos de tiempo de ejecución del entrenamiento.

Durante la primera media hora de ejecución no se apreciaron avances significativos en cuanto al comportamiento esperado por el agente.

Tabla 13: Ficha de Observación (Tiempo transcurrido 30m)

Ficha de Observación			
Fecha:	12/01/2022	N°	01
Observador:	Alex Mora		
Criterio a Observar	Comportamiento del agente.		
Observación		Detalle	
Tiempo de ejecución		30 minutos	
Número de pasos		30000	
Acciones tomadas por el agente		El agente mantiene su posición, no se observan avances significativos, al momento de intercambiar posiciones con el jugador existen interacciones que generan una falsa recompensa.	
Recompensas Recolectadas		0	
Penalizaciones recibidas.		0	

Los datos indican un aprendizaje lento, esto se debe al límite establecido de pasos para finalizar la sesión de entrenamiento, el resultado final de este entrenamiento arrojó una personalidad errática en cuanto a movimiento y ubicación del personaje dentro del entorno del videojuego.

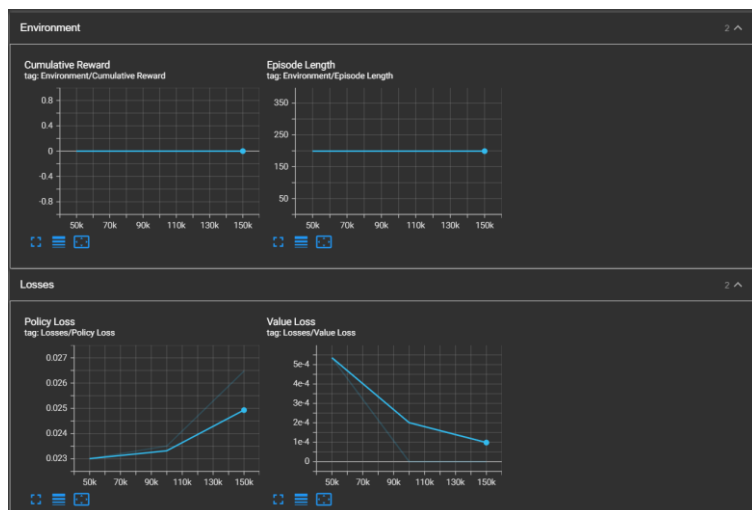


Figura 44: Resultado de PYtorch primera sesión de entrenamiento.

4.2.2. Una Hora de tiempo de ejecución del entrenamiento.

Tabla 14: Ficha de Observación (1 hora)

Ficha de Observación			
Fecha:	12/01/2022	N°	02
Observador:	Alex Mora		
Criterio Observar	a Comportamiento del agente.		
Observación		Detalle	
Tiempo de ejecución		60 minutos	
Número de pasos		250000	
Acciones tomadas por el agente		El agente se mueva en búsqueda del jugador, se observa una constante tendencia de búsqueda del jugador y obstáculos.	
Recompensas Recolectadas		60	
Penalizaciones recibidas.		60	

Después de una hora de entrenamiento el proceso es persistente en la demora de toma de decisiones el incremento de los pasos llevados a cabo no refleja el estado del proceso de entrenamiento, como resultado el comportamiento errático observado anteriormente ha disminuido en gran medida el agente ya es capaz de identificar con éxito al jugador, pero le cuesta moverse adecuadamente por escenario.



Figura 45: Gráfico indicativo de disminución de pérdida de datos.

En el tablero de TensorBoard se observa una creciente tendencia de disminución de pérdida de datos en relación a las primeras ejecuciones del aprendizaje, a su vez se visualiza que el tiempo de ejecución no es el idóneo.

4.2.3. Cuatro horas de ejecución del Entrenamiento.

Tabla 15: Ficha de Observación Cuarta hora.

Ficha de Observación			
Fecha:	12/01/2022	N°	03
Observador:	Alex Mora		
Criterio Observar	a Comportamiento del agente.		
Observación		Detalle	
Tiempo de ejecución	240 minutos		

Número de pasos	500000
Acciones tomadas por el agente	El agente se mueva en búsqueda del jugador, se observa una mejor respuesta por parte del agente, sigue acercando a los obstáculos
Recompensas Recolectadas	180
Penalizaciones recibidas.	40

Después de cuatro horas de entrenamiento persiste el tiempo de demora en relación a la toma de decisiones, el ambiente alcanzó el número de pasos máximos determinado en los parámetros de aprendizaje. Como resultado el entrenamiento tuvo que reiniciarse en dos ocasiones, a pesar de esto el resultado obtenido en esta sesión de entrenamiento son prometedores ya que se observa un movimiento más dinámico en cuanto a la toma de decisiones dentro del juego, el agente reacciona cada vez más rápido a acciones emprendidas por el jugador, pero persiste la falta de ubicación espacial del agente, continúa chocando y chocando los diversos obstáculos del mapeado.



Figura 46: Índice de pérdida de datos.

Los resultados arrojados por el TensorBoard Indican una persistente tendencia a la disminución de datos perdidos en relación del tiempo de ejecución del aprendizaje. Por lo cual se sugiere incrementar el tiempo de ejecución del aprendizaje con el fin de eliminar los problemas descritos anteriormente.

4.3. RESULTADOS DEL CUESTIONARIO REFERENTE AL COMPORTAMIENTO DEL AGENTE INTELIGENTE.

En base a las observaciones realizadas sobre el entrenamiento del agente se planteó un cuestionario destinado a recolectar la experiencia por parte de los testers del prototipo del videojuego. Dicho cuestionario consta de 8 preguntas enfocadas a calificar el comportamiento e interacción del agente inteligente con el usuario en base a la experiencia obtenida al momento de jugar.

Dentro de la primera pregunta se obtuvo el estado inicial del agente en relación al tiempo de reacción que este posee al momento de visualizar al jugador y las acciones tomadas por el mismo.

1. Considera que el tiempo de reacción del agente (enemigo del videojuego) es el adecuado.
5 respuestas

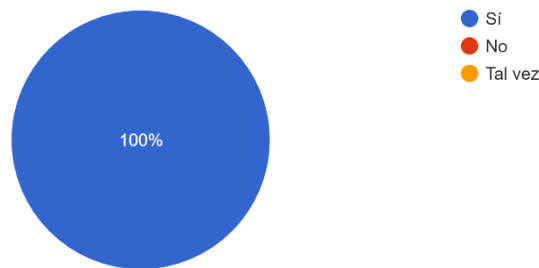


Figura 47: Resultado Primera Pregunta Cuestionario Testers.

La respuesta obtenida por parte de los testers afirma que el agente posee un comportamiento idóneo al momento de visualizar al jugador y que las acciones que ejecuta el mismo son las esperadas dentro de lo planteado al principio del proyecto.

En cuanto a la segunda pregunta, esta se deriva de las acciones evaluadas dentro de la primera pregunta y está buscó evaluar cómo el agente se traslada a través del mapa en relación a los movimientos realizados por el jugador.

2. Clasifique el movimiento del Agente.

5 respuestas

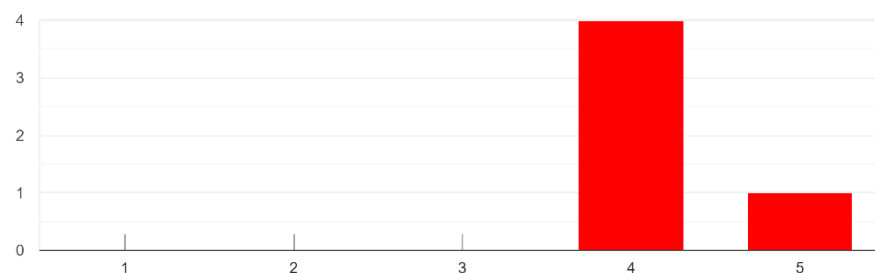


Figura 48: Resultado segunda Pregunta Cuestionario Testers.

Dentro de los cinco testers asignados a la evaluación del agente en su mayoría califican al movimiento del agente como fluido pero que este puede tener mejoras, el movimiento presenta fallos menores poco frecuentes al momento de acercarse demasiado al jugador, estos fallos son comportamientos esperados debido a la naturaleza del agente.

Siguiendo con las evaluaciones referentes a desenvolvimiento del agente a través del mapeado la tercera pregunta busca evaluar la capacidad del agente al momento de evitar los obstáculos establecidos en el mapa sin dejar de lado la tarea de persecución del jugador.

3. ¿El Agente es capaz de evitar obstaculos?

5 respuestas

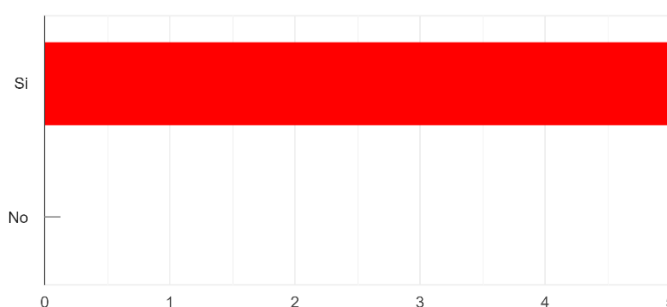


Figura 49: Resultado tercera Pregunta Cuestionario Testers.

En cuanto a la capacidad del agente de evitar obstáculos los testers afirman que este es capaz de evitar los obstáculos a medida que lleva a cabo las tareas asignadas al mismo afirmando que dentro de las pruebas realizadas el agente se desenvolvió de manera óptima dentro del mapa.

A partir de la cuarta pregunta el cuestionario buscó evaluar el nivel de eficacia del algoritmo planteado en relación a la forma en la que el agente genera estrategias y comprobar si este cumple una de las expectativas esperadas dentro del proyecto de investigación.

4. ¿Considera que los agentes en el mapa son capaces de generar estrategias de acuerdo a las Interacciones del Usuario?
5 respuestas

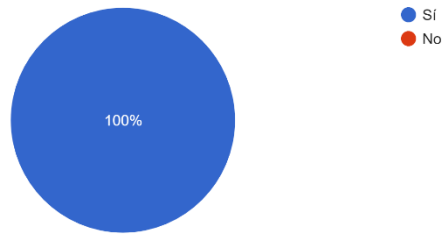


Figura 50: Resultado cuarta Pregunta Cuestionario Testers.

En relación a la cuarta pregunta planteada los testers supieron apreciar que el agente posee la cualidad de generar pequeñas estrategias en relación a las acciones tomadas por parte del jugador, este comportamiento se ve reflejado al momento que estos son capaces de acorralar al jugador y atacar.

La quinta pregunta buscó conocer la postura de los testers en cuanto al tiempo invertido en el entrenamiento del agente y si este refleja el comportamiento planteado dentro del proyecto de investigación.

5. ¿Cree usted que las horas empleadas en el entrenamiento del agente fueron suficientes para obtener el comportamiento esperado?
5 respuestas

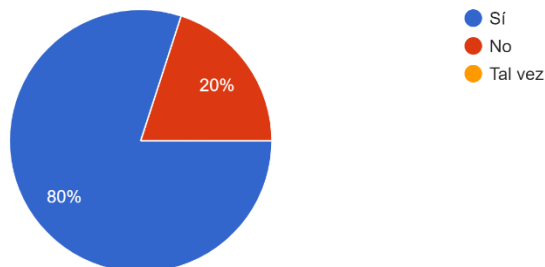


Figura 51: Resultado quinta Pregunta Cuestionario Testers.

Dentro de este parámetro de evaluación la mayoría de los testes afirman que el tiempo empleado cumple con el comportamiento esperado por parte del agente sin embargo un tester menciona que el comportamiento se apega a lo esperado, pero aún existe un margen de mejora.

En cuanto a la sexta pregunta se buscó evaluar el nivel de interacción que poseía el agente en relación de las acciones llevadas a cabo por el jugador, es decir se buscó evaluar si las acciones planteadas (atacar, perseguir, generar estrategias) son suficientes dentro del contexto de la investigación.

6. Clasifique el nivel de interacción con el usuario por parte del agente.
5 respuestas

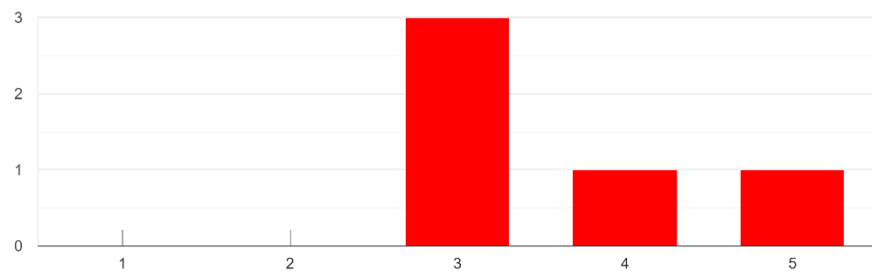


Figura 52: Resultado sexta Pregunta Cuestionario Testers.

En cuanto a este punto los testes afirman que las acciones planteadas son las que se requiere dentro del proyecto de investigación pero que es necesario plantear acciones más dinámicas en relación a las ejecutadas por el jugador.

En relación a lo planteado en la hipótesis del presente proyecto de investigación la séptima pregunta busca confirmar si el objetivo de incorporar machine learning dentro de un videojuego se puede realizar exitosamente.

7. Considera que la incorporación de machine learning dentro del prototipo se realizó con éxito.
5 respuestas

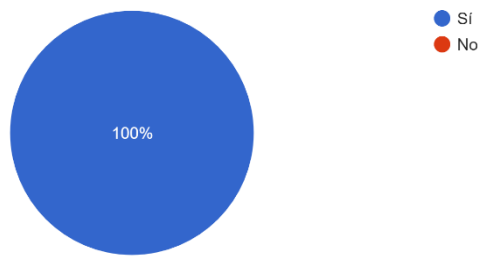


Figura 53: Resultado séptima Pregunta Cuestionario Testers.

En este caso los testers manifestaron que la incorporación de machine learning dentro del videojuego se realizó exitosamente y sin ningún contratiempo, las observaciones en las que se puede mejorar el comportamiento del agente para posteriores versiones recaen en la última pregunta.

8. ¿Que aspectos considera Usted que se pueden mejorar?

5 respuestas

El tiempo de reacción

El nivel de interacción con el Usuario

Que el agente no se monte con el avatar cuando ataquen

Mejorar las animaciones del agente

Las tareas asignadas al agente inteligente.

Figura 54: Octava Pregunta Cuestionario Testers.

Estas observaciones se tomarán en cuenta en futuras versiones del videojuego, debido a la naturaleza del presente trabajo de investigación.

5. CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES.

- La información recolectada de las diversas fuentes de información referente a machine learning aplicada a videojuegos facilito el desarrollo del proyecto de investigación a pesar que esta fue escasa y en su mayoría difícil de localizar.
- Mediante el uso de la metodología SUM se pudo generar las diversas interacciones y fases del prototipo, permitiendo así cumplir los tiempos planteados para el desarrollo del proyecto.
- La incorporación de machine learning dentro del prototipo fue posible gracias a la implementación de ML-Agents del propio Unity el cual basa su funcionalidad en algoritmos de aprendizaje basados en el refuerzo, imitación o neuro evolución. Para el proyecto solo se exploró los algoritmos de aprendizaje por refuerzo e imitación debido a la naturaleza del videojuego dando como resultado una mecánica de juego más interactiva a diferencia de algoritmos de comportamiento preestablecidos y no evolutivos.
- El tiempo empleado para el entrenamiento del agente fue mínimo para reflejar un comportamiento complejo dentro del ambiente en el que se encuentra con el fin de perseguir al jugador y evitar obstáculos del mapa.
- El uso del motor gráfico Unity agilizo en gran medida el proceso de desarrollo del prototipo del videojuego, siendo este pilar fundamental en la implementación de machine learning en el agente inteligente.

5.2. RECOMENDACIONES.

- Profundizar el uso de la metodología SUM para el desarrollo de videojuegos validando su usabilidad y factibilidad en el desarrollo de este tipo de proyecto.
- Contar con un grupo de desarrollo más amplio y multifacético con el fin de presentar un prototipo más apegado a la versión final del videojuego.
- Explorar más alternativas relacionadas al machine learning aplicado a videojuegos y la relación existente entre este tipo de tecnología y software dedicados al entretenimiento.
- Indagar en más opciones de motores de desarrollo para videojuegos, ya sean estos enfocados en proyectos en 2D o 3D con el fin de determinar cuál de estos se adapta mejor al propósito de la investigación.
- Incluir dentro de la malla curricular de la carrera de Sistemas de información asignaturas que orienten a los futuros profesionales al desarrollo de este tipo de software, ya que en la actualidad tiene gran acogida dentro del campo laboral.

6. BIBLIOGRAFÍA.

- [1] A. MARIA, C. M. ANGEL, C. OTTO y E. FRANCISCO, *Inteligencia artificial: modelos, técnicas y áreas de aplicación*, Madrid: International Thomson Ediciones Spain, 2003.
- [2] L. Cristian, «El videojuego como herramienta educativa. Posibilidades y problemáticas acerca de los,» *Apertura*, vol. 8, nº 1, pp. 1-15, 2016.
- [3] V. Julio, C. Raquel y G. José, «Universidad Carlos III de Madrid,» 2012. [En línea]. Available: <http://ocw.uc3m.es/ingenieria-telematica/inteligencia-en-redes-de-comunicaciones/material-de-clase-1/01-historia-de-la-inteligencia-artificial>.
- [4] G. Rodrigo, «EL TEST DE TURING: DOS MITOS, UN DOGMA,» *revista de Filosofía*, vol. 63, nº 63, pp. 37-53, 2007.
- [5] A. G. Serrano, *Inteligencia Artificial. Fundamentos, práctica y aplicaciones.*, Madrid: Alfaomega, 2016.
- [6] H. Thomas, «IA: Inteligencia Artificial,» *Revista de la Universidad Bolivariana*, vol. 1, nº 2, p. 0, 2001.
- [7] A. R. V. Alvarado, «ResearchGate,» mayo 2018. [En línea]. Available: [https://www.researchgate.net/publication/338518560_INTRODUCCION_AL_MACHIN E_LEARNING](https://www.researchgate.net/publication/338518560_INTRODUCCION_AL_MACHIN_E_LEARNING).
- [8] Unity, «Unity,» Unity Technologies, 2022]. [En línea]. Available: <https://unity.com/es/products/machine-learning-agents>.
- [9] E. José, R. Contreras y S. Lluís, «3Ciencias,» [En línea]. Available: <https://www.3ciencias.com/wp-content/uploads/2013/04/videojuegos.pdf>.
- [10] J. L. Gómez, R. S. Contreras y L. Solano, «Videojuego: Conceptos, Historia, y su potencial como herramienta para la educación.,» *3Ciencias*, vol. 1, nº 2, pp. 29-42, 2013.
- [11] ESRB, «Categorías de clasificación: sugieren la edad adecuada para el juego.,» Entertainment Software Association, 2021. [En línea]. Available: <https://www.esrb.org/ratings-guide/es/>.
- [12] Unity Technologies, «Unity Documentation,» Unity Technologies, 2020. [En línea]. Available: <https://docs.unity3d.com/es/current/Manual/index.html>.
- [13] S. Ana, «AkademUS,» El Blog de Akademos, 19 julio 2019. [En línea]. Available: <https://www.akademos.es/blog/emprendedores/unreal-engine-que-es-y-para-que-sirve/>.
- [14] Adobe, «Adobe,» 2021. [En línea]. Available: <https://www.adobe.com/la/products/illustrator.html>.

- [15] Adobe, «Adobe,» [En línea]. Available: <https://www.adobe.com/la/products/animate.html>.
- [16] Adobe, «Adobe,» [En línea]. Available: <https://www.adobe.com/la/products/photoshop.html>.
- [17] Aseprite, «Aseprite,» Igara Studio S.A., 2022. [En línea]. Available: <https://www.aseprite.org/docs/>.
- [18] Microsoft, «Microsoft DOcs,» 18 enero 2021. [En línea]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
- [19] R. Angel, «OpenWebinars,» OpenWebinars S.L., 22 julio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-cpp/>.
- [20] M. César, R. Manuela y C. Beatriz, «Mentes, videojuegos y sociedad: algunos puntos cruciales para el debate,» *Guillermo de Ockham*, vol. 7, nº 1, pp. 19-30, 2009.
- [21] C. Eduardo, «Universidad Nacional de Loja,» diciembre 2015. [En línea]. Available: <https://dspace.unl.edu.ec/jspui/bitstream/123456789/10948/1/Cartuche%20Granda%2C%20Eduardo%20Luis.pdf>. [Último acceso: 12 mayo 2021].
- [22] «Visual Studio Code,» Microsoft, 2021. [En línea]. Available: <https://code.visualstudio.com>. [Último acceso: 2021].
- [23] «Aseprite,» Igara Studio, 2021. [En línea]. Available: <https://www.aseprite.org/docs/>.

ANEXOS

ANEXO A. PRESUPUESTO DEL PROTOTIPO (PUNTOS DE FUNCIÓN)

1. Requerimientos:

- Mapeado de personajes
- Mapeado basado en Tilemap
- Motor de movimiento de los personajes
- Colisión con el ambiente
- Mecánicas de Interacción con el mapa
- Limitación del Mapeado
- Gestión de armas
- Combate en relación de objetos y entorno
- Interfaz de gestión.
- Motor de Texto.
- Integración del Agente Inteligente basado en Machine learning.

2. Clasificación de los requerimientos:

Archivo lógico externo (medio): Mapeado de personajes.

Archivo lógico interno (bajo): Mapeado basado en Tilemap.

Archivo lógico interno (alto): Motor de movimiento de los personajes.

Archivo lógico interno (medio): Colisión con el ambiente.

Archivo lógico interno (alto): Mecánicas de Interacción con el Mapa

Archivo lógico interno (bajo): Limitación del Mapeado

Archivo lógico interno (alto): Gestión de armas

Archivo lógico interno (alto): Combate en relación de objetos y entorno

Archivo lógico interno (medio): Interfaz de gestión.

Archivo lógico interno (medio): Motor de Texto.

Archivo lógico externo (alto): Integración del Agente Inteligente basado en Machine learning

3. Cálculo de los puntos de función.

Tabla 16: Cálculo de puntos de función.

Parámetro	Complejidad	Peso	Cantidad	Total= cantidad*peso
Ficheros Lógicos Internos	Alta	15	4	60
	Media	10	3	30
	Baja	7	2	14
Ficheros Lógicos Externos	Alta	10	1	10
	Media	7	1	7
	Baja	5	0	0
Entradas	Alta	6	0	0
	Media	4	0	0
	Baja	3	0	0
Salidas	Alta	7	0	0
	Media	5	0	0
	Baja	4	0	0
Consultas	Alta	6	0	0
	Media	4	0	0
	Baja	3	0	0
			Total	121

4. Cálculo de factor de ajuste (VAF)

Tabla 17: Valores del factor de Ajuste.

Valor	0	1	2	3	4	5
Significado	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Tabla 18: Características generales del Sistema (GSC's).

Características Generales del Sistema (GSC's)	Cuestiones	Valor
Comunicación de datos	¿Qué necesidades de comunicación requiere el sistema para transferencia o intercambio de información?	0
Procesamiento de datos distribuido	¿Existen funciones de procesamiento distribuido? ¿Cómo son manejados los datos distribuidos?	1
Rendimiento	¿Es importante el tiempo de respuesta? ¿Es crítico el rendimiento?	4
Uso del Hardware Existente	¿En qué medida se está utilizando la plataforma hardware en donde se ejecutará la aplicación?	5
Transacciones	¿Con qué frecuencia se ejecutan las transacciones? (diariamente, semanalmente, mensualmente, etc....)	0
Entrada de datos Interactiva	¿Requiere el sistema entrada de datos interactiva? ¿Cuánta información se captura on-line? (en %)	1
Eficiencia	¿Se diseñó la aplicación pensando en que fuera eficiente y fácilmente utilizable por el usuario?	5
Actualizaciones on-line	¿Cuántos ficheros Ficheros Lógicos	0

	Internos se actualizan interactivamente (por medio de transacciones on-line)?	
Complejidad de procesamiento	¿Existe mucha carga en cuanto a procesamiento lógico y/o matemático? ¿Es complejo el procesamiento interno?	3
Reusabilidad	¿Se desarrolló la aplicación para cumplir las necesidades de más de un usuario? ¿Se ha diseñado el código para ser reutilizable?	4
Facilidad de conversión e instalación	¿Cómo son de difíciles la conversión y la instalación? ¿Se ha incluido en el diseño la conversión y la instalación?	
Facilidad de operación	¿Requiere el sistema copias de seguridad y de recuperación fiables? ¿Cómo son de efectivos y qué grado de automatización tienen los procesos de arranque, copia de seguridad y recuperación de datos?	0
Múltiples instalaciones	¿Se diseñó y desarrolló el sistema para soportar múltiples instalaciones en diferentes organizaciones?	2
Facilidad de Mantenimiento	¿Se diseñó y desarrolló el sistema pensando en facilitar el posterior proceso de mantenimiento?	0

Total: 25

5. Puntos de Función ajustados

$$PFA = 121(0.65 + (0.01) * 25)$$

$$PFA = 108.9$$

6. Cálculo del esfuerzo horas/personas (Jornada de 8 horas de una persona)

$$esfuerzo = 108.9 / \left(\frac{1}{8}\right)$$

$$esfuerzo = 871.2$$

7. Cálculo del proyecto

En promedio un equipo de desarrollo puede alcanzar 32 puntos de función al mes por lo tanto el coste de producción de una sola persona sería el siguiente:

Considerando que un mes tiene 21 jornadas en promedio, podemos determinar el número de jornadas que se necesita para producir 32 puntos de función, a saber:

Para producir 32 puntos de función se necesita:

84 jornadas de desarrolladores de software.

42 jornadas de testing.

21 jornadas de Project Management.

Por lo tanto, la productividad del equipo es:

Gerente de proyecto: 1,52 puntos de función por jornada.

Desarrollo de software: 0,38 puntos de función por jornada.

Testing de Software: 0,76 puntos de función por jornada.

Asumiendo que el sueldo percibido por cada uno de los miembros es el básico (\$425).

El costo del personal sería \$7650

El costo por unidad de medida sería:

$$Costo U. medida = \frac{7650}{32}$$

$$Costo U. medida = 239 USD$$

Por lo tanto, al poseer 109 puntos de función el coste del proyecto sería el siguiente: 26051 USD.

ANEXO B. TABLA COMPARATIVA DE METODOLOGÍAS ÁGILES.

La presente tabla tiene como propósito evaluar las principales características pertenecientes a las principales metodologías referentes al desarrollo de software junto a la metodología experimental de SUM, la misma que enfoca su metodología netamente en el desarrollo de videojuegos a diferencia de SCRUM o XP.

Tabla 19: Tabla comparativa Metodologías Ágiles.

	SCRUM	Extreme Programming (XP)	SUM
Desarrollo iterativo e incremental	x	x	x
Duración de las interacciones menor a 3 semanas		x	x
Pruebas unitarias continuas		x	x
Comunicación con el cliente	x	x	
Corrección de errores	x	x	x
Refactorización del código		x	x
Propiedad del código compartida		x	x
Simplicidad en el código		x	x
Administración del proyecto	x		x
Grupos de trabajo		x	x
Validación de tareas	x		x
Secuencia de tareas		x	x
Orientado a videojuegos			x

ANEXO C. CUESTIONARIO REFERENTE AL COMPORTAMIENTO DEL AGENTE INTELIGENTE.



Valoración del Prototipo

El presente formulario tiene como objetivo calificar aspectos relacionados al desempeño del agente Inteligente dentro del prototipo del videojuego presentado.

1. Considera que el tiempo de reacción del agente (enemigo del videojuego) es el adecuado. *

- Sí
- No
- Tal vez
- Otra...

2. Clasifique el movimiento del Agente. *

- ⋮
- Poco fluido 1 2 3 4 5 Muy fluido
-

3. ¿El Agente es capaz de evitar obstáculos? *

- Si
- No

4. ¿Considera que los agentes en el mapa son capaces de generar estrategias de acuerdo a las Interacciones del Usuario? *

- Sí
- No
- Otra...

5. ¿Cree usted que las horas empleadas en el entrenamiento del agente fueron suficientes para obtener el comportamiento esperado?

- Sí
- No
- Tal vez

6. Clasifique el nivel de interacción con el usuario por parte del agente.

- | | | | | | | |
|-------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
| | 1 | 2 | 3 | 4 | 5 | |
| No adecuado | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy Adecuado. |

7. Considera que la incorporación de machine learning dentro del prototipo se realizó con éxito.

- Sí
- No

⋮

8. ¿Que aspectos considera Usted que se pueden mejorar?

Texto de respuesta larga
