UNIVERSIDAD DE ORIENTE
NÚCLEO DE SUCRE
ESCUELA DE CIENCIAS
DEPARTAMENTO DE INFORMÁTICA

VIDEOJUEGO DEL GÉNERO ROL AMBIENTADO EN EL CASCO HISTÓRICO DE LA CIUDAD DE CUMANÁ, ESTADO SUCRE

(Modalidad Tesis de grado)

VALENTINA JOSÉ AZÓCAR GÓMEZ

CUMANÁ JULIO 2021

TRABAJO DE GRADO PRESENTADO COMO REQUISITO
PARCIAL PARA OPTAR AL TÍTULO DE LICENCIADO
EN INFORMÁTICA

CUMANÁ, 2021

VIDEOJUEGO DEL GÉNERO ROL
AMBIENTADO EN EL CASCO HISTÓRICO
DE LA CIUDAD DE CUMANÁ, ESTADO SUCRE

APROBADO POR:

_____

Profa. Carmen Victoria Romero
Asesor

_____

Profa. Ana Teresa Fuentes
Co-asesor

_____

Jurado

_____

Jurado

INDICE

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN

Se desarrolló un juego de video del género *rol* en el motor de juegos Unity, usando herramientas como Blender, Magica Voxel y Zbrush para crear los modelos 3d, Substance Designer y Krita para las texturas, el lenguaje de programación C# para la creación de los scripts de las interacciones, elementos de UI y demás funciones del software. Se utilizó la metodología SUM (Acerenza, Copes, Mesa y Viera., 2009) para videojuegos, una versión de SCRUM adaptada para el desarrollo de videojuegos. Inicialmente se realizó en el sprint inicial la distribución de las actividades, con ayuda de la aplicación de gestión de tareas Taskade, logrando en los siguientes sprints diversas versiones del producto, los assets necesarios, nuevas funcionalidades o la mejora de funcionalidades ya terminadas. Se consiguió desarrollar un producto beta luego de varias fases de prueba entre diferentes usuarios, tomando en cuenta las correcciones y sugerencias de estos en cada iteración.

GLOSARIO

API (Application programming interface): es un interfaz que se utiliza como intermediario para ayudar a que dos aplicaciones puedan comunicarse.

Enemigo: es un tipo de NPC hostil hacia el jugador que puede atacarlo y ser atacado por el jugador.

Gameplay: es un patrón de interacción entre un jugador y el video juego, definido por una serie de reglas, desafíos y recompensas.

NPC (non-playable character): es cualquier personaje que no puede ser controlado por el jugador. Los NPC tienen una gran variedad de funciones en los videojuegos, como ayudantes durante una misión, fuentes de información, aliados, enemigos o simplemente personajes de relleno.

Interfaz: se conoce como interfaz de usuario al medio que permite a una persona comunicarse con una máquina.

Ítems: objetos que pueden ser almacenados en el inventario de un personaje. Un ítem puede ser de equipo, de consumo, un objeto clave para superar una misión o simplemente un objeto inútil.

Ítems de equipo: es un ítem que puede ser equipado al personaje principal, modificando sus estadísticas iniciales, obteniendo un beneficio al realizar esta acción.

Misión (quests): objetivo del juego en un determinado nivel o área.

Modelo 3D: representación en tres dimensiones de un objeto como un personaje, un edificio o un mueble, por ejemplo.

Pixel: en las imágenes digitales, un pixel es la unidad más pequeña dentro de una imagen.

Zoom: efecto de acercamiento o alejamiento de la imagen.

INTRODUCCIÓN

Los videojuegos, al igual que otros medios de entretenimiento han probado ser mucho más que una forma de distracción, y que pueden llegar a ser un canal de aprendizaje muy valioso debido a la posibilidad que tiene el usuario de interactuar con este y de adaptar el contenido a sus gustos e intereses. Se puede definir a los videojuegos como "una prueba mental, llevada a cabo frente a una computadora de acuerdo con ciertas reglas, cuyo fin es la diversión o esparcimiento" (Zyda, 2005).

Una de las características que tienen los juegos de vídeo como material pedagógico es su disponibilidad tanto para estudiantes como para profesores, también la facilidad que ofrecen para comunicar información de forma fragmentada, cuando esta puede ser parte de un contenido más amplio, haciéndola más fácil de digerir. Los videojuegos tienen una característica fundamental; el conocimiento es adquirido de manera implícita, los jugadores no se dan cuenta que mientras juegan adquieren una serie de conocimientos concretos, sino que se van apropiando de éstos en el transcurso natural del videojuego. Cabe destacar también la flexibilidad que ofrece este medio, ya que puede ser utilizado en diversas asignaturas y áreas del saber. Algunos videojuegos educativos, que tienen como objetivo que el jugador obtenga conocimientos sobre historia de diversas culturas son Total War Attila, cuya temática era sobre el ambiente cultural y político de los últimos años del imperio romano y Ultimate General Civil War, uno de los juegos que describe con mayor exactitud la guerra civil americana.

Existen numerosos géneros y subgéneros de videojuegos, estos se clasifican de acuerdo a su mecánica de juego, la estética y la temática. Los videojuegos de aventura y de rol presentan la dominante del descubrimiento y la construcción de una experiencia narrativa a través del juego, como finalidad esencial. (Latorre, 2011). Este género tiene la peculiaridad de permitir al usuario explorar su entorno y tener la oportunidad de aprender sobre el mismo, mientras cumple los objetivos del juego. Precisamente ésta característica del género rol da un espacio para que el usuario pueda obtener

conocimientos sobre la historia y la cultura del ambiente en donde se desarrolla el juego, y es una de las formas más populares de fomentar estos saberes entre el público más joven de una forma más entretenida y fácil de digerir con respecto a otras formas clásicas de aprender sobre historia.

El género del juego de rol se caracteriza por la adopción del jugador de una máscara ficcional, el rol de su personaje, y por la búsqueda de puntos de experiencia, que se obtienen por diversas acciones meritorias en el juego y que sirven para mejorar progresivamente las habilidades del personaje. El *gameplay* del juego de rol resulta más abierta que la del juego de aventura en virtud de una estructura de misiones (*quests*) diseminadas en el juego, entre las que el jugador tiene siempre un cierto margen de libertad en su elección y/o en el orden de abordarlas (Latorre, 2011).

Usualmente este tipo de juegos se desarrolla en una época distinta a la actual y se sitúan en sitios ricos en historia y cultura para explorar. Cuando se piensa en un lugar con estas características inmediatamente se puede pensar en lugares lejanos como la antigua Roma, el Japón feudal o incluso las ciudades Aztecas, sin embargo, se cuenta con un sitio con estas características mucho más cerca, en la ciudad Cumaná, estado Sucre.

Cumaná se destaca por haber sido la primera ciudad fundada del continente americano, en 1515, rica en historia, cultura y candidata a ser nombrada patrimonio de la humanidad por la UNESCO, específicamente por su casco histórico. Éste está lleno de edificaciones con un gran valor histórico, como la iglesia Santa Inés, las ruinas del Convento de San Francisco, una edificación tipo religiosa, declarada Monumento Histórico Nacional en 1960 y el castillo de San Antonio de la Eminencia, construido por los españoles en el siglo XVII, entre otras. (Teron, 2012). Nombrando solo algunas de las edificaciones se puede apreciar el potencial que tiene esta ciudad como sitio turístico, de exploración y sitio de referencia para la historia colonial de Venezuela.

El casco histórico de la ciudad de Cumaná cuenta con ciertas características que lo hacen un escenario ideal para explorar, adentrarse en él y sumergirse en un relato ambientado en el mismo, debido a sus antiguas edificaciones, con gran contenido histórico único en el continente y relatos que merecen llegar a una audiencia mayor. Su configuración arquitectónica y cultural son aspectos que le dan un valor estratégico para la puesta en escena de un videojuego en sus localidades, haciéndolo idóneo para el desarrollo de un juego de rol, que permita el conocimiento de sus monumentos, brindando la facilidad de que un jugador en cualquier lugar geográfico pueda aprender y conocer más sobre la ciudad, permitiéndoles interactuar de manera inmersiva con sus localidades, despertando en muchos la inquietud por aprender más sobre el valor histórico de Cumaná, y entre los Venezolanos fortaleciendo su identidad cultural. La cantidad de sitios históricos de gran relevancia con que cuenta el casco histórico de Cumaná hacen que valga la pena popularizarlos en un medio en el que se puede profundizar sobre distintos temas y ser llevados a un público masivo, como los videojuegos, en el que muy pocas veces se ha mencionado a Venezuela, y menos en el que una de sus ciudades haya sido protagonista de un juego de este estilo, bajo esta premisa y con un nivel de inmersión suficiente como para que el usuario pueda aprender y vivir su historia.

Esta investigación tuvo como propósito el desarrollo de un videojuego del género rol ambientado en el casco histórico de la ciudad de Cumaná, estado Sucre. Para esto se seleccionaron herramientas como el motor de videojuegos Unity y tecnología 3D, las herramientas de modelado y escultura 3d Blender y Zbrush y otras herramientas para la creación de los assets como Krita y Substance painter. La planificación del desarrollo del videojuego se ejecutó bajo la metodología SUM con la ayuda de las herramientas de manejo de proyectos Taskade y Notion. Posteriormente se evaluó el producto desarrollado en versión beta con un grupo de muestra de potenciales usuarios.

CAPÍTULO I. PRESENTACIÓN

PLANTEAMIENTO DEL PROBLEMA

El casco histórico de la ciudad de Cumaná, tiene un gran valor cultural, no solo para la región, sino para todo el país y el continente, teniendo un gran valor histórico para la nación, que merece un mayor reconocimiento entre el público general, pero que por distintos motivos no ha logrado entrar en la cultura popular del venezolano por completo después de tantos años.

Las diversas memorias de la ciudad son poco conocidas entre los habitantes de la misma y resultan desconocidas casi por completo para personas ajenas a Cumaná debido a que la documentación sobre ésta no tiene la suficiente publicidad, o que el acceso a esta información está limitado al público. Actualmente la única forma de conseguir información histórica exacta sobre la ciudad es visitando museos o bibliotecas, en donde el acceso a la documentación disponible está limitado por los horarios de éstos, y sólo se encuentra en formato físico, resultando difícil que una persona fuera de la ciudad o el país obtenga estos conocimientos. En las instituciones educativas de la ciudad pocos espacios se dedican a tratar sobre la historia de Cumaná, porque el tema no se encuentra plasmado en el programa educativo de las asignaciones de historia y no se realizan excursiones para conocer los monumentos de la ciudad.

La falta de acceso a los materiales informativos, documentación y promulgación de la historia tienen como consecuencia que la falta de interés y el desconocimiento sobre el patrimonio de Cumaná crezcan entre las nuevas generaciones, teniendo como resultado la perdida de la identidad cultural entre ellos. Esta falta de aprecio hacia los monumentos históricos de la ciudad por parte de sus habitantes y gobernantes ha producido en estas edificaciones, muchas con más de 200 años de antigüedad, un deterioro significativo que en los últimos años se ha precipitado de tal forma que, si no se ponen esfuerzos en

mantener la identidad de los Cumaneses y rescatar estas estructuras, pueden perderse no solo físicamente, sino también, las historias y el significado que tienen cada una de ellas.

En noviembre del año 2015, la ciudad cumplió 500 años de su fundación, y debido a la relevancia de la fecha se hicieron esfuerzos por parte de los entes encargados para compartir y promulgar la historia y cultura de la ciudad, realizando diversos actos y repartiendo materiales educativos impresos como panfletos informativos con breves reseñas históricas y libros sobre el tema, sin embargo, el alcance que tuvieron se limitó a la ciudad, ya que solo se utilizaron medios tradicionales para distribuir la información, no aprovechándose el uso de medios digitales.

La escaza existencia de material digital respecto a la historia de Cumaná, y la falta de distribución de documentos relacionados con este tema, que se encuentran actualmente en posesión de los entes encargados en la ciudad, dificulta que la información alcance a un mayor público fuera de la ciudad, perdiendo una gran oportunidad en una fecha significativa, como lo fue la celebración de los 500 años de su fundación, para reforzar y expandir el conocimiento histórico sobre la ciudad de Cumaná. Aparte de esto, el material que fue difundido en esta fecha pudo haber resultado difícil de digerir para una población más joven, acostumbrada a otro tipo de medios de aprendizaje, inclinados a representar la información de una forma más visual y fácil de entender para el usuario, utilizando herramientas más actuales.

El uso de medios como páginas web, programas multimedia, materiales interactivos y videojuegos, en muchas ocasiones se han centrado en el tema de explorar culturas y sitios históricos, pero muy pocas veces se ha mencionado a Venezuela, mucho menos teniendo la oportunidad de recorrer en un juego los lugares que ya se conocen en la vida real, ambientado en una época diferente y bajo una perspectiva distinta, teniendo la posibilidad de aprender más sobre estos. El desarrollo de un videojuego de rol, con propósito educativo, de exploración, ambientado en Cumaná, aparte de entretener un público más inclinado por este tipo de medio interactivo, puede motivarlos a aprender

más sobre el trasfondo de la ciudad, creando en ellos una identidad cultural bien definida y ayudándolos a apreciar más el valor histórico y cultural de la ciudad.

ALCANCE Y LIMITACIONES

Alcance

El presente trabajo tiene como objetivo el desarrollo de un juego de video ambientado en el casco histórico de la ciudad de Cumaná, con objetivos tanto educativos como de entretenimiento para una audiencia de todas las edades.

El software en cuestión está dirigido principalmente a un público joven, con limitado acceso a la información histórica de la ciudad, que tenga interés en aprender sobre ésta y no esté familiarizado o tengan dificultades de aprendizaje utilizando medios más convencionales como libros, manuscritos, mapas, entre otros. Sin embargo, este videojuego no se encuentra limitado al alcance de la población más joven, dado que también está adaptado para ser apropiado para un público mayor, que reconocería los lugares, símbolos y referencias dentro del videojuego y también llegaría a disfrutarlo.

Hay que resaltar también que mucha de la información sobre la ciudad se encuentra en el conocimiento popular de los ciudadanos más ancianos, y que mucha de ésta información no está registrada en algún medio donde se asegure su preservación.

Limitaciones

La principal limitación de este trabajo fue la falta de material histórico sobre la ciudad, específicamente sobre las edificaciones que se seleccionaron para ambientar el juego. Aparte de eso, la recolección del material educativo tomo una parte significativa del tiempo previo al desarrollo del videojuego, que también involucro el ajuste de los datos a la historia escrita para el juego de video, y el ajuste de la frecuencia con la que los datos históricos reales se mostraban en la narrativa del mismo, amalgamando ambas de

forma que la parte educativa del juego estuviera presente de forma implícita para el usuario.

Para el desarrollo del videojuego el equipo involucrado componía de una sola persona, que cubrió las diferentes disciplinas, aparte de la programación, que el involucran la creación de un juego de video, como la animación, el sonido, el diseño de *gameplay*, la narrativa, y el arte en general. Todo esto implicó una mayor carga de trabajo y tiempo adicional para adquirir los conocimientos necesarios en las áreas antes mencionadas. Además, existieron inconvenientes relacionados al equipo utilizado para el desarrollo, ya que la mayoría del *software* involucrado en la creación del videojuego contaba con requisitos mínimos bastante altos en cuanto a *hardware*.

CAPÍTULO II. MARCO DE REFERENCIA

MARCO TEÓRICO

Antecedentes de la investigación

Muñoz (2014) realizó una investigación que lleva por título "Simulación inmersiva de la iglesia Santa Inés de Cumaná, estado Sucre, Venezuela. Ambientada al año 1929", cuyo objetivo era desarrollar dicha simulación de la iglesia Santa Inés, ubicada en el casco histórico de la ciudad de Cumaná, estado Sucre. En esta investigación se estudiaron las memorias históricas de este monumento, se diseñó y recreó digitalmente la estructura de la iglesia Santa Inés y por último se simulo un recorrido en esta localidad. El uso de tecnología 3D es un elemento clave que se comparte con esta investigación, además de que se sitúa en uno de los escenarios del videojuego a desarrollar, logrando una simulación inversiva de gran atractivo gracias a la interactividad con el usuario de la aplicación. Además de explorar la arquitectura de la iglesia santa Inés, el usuario puede aprender sobre su pasado, ganando conocimientos históricos sobre, no solo la iglesia, sino también la ciudad, ya que históricamente, este monumento es uno de los más importantes en Cumaná.

Caballero (2014) en su investigación "Aplicación multimedia para la narración de la historia de la iglesia Santa Inés en el período de 1929 hasta la actualidad", desarrolla una aplicación con este propósito. Esta investigación utilizó como metodología MOOM. Se aborda un tema bastante similar al del videojuego a desarrollar en esta investigación; la historia de parte del casco histórico, en este caso, de la iglesia Santa Inés, tomando en cuenta diversas historias que giran en torno a este monumento, de gran importancia para la ciudad, con el objetivo de llevar estas historias a un medio de información con un mayor alcance, interactivo, visual y de fácil acceso para generaciones más jóvenes.

Galindo (2016) en su investigación titulada "Desarrollo de un videojuego indie, 3d, del género *role player*", la cual tuvo como objetivo desarrollar un videojuego del género

RPG y modalidad Indie, a partir de un concepto original establecido. La investigación planteaba la falta de participación y desarrollo en el área de los videojuegos por parte de la Universidad de Oriente Núcleo de Sucre, teniendo en cuenta que en el desarrollo de un videojuego participan distintas áreas y asignaturas impartidas en el pensum de la carrera de licenciatura en informática, dictada en esta universidad. El producto de su investigación es un juego del genero RPG en versión beta que comparte características similares con el videojuego a desarrollar en esta investigación, además de utilizar herramientas similares, como el motor de desarrollo de juegos Unity, y la metodología SUM, la cual permitió un desarrollo ágil, además de que el proyecto se mantuviera organizado, esta metodología tiene la característica de que se adapta al tipo de software que fue desarrollado en esta investigación.

Jensen (2016) realizó en Unity 3D Inside, un juego plataformas de aventuras y acertijos. Los jugadores controlan a un personaje en un mundo fantasioso, resolviendo acertijos ambientales. Este juego comparte similitudes con la modalidad de juego del software a desarrollar en esta investigación.

Moldenhauer (2017) desarrolló Cuphead, este es un videojuego de acción indie, basado en el estilo clásico de los dibujos animados de la década de 1930. Este estilo característico se logró a través de animación dibujada a mano y fondos de acuarela, acompañado de una banda sonora con toques de jazz que se grabó en vivo en un estudio. El uso de la herramienta Unity fue parte clave del desarrollo de este juego. Este motor de videojuegos fue seleccionado para el desarrollo del software descrito en esta investigación, debido a su flexibilidad, compatibilidad con varias plataformas, y extensa documentación.

Bases teóricas

A continuación, se describen conceptos relacionados con las herramientas utilizadas en este proyecto de desarrollo de videojuegos.

Este trabajo fue desarrollado utilizando un motor de videojuegos, que brinda herramientas que asisten en el desarrollo de videojuegos brindando funciones que se encargan de tareas tales como el renderizado, la detección de colisiones entre objetos, el manejo de la física del juego, entre otras. Existen actualmente muchos motores de videojuegos en el mercado, entre los más populares están Unity, Unreal. Godot y Cry Engine.

El motor seleccionado fue Unity 2020.1.12f1, debido a las facilidades que ofrece al momento de desarrollar y a la extensa documentación con la que cuenta. Este motor brinda un motor de simulación de físicas, facilidades y flexibilidad a la hora de crear el ambiente del juego, un punto muy importante para este proyecto, ya que se tenía como uno de los objetivos crear una versión estilizada de los edificios reales en los cuales fue inspirado el juego. Para el desarrollo de las interacciones en el videojuego, Unity utiliza un sistema de scripts que se agregan a los componentes en el editor de Unity. Estos scripts utilizan el API de Unity y el lenguaje de programación C#.

La unidad fundamental de trabajo en Unity son los GameObjects, estos pueden representar cualquier tipo de objeto dentro de una escena dependiendo de los componentes que le conformen, los cuales pueden cambiar y definirse de acuerdo a los requerimientos necesarios para el objeto creado. En Unity, las áreas de espacio son conocidas como escenas, en estas se pueden crear instancias de los objetos interactivos, ambientaciones y usarse para crear un juego completo o un nivel de un juego por escena, pudiendo crearse tantas escenas como sea necesario. Por default, una escena cuenta únicamente con una cámara, la cual es un componente que crea una imagen desde un ángulo determinado. Para crear una ambientación personalizada en una escena es necesario que sean importados assets a la escena. Los assets son archivos de data en distintos formatos, usualmente creados con otros programas, estos pueden ser modelos 3D, imágenes, sonidos o cualquier otro tipo de archivo que pueda ser usado para dar forma a los objetos dentro del juego. El flujo de trabajo usual para la creación de objetos que forman parte de una escena de Unity, es crear un GameObject a partir de un asset

exportado y agregarle los componentes necesarios de acuerdo a su propósito en el juego. En muchas ocasiones es necesario reutilizar los GameObjects en varias escenas de un juego o incluso crear varias instancias dentro de la misma escena, por ejemplo, cuando se trata de un objeto que se repite constantemente durante todo el nivel, en el caso de cierto tipo de árbol, o un poste de luz. Para estos casos Unity tiene una herramienta ideal para realizar este trabajo; los prefabs, estos actúan como plantillas de GameObjects que pueden reutilizar la data de los assets, las configuraciones de los componentes agregados y ayuda a mantener mayor grado de control entre los assets del proyecto. Los objetos agregados a una escena pueden ser organizados en la ventana de jerarquía de Unity, donde los GameObjects pueden ser reordenados, eliminados o agregados a otros objetos haciéndose hijos de los del nivel superior.

Unity utiliza render pipeline, para generar los contenidos de una escena del juego e imprimirlos en la pantalla. Un render pipeline cumple la función de un motor gráfico y se encarga de ejecutar una serie de operaciones que toman el contenido de una escena y lo muestran en la pantalla. Esto ejecuta operaciones tales como culling, rendering, y post-processing, este último agrega efectos y filtros a la imagen de la cámara antes de que aparezca en pantalla.

En Unity se puede elegir diferentes render pipelines, las cuales tienen diferentes capacidades y características de rendimiento, que se ajustan a diversos proyectos, aplicaciones y a las plataformas para las cuales se desea desarrollar. Para los fines de este proyecto se seleccionó el Universal render pipeline (URP).

URP es un render pipeline que provee flujos de trabajo que permiten crear gráficos optimizados y a la vez vistosos. Tiene la característica de que los ejecutables generados que usen este render pipeline puede funcionar de forma fluida y sin problemas en un gran rango de plataformas, desde dispositivos móviles hasta computadoras de gama alta y consolas de videojuegos.

Se seleccionó este pipeline debido a que el proyecto apuntaba dispositivos de escritorio desde gama media hasta gama alta, y este pipeline tiene como característica presentar un gran rendimiento en un extenso rango de dispositivos y a la ves gran fidelidad de los gráficos. El URP utiliza un acercamiento a los shaders distinto al que usa el pipeline de renderizado default de Unity, en lugar de esto, el URP utiliza un set de shaders propios.

Los shaders son pequeños scripts que contienen los cálculos matemáticos y algoritmos para calcular el Color de cada píxel procesado, en función de la entrada de iluminación y la configuración del Material.

Los materiales definen cómo se debe renderizar una superficie, incluyendo referencias a las Texturas que utiliza, tintes de color y más. Las opciones disponibles para un Material dependen del shader que el Material esté usando.

Texturas son imágenes. Un material puede contener referencias a las texturas, de modo que el shader del Material pueda usar las texturas al calcular el color de la superficie de un GameObject, también conocido como objeto del juego. Además del color básico (Albedo) de la superficie de GameObject, las texturas pueden representar muchos otros aspectos de la superficie de un material, como su poder de reflejo o rugosidad.

Los materiales especifican que Shader van a utilizar para la renderización de un objeto, y el Shader especifica cómo se comporta este material con la luz, que textura espera utilizar. Estos componentes trabajan en conjunto dentro del pipeline de renderizado para poder proyectar en la pantalla los gráficos y sus interacciones.

Los GameObject siempre tienen un componente de transformación, llamado Transform, el cual representa la posición y la orientación del GameObject en el escenario del juego. Este es el único componente fijo de todos los GameObject, los otros componentes pueden ser agregados luego para darle funcionalidades adicionales al GameObject dependiendo del propósito que se tenga con ellos.

Para manejar las físicas que tiene un GameObject en el escenario son importantes componentes como rigidbody y collider.

El Rigidbody controla la posición de un objeto mediante la simulación de física. Agregar este componente a un GameObject pondrá el movimiento de este bajo el control del motor de físicas de Unity.

Un collider es un componente que es añadido a los objetos que requieran movimiento para manejar la detección de colisiones en un GameObject.

Una vez que es posible mover los modelos con estos componentes, es necesario agregarles animaciones para darle más vida a la ambientación y personajes del juego, para esto se emplean una serie de componentes de Unity como son los animator controllers, las cuales son herramientas que ayudan a programar los clips de animaciones de un objeto por medio de máquinas de estado que definen la animación que se debe de ejecutar en cada momento dependiendo de ciertos parámetros que se definan. Los clips de animaciones usualmente son importados al proyecto cuando se agregan assets de modelos 3D creados con otras aplicaciones a Unity.

Los modelos para este proyecto fueron creados con distintos gráficos 3D, principalmente Blender, que fue utilizado para crear los modelos de los personajes, ítems y objetos decorativos. Blender es un programa de creación de assets 3D dedicado al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales en general.

Otro programa utilizado para los assets 3D fue MagicaVoxel, un software que permite crear modelos low poly, es decir, de una cantidad baja de polígonos e ideal para la optimización del juego, utilizando formas básicas de cubos conocidos como Voxels.

Para lograr un mayor detalle en los modelos low poly fueron utilizados los programas Zbrush, para hacer esculturas high poly a partir de los modelos base, es decir versiones

de una gran cantidad de polígonos y detalles de los modelos de bajos polígonos, y generar mapas de normales que contengan los datos de los modelos high poly, que luego fueron aplicados a materiales de los modelos low poly, conservando el detallado y optimizando la cantidad de polígonos en los objetos en las escenas de Unity. Además del mapa de normales, fue necesario crear texturas personalizadas para los modelos low poly, estas fueron hechas con el programa Substance painter, el cual permite pintar texturas en modelos 3D agregándole gran cantidad de detalles y estilización.

Muchos de los modelos 3D, sobretodo de los personajes interactivos y el personaje principal fueron realizados en Blender en la herramienta de pose, para esto fue necesario primer agregar un armature a los modelos base, estos componentes son similares a un esqueleto que se agrega al modelo tridimensional para agregarle articulaciones y poder moldearlos de acuerdo a una pose determinada. El proceso de agregar un armature a un modelo 3D es conocido como rigging. En el modo pose el proceso para crear la animación de un modelo, usualmente, es situarse en un frame del timeline, es decir, en una secuencia de la línea de tiempo de la herramienta de animaciones de Blender, cambiar la pose del personaje a la deseada, y capturar la posición y rotación de los huesos del armature. Este proceso se repite por cada frame que sea necesaria para la animación, un método parecido al de la animación clásica de stop motion, donde cada vez que el modelo cambia de pose, se toma una foto, y al juntar todas las fotos se genera una pelicula donde el modelo “se mueve”. De esta misma forma, al juntar todas las frames, es generada la animación deseada para el modelo.

Para la creación de este proyecto, aparte de los modelos 3D, fue necesario crear imágenes de dos dimensiones para la interfaz gráfica de usuario del juego. Las imágenes de ítems, personajes no jugable, fondos y decoraciones fueron creadas con Krita, un software de pintura digital 2D. Para diseñar las interfaces fue empleado el programa Adobe XD, una herramienta de diseño basada en vectores, utilizada para diseñar aplicaciones para distintos dispositivos.

Aparte de estos programas, para el desarrollo de este proyecto fueron utilizados paquetes adicionales de Unity descargados desde la asset store. Los paquetes de Unity son contenedores compuestos por varios assets, herramientas del editor de Unity, plantillas y colecciones de imágenes, sonidos, scripts, efectos y otros tipos de archivos. Los paquetes utilizados en este proyecto fueron Cinemachine, el cual es un paquete que cuenta con una serie de scripts ayuda al suavizado de la cámara y a agregarle funciones para cambiar ángulos y el zoom de acuerdo a los requerimientos necesarios en la escena. ProBuilder, que es una herramienta que permite crear prototipos 3D rápidamente dentro del mismo Unity, sin necesidad de importar modelos desde otros programas. Y ProGrids, una herramienta que crea un patrón de rejillas para asistir al desarrollador a colocar objetos dentro de la escena para mantener medidas y distancias constantes entre estos y evitar que la escena se vea desproporcionada.

Para la organización del proyecto fueron utilizadas Notion, una aplicación web de notas y creación de documentos, donde fueron creados los documentos de diseño del juego y guiones, y Taskade, una herramienta de organización de proyectos donde se gestionó el progreso del juego y las tareas concretadas y pendientes.

MARCO METODOLOGICO

Metodología del área aplicada

La metodología que se utilizó en el desarrollo de esta investigación fue SUM para desarrollo de videojuegos, Acerenza, et al., 2009.

La metodología SUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo (http://www.gemserk.com/sum/). SUM fue concebida para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente.

Esta metodología fue seleccionada ya que brinda flexibilidad para definir el ciclo de vida y puede ser combinada fácilmente con otras metodologías de desarrollo para adaptarse a distintas realidades. Tiene la ventaja de permitir el desarrollo del producto en tiempos cortos, con controles de calidad y manejo eficiente de los recursos.

Los roles en esta metodología guardan relación con los roles de la metodología Scrum. Esta metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador beta. El productor interno y el cliente se corresponden en forma directa con los roles de *Scrum master y product owner* de Scrum respectivamente. El equipo de desarrollo tiene las características del *Scrum team*, pero a diferencia de Scrum se definen subroles dentro del equipo, que para propósitos de esta investigación serán asumidos por una sola persona. Estos subroles son; el diseñador del juego, quien tiene como tarea diseñar el gameplay, historia, ambientación, personajes, niveles y todos los elementos que hacen a la experiencia del jugador. El programador tiene como principal responsabilidad implementar el software que compone al juego. El artista sonoro, encargado de los efectos de sonido y la ambientación. Finalmente, el artista gráfico,

debido a que el arte y la animación son gran parte del trabajo requerido para el desarrollo del videojuego. El rol de verificador beta tiene como responsabilidad realizar la verificación funcional del videojuego y comunicar su resultado.

El proceso de desarrollo se divide en cinco fases iterativas e incrementales que se ejecutan en forma secuencial.

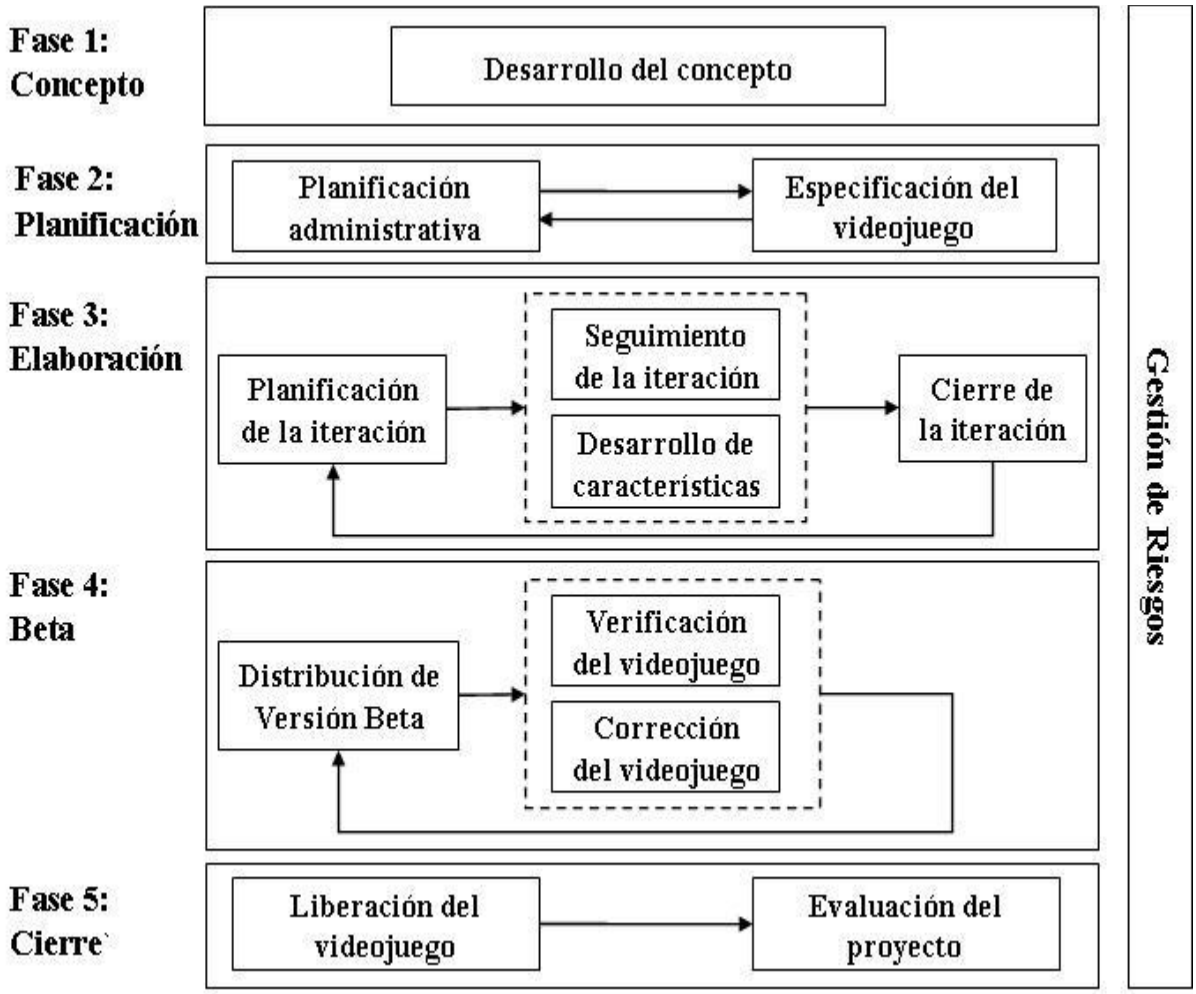


Figura 1: Flujo de trabajo de la metodología
Fuente: SUM (http://www.gemserk.com/sum/)

Fase 1: concepto. En esta fase es donde se define y acuerda el concepto de juego, se definen los aspectos técnicos y los aspectos de negocios. El concepto se divide en tres partes. La primera es definir los aspectos del juego, es decir la definición de la visión del

juego, el género, el gameplay, las características principales con las que contará el videojuego, su historia y ambientación. La segunda parte trata sobre definir los aspectos técnicos, tales como la plataforma que se tiene como objetivo, las tecnologías y las herramientas que se utilizaran. Finalmente, la última parte de la fase de concepto se definen los aspectos de negocios, tales como el modelo de negocios y el público objetivo.

Fase 2: planificación. Esta cuenta con dos fases, la primera es la planificación administrativa, donde se define el equipo de desarrollo, el cronograma, presupuesto y objetivos del proyecto. La segunda fase es la especificación del juego, donde se estiman las características propias del videojuego.

Fase 3: elaboración. Esta fase ocurre por iteraciones donde en cada una, se planifica la iteración correspondiente, se desarrollan las características y se cierra la iteración, dando inicio a la siguiente.

Fase 4: beta. En esta fase se eliminan la mayor cantidad de errores posibles y se evalúan distintos aspectos no funcionales del videojuego.

Fase 5: cierre. La fase final de la metodología, donde se pone a disposición del cliente la versión final del videojuego y son realizadas las evaluaciones del proyecto.
Esta metodología se escogió debido a su pertinencia en el área de los videojuegos, la adaptabilidad que tiene, y el desarrollo ágil propio de la metodología Scrum, dando resultados positivos en aplicaciones previas de desarrollo de videojuegos.

CAPÍTULO III. ELABORACIÓN

En este capítulo se describen cada una de las fases de la metodología SUM con enfoque al desarrollo de este proyecto.

FASE 1: CONCEPTO

En esta fase se describe el concepto general y visión del videojuego.

Definición de aspectos del juego

Se propuso el desarrollo de un videojuego del genero rol o RPG, principalmente, con características de los géneros aventura, puzle, y misterio; que tuviera el objetivo de motivar a la exploración de los escenarios del juego, guiado por la historia que se quiere contar en el mismo.

Visión del juego

Se desea que el videojuego brinde al jugador la experiencia de explorar edificios históricos de la ciudad, realizando quests, resolviendo puzles y sumergiéndose en la ambientación creada en el videojuego. El juego debe ofrecer una historia atractiva donde el jugador pueda identificarse con las acciones del personaje principal e intentar cumplirlos objetivos del juego, mientras avanza en la historia hasta el desenlace de esta.

Género

Se seleccionó como genero principal para el juego el de rol o RPG, por los sistemas de ítems, inventarios y misiones.

El diseño de juegos RPG suele ser bastante complejo ya que entrelaza narrativa, diseño de misiones o quests, y diseño de niveles. L quests dictan el escenario y contenido del nivel, además de servir como transporte para la historia del juego. Los niveles les proveen retos que superar a los jugadores en forma de quests. (Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games Gillian Smith, Ryan

Anderson, Brian Kopleck, Zach Lindblad, Lauren Scott, Adam Wardell, Jim Whitehead, and Michael Mateas, 2011)

Adicionalmente se integran al juego características de los géneros aventura ya que es un género que se envuelve en torno a la historia que se quiere contar en el videojuego; puzle ya que cuenta con mini juegos de acertijos y finalmente el género de misterio.

Otro videojuego que pertenecen a este género y cuenta con características similares es Chrono Trigger (1995), diseñado por Hironobu Sakaguchi. Que combina diversos rasgos de varios géneros, pero mantiene como genero principal el RPG.

Gameplay

En el videojuego el jugador moverse libremente en un entorno tridimensional. La cámara sigue al personaje y el ángulo de esta es manejado por el jugador con ayuda del mouse, con el cual también se controla la dirección del movimiento, tal como en los juegos en primera persona.

El jugador puede realizar diversas acciones para interactuar con los elementos que lo rodean al entrar en un rango definido por el objeto interactivo. Entre estos objetos interactivos están otros personajes NPC (non-player carácter), con los cuales puede conversar y los cuales serán vitales para brindarle datos importantes para la historia. También puede interactuar con edificios a los cuales puede entrar y con objetos que puede recoger para guardar en el inventario.

Los obstáculos que se encontrará serán las diferentes quests, o misiones que tendrá que completar en los diferentes niveles, de las que obtendrá rewards, ítems importantes y aprenderá más sobre los aspectos menos conocido de la historia de los monumentos, los cuales representan los niveles del juego.

En cuanto a la movilidad se tomó de referencia Metal Gear Solid (1998) creado por Hideo Kojima. Mientras que para el Sistema de inventario y misiones se tomó como inspiración The Legend of Zelda: Majora's Mask (2000) diseñado por Shigeru Miyamoto.

Características

El jugador podrá explorar libremente el casco histórico de Cumaná, el cual será la escenografía principal y donde más tiempo se ha de invertir en modelado, estilizado y desarrollo.

Juego impulsado por la historia. El jugador seguirá el transcurso de los hechos narrados dentro del videojuego y deberá completar ciertos objetivos para ir avanzando en la historia.

Interacción con personajes NPC, edificios, y objetos clave dentro del juego

Historia y ambientación

En el videojuego desarrollado el jugador deberá explorar edificios históricos, enfrentándose a los fantasmas que los habitan o completando misiones para ayudarlos a descansar. Está ambientado en el casco histórico de Cumaná y en sus edificios en diferentes periodos de tiempo, de tal forma que el jugador puede explorar estos monumentos en diferentes épocas.

La historia está enfocada y se cuenta desde la perspectiva de Pan, el personaje principal. Pan es un estudiante de segundaria, que debe presentarse en las noches de antaño en una obra de teatro. Las noches de antaño es una celebración anual que se realiza en Cumaná, con motivo del aniversario de la ciudad, durante esta fecha es costumbre que los asistentes usen trajes de la época, y realicen actividades culturales relacionadas al aniversario de la ciudad. Pan se ve accidentada de llegar hasta el sitio donde se realizará la obra de teatro y se pierde en el castillo Santa María de la Cabeza, donde comenzara su aventura en el videojuego, y es el primer edificio a explorar. Una vez ahí se da cuenta de

que de alguna forma regreso en el tiempo a la época colonial y que el lugar está habitado por los fantasmas que solían vivir en el castillo. En este sitio deberá resolver puzles y completar quests para lograr salir, encontrar una solución y volver a su época.

El escenario donde se desarrolla la historia es en el casco histórico de Cumaná, sitio donde se llevan a cabo las noches de antaño cada año. Este ambiente fue replicado de manera detallada y estilizada, manteniendo un estilo caricaturesco que coincida con el estilo de los personajes humanoides del videojuego.

Definición de aspectos técnicos

Plataformas

El videojuego estará disponible para computadoras con el sistema operativo Windows y varias distribuciones de Linux.

Tecnologías y herramientas

Como motor de juegos se seleccionó Unity versión 2020.1, el cual permite generar versiones ejecutables para las plataformas deseadas y brinda gran libertad al momento del desarrollo.

El software de modelado 3D usado fue Blender versión 2.9, que permite exportar modelos a diversas plataformas, y que, al ser un programa de código abierto, brinda muchas herramientas de aprendizaje y complementos que son de gran ayuda de acuerdo a las especificaciones requeridas para los modelos. Adicionalmente para el modelado de la mayoría de los edificios se usó el programa de modelado 3d de Voxel (cubos de construcción) Magica Voxel, el cual ayudo a agilizar la creación de la gran cantidad de modelos requeridos para las casas del escenario del casco histórico de Cumaná. Para los modelos 3d más detallados se utilizaron los programas Substance painter para las texturas y Zbrush, para escultura 3d. Los programas Krita y DragonBones fueron utilizados para las ilustraciones 2d y animaciones de la introducción de la historia.

Definición de aspectos de negocios

Modelo de negocios

La versión final del videojuego esta ideada para el modelo de negocios free to play, el cual permite a los jugadores descargar y jugar sin necesidad de realizar algún tipo de pago. Este modelo de negocios es común en las primeras versiones de videojuegos del genero indie, ya que permite al desarrollador obtener feedback y conocer mejor a su público objetivo, también permite la publicación de contenido extra relacionado al juego.

Público objetivo

El juego está orientado a un público joven, familiarizado con videojuegos de rol y aventura como The Legend of Zelda, aunque el propósito es que pueda ser disfrutado por jugadores de cualquier edad. Es un juego que requiere que el jugador esté dispuesto a explorar. En cada quest se van revelando fragmentos de la historia y cuentos poco conocidos, tocando el factor humano de la historia, los personajes, sus motivaciones y se intenta ir más allá de contar la historia como un libro, haciendo que el personaje del jugador se sienta parte de ella y explotando su curiosidad.

FASE 2: PLANIFICACIÓN

Especificación del videojuego

En esta sección se describen las características que se quieren desarrollar en la versión final del videojuego. Al ser un juego del genero rol debe contar con rasgos comunes de esta categoría de videojuegos, como lo es un sistema de vida del personaje, sistema de inventario y sistema de lucha.

Especificación de características

Características funcionales:

El personaje principal es controlado por jugador de manera libre.

Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario.
Objetos interactivos del juego
Inventario y uso de ítems.
Sistema de lucha con enemigos.
Ambientación relacionada a la temática del juego.

Características no funcionales:
Optimización para equipos de gama baja y media.
Fluidez al momento de ejecutar el videojuego.
Interfaz gráfica fácil de usar.
Modelos y animaciones.
Iluminación y graficas atractivas y estilizadas.

Criterios de evaluación

Tabla 1: criterios de evaluación

Característica	**Criterios de evaluación**
Control del movimiento y dirección del personaje principal.	El jugador puede cambiar la dirección del personaje con el mouse. El jugador hace que el personaje se mueva con las teclas de movimiento y hace que salte con la barra de espacio. Tiempo de respuesta rápido a la entrada del usuario.
Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario.	El jugador puede mover la cámara libremente con el mouse para detallar su entorno.

Objetos interactivos del juego.	Diálogos de personajes NPC. Sistema de patrullas de los NPC. Interacción con edificios, ver datos sobre un edificio, entrar al edificio.
Inventario y uso de ítems	Recoger ítems del escenario y guardarlos en el inventario. Equipar ítems del inventario. Eliminar ítems del inventario. Interfaz gráfica del inventario.
Sistema de lucha con enemigos	Atacar oponentes. Recibir daño y verlo reflejado en la barra de vida. Obtener curas y recuperar puntos de vida.
Ambientación relacionada a la temática del juego	Uso de modelos low poly. Crear modelos estilizados basados en los edificios reales del casco histórico. La estructura de la escenografía debe estar inspirada en las calles del casco histórico. Texturas estilizadas. Bloquear salida del escenario. Colisiones con el entorno.
Optimización para equipos de gama baja y media	Fluidez al momento de ejecutar el videojuego Al menos 30 FPS.
Interfaz gráfica	Diseño intuitivo. Alto contraste en la interfaz para fácil visualización. Fuentes legibles. Interfaz general del juego que refleje los

	estados actuales del personaje. Menú de pausa. Menú de inicio.
Modelos y animación	Modelos de ítems. Modelos humanoides básicos. Ridging de modelos de personaje. Animación de modelos de personajes. Modelos de objetos decorativos del juego. Texturas.
Iluminación y graficas atractivas y estilizadas	Sombreado correcto. Efecto de miniatura. Iluminación global.

Estimación de características

Estimación en tiempo de desarrollo de cada una de las características funcionales y no funcionales.

Tabla 2: Estimación de características

Característica	**Tiempo**
Control del movimiento y dirección del personaje principal	1 semana.
Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario	1 semana.
Objetos interactivos del juego	2 semanas.
Inventario y uso de ítems	2 semanas.
Sistema de lucha con enemigos	1 semana.
Ambientación relacionada a la temática del juego	4 semanas.

Interfaz gráfica	1 semana.
Modelos y animación	3 semanas.
Iluminación y graficas atractivas y estilizadas	1 semana.

Priorización de características

La ponderación de las características se basa en la ponderación de la metodología SUM que a su vez está basada en la Scrum, en esta se le asigna a cada característica un numero de la escala Fibonacci según su importancia, siendo los más importantes los que tienen un número más alto. Adicionalmente según la metodología SUM, los elementos que conforman el gameplay tienen mayor importancia y por lo tanto deben tener una ponderación mayor, en este caso, el control del personaje y las características relacionadas con interacciones e historia del juego. Las siguientes en importancia son las características basadas en la creación de modelos para el entorno, la interfaz gráfica y la ambientación, y por último los elementos decorativos del juego.

Tabla 3: Ponderación de características

Característica	**Ponderación**
El personaje principal es controlado por jugador de manera libre	377
Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario	233
Objetos interactivos del juego	144
Inventario y uso de ítems	233
Sistema de lucha con enemigos	55
Ambientación relacionada a la temática del juego	55
Interfaz gráfica	89
Modelos y animaciones	34
Iluminación y graficas atractivas y estilizadas	8

Planificación administrativa

1 Objetivos del proyecto

Después de determinar las características se pueden obtener los objetivos a desarrollar para este proyecto.

1. Desarrollar un controlador para el personaje principal que cumpla con los criterios deseados y utilice los controles mencionados.
2. Agregar una cámara que siga los movimientos del personaje y responda a los controles deseados. Además, ajustar la cámara para evitar que choque con objetos del juego.
3. Crear prefabs de los objetos interactivos y NPC del juego. Desarrollar los scripts necesarios para que, al estar en un rango de contacto con el jugador, los NPC u objetos interactivos disparen la interacción deseada.
4. Crear prefabs para los ítems del juego. Crear los scripts necesarios para el manejo de los ítems del inventario y la estructura de datos que permitirá guardar estos ítems.
5. Crear controlador y scripts para el manejo de la vida del personaje, sus estadísticas de ataque y defensa. Adicionalmente, manejar las estadísticas para los NPC enemigos.
6. Crear interfaces graficas necesarias.
7. Diseñar y crear los elementos del escenario y ambientación para el juego, de acuerdo con el concepto artístico previamente establecido.
8. Liberar beta del juego.

Criterios de evaluación:

Obtener una beta jugable que pueda ser probado por un grupo de jugadores y reciba aceptación entre ellos.

Brindarles a los jugadores una experiencia de juego entretenida y educativa.

Ambientación y edificaciones del juego deben ser fácilmente reconocibles para los jugadores que conozcan el lugar físico en el que está inspirado.

2 Equipo de desarrollo

Para el desarrollo de este proyecto el equipo estuvo conformado por una sola persona que asumió todos los roles establecidos en la metodología, excepto el de verificador de la versión beta. Se realizaron las tareas correspondientes a los roles desde la parte de diseño de juego, historia, creación de modelos 3D y desarrollo en Unity.

3 Cronograma

En esta sección fueron definidos el cronograma de desarrollo y el tiempo de duración de los sprints.

Cronograma de elaboración

Los sprints tendrán una duración de dos semanas, como es de costumbre en la metodología SCRUM. Las características que se estime que duren más de dos semanas serán divididas en dos partes, como es el caso de las características 6 y 9. Los puntos de estas categorías fueron divididos entre sus tareas en puntajes que pertenecen a la escala Fibonacci, utilizada en Scrum.

Para el caso de la característica 6: “Ambientación relacionada a la temática del juego” se estimaron 4 semanas, es decir dos sprints para su desarrollo. Debido a esto se ha dividido en dos tareas:

1. Estructura básica de calles y aceras del casco histórico. Luego de buscar referencias y armar bocetos del mapa general, se procedió a modelar el mapa principal del casco histórico. Los modelos que conforman el mapa del escenario principal fueron realizados dentro de Unity con la herramienta de modelado ProBuilder. Se modelaron las calles principales y se marcaron las zonas donde van los edificios, postes y vegetación. A esta base 3D se le agregó una grilla

proporcionada por la herramienta ProGrids para mantener las medidas y distancias de forma constante. Esta tarea se ponderó con 34 puntos.

2. Modelos de casas y edificaciones históricas. Para este sprint las herramientas principales fueron Magica Voxel para casas y estructuras básicas y Blender para los detalles de las estructuras, los tejados, puertas, ventanas, rejas, vegetación y postes de luz. Esta tarea se ponderó con 21 puntos.

En el caso de la característica 9: "Modelos y animación" se estimaron 3 semanas o un sprint y medio. Esta característica fue divida en dos modulos:

1. Modelos humanoides básicos para el personaje principal y NPC, Ridging del modelo humanoide. También se realizaron animaciones para el modelo humanoide tales como, la animación en modo de reposo, caminando, corriendo, saltando y hablando. Texturas para el personaje principal y los NPC. Esta tarea se ponderó con 21 puntos.

2. Modelos de ítems, objetos interactivos, objetos decorativos de la ambientación, texturas de modelos decorativos. Esta tarea se ponderó con 13 puntos.

La cantidad de sprints planificados para el desarrollo resulto ser de 8, lo que serían 16 semanas. Siendo el sprint inicial asignado a la característica 1, ya que fue la que obtuvo la mayor ponderación por su importancia para el gameplay.

Tabla 4: Cronograma de sprints

Sprint	**Fecha de inicio**	**Fecha de culminación**	**Descripción**	**Puntos**	**Puntos totales**
1	02/11/2020	15/11/2020	El personaje principal es controlado por jugador de manera libre	377	610

			Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario	233	
2	16/11/2020	29/11/2020	Inventario y uso de ítems	233	233
3	30/11/2020	13/12/2020	Objetos interactivos del juego	144	144
4	14/12/2020	27/12/2020	Interfaz gráfica	89	144
			Sistema de lucha con enemigos	55	
5	28/12/2020	10/01/2021	Estructura básica de calles y aceras del casco histórico.	34	34
6	11/01/2021	24/01//2021	Modelos de casas y edificaciones históricas.	21	21
7	25/01/2021	07/02/2021	Modelos humanoides, Ridging, animaciones y texturas	21	21
8	08/02/2021	21/02/2021	Modelos y texturas de ítems y objetos decorativos	13	21
			Iluminación y graficas atractivas y estilizadas	8	

Cronograma del beta

La fase del beta estará compuesta por 2 iteraciones, cada una de dos semanas. En cada sprint habrán 4 días de prueba, recepción del feedback de los testers y sus reportes, y el resto de los días para la corrección de los errores detectados durante las pruebas.

Tabla 5: Cronograma del beta

Sprint	Tipo de actividad	Fecha de inicio	Fecha de culminación
1	Pruebas, feedback y corrección de errores detectados	22/02/2022	07/03/2021
2	Pruebas, feedback y corrección de errores detectados	08/03/2021	21/03/2021

En la segunda iteración no debería de haber errores que afecten la jugabilidad. Si al concluir los 4 días de prueba no se reportan errores nuevos, se puede dar por concluida la fase beta, de lo contrario se deberá programar un nuevo sprint para esta fase.

Cierre del proyecto

La finalización del proyecto se estimó para luego de dos semanas de la última iteración de la fase beta, es decir el 03 de abril del año 2021.

Definir hitos

En esta sección se exponen los hitos de la evolución del proyecto.

Tabla 6: hitos

Hito	Descripción	Fecha
1	Personaje totalmente controlable	13/11/2020
2	Inventario funcional	27/11/2020
3	El jugador es capaz de interactuar con edificios y NPC	13/12/2020
4	Producto básico funcional listo para probar	26/12/2020

5	Modelos de personaje principal y NPC modelados y con texturas finalizadas	07/02/2021
6	Escenario final terminado con detalles decorativos y efectos visuales	21/02/2021
7	Videojuego funcional y listo para la fase beta	21/02/2021
8	Videojuego con correcciones listo para ser liberado	27/03/2021

FASE 3: ELABORACIÓN

Sprint 1

Fecha de inicio: 02/11/2020
Fecha fin: 15/11/2020

Descripción

Control del personaje principal es controlado por el jugador.
Control de cámara por parte del jugador, para aumentar la sensación de exploración y que el jugador pueda detallar más el escenario.

Planificación

Objetivos

Crear GameObject para el personaje del jugador con los componentes necesarios.
Controlar al personaje con las teclas de movimiento y que salte con la tecla de espacio.
Cambiar la dirección del personaje y la vista de la cámara usando el mouse.

Métricas de los objetivos

Tiempo de respuesta rápido a la entrada del usuario.
Comportamiento correcto de las colisiones al andar en un terreno con obstáculos.

Seleccionar y refinar características

Tabla 7: características sprint 1

1	Inicializar el proyecto.
2	Crear GameObject del personaje del jugador.
3	Controlar movimiento del personaje con el teclado.
4	Controlar cambios de dirección del personaje con el mouse.
5	Agregar cámara que siga al personaje principal.

Seguimiento y desarrollo del sprint

Característica 1: Inicializar el proyecto

Desarrollo

El proyecto de Unity fue creado en URP con los ajustes necesarios para las plataformas que se tenían como objetivo. Luego se descargaron los paquetes requeridos para este proyecto, como Cinemachine, ProBuilder y ProGrids.

Problemas

La conexión inestable a internet dificultó descargar algunos de los paquetes necesarios.

Estado de la característica

Completada el 03/11/2020.

Puntos quemados

89.

Característica 2: Crear GameObject del personaje del jugador

Desarrollo

Se creó el GameObject con los componentes necesarios, como el Rigidbody y el Capsule Collider Estos componentes permiten que el personaje sea afectado por las físicas del mundo simulado en el juego, y pueda moverse.

Problemas

Ninguno.

Estado de la característica

Completada el 04/11/2020.

Puntos quemados

89.

Característica 3: Controlar movimiento del personaje con el teclado

Desarrollo

Se agregaron los scripts necesarios para tomar las entradas del teclado y comunicar al personaje del jugador la dirección que debería tomar.

Problemas

Ninguno.

Estado de la característica

Completada el 07/11/2020.

Puntos quemados

144.

Característica 4: Controlar cambios de dirección del personaje con el mouse

Desarrollo

Al script de movimiento se le agregó la función de tomar como entrada el movimiento del mouse y rotar horizontalmente al personaje para cambiar su dirección.

Problemas

Ninguno.

Estado de la característica

Completada el 12/11/2020.

Puntos quemados

144.

Característica 5: Agregar cámara que siga al personaje principal

Desarrollo

En la jerarquía de objetos de Unity, al GameObject del personaje se le agregó un GameObject del tipo cámara. Y a su vez, al GameObject cámara se le agrego un componente Cinemachine, el cual es un paquete que cuenta con una serie de scripts ayuda al suavizado de la cámara y a agregarle funciones para cambiar ángulos y el zoom de acuerdo a los requerimientos necesarios en la escena.

Problemas

Detalles de colusión de la cámara con objetos muy altos cuando la cámara sigue al personaje. Estos detalles fueron resueltos durante este sprint.

Estado de la característica

Completada el 13/11/2020.

Puntos quemados

144.

Cierre de la iteración

Se realizaron las pruebas necesarias con el personaje del jugador moviéndose en un entorno de pruebas con objetos con los que podía colisionar y los resultados fueron los

esperados. Surgieron detalles que no se habían tomado en cuenta, como por ejemplo la colisión de la cámara con objetos dentro del escenario, pero estos detalles se solucionaron en este sprint.

Evaluación del estado del juego

En este sprint se obtuvo un personaje jugable totalmente manejable por el jugador. Esto es de gran importancia para el resto del desarrollo y para las pruebas de juego que se realizarán en cada sprint.

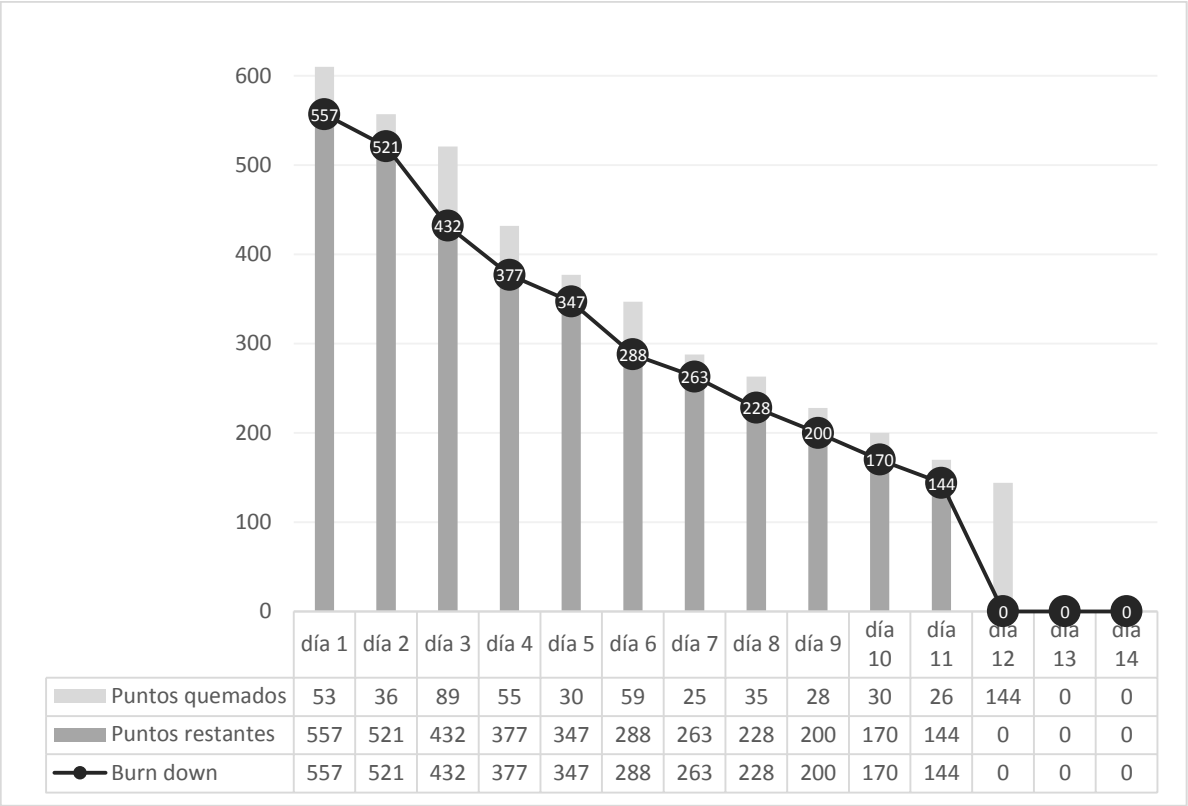
Evaluación de la iteración

El transcurso de este sprint se desarrolló sin inconvenientes importantes, se cumplieron las metas planificadas con tiempo de sobra para hacer las correcciones y ajustes necesarios. Durante este sprint se logró el primer hito del proyecto, al conseguir un personaje principal que el jugador pueda controlar totalmente.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 1	día 2	día 3	día 4	día 5	día 6	día 7	día 8	día 9	día 10	día 11	día 12	día 13	día 14
Puntos quemados	53	36	89	55	30	59	25	35	28	30	26	144	0	0
Puntos restantes	557	521	432	377	347	288	263	228	200	170	144	0	0	0
Burn down	557	521	432	377	347	288	263	228	200	170	144	0	0	0

Figura 2: burndown chart sprint 1
Fuente: propia

Sprint 2

Fecha de inicio: 16/11/2020.

Fecha fin: 29/11/2020.

Descripción

inventario y uso de ítems.

Planificación

Objetivos

Recoger ítems especiales.

Crear una estructura de datos para guardar y manejar los ítems recogidos.

Interfaz gráfica del inventario donde se visualicen los ítems guardados.
Equipar ítems del inventario.

Métricas de los objetivos

Al recoger los ítems del escenario, debe desaparecer el modelo 3d y mostrar el ítem en la interfaz del inventario.
Los ítems del inventario pueden ser equipados o eliminados.
La interfaz gráfica del inventario debe ser clara, fácil de usar y debe actualizarse cuando ocurra un cambio en el inventario.

Seleccionar y refinar características

Tabla 8: características sprint 2

1	Crear estructura de datos para el inventario.
2	Agregar y eliminar ítems del inventario.
3	Crear Prefab para usar en los modelos 3d de los ítems.
4	Hacer que el usuario pueda guardar los ítems en el inventario cuando el personaje está en el rango del modelo 3d del prefabs del ítem.
5	Crear interfaz gráfica del inventario donde se vean los ítems guardados.
6	Manejar ítems desde la interfaz gráfica.
7	Equipar ítems del inventario.

Seguimiento y desarrollo del sprint

Característica 1: Crear estructura de datos para el inventario

Desarrollo

Se creó una clase Ítem que fuera un Scriptable Object con los datos correspondientes al ítem, entre estos el nombre del ítem, la descripción, el tipo de ítem y un icono. Luego se creó la estructura de datos para el inventario, la cual está compuesta por una lista que toma objetos del tipo Ítem y un número entero que corresponde al espacio máximo del inventario.

Problemas

Ninguno.

Estado de la característica

Completada el 16/11/2020.

Puntos quemados

21.

Característica 2: Agregar y eliminar ítems del inventario

Desarrollo

Se agregó un script donde se creaba una instancia del inventario y funciones para agregar o eliminar los objetos de la lista de ítems.

Problemas

Ninguno.

Estado de la característica

Completada el 17/11/2020.

Puntos quemados

34.

Característica 3: Crear Prefab para usar en los modelos 3d de los ítems

Desarrollo

Para esta tarea se creó un GameObject vacío, y se le agrego de componente un script donde se le asignaba un Scriptable Object del tipo Ítem. En la jerarquía se le agrego el modelo del ítem deseado y finalmente se creó un Prefab de todo este objeto, con el fin de reutilizarlo para todos los ítems que serán recogidos para el inventario. El Prefab brinda la facilidad de cambiar el ítem en cuestión y su modelo 3d de manera fácil.

Problemas

Ninguno.

Estado de la característica

Completada el 18/11/2020.

Puntos quemados

34.

Característica 4: Hacer que el usuario pueda guardar los ítems en el inventario cuando el personaje está en el rango del modelo 3d del Prefab del ítem

Desarrollo

Al Prefab del ítem se le asignó un componente del tipo Collider con un rango determinado. A su script se le agrego una función que al detectar que el personaje entra en el rango del Collider, muestre un mensaje que indique al usuario que puede guardar el ítem en su inventario.

Problemas

Ninguno.

Estado de la característica

Completada el 19/11/2020.

Puntos quemados

21.

Característica 5: Crear interfaz gráfica del inventario donde se vean los ítems guardados

Desarrollo

Primero se diseñó la interfaz gráfica del inventario y sus componentes con el programa Adobe XD. Los componentes diseñados en Adobe XD fueron exportados en formato de imagen para ser utilizados en el diseño. La interfaz del inventario se creó en Unity con un objeto tipo Canvas. Dentro de este objeto se agregó un panel y como hijo de este, un componente del tipo grilla donde van los ítems del inventario. Finalmente se agregó un script que enlazaba la vista de la grilla de la interfaz con la instancia del inventario para que mostrara el contenido actual de este.

Problemas

Problemas menores en el escalado de la interfaz cuando se probaba en pantallas de baja resolución. Estos detalles fueron corregidos durante el sprint.

Estado de la característica

Completada el 25/11/2020.

Puntos quemados

55.

Característica 6: Manejar ítems desde la interfaz grafica

Desarrollo

A los objetos de la grilla se les asigno un script donde tomaban el ítem que correspondía y utilizaba funciones para eliminarlo del inventario o equiparlo al personaje, mostrando estas opciones de manera gráfica.

Problemas

Ninguno.

Estado de la característica

Completada el 26/11/2020.

Puntos quemados

34.

Característica 7: Equipar ítems del inventario

Desarrollo

Al seleccionar la opción de equipar un ítem en la interfaz gráfica, se dispara una función que agrega este ítem a un listado de ítems del tipo Equipo. Este listado está en un script nuevo que se agregó al GameObject del personaje principal.

Problemas

Ninguno.

Estado de la característica

Completada el 27/11/2020.

Puntos quemados

34.

Cierre de la iteración

Al completar las tareas del sprint se hicieron pruebas para corroborar que los objetivos se habían completado de manera exitosa, surgieron pequeños detalles en el escalado de la interfaz del inventario, pero fueron resueltos antes de concluir el sprint.

Evaluación del estado del juego

En este sprint se obtuvo un sistema de inventario totalmente completo, que permite agregar, eliminar o equipar ítems del inventario a través de una interfaz gráfica. También se creó un Prefab para Ítems que servirá de plantilla para otros objetos interactivos con características similares; compuestos por un Scriptable Object de un tipo, un rango de interacción y funciones definidas según el tipo de objeto.

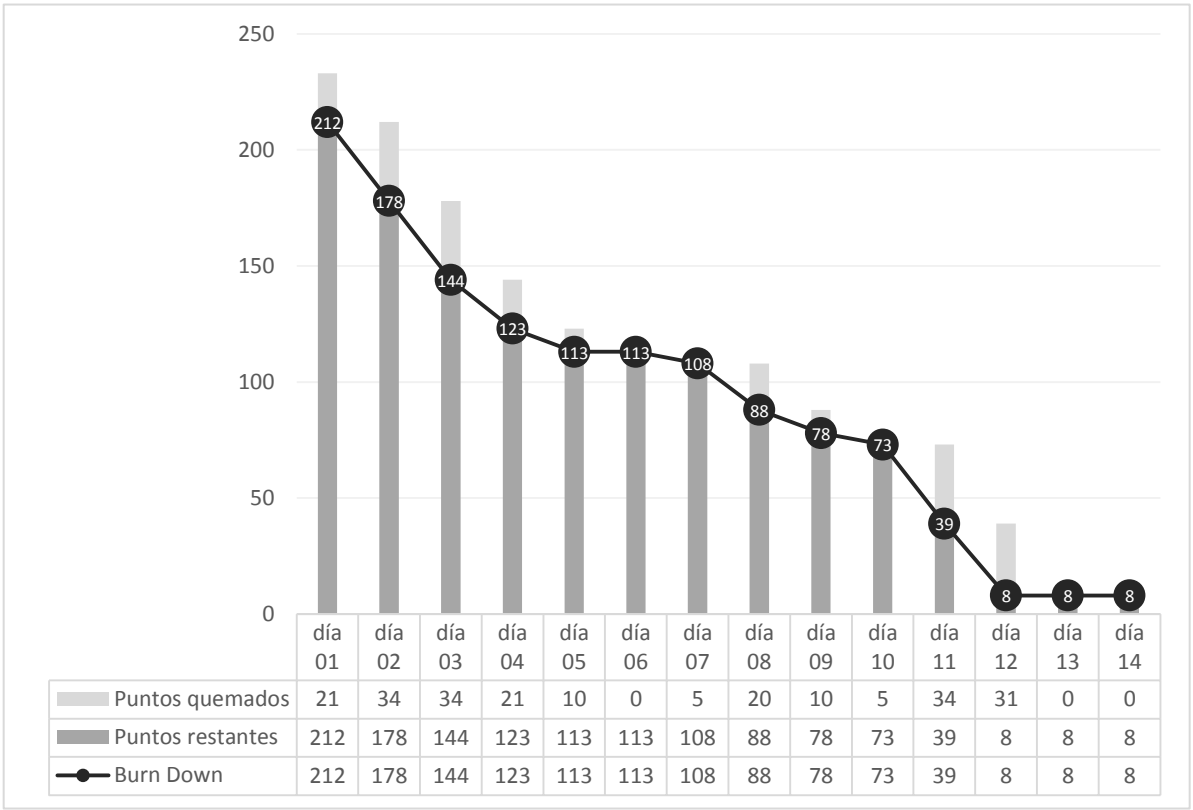
Evaluación de la iteración

El transcurso de este sprint se desarrolló sin inconvenientes importantes, se cumplieron las metas planificadas antes de que finalizara y en el tiempo restante se hicieron ajustes visuales en la interfaz del inventario. Durante este sprint se logró el segundo hito del proyecto, al lograr obtener un inventario completamente funcional.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	21	34	34	21	10	0	5	20	10	5	34	31	0	0
Puntos restantes	212	178	144	123	113	113	108	88	78	73	39	8	8	8
Burn Down	212	178	144	123	113	113	108	88	78	73	39	8	8	8

Figura 3: burndown chart sprint 2
Fuente: propia

Sprint 3

Fecha de inicio: 30/11/2020

Fecha fin: 13/12/2020

Descripción

Objetos interactivos del juego.

Planificación

Objetivos

Crear Prefab para edificios interactivos.

Crear Prefab para NPC.

Interfaz del dialogo al interactuar con un edificio con información correspondiente al edificio y opciones para interactuar con él.

Comportamientos de los NPC de patrulla y de dialogo para que interactúen con el personaje principal.

Métricas de los objetivos

Al entrar en el rango de interacción de los Prefabs, debe aparecer un dialogo indicando las acciones que se pueden ejecutar.

Los diálogos de los edificios deben mostrar información relevante sobre el edificio, como su nombre y una breve descripción.

Cada NPC debe tener diálogos propios de su personaje.

Cada NPC cuenta con un recorrido cíclico, con el fin de darle a cada uno un comportamiento distinto.

Tiempos de respuesta rápidos.

Seleccionar y refinar características

Tabla 9: características sprint 3

1	Crear Prefab para los edificios a partir del Prefab de ítems.
2	Crear Prefab para los NPC a partir del Prefab de ítems.

3	Crear interfaz para mostrar información de los edificios.
4	Crear interfaz para los diálogos de los NPC.
5	Sistema de dialogo para los NPC.
6	Sistema de patrullaje para los NPC.

Seguimiento y desarrollo del sprint

Característica 1: Crear Prefab para los edificios a partir del Prefab de ítems

Desarrollo

Primero se creó una clase Edificio, que fuera un Scriptable Object con los datos correspondientes al edificio, como el nombre y una breve descripción. Luego, a partir del Prefab de ítem, se creó el Prefab para los edificios, que al igual que el primero, contiene un Scriptable Object, que en este caso sería el de Edificio, y un Collider con un rango de interacción.

Problemas

Ninguno.

Estado de la característica

Completada el 30/11/2020.

Puntos quemados

12.

Característica 2: Crear Prefab para los NPC a partir del Prefab de ítems

Desarrollo

Se creó una clase NPC, que fuera un Scriptable Object. Luego, a partir del Prefab de ítem, se creó el Prefab para los NPC, que al igual que el primero, contiene un Scriptable Object, que en este caso sería el de NPC, y un Capsule Collider con un rango de

interacción. Además, se le agregaron los componentes Rigidbody y un NavMesh Agent que permiten que el NPC sea afectado por las físicas del mundo simulado en el juego, y pueda moverse esquivando objetos, esto último será necesario más adelante cuando se quiera hacer que los NPC deambulen en el escenario por su cuenta.

Problemas

Ninguno.

Estado de la característica

Completada el 01/12/2020.

Puntos quemados

24.

Característica 3: Crear interfaz para mostrar información de los edificios

Desarrollo

Se diseñaron las interfaces gráficas con el programa Adobe XD y luego se creó la interfaz en Unity con objetos tipo Canvas. Esta está compuesta de un panel que aparece información relevante al edificio.

Problemas

Problemas menores en el escalado de la interfaz cuando se probaba en pantallas de baja resolución. Estos detalles fueron corregidos durante el sprint.

Estado de la característica

Completada el 05/12/2020.

Puntos quemados

12.

Característica 4: Crear interfaz para los diálogos de los NPC

Desarrollo

Se diseñaron las interfaces gráficas de los diálogos con el programa Adobe XD, las imágenes de los NPC fueron dibujadas con Krita. Se creó la interfaz en Unity con objetos tipo Canvas. Esta está compuesta de un panel que aparece inicialmente con indicaciones para que el usuario interactúe con el NPC, y otro panel que aparece con la imagen del NPC y el dialogo, que aparece una vez se confirma que se desea interactuar.

Problemas

Ninguno.

Estado de la característica

Completada el 08/12/2020.

Puntos quemados

12.

Característica 5: Sistema de dialogo para los NPC

Desarrollo

Se creó un script a partir del script de objetos interactivos, en el que al entrar en el rango del NPC interactivo, muestre un indicador de que se puede interactuar con este apretando una tecla. En el script se define un entero que representaría la cantidad de bloques de textos del NPC, un ScriptableObject que corresponda al personaje NPC único, y un arreglo generado a partir de la cantidad de bloques de texto, donde deben ir los diálogos del personaje. Todos estos parámetros se definen en la configuración del script una vez es asignado a un objeto instanciado en la escena del juego. Al apretar la tecla para interactuar se muestra el primer bloque de texto del arreglo, al presionar la misma tecla cambia al siguiente bloque y así sucesivamente hasta que no hay más bloques de texto que mostrar y se cierra la interfaz de dialogo.

Problemas

Ninguno.

Estado de la característica

Completada el 11/12/2020.

Puntos quemados

44.

Característica 6: Sistema de patrullaje para los NPC

Desarrollo

Al Prefab del NPC se le agregó un script que utiliza el NavMesh Agent del NPC y una serie de puntos de guía para hacer que el personaje pueda moverse por su cuenta en el escenario. El NavMesh Agent se encarga de detectar las posibles colisiones y esquivarlas, mientras que los puntos de guía describen el recorrido que seguirá el NPC.

Problemas

Ajustes en el NavMesh Agent del prefabs del NPC, ya que no detectaba las colisiones con objetos de cierta altura y el NPC quedaba truncado mientras realizaba un recorrido.

Estado de la característica

Completada el 13/12/2020.

Puntos quemados

40.

Cierre de la iteración

Los objetivos del sprint fueron completados de forma exitosa. Las pruebas realizadas sobre los comportamientos de los NPC, el cual fue el punto de mayor complejidad

durante el sprint, resultaron siendo positivas. Los ajustes necesarios a estos comportamientos fueron realizados durante este sprint.

Evaluación del estado del juego

En este sprint se obtuvieron dos objetos interactivos del juego de gran importancia, los edificios y los NPC. Gracias a que se crearon utilizando prefabs, se facilitó mucho la tarea de crear la cantidad necesaria de estos objetos, pudiendo customizar las características de cada uno de acuerdo se necesiten.

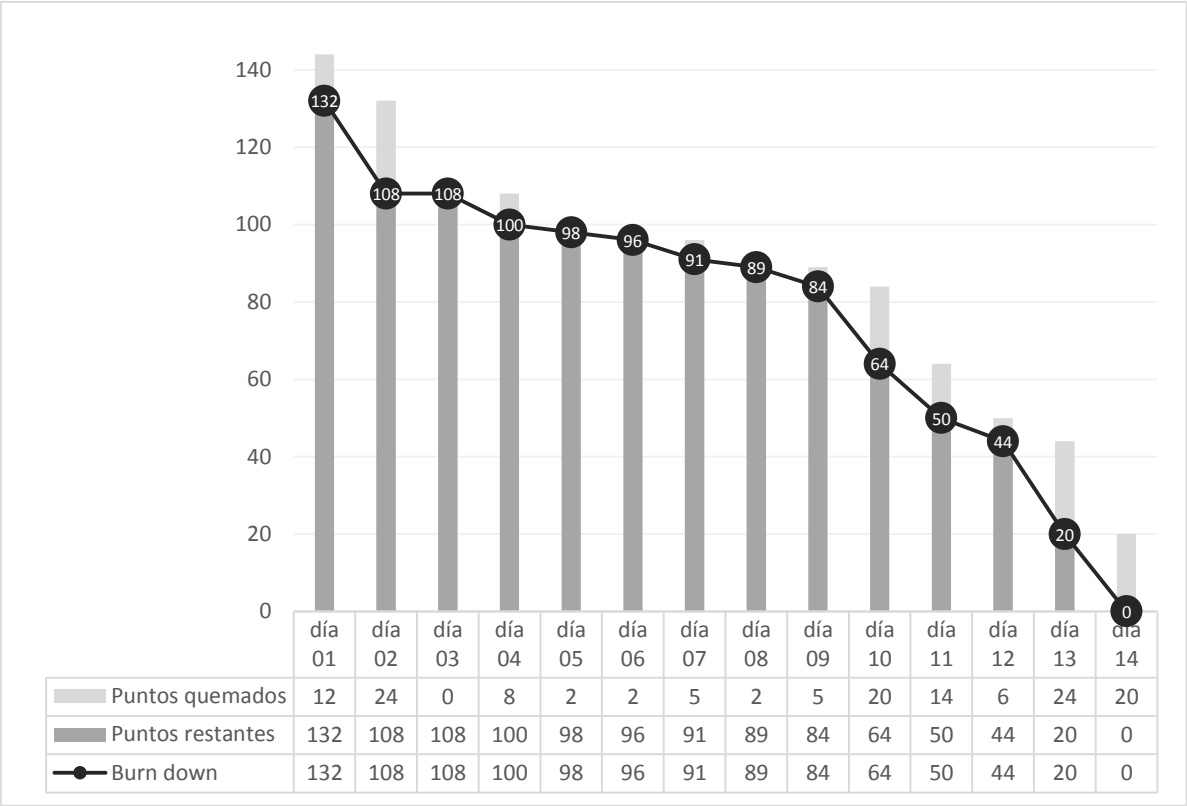
Evaluación de la iteración

El sprint se desarrolló sin mayores inconvenientes ni ningún problema que pudiera ocasionar retrasos al proyecto. En este sprint se logró el tercer hito del proyecto, al lograr que el usuario sea capaz de interactuar con edificios y habitar el escenario con NPC interactivos.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	12	24	0	8	2	2	5	2	5	20	14	6	24	20
Puntos restantes	132	108	108	100	98	96	91	89	84	64	50	44	20	0
Burn down	132	108	108	100	98	96	91	89	84	64	50	44	20	0

Figura 4: burndown chart sprint 3
Fuente: propia

Sprint 4

Fecha de inicio: 14/12/2020

Fecha fin: 27/12/2020

Descripción

Sistema de lucha con enemigos.

Interfaz gráfica.

Planificación

Objetivos

Crear sistema de lucha contra NPC enemigos, donde el personaje pueda infligir y recibir daño.

Crear interfaz gráfica general del juego donde se muestre la salud del personaje y su equipamiento actual.

Crear interfaz del menú de pausa.

Crear interfaz del menú de inicio del juego.

Métricas de los objetivos

El personaje puede infligir daño a los NPC enemigos y recibir daño de estos, la cantidad de daño está relacionada a las estadísticas de ataque y defensa de cada personaje.

Cuando la salud del personaje llegue a cero, debe mostrarse una pantalla informando al usuario que el juego termino con un botón de intentar de nuevo que reinicia el juego.

Cuando haya un cambio en la salud del personaje, debe actualizarse la barra de salud inmediatamente.

Cuando haya un cambio del equipamiento del personaje, los cambios deben ser reflejados en la interfaz.

Las interfaces deben ser intuitivas y fáciles de usar.

El menú de pausa debe mostrarse cuando el jugador oprima el botón de pausa.

El menú de inicio debe tener la opción para iniciar el juego.

Seleccionar y refinar características

Tabla 10: características sprint 4

1	Crear script de estadísticas de ataque, defensa y vida para el personaje principal y para el Prefab de los NPC
2	Crear sistema de lucha
3	Crear interfaz de la barra de salud del personaje
4	Crear interfaz de equipamiento actual del personaje
5	Crear menú de pausa
6	Crear menú de inicio

Seguimiento y desarrollo del sprint

Característica 1: Crear script de estadísticas de ataque, defensa y vida para el personaje principal y para el Prefab de los NPC

Desarrollo

En el GameObject del personaje se agregó un script para manejar sus puntos de vida, ataque y defensa. Estas características se declararon como variables flotantes que pudieran ser modificadas según el equipamiento que tuviera el personaje. Inicialmente e le asignan al personaje 10 puntos de vida, que pueden bajar si es atacado o aumentar hasta 10 sin consigue un ítem de curación. Al llegar a 0 puntos de vida el personaje muere y el juego termina.

Problemas

Se tuvo que editar el Prefab del NPC para agregar este script

Estado de la característica

Completada el 16/12/2020

Puntos quemados

30

Característica 2: Crear sistema de lucha

Desarrollo

La creación del sistema de lucha está conformado por dos acciones; infligir y recibir daño. Al Prefab del NPC enemigo se le agregó un script basado en los scripts de interacción anteriores, en el cual, al entrar en el rango de contacto del personaje principal, el enemigo comenzara a restarle puntos de vida al jugador, dependiendo estos puntos de ataques, de las estadísticas de daño del enemigo y las de defensa del personaje principal. Por otro lado, al personaje principal, se le agrego un script en el que, al estar

en el rango del personaje enemigo, pudiera restarle puntos de vida presionándola tecla de ataque, y al igual que en el caso anterior, el daño hecho se calcula con los puntos de ataques detallados en los stats del jugador principal y los puntos de defensa del NPC enemigo. Cuando el jugador está en el rango del enemigo, se necesitaba que el enemigo lo persiguiera para atacarlo, para esto, en el script de ataque del enemigo, fue utilizada una función del NavMesh agent, parecida a la de los NPC que recorren el juego, en este caso se modificó para que creara un recorrido desde su posición inicial hasta el target, que sería el jugador, y lo actualice cada vez que cambia la posición del jugador.

Problemas

Ninguno

Estado de la característica

Completada el 20/12/2020

Puntos quemados

25

Característica 3: Crear interfaz de la barra de salud del personaje

Desarrollo

Para crear la interfaz que mostrara la salud actual del personaje principal, primero se creó un nuevo Canvas en la jerarquía de Unity, donde se agregó un componente de Slider y un script que asociara la salud actual del personaje con la ubicación del Handler de la barra del Slider. Cada vez que ocurre un cambio en los puntos de salud del personaje, se dispara un script que actualiza la interfaz de la barra de salud.

Problemas

Problemas menores en el escalado de la interfaz cuando se probaba en pantallas de baja resolución.

Estado de la característica

Completada el 22/12/2020.

Puntos quemados

30.

Característica 4: Crear interfaz de equipamiento actual del personaje

Desarrollo

En la jerarquía de Unity se creó un nuevo Canvas con dos componentes de imágenes. Luego se agregó un script que revisaba el equipamiento actual del personaje y ubicaba las imágenes de arma actual y escudo actual en las imágenes de la interfaz correspondientes. Cada vez que ocurre un cambio en el equipamiento, se dispara esta función que verifica de nuevo el equipamiento actual y actualiza la interfaz gráfica.

Problemas

Problemas menores en el escalado de la interfaz cuando se probaba en pantallas de baja resolución.

Estado de la característica

Completada el 24/12/2020.

Puntos quemados

24.

Característica 5: Crear menú de inicio

Desarrollo

Se creó una nueva escena en Unity llamada “menú inicio”. Se diseñó la interfaz gráfica del menú y sus componentes con el programa Adobe XD. Los componentes diseñados en Adobe XD fueron exportados en formato de imagen para ser utilizados en el diseño. En la nueva escena se creó la interfaz con un objeto tipo Canvas que contiene dos botones. El primero “iniciar juego” enlaza el botón a la escena principal del juego. El segundo botón “salir” permite abandonar la aplicación y cerrarla.

Problemas

Ninguno.

Estado de la característica

Completada el 25/12/2020.

Puntos quemados

15.

Característica 6: Crear menú de pausa

Desarrollo

Primero se creó un GameObject vacío en Unity en la escena principal llamado “Game manager” se le agrego un script que permite verificar el estado de pausa del juego y controlar la pausa con la función Time.TimeScale de Unity. También en este script se agregó una función que detecta que cuando el botón de pausa sea presionado, se muestre la interfaz del menú de pausa. Luego de que se diseñara la interfaz y los componentes del menú de pausa en Adobe XD, se crearon en Unity los elementos necesarios. Los componentes diseñados en Adobe XD fueron exportados en formato de imagen para ser utilizados en el diseño. El menú de pausa del juego contiene 3 botones, el primero “continuar juego” permite reanudar el juego con Time.TimeScale. El segundo botón “volver al menú principal” que enlaza el botón a la escena de menú inicial, creada en la

tarea anterior. Finalmente, el tercer botón permite al jugador salir del juego y cerrar la aplicación.

Problemas

Ninguno.

Estado de la característica

Completada el 26/12/2020.

Puntos quemados

20.

Cierre de la iteración

Al finalizar cada tarea se fueron haciendo pruebas de funcionamiento y usabilidad en el caso de las interfaces. Todas las pruebas fueron exitosas. En el caso de la interfaz general surgieron ciertos problemas con el escalado en dispositivos pequeños, pero fueron resueltos durante este sprint si mayores problemas.

Evaluación del estado del juego

En este sprint se obtuvo el sistema de lucha completo que permite que el jugador entre en combate con los NPC hostiles, lo cual aumenta el nivel de interacción y jugabilidad en la aplicación. También se completó la interfaz principal del juego, que contiene la barra de salud, que es de gran importancia para el sistema de lucha, y la interfaz de equipamiento actual, que permite que el jugador pueda visualizar que arma y escudo está usando actualmente y sus estadísticas. Otros productos que se obtuvieron en este sprint fueron los menús de pausa y el menú de inicio del juego, que será lo primero que vea el usuario al abrir la aplicación.

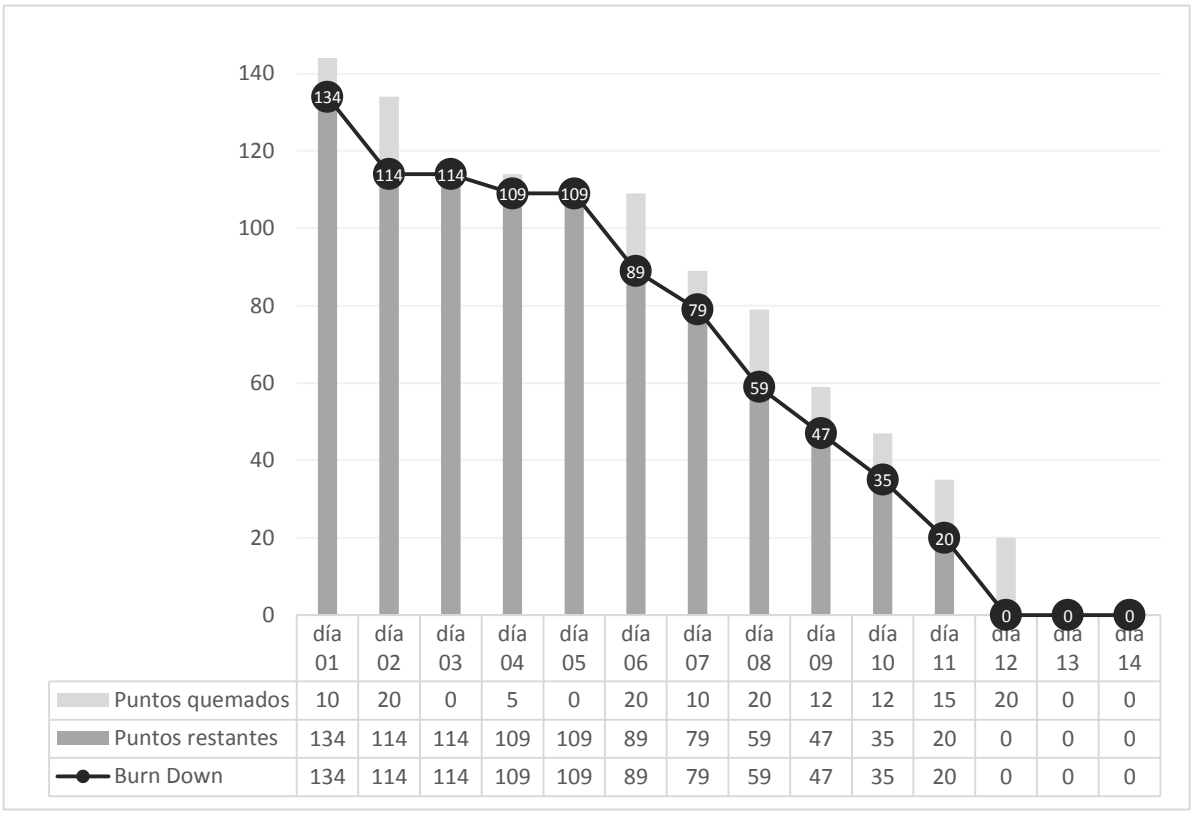
Evaluación de la iteración

El transcurso de este sprint se desarrolló sin inconvenientes importantes, se cumplieron las metas planificadas con tiempo de sobra para hacer correcciones y ajustes visuales en las interfaces gráficas. Al final de este sprint se cumplió con el cuarto hito del proyecto al lograr obtener un producto básico jugable, donde se pueden probar las características principales del juego.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	10	20	0	5	0	20	10	20	12	12	15	20	0	0
Puntos restantes	134	114	114	109	109	89	79	59	47	35	20	0	0	0
Burn Down	134	114	114	109	109	89	79	59	47	35	20	0	0	0

Figura 5: burndown chart sprint 4
Fuente: propia

Sprint 5

Fecha de inicio: 28/12/2020

Fecha fin: 10/01/2020

Descripción

Estructura básica de calles y aceras del casco histórico.

Planificación

Objetivos

Reunir referencias de las calles del casco histórico en las que se basará el escenario del juego.

Elaborar modelos de las calles basado en el casco histórico de Cumaná.

Métricas de los objetivos

El modelo del escenario debe tener límites donde el jugador pueda deambular.

Deben de respetarse los espacios donde se ubicarán los edificios, postes y vegetación.

Modelos 3D de bajos polígonos.

Texturas estilizadas.

Seleccionar y refinar características

Tabla 11: características sprint 5

1	Recopilar referencias y crear bocetos del mapa a modelar.
2	Crear modelo básico del mapa del casco histórico.
3	Agregar restricciones en los bordes del mapa.
4	Agregar los relieves de las aceras de la calle.
5	Agregar grilla.
6	Generar superficies caminables con el NavMesh.

Seguimiento y desarrollo del sprint

Característica 1: Recopilar referencias y crear bocetos del mapa a modelar

Desarrollo

Primero se realizaron bocetos iniciales en Krita, usando de referencia el mapa actual del casco histórico proporcionado por Google Maps. Luego sobre los bocetos iniciales se agregaron detalles de vegetación, postes eléctricos y edificios, para tomar en cuenta su ubicación en el mapa. Con fotografías de referencia más detalladas obtenidas de varias fuentes, se identificó cada edificio y su fachada actual para modelarlos en el siguiente sprint.

Problemas

Esta tarea dio bastante trabajo ya que fue difícil encontrar fotos actualizadas de la fachada del Casco histórico.

Estado de la característica

Completada el 01/01/2021.

Puntos quemados

10.

Característica 2: Crear modelo básico del mapa del casco histórico

Desarrollo

Con la herramienta ProBuilder se modelo la base del mapa de parte del casco histórico, esto incluye las calles sucre, santa Inés, Bolívar, La ermita y el callejón santa Inés. A los modelos se le agregaron texturas simples unicolor.

Problemas

Ninguno.

Estado de la característica

Completada el 04/01/2021.

Puntos quemados

10.

Característica 3: Agregar restricciones en los bordes del mapa

Desarrollo

Se agregó a la escena de Unity un cubo estándar que sirvió a modo de pared para limitar los bordes del mapa. Esta pared fue multiplicada 6 veces para formar un hexágono que encierra el escenario. A estos modelos se les agrego una textura transparente para que no se viera en el escenario.

Problemas

Ninguno.

Estado de la característica

Completada el 05/01/2021.

Puntos quemados

3.

Característica 4: Agregar los relieves de las aceras de la calle

Desarrollo

Con la herramienta ProBuilder se modelaron las aceras sobre los modelos del mapa del casco histórico, tomando en cuenta la posición de los edificios y otros elementos que serán agregados en los siguientes sprints.

Problemas

Ninguno.

Estado de la característica

Completada el 06/01/2021.

Puntos quemados

5.

Característica 5: Agregar grilla

Desarrollo

Con ayuda de la herramienta ProGrids se estableció una grilla que marca una cuadricula sobre los modelos, y ayuda a facilitar el posicionamiento de los modelos de los elementos a agregarse en los siguientes sprints. Esta grilla no aparece en el producto final, ya que solo es una herramienta para facilitar el desarrollo.

Problemas

Ninguno

Estado de la característica

Completada el 06/01/2021

Puntos quemados

3

Característica 6: Generar superficies caminables con el NavMesh

Desarrollo

Para generar el NavMesh del mapa, primero se seleccionaron los modelos que serían afectados, y luego se hicieron los ajustes correspondientes para que fuera compatible con el tamaño del NavMesh Agent.

Problemas

Fueron necesarios hacer varios ajustes en la configuración del generador de NavMesh, ya que, al probar las primeras veces, en ciertas superficies no era posible caminar con el personaje.

Estado de la característica

Completada el 07/01/2021.

Puntos quemados

3.

Cierre de la iteración

Los modelos requeridos en este sprint fueron completados en su totalidad. Durante la tarea de generar las superficies caminables por el jugador surgieron varios inconvenientes y fue necesario hacer ajustes en el modelo del personaje principal y en la configuración del NavMesh.

Evaluación del estado del juego

En este sprint se obtuvo un escenario base caminable para el jugador, basado en el casco histórico de Cumaná. También se recopilaron referencias que serán utilizadas durante el resto del proyecto para modelar el resto de los elementos del escenario.

Evaluación de la iteración

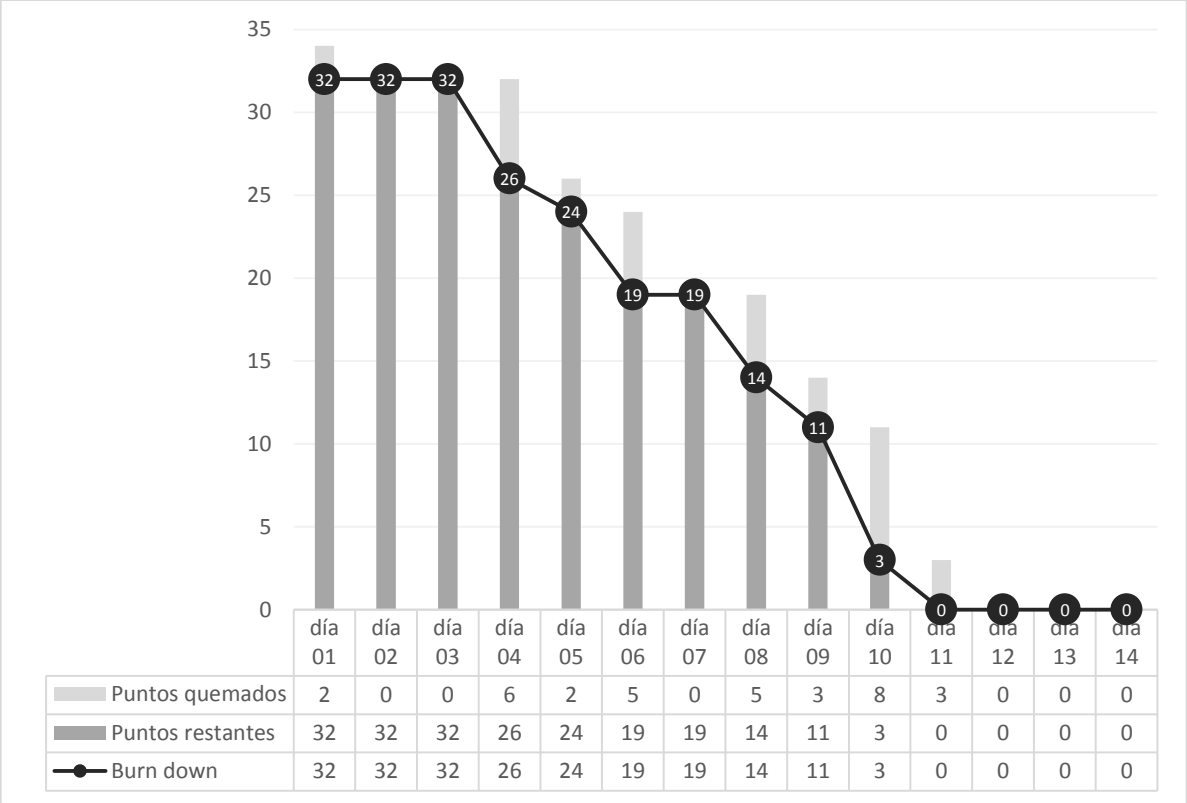
Los inconvenientes surgidos durante este sprint fueron resueltos sin mayores problemas, y los objetivos planteados fueron completados antes de la fecha límite del sprint. El

tiempo sobrante del sprint fue empleado para continuar investigando y buscando referencias fotográficas e históricas sobre el casco histórico.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	2	0	0	6	2	5	0	5	3	8	3	0	0	0
Puntos restantes	32	32	32	26	24	19	19	14	11	3	0	0	0	0
Burn down	32	32	32	26	24	19	19	14	11	3	0	0	0	0

Figura 6: burndown chart sprint 5
Fuente: propia

Sprint 6

Fecha de inicio: 11/01/2021

Fecha fin: 24/01/2021

Descripción

Modelos de casas y edificaciones históricas.

Planificación

Objetivos

Elaborar modelos de los edificios de las calles que forman parte del casco histórico de Cumana.

Métricas de los objetivos

Modelos 3D de bajos polígonos.

Texturas estilizadas.

Seleccionar y refinar características

Tabla 12: características sprint 6

1	Crear modelo del castillo santa María de la cabeza.
2	Crear modelo de la iglesia santa Inés.
3	Crear modelo de la plaza santa Inés.
4	Modelar componentes modulares de las casas.
5	Crear modelos de las casas de la calle Sucre.
6	Crear modelos de las casas de la calle Santa Inés.
7	Crear modelos de las casas del callejón Santa Inés.
8	Crear modelos de las casas de la calle Bolívar.
9	Crear modelos de las casas de la calle La Ermita.

Seguimiento y desarrollo del sprint

Característica 1: Crear modelo del castillo santa María de la cabeza

Desarrollo

Con las referencias recopiladas en el sprint anterior, se crearon los modelos componentes del castillo santa maría de la cabeza. Este sprint se dividió en tres partes. La primera parte consistió en modelar la estructura básica del castillo, compuesta por la base, el borde de mampostería y una columna de mampostería que sería reutilizada para el modelo final. Las texturas de estos componentes fueron creadas en Substance Painter. La segunda, compuesta por la ermita del Carmen, ubicada en la parte superior del castillo. Finalmente, la tercera parte consistió en modelar los detalles adicionales como la escalera exterior, la estructura de piedra donde se encuentra la estatua de la virgen de Lourdes y el portón metálico de la entrada del castillo. Para mantener los modelos bajos en polígonos, fue necesario modelar estructuras muy básicas en Blender y agregar todos los detalles necesarios en Zbrush a partir del modelo de Blender. Del modelo detallado creado en Zbrush, se obtuvo un mapa de normales, en formato PNG que contenía todos los detalles del modelo. Este mapa fue exportado a Substance painter donde se crearon las texturas que luego fueron exportadas y agregadas al material de cada elemento en Unity y aplicadas al modelo de bajo polígonos.

Problemas

Ninguno.

Estado de la característica

Completada el 15/01/2021.

Puntos quemados

4.

Característica 2: Crear modelo de la iglesia santa Inés

Desarrollo

Para este modelo, se creó la estructura básica en Magica Voxel y se exportó en formato OBJ a Blender, donde encima de este, se modelaron las partes más complejas de la

estructura, como las torres, las columnas de la entrada y los escalones. No fue necesario utilizar Zbrush para agregar detalles. Las texturas fueron creadas con Magica Voxel.

Problemas

Ninguno.

Estado de la característica

Completada el 16/01/2021.

Puntos quemados

3.

Característica 3: Crear modelo de la plaza santa Inés

Desarrollo

Esta estructura fue modelada en Blender, y sus texturas fueron creadas con Substance Painter. La vegetación de la plaza fue agregada utilizando modelos externos.

Problemas

Ninguno.

Estado de la característica

Completada el 18/01/2021.

Puntos quemados

3.

Característica 4: Modelar componentes modulares de las casas

Desarrollo

Se concluyó en que el mejor enfoque para agilizar la creación de los modelos de las casas sería crear modelos reutilizables para las puertas, ventanas y rejas, en al menos 4 modelos distintivos para cada componente. Para las bases de las casas, se hizo un modelo plantilla en Magica Voxel, en esta plantilla se basaría la altura, el ancho, y el tamaño de puertas y ventanas del resto de los modelos de las casas. A partir de las medidas de las puertas y ventanas, se modelaron estos componentes en Blender, al igual que los modelos de las rejas. Las texturas fueron creadas en Substance Painter.

Problemas

Ninguno.

Estado de la característica

Completada el 20/01/2021.

Puntos quemados

6.

Característica 5: Crear modelos de las casas de la calle Sucre

Desarrollo

A partir de la plantilla de casa, se hicieron los modelos de 12 casas, incluyendo el museo de arqueología del estado sucre y la casa Ramos Sucre. Los modelos adicionales, como los detalles en la fachada de la casa Ramos Sucre, y los letreros como los de la posada santa Inés y el museo de arqueología fueron hechos en Blender

Problemas

Ninguno.

Estado de la característica

Completada el 22/01/2021.

Puntos quemados

1.

Característica 6: Crear modelos de las casas de la calle Santa Inés

Desarrollo

A partir de la plantilla de casa, se hicieron los modelos de 8 casas. Los detalles, como el letrero de la escuela de inglés Traduce, fueron hechos en Blender.

Problemas

Ninguno.

Estado de la característica

Completada el 23/01/2021.

Puntos quemados

1.

Característica 7: Crear modelos de las casas del callejón Santa Inés

Desarrollo

A partir de la plantilla de casa, se hicieron los modelos de 12 casas.

Problemas

Ninguno.

Estado de la característica

Completada el 23/01/2021.

Puntos quemados

1.

Característica 8: Crear modelos de las casas de la calle Bolívar

Desarrollo

A partir de la plantilla de casa, se hicieron los modelos de 7 casas. Los detalles del Palacete Briceño se modelaron en Blender.

Problemas

Ninguno.

Estado de la característica

Completada el 24/01/2021.

Puntos quemados

1.

Característica 9: Crear modelos de las casas de la calle La Ermita

Desarrollo

A partir de la plantilla de casa, se hicieron los modelos de 6 casas. Adicionalmente se modelo en Magica Voxel la estructura del resto de la calle y las escalinatas del castillo san Antonio de la eminencia.

Problemas

Ninguno.

Estado de la característica

Completada el 24/01/2021.

Puntos quemados

1.

Cierre de la iteración

Para este sprint no fue necesario realizar pruebas ya que los productos de este sprint fueron solamente los modelos 3d con sus texturas.

Evaluación del estado del juego

En este sprint se obtuvieron los modelos de los edificios del escenario. Ahora el entorno del juego es fácilmente reconocible, ya que cuenta con los edificios y puntos de referencia más representativos del casco histórico.

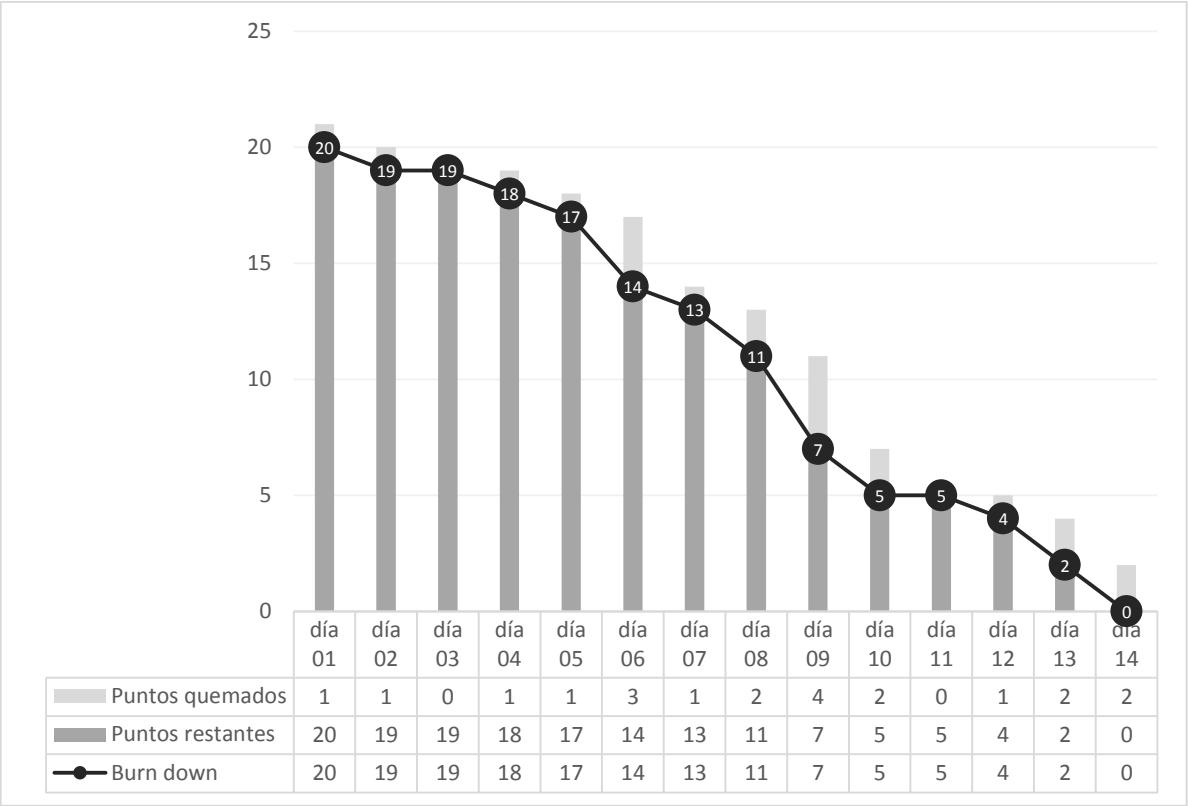
Evaluación de la iteración

Los objetivos planteados para este sprint fueron completados en el tiempo estimado y no surgieron inconvenientes importantes.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	1	1	0	1	1	3	1	2	4	2	0	1	2	2
Puntos restantes	20	19	19	18	17	14	13	11	7	5	5	4	2	0
Burn down	20	19	19	18	17	14	13	11	7	5	5	4	2	0

Figura 7: burndown chart sprint 6
Fuente: propia

Sprint 7

Fecha de inicio: 25/01/2021

Fecha fin: 07/02/2021

Descripción

Modelos humanoides, Ridging, animaciones y texturas.

Planificación

Objetivos

Crear modelo base para los personajes humanos.

Agregar armazón al modelo humanoide.

Hacer animación de estado de reposo.

Hacer animación de caminata.

Hacer animación de salto.

Hacer animación de charla.

Hacer modificaciones en el modelo base para cada personaje.

Crear texturas para cada personaje.

Métricas de los objetivos

Animaciones fluidas y suaves.

Modelos estilizados.

Texturas estilizadas y con detalles.

Seleccionar y refinar características

Tabla 13: características sprint 7

1	Crear modelo humano base.
2	Ridging del modelo base.
3	Animación de estado de reposo.
4	Animación de caminata.
5	Animación de salto.
6	Animación de charla.
7	Creación del modelo del personaje principal.
8	Creación de los modelos de los NPC.
9	Creación de NPC genéricos.
10	Agregar modelos nuevos al escenario del juego.

Seguimiento y desarrollo del sprint

Característica 1: Crear modelo humano base

Desarrollo

El modelo del personaje humanoide se creó en Blender en base a un modelo humano estilizado, para que fuera acorde con la estética del juego.

Problemas

Ninguno.

Estado de la característica

Completada el 27/01/2020.

Puntos quemados

7.

Característica 2: Ridging del modelo base

Desarrollo

Con la herramienta de armazón de Blender, se agregó un esqueleto al modelo base. Gracias este componente, se realizaron las animaciones de las siguientes tareas.

Problemas

Ninguno.

Estado de la característica

Completada el 28/01/2021.

Puntos quemados

2.

Característica 3: Animación de estado de reposo

Desarrollo

En el modo de pose de Blender, se modificó el esqueleto del modelo base para que en cada frame del timeline tuviera la pose requerida, luego de varios frames, se obtuvo la animación deseada.

Problemas

Ninguno.

Estado de la característica

Completada el 29/01/2021.

Puntos quemados

2.

Característica 4: Animación de caminata

Desarrollo

Para crear la animación, en el modo de pose de Blender, se modificó el esqueleto del modelo base para que en cada frame del timeline el modelo tomara la pose del esquema del libro citado, luego de aproximadamente 6 poses se obtuvo la animación requerida.

Problemas

Ninguno.

Estado de la característica

Completada el 29/01/2021.

Puntos quemados

1**.**

Característica 5: Animación de salto

Desarrollo

Esta animación fue creada en el modo pose de Blender de la misma forma que las animaciones anteriores.

Problemas

Ninguno.

Estado de la característica

Completada el 29/01/2021.

Puntos quemados

1.

Característica 6: Animación de charla

Desarrollo

Esta animación fue creada en el modo pose de Blender de la misma forma que las animaciones anteriores.

Problemas

Ninguno.

Estado de la característica

Completada el 29/01/2021.

Puntos quemados

1.

Característica 7: Creación del modelo del personaje principal

Desarrollo

En base al diseño del personaje principal, se modificó el modelo base y se hicieron algunos arreglos en las animaciones para que fueran más distintivas. Las texturas fueron creadas con Substance Painter.

Problemas

Ninguno.

Estado de la característica

Completada el 01/02/2020.

Puntos quemados

2.

Característica 8: Creación de los modelos de los NPC

Desarrollo

Para la creación de estos modelos, se modificó el modelo base para cada diseño y se agregaron las texturas creadas en Substance Painter.

Problemas

En algunos modelos fue necesario hacer ajustes en las animaciones, ya que al cambiar el tamaño de ciertas partes del alguno de los modelos las animaciones tuvieron fallas.

Estado de la característica

Completada el 04/02/2021.

Puntos quemados

3.

Característica 9: Creación de NPC genéricos

Desarrollo

Con el objetivo de poblar el escenario, se determinó que era necesario crear NPC sin dialogo que deambularan por las calles. Para la creación de estos modelos, se modificó el modelo base y se crearon 5 texturas para las 5 variaciones de estos NPC.

Problemas

Ninguno.

Estado de la característica

Completada el 06/02/2021.

Puntos quemados

1.

Característica 10: Agregar modelos nuevos al escenario del juego

Desarrollo

Los modelos de los personajes fueron exportados en formato FBX a Unity, cada uno en una carpeta con sus texturas en formato PNG para mantener orden dentro del proyecto. Luego de asignar a cada modelo un material con sus texturas, se creó para cada uno un componente Avatar, que permite controlar cada parte del cuerpo del personaje. Para cada uno se creó un Animator Controller del tipo humanoide donde se manejan las transiciones entre las animaciones. De acuerdo a la animación requerida, los parámetros para reproducirla se manejan vía script.

Problemas

Ninguno.

Estado de la característica

Completada el 07/02/2021.

Puntos quemados

1.

Cierre de la iteración

Los objetivos del sprint fueron completados de forma exitosa. Las pruebas de animación dentro de Unity no tuvieron problemas importantes, y los ajustes necesarios durante esta tarea fueron realizados sin mayores complicaciones.

Evaluación del estado del juego

Los productos obtenidos en este sprint fueron los modelos humanos que ahora pueblan el escenario del juego y más importante aún, se creó el modelo del personaje principal, el cual reemplazó al antiguo modelo con el que se estaban haciendo las pruebas, que era un cilindro de Unity, el cual era funcional para las pruebas necesarias.

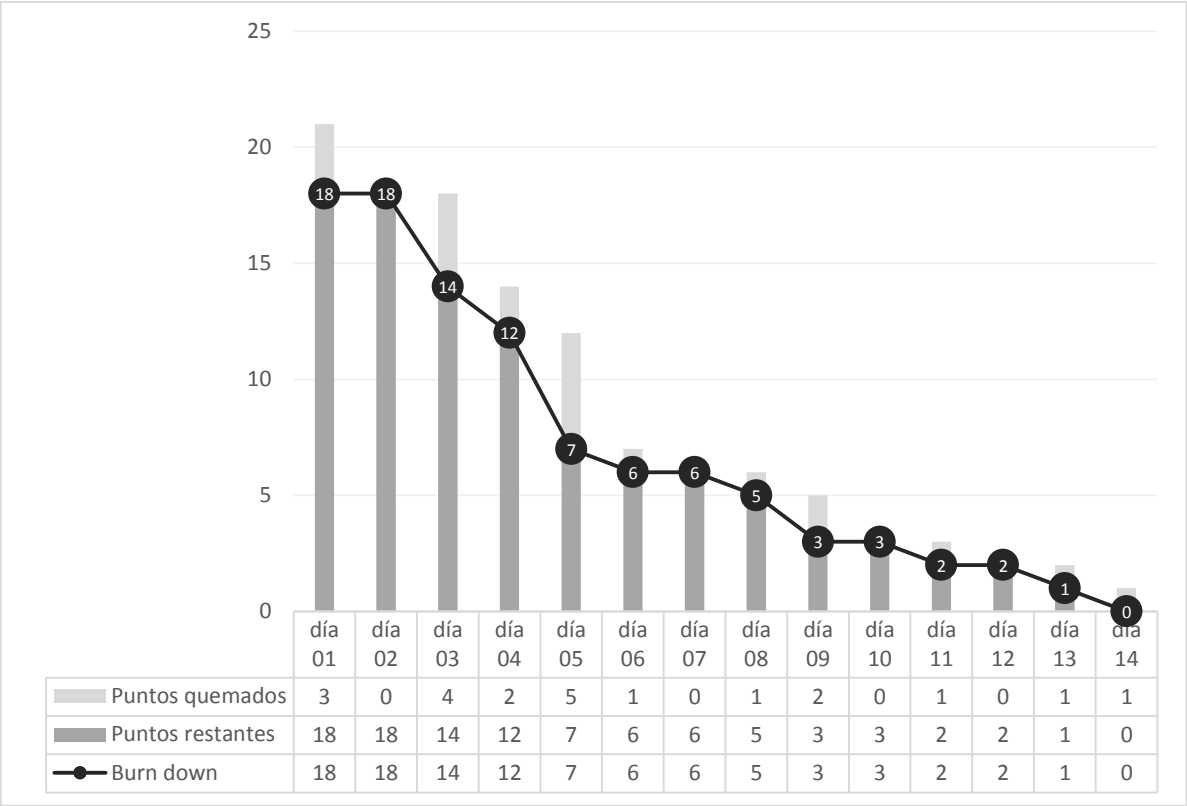
Evaluación de la iteración

El sprint se desarrolló sin mayores inconvenientes ni ningún problema que pudiera ocasionar retrasos al proyecto. En este sprint se logró completar el quinto hito del proyecto, al obtener los modelos de personaje principal y NPC modelados y con texturas finalizadas.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	3	0	4	2	5	1	0	1	2	0	1	0	1	1
Puntos restantes	18	18	14	12	7	6	6	5	3	3	2	2	1	0
Burn down	18	18	14	12	7	6	6	5	3	3	2	2	1	0

Figura 8: burndown chart sprint 7
Fuente: propia

Sprint 8

Fecha de inicio: 08/02/2021

Fecha fin: 21/02/2021

Descripción

Modelos y texturas de ítems y objetos decorativos.

Iluminación y graficas atractivas y estilizadas.

Planificación

Objetivos

Crear modelos de ítems.

Crear modelos decorativos del escenario.

Agregar vegetación al escenario.

Agregar modelos decorativos al escenario.

Ajustar iluminación global.

Agregar efectos visuales.

Métricas de los objetivos

Modelos estilizados.

Texturas estilizadas y con detalles.

Efectos visuales optimizados y que no afecten el rendimiento del juego.

Fluidez al momento de ejecutar el videojuego al menos 30 FPS.

Seleccionar y refinar características

Tabla 14: características sprint 8

1	Crear modelos de armas.
2	Crear modelos de escudos.
3	Crear modelos de otros tipos de ítems.
4	Crear modelos decorativos del escenario.
5	Ajustar iluminación.
6	Agregar efectos visuales.

Seguimiento y desarrollo del sprint

Característica 1: Crear modelos de armas

Desarrollo

Los modelos de las armas, principalmente espadas y lanzas, fueron creados en Blender, con detalles en Zbrush, para mantener los modelos bajos en polígonos, pero conservando los detalles. Las texturas fueron creadas en Substance painter a partir de los mapas de normales generados con el modelo detallado de Zbrush.

Problemas

Ninguno.

Estado de la característica

Completada el 09/02/2021.

Puntos quemados

2.

Característica 2: Crear modelos de escudos

Desarrollo

A partir de un modelo base de escudo en Blender, se crearon variaciones con diferentes texturas hechas con Substance painter.

Problemas

Ninguno.

Estado de la característica

Completada el 10/02/2021.

Puntos quemados

2.

Característica 3: Crear modelos de otros tipos de ítems

Desarrollo

Estos modelos fueron hechos en Blender, con detalles modelados en Zbrush. Las texturas fueron creadas en Substance painter a partir de los mapas de normales generados con el modelo detallado de Zbrush. Se crearon modelos para botellas de pociones, comida, libros, botas y antorchas.

Problemas

Ninguno.

Estado de la característica

Completada el 13/02/2021.

Puntos quemados

3.

Característica 4: Crear modelos decorativos del escenario

Desarrollo

Estos modelos fueron creados con un procedimiento parecidos al de la tarea anterior. Entre los modelos obtenidos en esta tarea se encuentra el modelo de poste eléctrico repetido durante todo el escenario, la estatua de la virgen de Lourdes ubicada en la gruta de Lourdes, la estatua de la virgen del Carmen, ubicada en la gruta del Carmen, bancos, macetas, señales de transito y de ubicación, automóviles y autobuses.

Problemas

Ninguno.

Estado de la característica

Completada el 16/02/2021.

Puntos quemados

5.

Característica 5: Ajustar iluminación

Desarrollo

Los ajustes de iluminación fueron realizados en la interfaz de iluminación de Unity, donde se controla el tono, color e intensidad de la luz, los ángulos de iluminación y calidad de sombras. Las sombras fijas se generaron en esta interfaz con un mapa de sombras, creado en Unity, este mapa contiene la información de la ubicación de las sombras en el escenario, lo que ahorra que se calculen las sombras en cada frame del juego y representa una mejora significativa del rendimiento del videojuego.

Problemas

Ninguno.

Estado de la característica

Completada el 19/02/2021.

Puntos quemados

6.

Característica 6: Agregar efectos visuales

Desarrollo

Con la herramienta nativa de Unity, Post-processing Stack, se agregaron varios efectos visuales que mejoraron considerablemente las gráficas del videojuego. El Post-processing Stack contiene filtros y efectos nativos y los carga en la cámara principal del juego justo antes de iniciar el renderizado.

Problemas

Ninguno.

Estado de la característica

Completada el 21/02/2021.

Puntos quemados

3.

Cierre de la iteración

¿Funciono todo? Resultado de las pruebas. Observaciones.

Para la primera parte de este sprint no fue necesario realizar pruebas ya que los productos fueron solamente los modelos 3d con sus texturas. Para las tareas de efectos visuales, se realizaron varias pruebas de rendimiento, obteniendo un resultado positivo luego de varios ajustes en la configuración de los efectos. Luego de estos ajustes la aplicación pudo ser ejecutada en computadoras desde gama baja hasta gama alta sin problemas mayores.

Evaluación del estado del juego

En este sprint se terminó de armar el escenario principal del juego creando y agregando los modelos decorativos restantes. También se modelaron los ítems de equipamiento del personaje principal. Fueron agregados los efectos necesarios al juego y se arregló su iluminación global, mejorando por mucho la presentación visual del videojuego.

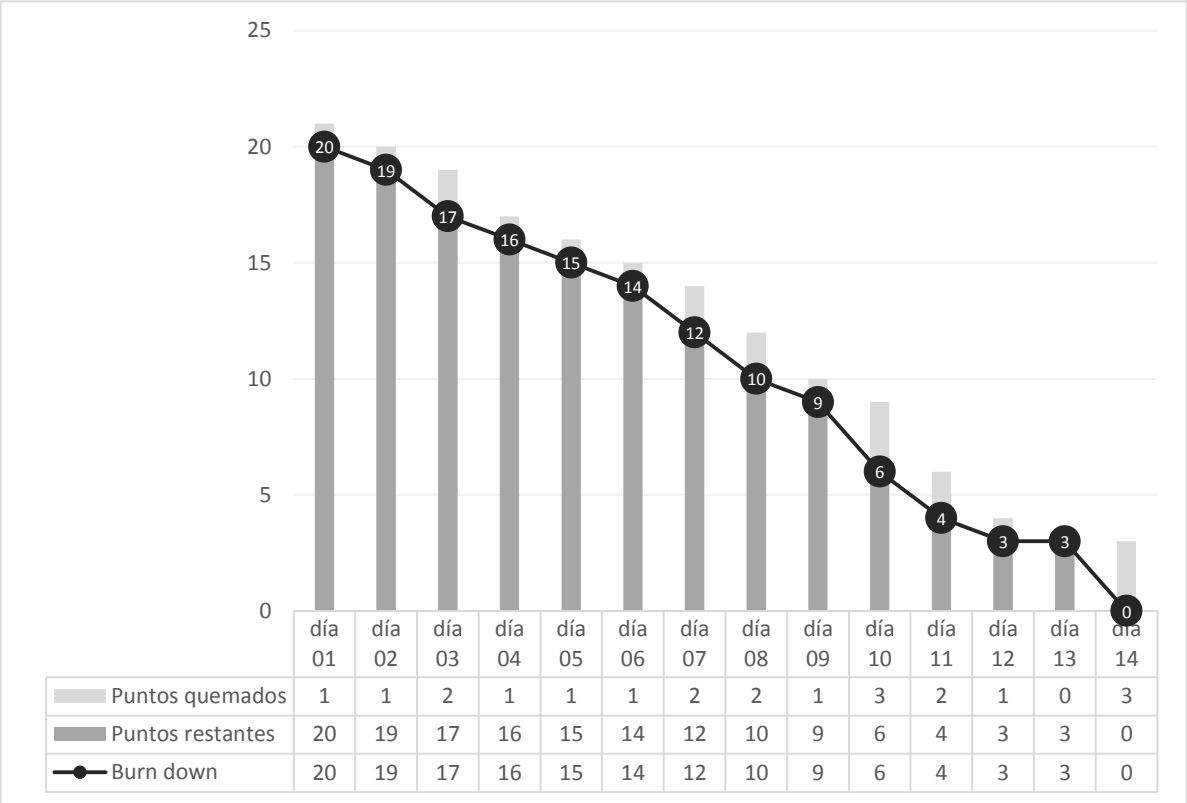
Evaluación de la iteración

Los objetivos planteados para este sprint fueron completados en el tiempo estimado y no surgieron inconvenientes importantes. Al final de este sprint se cumplieron dos hitos importantes, se logró obtener un escenario final terminado con detalles decorativos y efectos visuales y también un videojuego funcional listo para la fase beta, concluyendo la fase principal de desarrollo del proyecto.

Actualizar plan del proyecto

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron en el plazo estimado.

Burndown Chart



	día 01	día 02	día 03	día 04	día 05	día 06	día 07	día 08	día 09	día 10	día 11	día 12	día 13	día 14
Puntos quemados	1	1	2	1	1	1	2	2	1	3	2	1	0	3
Puntos restantes	20	19	17	16	15	14	12	10	9	6	4	3	3	0
Burn down	20	19	17	16	15	14	12	10	9	6	4	3	3	0

Figura 9: burndown chart sprint 8
Fuente: propia

FASE 4: BETA

Planificación del sprint 1 de la fase beta

En esta sección se describe el proceso que fue planificado para el primer sprint de la fase beta.

Aspectos funcionales y no funcionales

Los aspectos a tomar en cuenta a la hora de verificar el videojuego deben ser las mecánicas principales del juego, las gráficas y el rendimiento general en los equipos de los verificadores.

Medios de distribución

El medio de distribución para el juego de video es computadoras con Windows 10 o Debian. Los requisitos mínimos aún están por determinarse. El producto beta fue probado en la fase de desarrollo en equipos de gama alta con 16gb de memoria RAM y en equipos con 4gb de memoria RAM y no tuvo mayores inconvenientes. Debido a esto se determinó que se podían usar las características del dispositivo con menor memoria como las medidas para los requisitos mínimos. Para el primer sprint de la fase de pruebas se realizó un beta cerrado con 5 voluntarios a los que se les envió por correo un enlace a una página provisional donde podían descargar los ejecutables del juego para Windows o Debian.

Tabla 15: Estimación de requisitos mínimos

Recurso	**Mínimo**
CPU	Intel Celeron N830
RAM	4 GB

Medios de reporte de errores

Los errores para esta fase fueron reportados vía mensaje privado en Twitter por los voluntarios seleccionados. Se recopilaron los bugs detectados, sugerencias y reviews del beta del juego.

Equipo de verificadores beta

Como equipo de verificadores fueron seleccionados 5 voluntarios de edades entre los 12 y 35 años, entre los cuales, varios tenían tiempo siguiendo el desarrollo del proyecto por Twitter. Se utilizó esta plataforma para mantener el contacto con los verificadores.

Distribución

En esta sección se describen los medios de comunicación entre el equipo de verificadores y desarrolladores y los aspectos a verificar por los primeros.

Medios de comunicación.

Se utilizó la plataforma de mensajes privados de Twitter para mantener contacto directo con los verificadores.

Aspectos a verificar.

Los aspectos más importantes para verificar y en los que se necesitaba que los verificadores se enfocaran fueron las mecánicas principales del juego como el movimiento y control del personaje principal, las interacciones con edificios, objetos y NPC y manejo del inventario. También se deben verificar otros aspectos como el look and feel del juego, sus gráficas, jugabilidad, entretenimiento, aprendizaje y rendimiento en sus equipos.

Verificación

Evaluación y verificación.

Se evaluaron los aspectos mencionados anteriormente, recibiendo resultados satisfactorios y sugerencias en cuanto a la jugabilidad e historia del juego. El rendimiento del videojuego en los equipos de los verificadores no tuvo problemas ni bugs relevantes que afecten mucho el desarrollo del proyecto.

Reporte de resultados

Se reportaron algunos errores con las transiciones de las animaciones del personaje principal; se seguía reproduciendo la animación de caminata aun cuando el personaje estaba parado. También algunos efectos tuvieron detalles menores, como el del rastro de partículas que deja el personaje principal mientras camina.

Correcciones

Al hacer pruebas en Unity sobre los errores detectados se concluyó en que los orígenes de los problemas de animación venían del manejo de transiciones del animator controller, en donde la animación de caminata era disparada con un booleano que era true cuando el movimiento del personaje era distinto de 0. La solución para este problema fue hacer en el animator controller que la animación se disparara con un flotante con valores entre 0 y 1 que regulaba la velocidad de la reproducción de la animación, de esta forma, cuando el personaje se estuviera moviendo lentamente la animación se reproducía de manera pausada, y cuando el personaje se moviera rápido la animación se reproducía de forma que pareciera que el personaje estaba “corriendo”, esto mejoro significativamente los visuales del personaje principal. Se solucionó el inconveniente y se realizaron nuevas pruebas hasta comprobar que el error había desaparecido. El rastro de partículas también era disparado por un booleano que era true cuando el movimiento era diferente a cero. Se utilizó la misma variable que disparaba la animación de movimiento para regular la cantidad de partículas que dejaba el rastro del personaje cuando se movía y de esta forma se solucionó también este problema.

Planificación del sprint 2 de la fase beta

Aspectos funcionales y no funcionales

Los aspectos a verificar en este sprint son los mismos aspectos que en la iteración anterior, ahora con la corrección de los errores detectados previamente.

Medios de distribución

Para este sprint se creó una página web básica para distribuir el beta de forma abierta, con un formulario de contacto para facilitar el reporte de errores, reviews y sugerencias. Los requerimientos mínimos para ejecutar la aplicación siguen siendo los mismos que en el sprint pasado.

Medios de reporte de errores

Los errores para esta fase fueron reportados por el formulario de contacto de la página creada para el beta abierto y también por mensaje privado en Twitter por los voluntarios seleccionados en el sprint anterior. Se recopilaron los bugs detectados, sugerencias y reviews del beta del juego.

Equipo de verificadores beta

Se mantuvo el equipo de verificación del sprint pasado formado por 5 voluntarios, y adicionalmente se contó con 6 verificadores en este sprint que dieron su feedback por la pagina creada para el beta abierto.

Distribución

En esta sección se describen los medios de comunicación entre el equipo de verificadores y desarrolladores y los aspectos a verificar por los primeros.

Medios de comunicación.

Se utilizó el form de contacto de la página del beta abierto, y la plataforma de mensajes privados de Twitter para mantener contacto directo con los verificadores.

Aspectos a verificar.

Los aspectos más importantes para verificar fueron, de nuevo, las mecánicas principales, el rendimiento y los aspectos visuales y de interfaz del juego.

Verificación

Evaluación y verificación.

Los resultados de la evaluación fueron generalmente satisfactorios. Los verificadores hicieron sugerencias sobre mejoras que realizar a nivel de interfaces y sobre agregar elementos visuales más llamativos a la narrativa del juego.

Reporte de resultados

No se detectaron errores o bugs que puedan afectar la funcionalidad del juego ni la jugabilidad. Los verificadores aconsejaron hacer varias mejoras en cuanto a la interfaz gráfica, sugiriendo hacerla más llamativa y menos minimalista. También sugirieron mejoras en cuanto a la narración de la introducción del juego, para hacer esta sección más parecida a una novela visual, haciéndola más llamativa y para evitar que otros jugadores la pasen por alto, ya que es importante para dar contexto al juego.

Correcciones

Las correcciones de interfaz consistieron agregar más elementos visuales para que fuera más atractiva. La introducción del juego originalmente fue hecha únicamente como texto y música de fondo, pero luego de las sugerencias de los verificadores se determinó que sería mejor realizar ilustraciones y animaciones mínimas para hacer esta sección del juego más interactiva y entretenida. Estas sugerencias fueron realizadas dentro del plazo de correcciones dado para este sprint, en donde se ilustraron las imágenes de la introducción del juego y fueron diseñados los nuevos elementos de la interfaz gráfica, cambiando la barra de vida por corazones y agregando pequeñas ilustraciones a la interfaz general y de inventario.

FASE 5: CIERRE

En esta fase se pone a disposición del público la versión final del videojuego y se evalúa el proyecto. Esta fase se llevó a cabo dos semanas luego del cierre del proyecto, entre el 03 y el 10 de abril del 2021.

Liberación del videojuego

Entrega final

En esta fase se pone a disposición del público la versión final del videojuego en las formas establecidas. El videojuego en fase beta corregido fue entregado en la página web creada para el videojuego en la fase anterior en modalidad de beta abierto.

Definir entregable

Los productos a entregar serán los ejecutables para Windows y Debian, disponibles para descargar de manera gratuita y publica en la página web creada para el proyecto.

Realizar entregable

En esta fase se realizaron las tareas necesarias para incorporar todos los elementos que contiene el entregable final. Los entregables en versión final para Windows y Debian fueron exportados desde Unity.

Validar entregable

El juego hasta la fecha ha sido descargado por aproximadamente 15 personas desde la página web, sin contar al equipo de verificadores de la fase anterior.

Evaluación del proyecto

En esta fase se evalúa el proyecto y se extraen conclusiones sobre los éxitos y problemas ocurridos.

Evaluación postmorten

Evaluar proyecto

En esta sección se evaluó lo sucedido durante el desarrollo identificando problemas surgidos, cumplimiento de objetivos, y certeza de las estimaciones.

Antes de iniciar este proyecto se realizaron al menos dos proyectos en Unity con mecánicas de juego muy básicas, donde surgieron muchos inconvenientes y donde el equipo de desarrollo aprendió a manejar las herramientas inicialmente. Debido a esto, la curva de aprendizaje de Unity fue suavizada y fue más fácil estimar los tiempos para completar las tareas de los sprints de este proyecto. Las tareas que dieron más trabajo durante este proyecto y que consumieron la mayor parte del tiempo fueron las de investigación para la historia, búsqueda de referencias, creación de modelos 3d, escenarios, assets y diseño de interfaces, ya que las creaciones de estos elementos involucraban una cantidad de programas considerable y varios de estos programas nunca habían sido usados antes por el equipo de desarrollo. Pese a este detalle, los plazos planificados fueron cumplidos de forma exitosa en los tiempos estimados y el producto fue evaluado por los verificadores y jugadores del beta abierto, obteniendo en promedio una puntuación positiva.

Registrar lecciones aprendidas

Durante el desarrollo, las lecciones aprendidas, los problemas y soluciones surgidos y las notas recogidas a lo largo del proyecto fueron registrados en una página de Notion a modo de devblog. Estas notas fueron luego compartidas de forma pública como links externos en la página oficial del juego donde pueden ser de utilidad para futuros proyectos del equipo de desarrollo o para otros desarrolladores. También, el proceso del desarrollo fue registrado en Twitter, donde el proyecto genero varios seguidores**,** entre los que se encontraban los que luego serían los verificadores de la fase beta del proyecto.

Proponer mejoras a la metodología

La metodología aplicada se ajustó bien de forma general con el proyecto y la forma de trabajar del equipo. Sin embargo, una forma de hacer más efectivo el flujo de trabajo y

de manejar el seguimiento del sprint seria recolectar la información de las tareas completadas al final del día con un Daily Scrum, que es un elemento de la metodología SCRUM, de donde parte la metodología SUM, que fue implementada en el proyecto.

Otra mejora que se podía realizar en esta metodología seria eliminar la asignación de puntos y la forma en la que se ordena cuales tareas tienen más prioridad en base a este puntaje. Esto se debe a que durante el desarrollo de videojuegos pueden surgir tareas que no estaban planificadas o tareas planificadas que tienen un nivel de complejidad mayor al de otras que se creían más difíciles de completar y fueron ponderadas con un puntaje más alto. También hay tareas con cierto nivel de incertidumbre, lo que hace que el equipo las pondere de forma muy alta, y al momento de atacarlas, puede suceder que son más fáciles de completar que lo estimado inicialmente.

Por ejemplo, durante este proyecto, fueron ponderadas con un puntaje mayor las tareas que involucraban el desarrollo de mecánicas del juego, estas tareas fueron completadas en un plazo menor al planificado y sin inconvenientes importantes, en cambio otras tareas como las de la creación de los modelos y assets, fueron más difíciles de terminar y consumieron más tiempo que las otras tareas, aunque no se salieron de los plazos estimados.

CONCLUSIONES

El uso de la metodología para desarrollo de videojuegos SUM, permitió un desarrollo flexible, organizado y que se ajustó bastante bien al proyecto. Cada fase de la metodología ayudo a moldear lo que fue el producto final, comenzando con la fase de concepto, en la se establecieron las bases del proyecto, modelando los requerimientos en las áreas de desarrollo, modelado 3d, ilustraciones, y otros assets necesarios, además, en esta etapa se logró buscar una manera de adaptar el flujo de trabajo de la metodología al proyecto, lo cual contribuyo a minimizar los posibles retrasos y errores durante la producción. Durante la fase de desarrollo, se crearon múltiples productos de gran valor para el proyecto, y con la ventaja de que pueden ser reutilizados en otros niveles del juego e incluso en otros proyectos. La fase de cierre ayudo a que el proyecto pudiera crecer y mejorar gracias al feedback y las recomendaciones dadas por el equipo de testers y demás personas que dieron sus opiniones sobre el juego.

El uso del motor de juegos Unity permitió que este proyecto pudiera ponerse en marcha rápidamente, brindando facilidades de aprendizaje y una detallada documentación. Unity es sumamente flexible, cuenta con herramientas y funciones que son de gran valor no solamente en el área de desarrollo de videojuegos, sino, también en el área de simulaciones en general. Gracias a sus funciones nativas se agilizo en gran parte del desarrollo de este juego de video. Las herramientas de medición de recursos que ofrece fueron de gran utilidad al hacer ajustes de rendimiento durante toda la producción, lo que contribuyó a obtener un producto estable, de buen funcionamiento y con gran capacidad de ejecución en los distintos equipos en los que fue probado. Además, al poder generar ejecutables para distintas plataformas, el videojuego pudo ser probado en varios sistemas operativos Linux aparte de Windows 10, obteniendo resultados exitosos.

Durante el desarrollo de este proyecto, uno de los retos más grandes fue lograr manejar adecuadamente las herramientas y software involucrado en la creación de assets del juego. Afortunadamente, la herramienta de modelado 3D Blender cuenta con una extensa documentación, ejemplos y demás facilidades a la hora de aprender a manejarla,

lo cual ayudo considerablemente a suavizar la curva de aprendizaje de este programa, el cual fue utilizado para modelar gran parte de los assets 3d del video juego.

El programa de modelado Magica Voxel, fue de gran utilidad para crear los edificios que poblaban el escenario del videojuego, las herramientas que ofrece este programa, basado en el modelado 3d en base a cubos, ayuda a que el proceso de creación de modelos sea bastante rápido y fluido. Este punto era un reto bastante importante, debido a la gran cantidad de edificios que componen el escenario del juego y el tiempo limitado para modelarlos.

Substance Painter y Zbrush, fueron dos programas vitales para la creación de los detalles de los modelos 3d, siendo usados para crear las texturas y los modelos detallados, respectivamente. Además, al crear los detalles de los modelos con Zbrush mediante un mapa de normales, se ahorra una gran cantidad de polígonos en los modelos, aportando de gran forma al correcto desempeño del programa a la hora de ser ejecutado.

RECOMENDACIONES

Es recomendable agregar funcionalidades de accesibilidad como un modo para daltónicos, ajustes del brillo, y poder cambiar las fuentes de los textos a una más amigable con personas que sufran de dislexia.

Promover el desarrollo de juegos de video, proyectos audiovisuales e interactivos en la Universidad de Oriente como área de investigación y desarrollo.

Agregar funciones que permitan compartir capturas de pantalla de los edificios históricos en redes sociales.

Crear niveles adicionales que exploren a detalle los edificios del casco histórico y otras zonas históricas de Cumaná.

BIBLIOGRAFÍA

Acerenza, N., Coppes, A., Mesa, G., Viera, A., Fernández, E., Laurenzo, T., & Vallespir, D. (2009). Una Metodologıa para Desarrollo de Videojuegos. *38º JAIIO - Simposio Argentino de ingenieria de Software*, (págs. 171-176). Mar del Plata. Recuperado el 11 de Julio de 2018

Caballero, J. A. (2014). *APLICACIÓN MULTIMEDIA PARA LA NARRACIÓN DE LA HISTORIA DE LA IGLESIA SANTA INÉS EN EL PERÍODO DE 1929 HASTA LA ACTUALIDAD.* Proyecto de trabajo de grado, modalidad tesis de grado, Universidad de Oriente, Departamento de informática, Cumaná. Recuperado el 15 de Julio de 2018

Galindo, J. D. (2016). *DESARROLLO DE UN VIDEOJUEGO INDIE, 3D, DEL GÉNERO ROLE PLAYER.* Proyecto de trabajo de grado, modalidad tesis de grado, Universidad de Oriente, Departamento de informática, Cumaná. Recuperado el 15 de Julio de 2018

Jensen, A. (2016). Inside. Recuperado el 20 de Julio de 2018

Latorre, Ó. P. (2011). Géneros de juegos y videojuegos. Una aproximación desde diversas perspectivas teóricas. *Revista de Recerca i d'Anàlisi*, 127-146. Recuperado el 20 de Junio de 2018

Moldenhauer, J. (29 de Julio de 2017). Cuphead. (StudioMDHR, Ed.) Recuperado el 20 de Julio de 2018

Muñoz, L. D. (2014). *SIMULACION INMERSIVA DE LA IGLESIA SANTA INES DE CUMANÁ, ESTADO SUCRE, VENEZUELA. AMBIENTADA AL AÑO 1929.* Proyecto de trabajo de grado, modalidad tesis de grado, Universidad de oriente, Departamento de informática, Cumaná. Recuperado el 15 de Julio de 2018

Teron. (2012). *Casco Histórico de la ciudad de Cumaná.* Recuperado el 12 de Abril de 2018, de Sucre turístico: http://www.sucreturistico.com/casco.html

Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer, 38*(9), 25-32. Recuperado el 10 de Agosto de 2018

APENDICES

DOCUMENTO DE DISEÑO DEL JUEGO

Sumario

Follow the lights es un juego de aventura donde el jugador deberá explorar edificios históricos, enfrentándose a los fantasmas que los habitan o completando misiones para ayudarlos a descansar. Está ambientado en el casco histórico de Cumaná y en sus edificios en diferentes periodos de tiempo, de tal forma que el jugador puede explorar estos monumentos en diferentes épocas. En la versión demo el jugador puede explorar el casco histórico y ver las fachadas actuales de los edificios a los que podrá acercarse y obtener datos sobre el edificio en cuestión. También en la versión demo podrá probar las mecánicas básicas, como las de ataque, recolección de ítems e interacción con otros personajes.

Gameplay

En la versión final del juego, el rol del jugador será el de restaurar los monumentos a medida que va avanzando por los niveles, logra hacer esto reuniendo calicanto (un material de construcción de la época) o reuniendo los materiales para fabricarlo. La meta final es reconstruir por completo los edificios del casco histórico. Los obstáculos que se encontrará serán las diferentes quests, o misiones que tendrá que completar en los diferentes niveles, de las que obtendrá rewards, ítems importantes y aprenderá más sobre los aspectos menos conocido de la historia de los monumentos, los cuales representan los niveles del juego.

Mindset del jugador

Es un juego que requiere que el jugador esté dispuesto a explorar. En cada quest se van revelando fragmentos de la historia y cuentos poco conocidos, tocando el factor humano de la historia, los personajes, sus motivaciones y se intenta ir más allá de contar la historia como un libro, haciendo que el personaje del jugador se sienta parte de ella y explotando su curiosidad.

Comportamientos

Jugador

- Puede ser controlado por el sistema de input.
- Movimiento.
- Salto.
- Uso de ítems.

- Atacar.

Enemigos

- Ataque a distancia corta.
- Persigue al objetivo si está en su campo de visión.

GUION DEL NIVEL DE INTRODUCCIÓN

Follow the lights: nivel inicial e introducción

El usuario aprende a utilizar los controles y a interactuar con el escenario. En este nivel se aprenden las mecánicas básicas como:

- Control del personaje
- Control de la cámara
- Saltar
- Interacción principal
- Guardar ítem en el inventario
- Abrir inventario
- Equipar ítem

Mensaje de la interfaz: ¨debes encontrar a tus compañeros, muévete usando las teclas de dirección. Puedes interactuar con otros personajes acercándote a ellos y apretando la tecla E¨.

El personaje es guiado por letreros y flechas desde el punto inicial hasta el castillo Santa María de la Cabeza. En el camino se encuentra distintos personajes con los que puede interactuar, que le explicarán los controles del juego y le darán datos históricos sobre los edificios del lugar.

Tu: debo ir al fuerte santa maría, creo que está al lado de la iglesia.

Señora: ¿puedes alcanzarme ese mango? Intenta saltar al techo usando la barra de espacio.

Bachiller Zerpa: cuidado con acercarte a las áreas oscuras, dice la leyenda que en las noches ahí aparecen animas. Puedes espantarlas con tu linterna y atacarlas con la tecla E.

Hombre malhumorado: ¡no me molestes!

Vendedor de pinchos: toma una muestra gratis, puedes guardar este pincho para comerlo más tarde y verlo en tu inventario apretando la tecla Q.

Vendedor ambulante: puedes interactuar con objetos acercándote a ellos y apretando E.

Personaje de relleno: Puedes mover la cámara moviendo el mouse … espera, ¿qué mouse? ¿de qué hablo?

Vendedor de libros: ¿quieres ver los libros más de cerca? Intenta hacer zoom con la ruedita del mouse.

Llegas a la entrada del castillo Santa María de la cabeza, donde te encuentras con tu profesor y otros estudiantes.

Niño vestido de soldado**:** mi mamá me hizo este traje, ¿de que estas disfrazada?

Niña vestida de dama antañona**:** hoy día de Santa Inés, patrona de Cumaná… Oh espera, esa canción es para otro día.

Niño vestido de árbol**:** no te preocupes si aún no te sabes tus líneas del dialogo. Aun no se me las mías.

Profesor: ¡es hora de comenzar la obra! ¡apúrense! ¡tú! Ve a buscar tu guion, está dentro del castillo, la reja se ha cerrado, así que debes saltar para poder entrar.

Saltas la reja y entras al castillo Santa María de la Cabeza y escuchas un estruendo. En este sitio es donde comienza el primer nivel del juego y cambia a la siguiente escena.

HOJA DE METADATOS

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 1/6

Título	Videojuego del género rol ambientado en el casco histórico de la ciudad de Cumaná, estado Sucre
Subtítulo	

Autor(es)

Apellidos y Nombres	**Código CVLAC / e-mail**	
Valentina Azócar Gómez	**CVLAC**	**24.753.255**
	e-mail	**pizenblues@gmail.com**
	e-mail	
	CVLAC	
	e-mail	
	e-mail	

Palabras o frases claves:

Videojuego, Juego rol, Casco Histórico, Cumaná.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 2/6

Líneas y sublíneas de investigación:

Área	Subárea
Ciencias	Departamento de informática

Resumen (abstract):

Se desarrolló un juego de video 3D del género *rol* en el motor de juegos Unity, el cual brinda un motor de simulación de físicas, facilidades y flexibilidad a la hora de crear el ambiente del juego, un punto muy importante para este proyecto, ya que se tenía como uno de los objetivos crear una versión estilizada de los edificios reales del casco histórico de Cumaná, ambiente en el que transcurre el juego. Para el desarrollo de las interacciones en el videojuego, Unity utiliza un sistema de scripts que se agregan a los componentes de cada GameObject (objetos base de Unity que cambian según las propiedades y componentes que se les dé) en el editor de Unity. Estos scripts utilizan el API de Unity y el lenguaje de programación C#, por lo que la totalidad del proyecto fue desarrollado en este lenguaje de programación. Los modelos 3D fueron creados utilizando programas como Blender, Magica Voxel, Zbrush, Substance Designer y Krita. Para la planificación del desarrollo se utilizó la metodología SUM (Acerenza, Copes, Mesa y Viera., 2009) para videojuegos, una versión de SCRUM adaptada para el desarrollo de videojuegos. El software en cuestión se desarrolló dirigido a un público joven, con limitado acceso a la información histórica de la ciudad y no esté familiarizado o tengan dificultades de aprendizaje utilizando medios más convencionales como libros, manuscritos, mapas, entre otros. Sin embargo, este videojuego también fue adaptado para que pudiera ser disfrutado por un público mayor, que reconocería los lugares, símbolos y referencias dentro del videojuego y brindarles la oportunidad de revisitar el ambiente. Inicialmente se realizó en el primer sprint la distribución de las actividades, aparte de la inicialización del proyecto. En los siguientes sprints se lograron diversas versiones del producto, comenzando por las funciones y mecánicas básicas para tener un producto funcional jugable y listo para probar, posteriormente se trabajó en la creación de los assets necesarios como modelos 3d, imágenes de las interfaces, animaciones y diseños requeridos. También se siguió trabajando continuamente en nuevas funcionalidades de menor importancia y la mejora de funcionalidades ya terminadas. La fase de cierre ayudo a que el proyecto pudiera crecer y mejorar gracias al feedback y las recomendaciones dadas por el equipo de testers y demás personas que dieron sus opiniones sobre el juego. Al culminar se consiguió desarrollar un producto beta liberado y disponible para la descarga gratuita, tomando en cuenta las correcciones y sugerencias de estos en cada iteración.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 3/6

Contribuidores:

Apellidos y Nombres	ROL / Código CVLAC / e-mail	
Romero Carmen	ROL	CA ☐ AS ☐ TU ☐ JU ☐
	CVLAC	10.947.403
	e-mail	cvromerob@gmail.com
	e-mail	
Fuentes Ana	ROL	CA ☐ AS ☐ TU ☐ JU ☐
	CVLAC	12.666.425
	e-mail	afuentesmarquez28@gmail.com
	e-mail	

Fecha de discusión y aprobación:

Año	Mes	Día
2021	07	08

Lenguaje: **SPA**

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 4/6

Archivo(s):

Nombre de archivo	Tipo MIME
Valentina_Azócar_Gómez_Videojuego_del_genero_rol.docx	Aplication/Word

Alcance:

Espacial : Nacional (Opcional)

Temporal: Temporal (Opcional)

Título o Grado asociado con el trabajo:
Licenciatura en Informática

Nivel Asociado con el Trabajo:
Licenciatura

Área de Estudio:
Informática

Institución(es) que garantiza(n) el Título o grado:
Universidad de Oriente

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 5/6

UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

CU Nº0975

Cumaná, 04 AGO 2009

Ciudadano
Prof. JESÚS MARTÍNEZ YÉPEZ
Vicerrector Académico
Universidad de Oriente
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda **"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC Nº 696/2009".**

Leído el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.

Comunicación que hago a usted a los fines consiguientes.

UNIVERSIDAD DE ORIENTE
SISTEMA DE BIBLIOTECA
RECIBIDO POR
FECHA 5/8/09 HORA 5:30

Cordialmente,

JUAN A. BOLAÑOS CURIEL
Secretario

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD DE ORIENTE
SECRETARÍA
CONSEJO UNIVERSITARIO

C.C: Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.

JABC/YGC/maruja

Apartado Correos 094 / Telfs: 4008042 - 4008044 / 8008045 Telefax: 4008043 / Cumaná - Venezuela

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 6/6

Articulo 41 del REGLAMENTO DE TRABAJO DE PREGRADO (vigente a partir del II Semestre 2009, según comunicación CU-034-2009): "Los trabajos de grados son de la exclusiva propiedad de la Universidad de Oriente, y solo podrá ser utilizados para otros fines con el consentimiento del Concejo de Núcleo respectivo, quien deberá participarlo previamente al Concejo Universitario, para su autorización".

Br. Valentina Azócar
Autor

Profa. Carmen Victoria Romero
Asesor

Profa. Ana Teresa Fuentes
Co-asesor