



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo Terminal

“Yolotl: un videojuego para fomentar la cultura”

2017-A035

Presentan
Hernández Bautista Yasmine Pilar

Márquez Hernández Karla Rocío

Directores
M. en C. Rafael Norman Saucedo Delgado.

Lic. Ulises Vélez Saldaña.



Junio 2018



No. de TT:2017-A035

17 de noviembre de 2017

Documento Técnico Parte A

“Yolotl: un videojuego para fomentar la cultura”

Presentan
Hernández Baustista Yasmine Pilar¹

Márquez Hernández Karla Rocío²

Directores

M. en C. Rafael Norman Saucedo Delgado. ***Lic. Ulises Vélez Saldaña.***

RESUMEN

En México la industria de videojuegos tiene una alta demanda de consumo; sin embargo, existen pocos estudios que desarrollen videojuegos basados en la cultura mexicana. Actualmente en México existe un fuerte desinterés en la cultura nacional. El presente trabajo terminal consiste en el desarrollo de un videojuego que fomente la cultura con temática de la cultura mexica.

Palabras clave. – Cultura mexica, desarrollo tecnológico, ingeniería de software, videojuego.

¹daughterofthewind10@gmail.com

²winged.zelda@gmail.com

Capítulo 1

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.” La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen. Información adicional sobre este reporte técnico podrá obtenerse en: La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Índice general

Capítulo 2

Introducción

En la época actual nos vemos rodeados por tecnología por todos lados, esta ya es parte de nuestra vida diaria, de nuestras actividades tanto de trabajo, de nuestro medio de comunicación, como medio de información e incluso de nuestro medio de entretenimiento. Las generaciones recientes han crecido con la evolución de la tecnología a un ritmo acelerado, a tal punto que la tecnología ya es parte de su cultura.

Dentro de la evolución de la tecnología se encuentra los videojuegos, una industria de entretenimiento. Nos daremos a la tarea de investigar los puntos de impacto que tiene. Pues a primera instancia podremos observar el tipo de personas que juegan, los ingresos que se generan en esta industria, los tipos de industria que existen, las consecuencias positivas, las consecuencias negativas que generarían, las ganancias como profesionista en esta rama y como realizar un proyecto de esta naturaleza.

Los videojuegos hacen al jugador involucrarse con varios sentidos a la vez en lo que se le presenta, así se crea una experiencia propia como cualquiera de la vida pues provoca la inmersión. El INJUVE ha dado a conocer información donde se puede observar quelos jóvenes y los videojuegos llevan una interacción diaria y sobre cualquier tema, desde educativo hasta de ocio.

Por otra parte tenemos que la sociedad ignora en su mayoría los aspectos históricos culturales propios de su país, específicamente de México. Se denota un gran desinterés por parte de la gente el siquiera conocer su legado.

Tomando en cuenta todos los aspectos anteriores, el proyecto consistirá en juntar ambas ideas para usar el videojuego como difusión del aspecto cultural mexicano. Se investigará más a fondo la historia de los videojuegos para elegir el público objetivo potencial o las personas alcanzables, los procesos y metodología para poder realizar un videojuego, aspectos a considerar para el proyecto, la cultura y ramas que abarca dentro de la sociedad, la cultura y la tecnología como se relacionan entre sí, las herramientas con las que se cuentan para la realización del videojuego, las teorías que pueden usarse, los conflictos que pueden ocurrir, como solucionar los problemas que se nos presenten y por su puesto el cambio o impacto que se tenga al presentarlo al público.

Al final se presentará un videojuego como producto con pruebas de diferentes tipos para demostrar el resultado ante la sociedad y los cambios que se presentaron en las personas.

Capítulo 3

Antecedentes

En este capítulo se presenta el marco teórico, la propuesta y el avance realizado durante TT1. Primeramente se presenta la propuesta del trabajo. Después se definen algunos conceptos como son el videojuego, la cultura, las metodologías de desarrollo de videojuegos, es decir el marco teórico. Seguido de esto se introduce la solución de la problemática a tratar. Una vez hecho esto, se mencionan los ajustes que se realizan en el proyecto para trabajo terminal 2, tales como la asignación de trabajo, el enfoque del problema y la actualización del motor de juego. Lo siguiente en ser presentado son las observaciones realizadas por los sinodales en la presentación del trabajo terminal 1 y como fueron solucionadas. Finalmente se hace un resumen del trabajo realizado durante el trabajo terminal 1, correspondiente a la etapa de preproducción y a los dos primeros sprints de la etapa de producción.

3.1. Propuesta

En esta sección se presenta a manera de resumen las propuestas y los conceptos definidos durante el trabajo terminal 1, tales como el planteamiento del problema, conceptos y definiciones referentes al videojuego y su desarrollo, la definición y delimitación de la cultura y el planteamiento de la solución que se desarrolla durante el trabajo terminal.

3.1.1. Planteamiento del problema

En México existe un fuerte desinterés y desconocimiento hacia su cultura e historia nacional. De acuerdo con la Tercera Encuesta Nacional de Cultura Constitucional, el 52.7 % de los encuestados desconoce el año en que se aprobó la constitución nacional y no la relaciona con la Revolución Mexicana **RefConsti**. Con base en la encuesta realizada por Parametría, empresa dedicada a la investigación estratégica de la opinión y análisis de resultados, solo el 32 % de su encuestados supo que México se independizó de España, el 51 % desconoce el país del que se independizó México, mientras que el resto del porcentaje de los encuestados piensa que México se independizó de otro país que no es España; la misma encuesta realizada por Parametría señala que el 25 % de los encuestados mencionaron personajes históricos ajenos a la independencia de México como partícipes de ésta y el 12 % respondió no saber qué personajes históricos participaron en la independencia **RefParametria**.

Con base en lo anterior, el problema que se plantea solucionar en el presente trabajo terminal es el fomento de la cultura histórica nacional de México.

3.1.2. Marco Teórico

En esta sección se presentan los conceptos básicos para comprender el trabajo realizado durante el desarrollo del trabajo terminal, tales como la definición del videojuego, sus características, su clasificación, las metodologías de desarrollo, las herramientas para el desarrollo y la cultura.

Videojuego

El grupo de periodista especializado en tecnología y desarrollo de software Carricay define al videojuego como: una aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico"**Ref_DefVideo**

Al igual que con otros productos tecnológicos, la evolución de los videojuegos ha sido vertiginosa, resultando complicado mencionar características comunes para todos los videojuegos. Sin embargo, en el libro "*Marketing y videojuegos: Product placement, in-game, advertising y advergaming*" se menciona que existen seis características comunes en los videojuegos: Interactividad, entretenimiento, jugabilidad, simulación\virtualidad, inmersión y multiplataforma**RefCarac** a continuación se menciona en qué consisten cinco de las seis características, esto debido a que la última no se encuentra presente en todos los juegos y el mismo autor de la obra la menciona como una característica opcional a tomar en cuenta:

- **Interactividad:** En el artículo "*Defining Virtual Reality: Dimensions Determining Telepresence*" se define la interactividad como la capacidad de los usuarios para participar y modificar la forma y el contenido de un entorno mediado en tiempo real**RefInteractividad**
- **Entretenimiento:** en el artículo "Las Tecnologías del Entretenimiento: Pasado, Presente y Futuro", el entretenimiento "se asocia, usualmente, de hacer algo que nos divierte, algo que podemos hacer solos o con otros, para entretenernos o divertirnos, en nuestro tiempo libre, o tal vez, algo que nos relaje o que nos haga reír"**RefEntretenimiento**
- **Jugabilidad:** en el libro "*Marketing y videojuegos: Product placement, in-game, advertising y advergaming*" se define la jugabilidad como "la relación que existe entre todas las acciones reacciones e interacciones tanto del videojugador como el videojuego como entre los propios sistemas y subsistemas programados en el videojuego"**RefCarac**
- **Simulación \Virtualidad:** La simulación "se trata de una representación a medida cuyo objetivo nos permite interactuar y relacionarnos con lo representado según nuestros intereses"**RefCarac**
- **Inmersión:** Con base en el libro "La vida en la pantalla: La construcción de la identidad en la era de internet", la inmersión es un proceso psicológico que se produce cuando la persona deja de percibir de forma clara su medio natural al concentrar toda su atención en un objeto, narración, imagen o idea que le sumerge en un medio artificial **RefInmersion** Por otra parte en la tesis "Libertad dirigida: Análisis formal del videojuego como sistema, su estructura y su avataridad", la inmersión se entiende como la coherencia de la ficción del juego y su aceptación por el jugador.**refInmersionNavarro**

Los videojuegos pueden ser clasificados con base a su jugabilidad, en el libro "Juego. Historia, Teoría y Práctica del Diseño Conceptual de Videojuegos" **Ref_JuegoDisenio** se propone la siguiente clasificación.

- **Juegos de acción:** Son juegos usualmente de temática violenta. El jugador lucha por su supervivencia, para ello se vale de armas o habilidades de combate.
- **Juegos de estrategia:** Para que el jugador logre sus objetivos en este tipo de juegos, éste debe de planear una estrategia, normalmente a largo plazo.
- **Juegos de Rol:** La mecánica de los juegos de rol gira en torno a un grupo de héroes, con habilidades y progresión definidos; el grupo de héroes debe de trabajar coordinadamente para cumplir un objetivo; estos héroes pueden ser controlados por un solo jugador o por varios. El jugador deberá explorar un mundo de gran tamaño haciendo evolucionar a sus personajes y sus habilidades.
- **Videojuego de aventura:** Son parecidos a los juegos de Rol; con la peculiaridad de que tienen una progresión más lineal y no se hace tanto énfasis en los combates, siendo su eje principal la narrativa.
- **Videojuegos de deportes:** Son todos aquellos videojuegos que tratan sobre deportes que no involucren la conducción de un vehículo. Pueden ser juegos sobre fútbol, fútbol americano, tenis, etc.
- **Videojuegos de carreras de vehículos:** Son todos aquellos se centran en las carreras con todo tipo de vehículos, mayoritariamente automóviles.
- **Videojuegos puzzle:** Este tipo de juego involucra la resolución de un problema a partir de la utilización de una serie limitada de recursos, por lo que si los recursos no se utilizan de la manera correcta el problema no podrá ser solucionado.

Dentro de la clasificación de los juegos de acción se encuentran los juegos de plataforma, definidos por una jugabilidad donde el jugador debe de controlar a un personaje con el que se desplaza saltando entre plataformas y esquivando todo tipo de obstáculos y enemigos **Ref_JuegoDisenio**

Es importante que se entienda el concepto del videojuego, sus características, su clasificación y la jugabilidad básica de un juego de plataforma ya que el presente Trabajo Terminal gira entorno al desarrollo de un videojuego de plataforma que cumple con cinco de las seis características de los videojuegos.

Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software **Ref_metodología** Para el presente trabajo terminal se consideran las siguientes metodologías como candidatas a implementar para guiar el desarrollo:

- **Metodología en cascada:** Sigue una progresión lineal por lo que cualquier error que no se haya detectado con antelación afectará todas las fases que le sigan provocando una redefinición en el proyecto y por ende un aumento en los costos de producción del sistema **Ref:CarCascada** Esta metodología se divide en las siguientes etapas:

- Análisis de los requisitos del software.
 - Diseño.
 - Codificación.
 - Pruebas.
 - Mantenimiento.
- **Metodología en Scrum:** Scrum parte de la visión general que se desea que el producto alcance; a partir de la visión se inicia la división del proyecto en diferentes módulos. Scrum implementa una jerarquía entre los módulos en donde los de mayor jerarquía son los que se desarrollan al inicio del proyecto o durante las primeras iteraciones o *sprints* Ref_ScrumRefCada sprint se compone de las siguientes fases:
- Concepto.
 - Especulación.
 - Exploración.
 - Revisión.
 - CierreRef_ScrumGuia
- **Metodología de Programación extrema:** Es una metodología de desarrollo ágil y adaptable, soporta cambios de requerimientos sobre la marcha. Su principal objetivo es aumentar la productividad y minimizar los procesos burocráticos, por lo que el software funcional tiene mayor importancia que la documentaciónRef_XP
- **Metodología Huddle:** Es una metodología cuya funcionalidad se basa en la metodología Scrum, con la diferencia de que está orientada al desarrollo de videojuegos. De naturaleza ágil, resulta óptimo para equipos multidisciplinarios de 5 a 10 personas; es iterativa, incremental y evolutiva Ref_Huddle Huddle se divide en las siguientes etapas:
- Preproducción.
 - Producción.
 - Postmorten.

Tras un análisis comparativo entre metodologías, se elige a *Huddle* como la metodología a guiar el desarrollo del Trabajo Terminal; esta elección se basa principalmente en que dicha metodología está enfocada a videojuegos y no requiere ser adaptada por lo que se puede llevar a cabo el proyecto de manera directa sin tener que invertir tiempo en adaptar la metodología a las necesidades del desarrollo de un videojuego.

Herramientas de desarrollo

Como cualquier desarrollo de software, el desarrollo de un videojuego requiere de software especializado tal como un motor de juego, editores de imágenes, software de diseño, de edición de audio, etc. En este apartado se van a definir algunas de las herramientas utilizadas durante la elaboración del trabajo terminal.

La primera herramienta a definir es el del motor de juego. El motor de juego, también conocido como *Game Engine*, parte del concepto de reutilización; es decir, es posible generar juegos a partir de un código base y común mediante una separación adecuada de los componentes fundamentales, tal como visualización de gráficos, control de colisiones, físicas, entrada de datos etc **Ref:MotorGraf** esto permite a quienes trabajen en un juego puedan centrarse en todos aquellos detalles que hacen al juego único. Dentro del mercado existen diferentes opciones de motores de juego tales como *Unity3D*, *UnrealEngine* y *CryEngine*, por citar algunos. Para el presente trabajo terminal se decide por utilizar *Unity3D* ya que ofrece:

- Desarrollo multiplataforma, lo que permite aumentar la escalabilidad del proyecto.
- Curva de aprendizaje rápido.
- Comunidad de desarrolladores activa.
- Tres opciones de lenguajes de programación para utilizar: *C#, JavaScript* y *Boo*.
- No requiere de muchos recursos para su instalación.
- Uso de diferentes tipos de licencia lo que permite contar con una licencia gratuita, de pago y una de negocios. No existiendo mucha diferencia de funcionalidad entre la licencia libre y la de pago.

En lo que refiere a la creación del entorno gráfico del videojuego, es decir de sus sprites, se decide utilizar los *software* de diseño *Adobe Photoshop* y *Corel Draw*. Ya que al momento de elegir dichos softwares ya se contaba con experiencia previa sobre su funcionamiento y no requiere ningún tipo de periodo de prueba para familiarizarse con su funcionamiento. Ambos *softwares* son de pago y para el desarrollo del presente trabajo terminal se utiliza una licencia personal por lo que si se desea comercializar el juego va a ser necesario adquirir otro tipo de licencia para la generación de *sprites*.

Cultura

Una vez explicado lo que es el videojuego, su metodología de desarrollo y las herramientas a usar para desarrollarlo, es preciso definir lo que es la cultura; para tal objetivo el presente trabajo se vale de la definición propuesta por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO, por sus siglas en inglés). La UNESCO define la cultura como “el conjunto de los rasgos distintivos, espirituales y materiales, intelectuales y afectivos que caracterizan a una sociedad o un grupo social. La cultura engloba, además de las artes y las letras, los modos de vida, los derechos fundamentales al ser humano, los sistemas de valores, las tradiciones y las creencias; de igual forma la cultura da al hombre la capacidad de reflexionar sobre sí mismo**RefCultura**”. Bajo su misma definición la UNESCO, se plantea que la importancia de la cultura radica en su capacidad de hacer a los seres humanos racionales, críticos y éticamente comprometidos; ya que, través de ella se disciernen los valores y se efectúan opciones. Siendo por medio de ella que el hombre se expresa, toma conciencia de sí mismo, se reconoce como un proyecto inacabado, pone en cuestión sus propias realizaciones, busca incansablemente nuevas significaciones, y crea obras que lo trascienden **RefCultura**

Para efectos del presente trabajo terminal, esté únicamente va a abordar la cultura de carácter histórica, es decir la cultura que hace referencia a la herencia social, es decir aquella que relaciona a la sociedad con su pasado **RefculturaClasificacionEl**

3.1.3. Planteamiento de la solución

Con el fin de fomentar la cultura y la historia se desarrolla Yolotl, un videojuego de plataforma y aventuras en dos dimensiones para dispositivos móviles android 5.1 de gama media alta.

Las razones por las que se aborda la solución del problema con un videojuego se debe principalmente a diferentes factores tales como:

- **El estado de la industria mexicana de los videojuegos:** En el 2017 México ocupó el 12 puesto en cuanto a consumo de videojuegos percibiendo un ingreso de 1.4 mil millones de dólares en esta industria. A su vez México cuenta con 49.2 millones de jugadores [Ref_JuegosGanancia](#)
- **El auge de los juegos para teléfonos móviles:** En el 2017 la industria del videojuego tuvo ganancias de 108.9 mil millones de dólares de los cuales el 32 % de las ganancias fueron generadas por los teléfonos inteligentes y un 10 % por las tablets; con este porcentaje los teléfonos superaron a las consolas de mesa en ingresos [Ref_JuegosGanancia](#)
- **El consumo de teléfonos móviles en México:** En el 2017 México contaba con 52 millones de usuarios de teléfonos móviles, lo que lo ubicó en el 9 puesto a nivel mundial en el consumo de teléfonos inteligentes [Ref_TelefonosGanancia](#)
- **La interactividad de un videojuego:** Como se menciona en el artículo *Identification with the Player Character as Determinant of Video Games Enjoyment*: en los videojuegos, la interactividad juega un papel importante para identificar y adoptar un determinado concepto, ya que dentro del videojuego el jugador no es un espectador, pues participa directamente en la historia e interactúa con el mundo del personaje; esto genera una relación íntima entre el jugador y el personaje puesto que es gracias al jugador que el personaje puede avanzar en la historia y a su vez es gracias al personaje que el jugador puede interactuar con la historia [PlayerIdentification](#)

3.2. Ajustes al trabajo terminal

En esta sección se definen todas las nuevas estrategias a seguir para agilizar y optimizar el desarrollo del juego.

3.2.1. Corrección del enfoque de la solución

El enfoque de gamificación no permitía responder la pregunta de “¿Se puede gamificar un videojuego?”, ya que definiendo la gamificación, ésta queda dada como el uso de elementos del juego y el pensamiento basado en juego en entornos no relacionados con el juego para aumentar el compromiso o modificar el comportamiento [RefIntroGamificacion](#) la gamificación de un juego resulta en una contradicción del mismo concepto ya que gamificación solo es aplicable a entornos ajenos a juegos. Por su parte *Serious Games* queda definido como juegos cuyo propósito va más allá del entrenamiento [Ref:Serious](#) Los *Serious Games* pueden ser utilizados para la educación, la capacitación, la medicina como medio terapéutico y en última instancia para la persuasión, es decir un juego puede hacer consciente al jugador de algún problema [ferreira2002serious](#) Desde el punto de vista de la gamificación, el presente trabajo terminal debería desarrollar una aplicación con elementos del juego para facilitar la transmisión de la cultura; mientras desde el enfoque de los

Serious Games, la cultura se ayuda de un videojuego para ser transmitida. El concepto y el enfoque de *Serious Games* se ajusta al que aborda el juego desarrollado durante el presente trabajo terminal por lo que se decide modificar el enfoque de gamificación al de *Serious Games*.

3.2.2. Nueva división de trabajo

Para realizar la segunda parte del trabajo terminal se decidió cambiar la división del trabajo con el objetivo de agilizar el desarrollo, ya que al terminar los sprints anteriores se podía observar que la realización de un nivel estaba llevando demasiado tiempo. Por tal motivo se decide reorganizar la asignación de tareas; en lugar de que los miembros del equipo de desarrollo se encarguen del mismo nivel, se reparten los niveles restantes del desarrollo entre los integrantes del equipo. Quedando la asignación de los niveles como se ve en la figura ?? .

Asignación de los niveles del juego	
Karla Rocío Márquez Hernández	Yasmine Pilar Hernández Bautista
Nivel 01, la chica y el perro (Ciudad)	Nivel 02, Nadie cruza mis dominios (Plataforma)
Nivel 01, la chica y el perro (Selva)	Nivel 02, Nadie cruza mis dominios (Jefe)
Nivel 03, la guarida del jaguar (Plataforma)	Nivel 04, Alas de obsidiana (Plataforma)
Nivel 03, la guarida del jaguar (Jefe)	Nivel 04, Alas de obsidiana (Jefe)
Nivel 05, el viento del norte (Plataforma)	Nivel 06, Sin gravedad (Plataforma)
Nivel 05, el viento del norte (Jefe)	Nivel 06, Sin gravedad (Jefe)
Nivel 07, castigo (Plataforma)	Nivel 08, la última batalla del jaguar (Plataforma)
Nivel 07, castigo (Jefe)	Nivel 08, la última batalla del jaguar (Jefe)
Nivel 09, el último caballero del rey (Plataforma A)	Nivel 10, el rey del Mictlán (Jafe fase 01)
Nivel 09, el último caballero del rey (Plataforma B)	Nivel 10, el rey del Mictlán (Jafe fase 02)
Nivel 09, el último caballero del rey (Jefe A)	Nivel 10, el rey del Mictlán (Jafe fase 03)

Figura 3.1: Asignacion de tareas

Esta división de trabajo permite que los niveles se desarrollen de manera paralela y no de manera secuencial como se había trabajado hasta este *sprint*; simulando de esta forma un flujo de trabajo similar a procesamiento multihilo, en el que cada integrante del equipo es un hilo y desarrolla sus tareas de manera paralela al otro.

3.2.3. Actualizando el motor de juego

Se decidió cambiar la versión 5.4 de *Unity* por la versión 2017.3.1f de *Unity3D*. Esta versión incluye herramientas que agilizan la creación de niveles como el uso de:

- **Tilemap:** Herramienta para el mapeado de niveles. Esta herramienta facilita la creación de mapas al crear una malla sobre la que se arrastran diferentes *Sprites* que se hayan importado previamente al tilemap (ver figura ??). En la sección ?? se profundizará su funcionamiento.



Figura 3.2: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

- **Cinemachine:** Asset que permite controlar la cámara de la escena, con este asset se le puede indicar que objeto se desea que la cámara siga y se puede asignar un área que limitará el movimiento de la cámara (ver figura ??). *Cinemachine* se descarga directamente desde la tienda de assets de *Unity* y fue desarrollado por los ingenieros de *Unity*, lo que significa que no genera conflictos o no requiere de configuraciones extras al proyecto para importar. En la sección ?? se profundizará su funcionamiento.

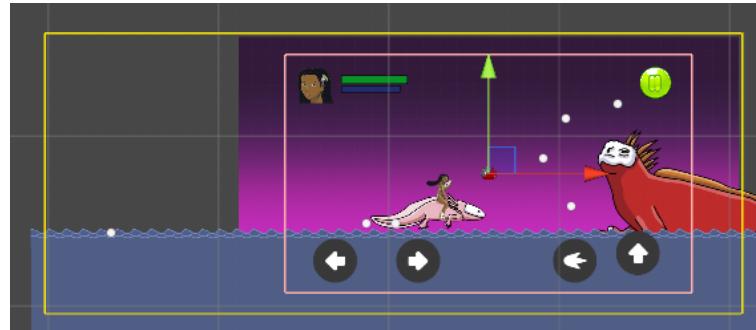


Figura 3.3: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

- **Sprite Packer:** Si bien no es una herramienta para construcción de niveles o un *asset*, esta herramienta es una de las más útiles que se agregó a la nueva versión de *Unity* ya que, como su nombre lo indica, permite el empaquetado de *sprites* (ver figura ??). Empaquetar los *sprites* es una práctica que optimiza el renderizado de objetos, pues el controlador de gráficos de *Unity* realiza una sola llamada por paquete cuando renderiza los objetos y con esa única llamada renderiza todos los objetos de la escena que se encuentren en ese paquete; si los *sprites* no se encontraran dentro de un paquete el controlador de gráficos de *Unity* haría una llamada por cada *sprite* aumentando el tiempo de renderizado.

Por el impacto que tendrían las nuevas herramientas de la versión de *Unity*, se propone utilizarla en lugar de la versión 5.6.2f1. Antes de actualizar la versión de *Unity* se investigó si el proyecto sufriría algún impacto negativo como falta de compatibilidad de componentes por la diferencia de

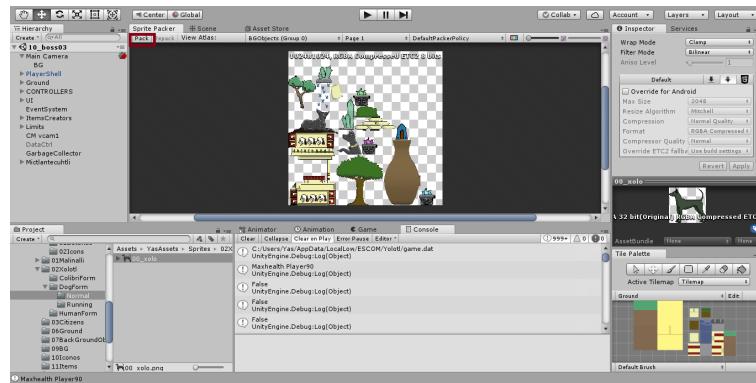


Figura 3.4: Vista de la pestaña del *Sprite Packer*.

versiones. Al comprobar que existía una total compatibilidad entre ambas versiones en cuanto a trasladar un proyecto de la versión 5.6.1f a la versión 2017.3.1f. Se determinó que la nueva versión de *Unity* sería la que se emplearía para el resto del desarrollo del juego.

3.2.4. Máquina de estados

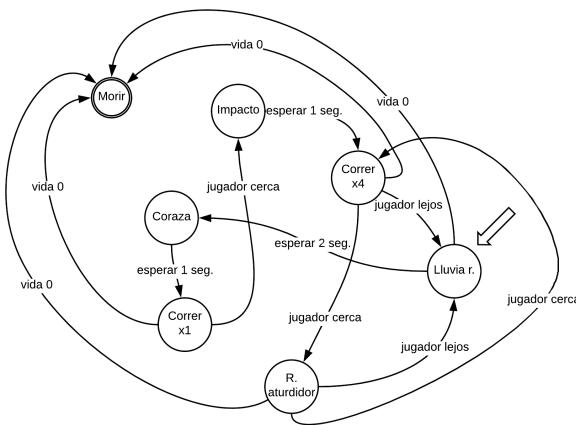
Dentro del desarrollo de videojuegos existen personajes no jugables, abreviados como NPC del inglés (no playable character) y para su funcionamiento se utilizan diferentes métodos o teorías. Dado que existía como riesgo acciones lineales o repetitivas para los jefes enemigos se decidió por emplear la teoría de autómatas. Para las acciones del jefe en los niveles tres, cinco, siete y nueve se utiliza un autómata finito. El estado final es "Morir", donde este representa la destrucción del objeto.

Jefe nivel 3

Dada las acciones descritas anteriormente en el documento de diseño, se tiene coraza, impacto, lluvia de rocas, rugido. Además se agregan otras acciones no contempladas para la ayuda de transiciones. La nomenclatura queda como sigue:

- Morir ->Mo
- Coraza ->Co
- Impacto ->Im
- Correr x4 ->Cx4
- Correr x1 ->Cx1
- Rugido ->RA
- Lluvia de rocas ->LR
- Vida 0 ->V0
- Esperar 2 seg. ->E2s

Figura 3.5: Máquina de estados que realiza el jefe del nivel 3



- Esperar 1 seg. ->E1s
- Jugador cerca ->Jc

Los estados son:

$$Q = \{M_o, C_o, I_m, C_x4, C_x1, R_A, L_R\}$$

El alfabeto son: $J_l, V_0, E_{2s}, E_{1s}, J_c$

El estado inicio es: $q_0 = L_R$

El estado final es: $F = M_o$

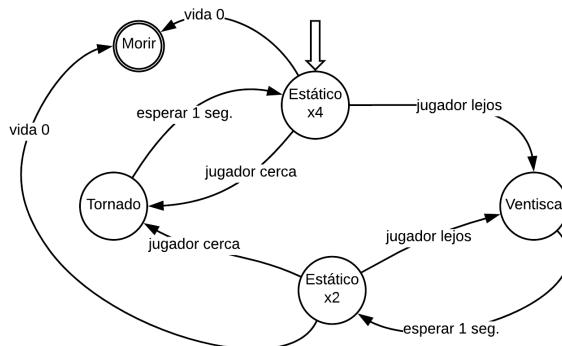
Como vemos en la imagen ?? queda representada la máquina de estados.

Jefe nivel 5

Dada las acciones descritas anteriormente en el documento de diseño, se tiene ventisca y tornado. Además se agregan otras acciones no contempladas para la ayuda de transiciones. La nomenclatura queda como sigue:

- Morir ->Mo
- Estático x4 ->Ex4
- Estático x2 ->Ex2
- Tornado ->To
- Ventisca ->Ve
- Vida 0 ->V0
- Jugador lejos ->Jl
- Esperar 1 seg. ->E1s

Figura 3.6: Máquina de estados que realiza el jefe del nivel 5



- Jugador cerca ->Jc

Los estados son: $Q = Mo, Ex4, Ex2, To, Ve$

El alfabeto son: $Jl, V0, E1s, Jc$ El estado inicio es: $q0 = Ex4$

El estado final es: $F = Mo$

Como vemos en la imagen ?? queda representada la máquina de estados.

Jefe nivel 7

Dada las acciones descritas anteriormente en el documento de diseño, se tiene lluvia de flechas, manotazo, lava. Además se agregan otras acciones no contempladas para la ayuda de transiciones. La nomenclatura queda como sigue:

- Morir ->Mo
- Estático x4 ->Ex4
- Estático x3 ->Ex3
- Estático x2 ->Ex2
- Manotazo ->Ma
- Lava ->La
- Lluvia flechas ->Lf
- Vida 0 ->V0
- Rand 0 ->R0
- Rand 1 ->R1
- Rand 2 ->R2

Los estados son: $Q = Mo, Ex4, Ex2, Ex3, Ma, La, Lf$

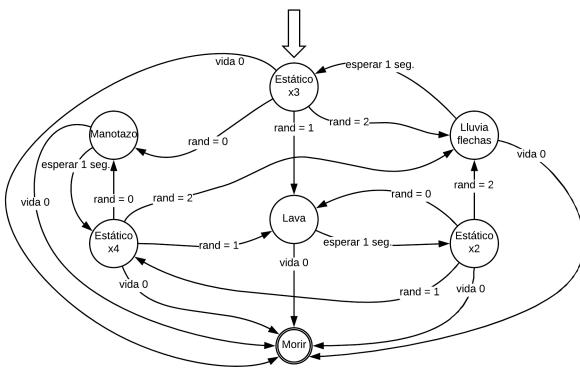
El alfabeto son: $V0, R0, R1, R2$

El estado inicio es: $q0 = Ex3$

El estado final es: $F = Mo$

Como vemos en la imagen ?? queda representada la máquina de estados.

Figura 3.7: Máquina de estados que realiza el jefe del nivel 7



Jefe nivel 9

Dada las acciones descritas anteriormente en el documento de diseño, se tiene estocada, filo y sablazo. Además se agregan otras acciones no contempladas para la ayuda de transiciones. La nomenclatura queda como sigue:

- Morir ->Mo
- Estocada ->Es
- Estático x4 ->Ex4
- Sablazo ->Sa
- Filo ->Fi
- Jugador lejos ->Jl
- Rand 1 ->R1
- Rand 0 ->R0
- Vida 0 ->V0
- Esperar 1 seg. ->E1s
- Jugador cerca ->Jc

Los estados son: $Q = Mo, Ex4, Es, Sa, Fi$

El alfabeto son: $Jl, Jc, E1s, R1, R0$

El estado inicio es: $q0 = Ex4$

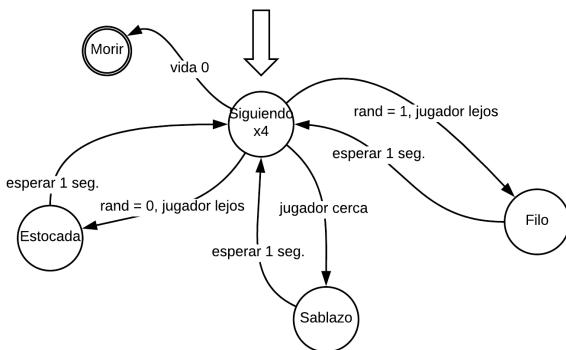
El estado final es: $F = Mo$

Como vemos en la imagen ?? queda representada la máquina de estados.

3.3. Contribuciones

En esta sección se presentan las soluciones a las observaciones realizadas por los sinodales durante la presentación del trabajo terminal 1.

Figura 3.8: Máquina de estados que realiza el jefe del nivel 9



3.3.1. El juego

El juego definida por la RAE es una actividad recreativa o de competición sometida a reglas por el entretenimiento. Sin embargo más que ello es parte fundamental para el desarrollo y aprendizaje de cualquier individuo. Esta actividad contribuye a la maduración, potencia cognitiva, desarrollo emocional, vehículo emocional que contribuye para aprender nuevas habilidades y conceptos a través de su propia experiencia.

El juego refleja la percepción de sí mismos, de otras personas y del mundo que nos rodea. Por ello mismo cuenta con 5 grandes ventajas según Pilar pedagoga y formadora **pilarjim\IeC {\'e}nez2015**

- El juego otorga placer y felicidad.
- En el juego no se tiene miedo al error.
- Fomenta la creatividad.
- Práctica de creación de estrategias y colaboración.
- El juego es el aprendizaje natural de las personas.

Teorías del aprendizaje

Es el estudio del aprendizaje que concierne al proceso por el que ocurre según características principales y ejes de análisis de las distintas teorías del aprendizaje. "orientaci\IeC {\'o}n\IeC {\'u}jar2015" Pues se necesita comprender algunas suposiciones generales de las teorías que sustentan el aprendizaje humano y de la forma en la que se construyen sus principios.

Las teorías más reconocidas sobre el aprendizaje son:

- Gestalt: reestructuración perceptual.
- Piaget: Constructivismo genético.
- Vygotsky: Teoría sociocultural.
- Ausbel: Teoría del aprendizaje significativo.
- Bruner: Teoría cognitiva.

Aquella más cercana al juego para el trabajo a realizar es la teoría de Bruner. Pues el jugador que contempla es epistémico social, inserto en una cultura y estructurado por un lenguaje. La inteligencia esta relacionada con 3 etapas de desarrollo para conocer: ejecución, impresión o imagen y significado simbólico. La evaluación está enfocada al estudio integral de los procesos cognoscitivos y los cambios que se originan.

Los videojuegos como medio de comunicación

Los videojuegos gracias a sus características de alcance masivo y presentación interactiva al usuario, son considerados parte de las TIC (tecnologías de la información y comunicación). Estos son más atractivos e influyentes dado que se enfoca a el ocio y entretenimiento de las personas. Dada las lecturas de artículos científicos sobre la relación de los jóvenes en las nuevas maneras de comunicarse entre sí **castellana2007adolescente** y las nuevas estrategias que han tomado los medios para adaptarse a ellos **ignasidebofarull2005** se comenta este apartado. Es así como podemos ver incluso a los videojuegos usados como publicidad, puede ser de manera implícita donde se muestre marcas o productos dentro de un escenario o situación del juego o explícita donde el mismo juego presenta a la marca mostrando sus cualidades y ventajas (en la mayoría de los casos de forma exagerada). Además podemos ahora combinar la expansión que nos da el internet junto con la diversión de un videojuego, posibilitando a los jugadores la capacidad de promover los productos que han probado y enseñarlo a los demás jugadores.

Serious games

Aquí podemos aprovechar la creación de un serious game, pues según Contreras **contreras2016investigacion** son los juegos digitales con una finalidad explícita para el aprendizaje más allá del entretenimiento sin ser pensados en la diversión. Tienen su interés en el desarrollo de las competencias, mediante actividades interactivas basadas en el juego según el artículo de una revista especializada en educación **romero2015serious** Contribuyen al desarrollo de la coordinación ojo-mano, agudeza visual, reacción, atención múltiple, aptitud relacional, motivación, tolerancia a la frustración, toma de riesgos, resolución de problemas y toma de decisiones, así como la reflexión estratégica, la creatividad, cooperación y sentido de innovación según Marqués **marquilec {\'e}s2012** Así mismo el jugador mejora el desempeño y se adentra a la experimentación, dada una situación simulada en la realidad virtual sin tener que enfrentar los riesgos de la realidad.

La gamificación y game-based learning son herramientas que persiguen el mismo objetivo de atraer y hacer practicar experiencias para memorizar y retener contenidos. Pueden usarse como ayuda para crear un serious game.

La gamificación es el uso de elementos de juego y técnicas de diseño para potenciar la motivación y compromiso de los jugadores. Mientras game-based learning se refiere al área cognitiva y apariencia donde debe crearse una experiencia de aprendizaje positiva.

En el siguiente cuadro **gabale** establecemos las diferencias más destacadas en ambas técnicas dadas en la Conferencia de Toronto por Perera **jorgepereragonzlez2016**

Cuadro 3.1: Diferencias entre gamificación y game-based learning

Gamificación

- Incluir los mecanismos de los juegos a situaciones de aprendizaje
- Existen como motivadores puntos de experiencia, logros e incentivos
- Enriquece la ambientación y simulación del aprendizaje

Game-based learning

- Usar los juegos para crear una experiencia
- La experiencia va dirigida al pensamiento
- Ambientación y simulación controladas

Figura 3.9: Rueda de motivos de Beatris Valderrama



Motivos para jugar

El área de interés para el desarrollo del trabajo es la gamificación, para ello la parte importante a conocer son los diferentes motivantes que tiene una persona al jugar.

Para determinar el perfil motivacional se tomará la rueda de motivos^{??} definida por Valderrama^{valde} donde se define motivos de aproximación; aquellas personas sociales y buscan la convivencia y motivos de evitación; aquellas personas que prefieren la seguridad y estancia individual.

Es así como tenemos en contra partes diferentes motivos dependiendo del jugador objetivo, que son la búsqueda de:

- Logros o hedonismo
- Exploración o seguridad
- Contribución o conservación
- Autonomía o afiliación
- Poder o cooperación

3.3.2. Modelo de negocios en un videojuego

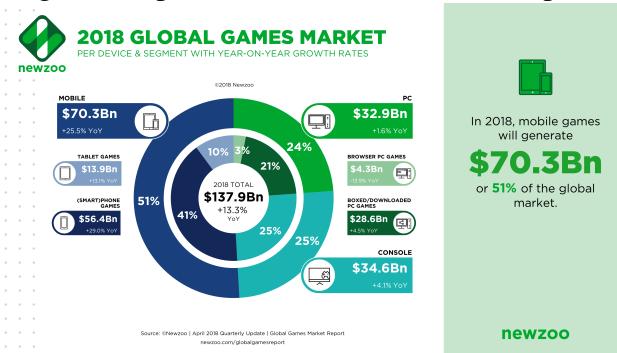
Para sustentar un proyecto o producto económico se debe tener claro un modelo de negocios. En el mundo de los videojuegos no existe la excepción, pero también debe considerarse que existen formas muy diferentes de adquirir el ingreso.

Incluso el mismo juego puede estar involucrado en un ingreso directo del servicio.

Mercado global

Se reporta segun Newzoo **newzoo2018** que 2.3 billones de jugadores en todo el mundo gastarán \$ 137.9 billones en juegos en 2018. Esto representa un aumento de + 13.3 % en comparación con

Figura 3.10: M rcado global al primer trimestre del a o 2018 por Newzoo newzoo2018



el a o anterior, o \$ 16.2 billones. Los ingresos por juegos digitales tomar n el 91 % del mercado global con \$ 125.3 mil millones, como podemos ver en la **fig:merglo**

Por primera vez, m s de la mitad de todos los ingresos del juego provendr n del segmento m vil como vemos en la imagenasdasd. Los tel fonos inteligentes representar n el 80 % de esto, o \$ 56.4 mil millones, con el 20 % restante proveniente de tabletas.

Salarios

Para realizar un videojuego se necesita de diferentes profesiones para llevarlo a cabo. En el siguiente cuadro se mostrar a la profesion y salario a recibir en la industria del videojuego en una empresa ya establecida al a o 2014 segun la encuesta con una relaci n definida en experiencia.

Cuadro 3.2: Tabla de salarios dados en una empresa formal de videojuegos

Rama	Profesi�n	Salario con <3a�os de exp.	Salario con 3-6a�os
Programadores e ingenieros	Programador	\$71,855 USD	\$79,877 USD
	Programador principal		\$94,877 USD
	Director t�cnico		
Artistas y animadores	Animador	\$50,000 USD	\$55,547USD
	Artista principal		\$71,029 USD
Dise�nadores de juego	Director de arte		
	Dise�nador de juego	\$53,000 USD	\$65,516 USD
	Director creativo		\$68,654 USD
Productores	Productor asociado		\$59,079 USD
	Lider de proyecto		\$73,500 USD
Profesional de audio	Productor ejecutivo		
	Director de sonido		
Testers	Tester		\$38,833 USD
	Lider de control de calidad		\$60,417 USD
	Marketing		\$73,500 USD
Negocios y administraci�n	CEO		
	Gerente ejecutivo		

Presentación al cliente

Un videojuego como cualquier software al momento de ser vendido al cliente puede encontrarse en dos presentaciones, una versión física o solo digital. En la siguiente tabla **fiDi** se muestran las diferencias más destacables dadas por experiencia empresarial en el desarrollo por Velneo **velneo2015**

Cuadro 3.3: Tabla comparativa de un producto físico o digital por Velneo **velneo2015**

	Físico	Digital
Coste de desarrollo	sí	sí
Coste de producción	sí	no
Coste de envío	sí	no
Riesgo de sobra/infra producir inventario	sí	no
Facturación por unidad vendida	mucho mayor	mucho menor
Unidad vendida costea soporte	generalmente sí	imposible
Porcentaje del precio de venta que percibe la empresa	30 %-40 %	70 %
Tiempo de cobro	90 días o más	30 días
Capacidad de llegar al público con marketing	caro	difícil

Aún así cabe mencionar que este es un aspecto general que involucra a cualquier software.

Formas de ingreso

Dentro de los videojuegos existen modelos de negocio que han ido cambiando a lo largo de los años y muchas de las veces depende del tipo del juego. Pero podemos definir las siguientes conforme lo visto y consumido en los últimos 5 años a la fecha del proyecto a presentar y con el apoyo de un artículo de la fundación UADE **fundaci\IeC {\v{o}}nuade2014**

Ingredientes de monetización

Los modelos de negocio anteriores pueden ser combinables con otros "ingredientes" de monetización para acrecentarlos ingresos.

- **Dinero virtual:** Es el medio de intercambio que utiliza un videojuego para poder formalizar las compras dentro de él. A menudo se suele diferenciar el virtual currency (dinero virtual que se consigue por las propias mecánicas del juego y con abundancia) y el hard currency (dinero virtual premium que se consigue con dinero real o con acciones muy concretas y con mucha escasez).
- **Bienes virtuales:** Son objetos intangibles que son comprados e intercambiados que sólo tienen sentido dentro del juego, muchas veces estos son comprados con dinero virtual.
- **Publicidad y patrocinio:** Anuncios o productos presentados en el juego para darse a conocer.
- **Bonificaciones y servicios virtuales:** Son aceleradores de juego o servicios que mejoran el desempeño o facilitan en el juego.
- **DLC (downloadable content):** Es un contenido de descarga digital exclusivo y adicional de un videojuego que se vende por separado y posterior al lanzamiento de este. Suele lanzarse

Cuadro 3.4: Tabla comparativa de ventajas y desventajas de modelos de negocios de videojuegos de autoría propia

Nombre Descripción

Pay-to-play Se debe pagar contenido y uso del videojuego

Free-to-play Ofrece gratis contenido y uso del videojuego en su totalidad, se monetiza con publicidad y con compras dentro del juego.

Freemium Ofrece gratis el uso del videojuego pero no se accede a todo su contenido, establece jerarquizaciones de pago.

Suscripción Se debe pagar el contenido y uso del videojuego pero con limitaciones.

Cuadro 3.5: Tabla de salarios dados en una empresa formal de videojuegos			
Rama	Profesión	Salario con <3años de exp.	Salario con 3-6años
Programadores e ingenieros	Programador	\$71,855 USD	\$79,877 USD
	Programador principal		\$94,877 USD
	Director técnico		
Artistas y animadores	Animador	\$50,000 USD	\$55,547 USD
	Artista principal		\$71,029 USD
	Director de arte		
Diseñadores de juego	Diseñador de juego	\$53,000 USD	\$65,516 USD
	Director creativo		\$68,654 USD
	Productor asociado		\$59,079 USD
Productores	Lider de proyecto		\$73,500 USD
	Productor ejecutivo		
	Director de sonido		
Profesional de audio	Tester		\$38,833 USD
	Lider de control de calidad		\$60,417 USD
	Marketing		\$73,500 USD
Negocios y administración	CEO		
	Gerente ejecutivo		

para alargar la longevidad del videojuego y para aprovechar su éxito comercial. Su adquisición no tiene sentido sin tener antes el videojuego ya que es un producto complementario y dependiente a él.

3.3.3. Costo de hacer un videojuego

Salarios

Para realizar un videojuego se necesita de diferentes profesiones para llevarlo a cabo. En el siguiente cuadro se mostrará la profesión y salario a recibir en la industria del videojuego en una empresa ya establecida al año 2014 segun la encuesta con una relación definida en experiencia.

3.3.4. Modelo de datos

La primera observación en atender fue el modelo de datos del juego, dicho modelo de datos se realizó utilizando un modelo entidad relación de base de datos (Ver Anexo ??) ya que al modelarse de esta forma hace escalable el juego si se deseará en algún futuro emplear una base de datos para mejorar el almacenamiento de datos y el manejo de más usuarios para ofrecer un modo online. El modelo de datos está basado en el modelo de clases y contiene únicamente a las clases actoras. Toda entidad actora se define como una especialización de una entidad base llamada GameObject, esta entidad está definida por como su identificador y por otras entidades como GameObjectPosition, Level, Tag, AnimationMachine, entre otros.

3.3.5. Estrategias para combatir la adicción entre los usuarios

La segunda observación sobre la que se trabajo fue como disminuir la adicción del jugador al videojuego Yolotl. Esta observación dio lugar a una investigación sobre la adicción a los video-

juegos ya que antes de proponer alguna solución se debía conocer cómo se definía, las causas y las consecuencias de la adicción al videojuego. Al final de la investigación se pudieron formular tres posibles soluciones para evitar la adicción del jugador; sin embargo, dado que este tópico no estaba en la planeación original del proyecto y por las implicaciones que conllevaban cada una de las soluciones se decidió únicamente describir las soluciones y sus implicaciones sin desarrollar ninguna de las tres. A continuación, se describen a manera de resumen las soluciones (nuevamente si se dese a profundizar en la investigación realizada y las soluciones se puede consultar el Anexo ??):

- **Notificación de confirmación para continuar la partida.** Esta solución propone que el juego solicite la confirmación del usuario para continuar una vez que éste ha detectado que el jugador ha estado jugando durante un tiempo prolongado como una hora.
- **Control paterno.** El juego le envía un formulario al tutor del jugador por medio de un correo electrónico. En este formulario el tutor podrá decidir cuánto tiempo al día la aplicación podrá estar abierta.
- **Sistema de vidas.** El jugador tiene una cantidad de vidas limitadas. Cada vez que el jugador ingresa a un nivel o muere dentro de uno y reinicia la partida se gasta una vida. Para recuperar vidas el jugador deberá de esperar un determinado tiempo.

3.4. Trabajo realizado durante trabajo terminal 1

En esta sección se habla a manera de resumen el trabajo realizado durante el periodo correspondiente a trabajo terminal 1. La división de esta sección queda organizada en dos subsecciones: una para la etapa de preproducción y otra para los dos primeros *sprints* de la etapa de producción.

3.4.1. Presentado en TT1

A continuación se muestra lo presentado en TT1, que representa la investigación, análisis y prototipos del proyecto.

Contexto

Primero nos encontramos a determinar qué es la cultura; aquella que define la identidad de un individuo por sus creencias religiosas, de pensamiento, sentimentales y sociales. Mientras que la historia son todos los sucesos pasados dentro de un espacio específico. Entonces la cultura histórica es aquellos aspectos arquitectónicos, de pensamiento, éticas y morales, grupos de pertenencia y convivencia. Dentro de la cultura histórica encontramos que el 41 % de los mexicanos no asisten a eventos culturales y presentan una gran desinformación de ella.

Luego nos encontramos que existen formas de presentar cualquier tema o actividad dentro de los juegos. A partir de eso investigamos sobre los videojuegos, juegos que se presentan a través de un medio visual o auditivo en el que existe interacción por diferentes dispositivos de entrada. Dentro de los videojuegos hay clasificaciones por contenido, que son las utilizadas para determinar el tipo de público al que va dirigido, además de que es una clasificación conveniente para limitar compra y venta. Sin embargo existen muchas más clasificaciones, en donde el tipo de contenido, dispositivo u objetivo es su pertenencia.

Viabilidad

Con esta información ya en mano, buscamos sobre la situación actual de comercio de los videojuegos. Pasando primero por los ingresos a nivel mundial y en que dispositivos en el año 2017; a nivel mundial se tiene un ingreso de \$108.9 mil millones de dólares, el 42 % está en los dispositivos móviles, tanto teléfonos inteligentes como tabletas, el resto queda en la computadora y consolas. Así México queda en el doceavo lugar de consumo de videojuegos a nivel mundial, con un ingreso percibido de \$1.4 mil millones de dólares en una población de 130 millones de mexicanos. De la población en México el 20 % juega videojuegos, de ese porcentaje 45 % juega en el celular y 40 % juega de una a dos veces por semana.

Análisis

Una vez contemplada la información, se decide que el proyecto se haga en un dispositivo móvil android, dado que el 86 % de los usuarios de dispositivos móviles tienen android y como versión mínima 5.0 lolipop con su uso de 32 % de las personas.

La metodología a usar será Hundle, que consiste en un parecido a la metología scrum solo que está adaptada a la creación y desarrollo de videojuegos, donde se establece un proceso general de preproducción, producción y postmortem. Destacando aquí que se hicieron algunos ajustes dentro de la parte de preproducción agregando y modificando apartados dado que eran necesarios aclarar antes de empezar con el proyecto.

Luego se pasó a definir las herramientas de desarrollo, donde se escogió debido a su flexibilidad multiplataforma el motor de juego Unity y como herramientas de dibujo corel draw x5 y photoshop.

Como arquitectura a usar se eligió modelo vista controlador, donde se dividirá el código en controladores, actores y auxiliares, estos últimos ayudarán a los controladores y actores en situaciones específicas donde no se encuentre en ninguna de estas características.

Progresión y prototipos

Ya realizadas las investigaciones y análisis previo para el proyecto, se estableció la progresión que iba a tener el juego, junto con definir las interfaces que contendría y su interacción entre ellas. Se estableció la mecánica del juego donde se definía los botones de acción del personaje, el espacio de características del personaje y un apartado para determinar otros factores como items o vida del enemigo.

Ya en los prototipos se pasó a el maquetado de los niveles uno y dos, determinando personajes y eventos, así como prototipos de uso de la herramienta Unity. Al final dando como resultado dos prototipos conteniendo el nivel uno y dos y un prototipo de uso de la herramienta.

3.4.2. Etapa de Preproducción

Esta etapa corresponde a la planeación análisis y diseño del juego. Como lo indica la metodología *Huddle*, para esta etapa se trabaja en el desarrollo del documento de diseño del juego. Esta etapa queda del desarrollo queda dividida en cuatro *sprints*.

Primer Sprint Huddle de Preproducción

Antes de iniciar el diseño del juego se realiza un trabajo de investigación sobre la cultura azteca. Esta investigación abarca:

- **La sociedad mexica:** su historia tradiciones y clases sociales.
- **Mitología mexica:** Dioses, mito de los cinco soles, mito de la creación del hombre del maíz, el Mictlán.
- **Historia de la Malinche:** Historia del personaje antes y después de la llegada de los españoles.

Durante la etapa de investigación se selecciona la información histórica que sera relevante y útil para la narrativa del juego y el diseño de su jugabilidad. Para la investigación histórica de esta etapa se consultan libros, códices, páginas de Internet, artículos de investigación e incluso se visitan museos como el templo mayor.

Segundo Sprint Huddle de Preproducción

En este *sprint* se redactan las primeras secciones del documento de diseño del juego *Yolotl*. Se inicia con la idea concepto y con el tema del juego. De igual forma se selecciona un nombre para el juego a desarrollar: *Yolotl*. Para algunos juegos la mecánica es la primera es ser definida; no obstante, por la naturaleza del juego como herramienta de transmisión de cultura, *Yolotl* nace con su historia. La historia de *Yolotl* pasa por diferentes etapas de diseño; siendo modificada gradualmente, pero manteniendo algunos elementos clave como la lucha contra la divinidad.

En la etapa del concepto también se define la plataforma para la que será el juego: dispositivos móviles con sistema operativo Android 5.2. Por su parte se decide utilizar un motor de juego como herramienta de desarrollo, pues esto permite centrarse en el diseño e implementación de aquellos elementos que diferencien a *Yolotl* del resto de juegos, tal como su mecánica, sus personajes, etc. Luego de investigar sobre los motores de juegos disponibles, se elige Unity 3D como ambiente de desarrollo.

Una vez teniendo la idea concepto se define la visión del juego y sus mecánicas. En cuestión de las mecánicas el enfoque por el que se opta es el de mantener el juego con mecánicas simples y familiares para aquellos jugadores que ya habían tenido alguna experiencia con algún juego de plataformas, sin descartar algunos detalles que le dieran identidad al juego en cuanto a su jugabilidad. Paralelamente a la preproducción, se inicia el desarrollo de un primer demo con el fin de familiarizarse con la herramienta de Unity3D, este demo incluye las mecánicas más simples del juego.

Con la historia, la visión y la mecánica definidas se procede a puntualizar los estados del juego, diseñar las interfaces gráficas de navegación y de interacción con el personaje. Para ver la versión final de las interfaces se puede consultar anexo ??.

Tercer Sprint Huddle de Preproducción

En el tercer Sprint se definen la cantidad de niveles y en que consiste cada uno, de igual forma se establecen los objetivos de cada nivel, la recompensa a obtener una vez completado el mismo, los enemigos a vencer y las cinemáticas que fungen como transiciones entre niveles.

Al mismo tiempo que se diseñan los niveles, se detallan los personajes tanto a nivel narrativo como a nivel de jugabilidad, definiendo habilidades para los enemigos, los niveles en los que

parecerían y sus acciones dentro de la historia. Para esta parte se trata de obtener la mayor fidelidad posible a los mitos y códices. En el anexo ?? se habla a mayor detalle sobre el diseño de los personajes.

Cuarto Sprint Huddle de Preproducción

En el cuarto sprint se termina de escribir el argumento del juego, de esta etapa se obtiene el guión literario del juego. En este *sprint* también se definen elementos de ambientación para el juego tales como la música de fondo, los efectos de sonido y los efectos especiales.

De igual forma, en este *sprint* se especifican las armas de los personajes, los ítems; quedando diseñados tanto a nivel de comportamiento como a nivel visual. Al igual que con los personajes se busca que las armas, tanto en comportamiento como en diseño, se mantengan lo más fiel posible a los mitos y leyendas de donde se basaron.

Con el cuarto *sprint* se finaliza la etapa de preproducción, obteniendo así un documento de diseño lo suficientemente detallado como para iniciar el diseño del juego a nivel de ingeniería.

3.4.3. Etapa de producción

En esta sección se habla del trabajo realizado durante los dos primeros *sprints* de esta etapa, ya que fueron desarrollados durante los meses correspondientes al trabajo terminal 1. Todos los *sprints* de la etapa de producción posteriores al segundo *sprint* son abordados en la sección ??.

Primer Sprint Huddle de Producción.

En este *sprint* se realiza un análisis del documento de diseño, en consecuencia de este análisis se diseña el videojuego en materia de las clases que lo componen y el modelo bajo el que funcionaría el juego a nivel de programación.

Haciendo uso del paradigma orientado a objetos se propone emplear tres tipos de clases:

- **Actores:** Son las clases que modelan a los enemigos, los ítems, los colecionables, los check-points y al jugador.
- **Controladores:** Son las clases encargadas de gestionar la partida y la navegación entre interfaces. Estas clases desencadenan eventos conforme a las acciones de las clases actoras. Estas clases también son las encargadas de verificar que se cumplan las reglas de los niveles.
- **Auxiliares:** Estas clases ayudan al funcionamiento de los actores y los controladores. Estas clases también se encargan de vincular datos con las clases controladoras como efectos de sonido, música, datos para la progresión entre niveles.

El modelo planteado permite reutilizar parte del demo generado durante la etapa de preproducción. Por lo que en este *sprint* se inicia la integración del código del primer demo con el comportamiento modelado por las clases definidas en el párrafo anterior.

En el primer *sprint* de Producción también se crean los *sprites* del primer nivel utilizando la herramienta de modelado en *3D Blender*. En la figura ?? se pueden observar algunos de los modelos

creados. Al finalizar este *sprint* se determina la no viabilidad del modelado en 3D de los *sprites* por cuestiones de tiempos; en consecuencia, se descarta este método para generar los *sprites* y se inicia el desarrollo de los *sprites* a partir de otras técnicas de animación más tradicionales.

Segundo Sprint Huddle de Producción.

En este *sprint* se inicia el desarrollo de los *sprites* con *Adobe Photoshop* y *Corel Draw*. A la par se inicia la maquetación de la etapa de selva del nivel uno. En este sprint se logran terminar todos los *sprites* referentes al primer nivel del juego tales como:

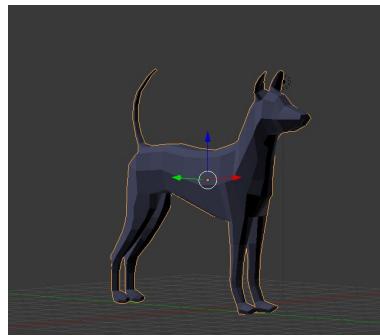
- Objetos de fondo: Arbustos, árboles, jarrones y cajas.
- Imagen de fondo: Fondo de la selva, la ciudad y el menú principal.
- Ciudadanos del mercado: Comerciantes, nobles y esclavos.
- *Xólotl* en su forma *xoloitzcuintle*: Bloques de animacion para correr y normal.
- *Malinalli* sin la caracola: Bloques de animación correr, saltar y normal.

Una vez terminados los *sprites* referentes al nivel uno estos se integran al código permitiendo tener un segundo prototipo con la siguiente funcionalidad:

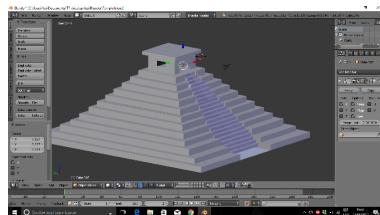
- Control de personaje por medio de la GUI.
- Transiciones entre interfaces.
- Personaje seguible que aparece en el primer nivel funcional.
- Funcionamiento básico del controlador de diálogos.



(a) Modelo de *Malinalli* generado en *Blender*.



(b) Modelo de *Xólotl* generado en *Blender*.



(c) Modelo de un templo generado en *Blender*.



(d) Modelo de una mujer comer-
ciante generado en *Blender*.

Figura 3.11: Modelos de personajes y objetos crados en *Blender* (Autoria propia).

Capítulo 4

Resultados obtenidos

En este capítulo se habla de los resultados obtenidos durante trabajo terminal 2. Por tal motivo en este capítulo se abordan las pruebas realizadas y las características que tienen los niveles para ser considerados como acabados.

4.1. Pruebas

En esta sección se reportan todos los tipos de pruebas a los que se sometió el juego para probar tanto su funcionalidad como su desempeño y el impacto que tuvo en los jugadores.

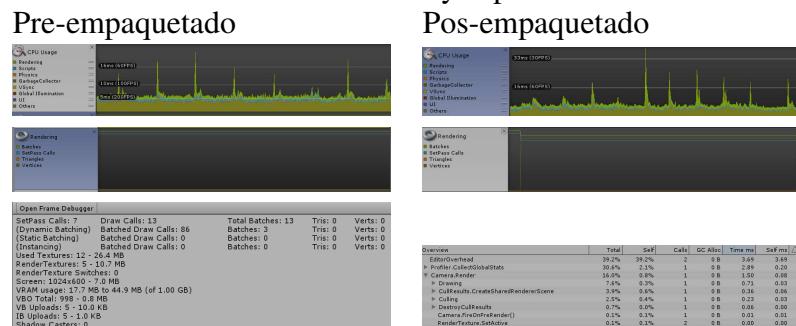
4.1.1. Sprite packer

El objetivo de esta prueba es la optimización de carga de imágenes al crear paquetes de sprites, en unity conocido como sprite packer. Se usarán pruebas automáticas con la herramienta profiles de Unity.

Una parte significativa de una textura de sprite a menudo será ocupado por el espacio vacío entre los elementos gráficos y este espacio va a resultar en memoria de video gastada en el tiempo de ejecución. Para un rendimiento óptimo, lo mejor es empacar gráficas de varias texturas de sprite muy juntas dentro de una misma textura conocida como un atlas. Unity proporciona una utilidad Sprite Packer para automatizar el proceso de generar atlases de las texturas de sprite individuales.

Al final podemos determinar que si existe un menor porcentaje de carga en imágenes debido al empaquetado. En este caso el empaquetado fue en sprites estáticos, una mayor percepción de carga menor se puede observar al tener un empaquetado de animaciones, recordando que un movimiento

Cuadro 4.1: My caption



en una imagen puede contener un número determinado de cuadros por segundo. Se deduce que si ha habido una optimización.

4.1.2. Mecánicas del juego

En esta parte se incluirán las pruebas de los componentes jugables que forman parte del juego. Como se desarrollan e interactúan a lo largo del juego.

Salto del jugador

El objetivo de la prueba consiste en verificar la condición inicial del salto, en donde debe realizarse un doble salto. Esta es una prueba funcional del juego, la herramienta a utilizar es el mismo programa de Unity, dentro del computador. Al aplicar la prueba se detecta un salto ilimitado de veces en el jugador, debido a que el valor de verdadero o falso para permitir un salto extra no está siendo detectado por el método de salto. Se hace un reporte final del fallo de salto extra.

Plataforma en movimiento

El objetivo de la prueba es comprobar que funcione la plataforma con movimiento tanto horizontal como vertical como se ha determinado. La herramienta a utilizar sigue siendo dentro del computador Unity. Como prueba dentro del funcionamiento del juego el jugador se posiciona sobre la plataforma, la plataforma se desplaza de forma adecuada y en la dirección establecida, tanto el movimiento horizontal como el movimiento vertical. Pero se detecta que el desplazamiento es independiente al personaje, por lo que el jugador debe manualmente seguir la trayectoria de la plataforma, en este caso particular, la plataforma horizontal. Mientras que en el caso de la plataforma vertical se detecta una disminución de velocidad al realizar una ascensión y un aumento de velocidad al descender. Se concluye que los componentes funcionan de manera independiente en cuanto a su movimiento, la solución a proponer consistiría en un código por parte de la plataforma en movimiento que cuando detecte al jugador le aplique la misma velocidad que este realice sumándosela a la que pudiera el jugador tener en ese momento y esto sería solo al contacto de la plataforma.

4.1.3. Dinámicas del juego

En esta parte las pruebas van dirigidas a las situaciones que crea el jugador con lo que se le ha permitido dentro del videojuego.

Salto bajo

El objetivo de la prueba consiste en entender la percepción del usuario ante las acciones que realiza el personaje dentro del juego. La herramienta a utilizar es en el computador en el programa Unity. El jugador se presenta en diferentes escenarios de los niveles para que pueda moverse a voluntad propia dentro de lo que establece el juego. Las situaciones a las que se presenta el jugador abarcan saltar en plataformas estáticas, plataformas móviles, camino horizontal y camino ascendente. Al final el usuario reporta que en su propia percepción el salto o elevación del personaje es menor al que le gustaría o en algunas ocasiones al que necesita. Las soluciones a proponer consisten en aumentar la variable que controla el movimiento ascendente del jugador o la variable que controla la gravedad proporcionada en el juego.

Instrucciones del juego

El objetivo de la prueba consiste en entender la percepción del usuario ante las acciones que realiza dentro del nivel introductorio del juego. La herramienta a utilizar es en el computador en el programa Unity. Al usuario se le presenta el nivel introductorio del juego para que termine con este mismo. Al final reporta que por prueba y error ha sabido el funcionamiento de los botones que se le presenta, según a opinión del usuario le gustaría dentro del juego una demostración o explicación de los mismos botones y lo que hacen. Al final se propone como solución dentro de los mismos diálogos explicar el funcionamiento de los botones, o el realizar un video explicativo de los botones o indicarle al usuario que realice actividades determinadas para la prueba de los botones.

Desbloqueo de personas

El objetivo de la prueba consiste en entender la percepción del usuario ante las acciones que realiza dentro del nivel introductorio del juego. La herramienta a utilizar es en el computador en el programa Unity. Al usuario se le presenta como situación seguir avanzando al siguiente nivel dentro del nivel introductorio, donde en una parte el jugador debe desbloquear al menos cinco diálogos. El jugador reporta que aunque sea el mismo diálogo que active una y otra vez, este se contabiliza sin ningún problema. La solución a proponer es colocar una variable de verdadero y falso para indicar si el diálogo ha sido activado y así no contabilizarlo nuevamente.

4.1.4. Estética del juego

En esta parte las pruebas van dirigidas a la respuesta del jugador ante lo que se le presenta directamente como interacción dentro del juego, ya sea auditiva o visualmente.

Ilustraciones de items

El objetivo de la prueba consiste en entender la percepción del usuario ante lo que se le muestra a lo largo del juego. La herramienta a utilizar es en el computador en el programa Unity. Dentro de los niveles impares el jugador reporta un poco confuso a primera vista el item que incrementa el tonalli, al no reconocer qué objeto lo representaba. El objeto a querer mostrar es una flor de vainilla, al mostrar la imagen por separado al jugador y presentarlo en un tamaño más grande el usuario reporta como confuso las ramas u hojas de la imagen a lado de la flor. Como propuesta de solución se tiene la eliminación de las ramas u hojas de la imagen, dejando solo la flor visible o hacer mucho más grande la imagen presente en el juego.

Menú principal

El objetivo de la prueba consiste en entender la percepción del usuario ante lo que se le muestra en el menú principal. La herramienta a utilizar es en el computador en el programa Unity.

4.1.5. Prueba unitaria

Las primera prueba unitaria fue sobre los actores. En esta sección se describe cómo se realiza la prueba y cómo se solucionan los errores encontrados a partir de ella.

Objetivo de la prueba

Verificar el funcionamiento lógico de los componentes del nivel y definir los valores a algunos atributos para el funcionamiento correcto de algunos actores.

Herramientas utilizadas durante la prueba

Unity.

Aplicación de la prueba

Para esta prueba se evaluó el comportamiento de los actores antes de su integración a los niveles y como estos interactúan con el jugador. Unity permite ver los valores que adquieren los atributos durante su ejecución, ver figura ???. Así que para esta prueba basta con ejecutar la escena base y observar como responden los actores.

Primeramente se revisa que los enemigos y las plataformas sigan sus patrones de movimiento definidos. Después se verifica que los enemigos, obstáculos e ítems afecten la cantidad de vida del jugador y en algunos casos su cantidad de *tonalli*. En el caso de los jefes se verifica que sus ataques no se vean interrumpidos por nuevos ataques de la máquina de estados. En el caso particular de obstáculos como *WindCreator* se definen los intervalos de tiempo para su correcto funcionamiento.

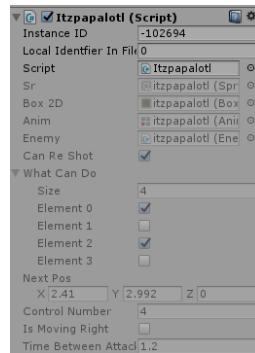


Figura 4.1: Unity permite ver los valores de los tributos de las clases en ejecución.

Conclusiones de la prueba

En esta prueba se observan diferentes problemas en el comportamiento de los actores que se solucionan, a continuación se mencionan los errores encontrados y como se solucionaron:

- El marcador se actualiza al doble cuando el jugador cae sobre un objeto colecciónable: Este error resulta producto de utilizar un *GameObject* auxiliar para la detección de las colisiones del suelo. El error se soluciona fácilmente al agregar un componente de tipo *rigidbody 2D* al *GameObject* auxiliar para la detección de las colisiones del suelo.
- Los ítems restauran el doble de vida cuando el jugador cae sobre ellos: Este error es generado por las mismas causas que el de los objetos colecciónables así que al solucionar el de los objetos colecciónables se soluciona éste.

- Los ataques de los jefes generados por corrutinas se empalman con otros ataques o interrumpen los que ya se están ejecutando: Esto se soluciona al detener todas las corrutinas generadas por el jefe cuando se ejecuta un ataque.

4.1.6. Prueba de integración

Esta prueba se realiza una vez se integraron los actores y controladores a los niveles.

Objetivo de la prueba

Verificar el funcionamiento lógico de los componentes del nivel al ser integrados para formar un nivel entero.

Herramientas utilizadas durante la prueba

Unity.

Aplicación de la prueba

Para realizar esta prueba es necesario jugar los niveles para observar que el comportamiento de los controladores y los actores se ejecutan correctamente al integrarse con otros actores. En esta prueba también se ajustan las áreas activas de las plataformas a fin de que su funcionamiento no se detenga si se alejan mucho del jugador al realizar su recorrido.

Conclusiones de la prueba

Al finalizar esta prueba se pudo verificar que los controladores funcionan de manera correcta; sin embargo, es necesario realizar ajustes referentes a los tiempos de transiciones entre escenas y los valores de las áreas activas de varias plataformas y obstáculos ya que con sus valores iniciales algunas plataformas se detenían al realizar su recorrido dado que el jugador se salía de su área activa y se volvía inalcanzable. En cuanto al obstáculo de *WindCreator* se ajustó el tamaño del área activa garantizando que el obstáculo se encuentre activo cuando el jugador llegue a donde se encuentra éste.

4.1.7. Prueba de sistema

Esta prueba se realiza una vez se integraron los actores y controladores a los niveles.

Objetivo de la prueba

Verificar el flujo de la navegación del juego.

Herramientas utilizadas durante la prueba

Unity.

Aplicación de la prueba

Esta prueba inicia desde la escena de menú principal en donde se verifica que el controlador del menú realiza las validaciones correspondientes antes de crear o cargar una partida; de igual forma de verifica que aparezcan los mensajes de confirmación a cada caso, sea el de confirmación de la nueva partida o el que notifica que no hay datos previamente guardados.

La siguiente escena a probar es el menú de selección de nivel. En este se verifica que la información mostrada por la interfaz corresponda al nivel que se desea acceder. Después, se verifica que en efecto el juego no permita acceder a niveles que aun no se desbloquean.

Para finalizar la prueba se verifica que se realicen las transiciones entre niveles y cinemáticas. De igual forma se prueba la funcionalidad de botones de navegación de los niveles referentes al panel de pausa, fin de partida y nivel completado.

Conclusiones de la prueba

Al finalizar esta prueba se pudo confirmar que las transiciones entre escenas se realiza de manera correcta; salvo en algunos casos pero fue debido a que el nombre de la escena a la que se debería redirigir no estaba escrito correctamente o no coincidía con el nombre de la escena a la que debía ir. Con esto se puede concluir que se cumple el mapa de navegación que se propuso en el documento de diseño realizado en trabajo terminal 1.

4.1.8. Prueba de rendimiento

Esta prueba se realiza una vez que hechas las modificaciones como producto de las pruebas unitarias, de integración y de sistema.

Objetivo de la prueba

Verificar el uso del GPU.

Herramientas utilizadas durante la prueba

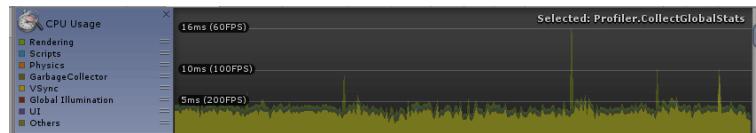
Profiler de Unity.

Aplicación de la prueba

Esta prueba inicia desde el menú principal y con la herramienta *profiler* se observa el desempeño del GPU de la maquina al simular el juego. La ventaja de utilizar *Profiler* es que indica que elementos de la escena son los que están consumiendo un determinado porcentaje del GPU. En las figuras ??, ?? y ?? se muestran los resultados de la herramienta *Profiler*.

Conclusiones de la prueba

Al observar el desglose del uso del GPU en las diferentes escenas que se probaron, se identifica al *EditorOverHead* como uno de los principales consumidores de recursos; investigando en la documentación de *Unity*, se detecta que este elemento es producto de un error de rendimiento en la versión 2017 pero que se puede solucionar al descargar uno de los parches que *Unity* proporciona



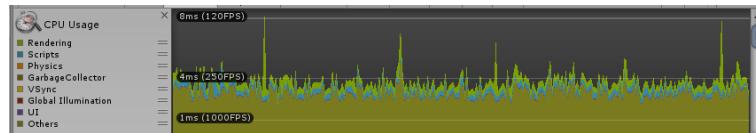
(a) Vista general del uso del GPU.

	Total	Self	Calls	GC Alloc	Time ms	Self ms	△
Profiler.CollectGlobalStats	70.1%	0.7%	1	0 B	11.78	0.12	
Profiler.CollectMemoryAllocationStats	59.3%	59.3%	1	0 B	9.97	9.97	
Profiler.CollectAudioStats	9.8%	9.8%	1	0 B	1.65	1.65	
Profiler.CollectDrawStats	0.1%	0.1%	1	0 B	0.02	0.02	
EditorOverhead	23.1%	23.1%	2	0 B	3.88	3.88	
Camera.Render	2.3%	0.3%	1	0 B	0.38	0.06	
PostLateUpdate.UpdateAudio	0.4%	0.0%	1	0 B	0.07	0.00	

(b) Desglose de los porcentajes del uso del GPU

```
Textures: 2186 / 151.5 MB
Meshes: 205 / 1.0 MB
Materials: 45 / 60.0 KB
AnimationClips: 29 / 148.0 KB
AudioClips: 0 / 234.0 KB
Assets: 3383
GameObjects in Scene: 54
Total Objects in Scene: 523
Total Object Count: 3906
GC Allocations per Frame: 150 / 5.0 KB
```

(c) Desglose de los porcentajes del uso de memoria

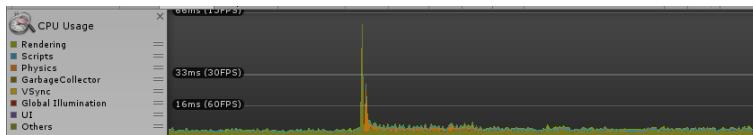
Figura 4.2: Resultados de la herramienta *profiler* al analizar el menú de selección.

(a) Vista general del uso del GPU.

```
Textures: 2191 / 118.1 MB
Meshes: 32 / 384.0 KB
Materials: 45 / 60.0 KB
AnimationClips: 29 / 148.0 KB
AudioClips: 0 / 0 B
Assets: 3444
GameObjects in Scene: 80
Total Objects in Scene: 454
Total Object Count: 3898
GC Allocations per Frame: 150 / 5.0 KB
```

(b) Desglose de los porcentajes del uso de memoria

Figura 4.3: Resultados de la herramienta *profiler* al analizar una cinemática.



(a) Vista general del uso del GPU.

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
EditorOverhead	37.8%	37.8%	2	0 B	4.05	4.05
Profiler.CollectGlobalStats	34.6%	1.4%	1	0 B	3.71	0.15
Profiler.CollectAudioStats	22.8%	22.8%	1	0 B	2.45	2.45
Profiler.CollectMemoryAllocationStats	9.6%	9.6%	1	0 B	1.03	1.03
Profiler.CollectDrawStats	0.6%	0.6%	1	0 B	0.07	0.07
GUI.Repaint	6.0%	1.5%	1	1.1 KB	0.64	0.17
Camera.Render	5.5%	0.7%	1	0 B	0.59	0.07

(b) Desglose de los porcentajes del uso del GPU

```
Used Total: 202.9 MB  Unity: 95.5 MB  Mono: 18.9 MB  GfxDriver: 90.3 MB  FMOD: 1.6 MB  Video: 0 B  Profiler: 17.1 MB
Reserved Total: 377.7 MB  Unity: 263.4 MB  Mono: 25.1 MB  GfxDriver: 90.3 MB  FMOD: 1.6 MB  Video: 0 B  Profiler: 24.0 MB
Total System Memory Usage: 0.93 GB
```

```
Textures: 2204 / 117.7 MB
Meshes: 158 / 1.0 MB
Materials: 48 / 63.0 KB
AnimationClips: 29 / 148.0 KB
AudioClips: 8 / 217.0 KB
Assets: 3632
GameObjects in Scene: 222
```

(c) Desglose de los porcentajes del uso de memoria

Figura 4.4: Resultados de la herramienta *profiler* al analizar un nivel.

desde su sitio web.

Con las pruebas de *profiler* se puede concluir que el juego tiene un buen rendimiento en cuanto a uso de recursos puesto que no presenta caídas dramáticas en cuanto a desempeño.

4.1.9. Prueba de rendimiento

Esta prueba se realiza una vez que hechas las modificaciones como producto de las pruebas unitarias, de integración y de sistema.

Objetivo de la prueba

Verificar el uso del GPU y el nivel de batería del teléfono que utiliza mientras la aplicación este funcionando.

Herramientas utilizadas durante la prueba

Opciones de desarrollador del teléfono Huawei TAG-L13 y *Battery Doctor*.

Aplicación de la prueba

Para esta prueba se debe de instalar la apk del juego en el dispositivo, activar las opciones de desarrollador e instalar la aplicación *Battery Doctor*. Una vez hecho esto se juega el juego y se mide el desempeño desde el teléfono. En las figura ?? se muestra el uso del GPU en distintos momentos de la partida. Por otra parte en la figura ?? se muestra el uso de la batería y el uso promedio de GPU que mide *Battery Doctor*.



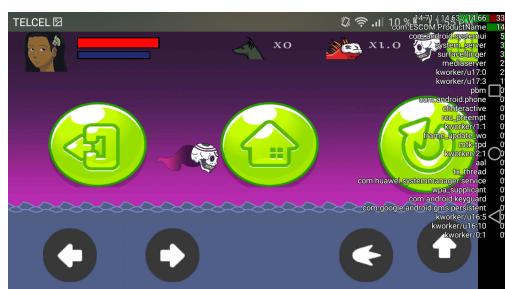
(a) Uso del GPU desde el menú principal.



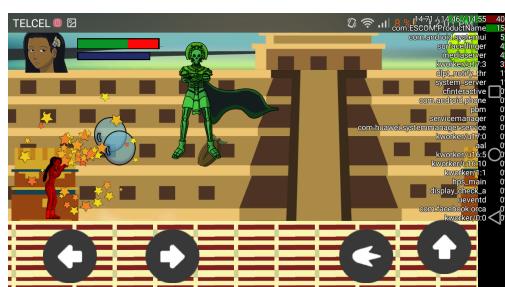
(b) Uso del GPU desde el menú de selección de nivel.



(c) Uso del GPU desde una cinematográfica.



(d) Uso del GPU desde un nivel de plataforma.



(e) Uso del GPU desde un nivel de jefe.

Figura 4.5: Resultados del rendimiento del GPU del dispositivo Huawei



Figura 4.6: Pantalla de la aplicación *Battery Doctor* para medir el rendimiento del juego.

Conclusiones de la prueba

De esta medición del desempeño del GPU se puede observar que la aplicación utiliza un mínimo del 9 % del GPU y hasta un máximo del casi el 30 %. Por su parte, el juego utiliza en promedio un 40 % de la batería. Estas cifras son buenas si se considera que otras aplicaciones como *Messenger de Facebook* llega a utilizar el 50.6 % del GPU y casi el 60 % de la batería del teléfono, ver figura ??.



Figura 4.7: Pantalla de la aplicación *Battery Doctor* para medir el rendimiento de *Messenger de Facebook*.

4.1.10. Prueba de Disfrute

Esta prueba esta basada en modelo de pruebas *Game Flow* para medir el disfrute de un juego con base en estrategias heurísticas de usabilidad y experiencia de usuario. El modelo de *Game Flow* permite a su vez, evaluar el diseño de interfaces, las mecánicas y la jugabilidad **gameflow**

Objetivo de la prueba

Obtener la opinión de los usuarios sobre los elementos de un nivel, tales como la mecánica, la jugabilidad, las interfaces, los enemigos, etc.

Herramientas utilizadas durante la prueba

Apk del juego, cuestionario(ver anexo ??) y encuesta en *Google Docs*.

Aplicación de la prueba

Para esta prueba se requieren grupos de personas para probar los niveles del juego. Dado que se buscaba evaluar las impresiones del jugador por nivel, se le pide a los encuestados que llenar la encuesta por nivel terminado. Para realizar la prueba se le proporciona al jugador el link para descargar la apk del juego y el link de la encuesta. Esta prueba esta diseñada para ser la más larga, ya que se busca que el mayor numero de personas puedan probar el juego. Esta prueba se realiza de dos maneras diferentes:

- Publicando los links de la apk y de la encuesta en redes sociales, indicando las intrucciones de responder la encuesta por nivel terminado.
- Realizando pruebas presenciales a grupos de personas.

En el caso de las pruebas presenciales, ademas de la encuesta se puede observar las reacciones reales de lo jugadores mientras prueban el juego. En muchos casos se pudo observar a diferentes grupos de amigos compitiendo por acabar el nivel, jugadores gritando de alegría al acabar un nivel que les había costado mucho esfuerzo o exclamaciones llenas de emoción al ser derrotados de ultimo momento por un jefe (ver figura ??).

Conclusiones de la prueba

A continuación se presentan algunos de los resultados de la encuesta y las conclusiones que se obtuvieron de estos:

- La principal marca de dispositivos con el que fue probado el juego fue Motorola con sistema operativo Android 7.
- La mayoría de los usuarios consideran como bueno el movimiento del personaje, pero consideran que haciendo más estable el salto el control del personaje mejoraría.
- La mayoría de los usuarios consideran que la respuesta de la *GUI* es buena; sin embargo, recomiendan mejorar el tiempo de respuesta de ésta y agregar una animación que indique que un botón ha sido oprimido.
- La mayoría de los usuarios opina que la actualización de la barra de tonali es buena pero les gustaría que existiera un indicador numérico para ver la cantidad de disparos que les queda.
- La mayoría de los usuarios considera que lo hace débil a un personaje es su patrón de movimiento y no la cantidad de daño que pueda generar; por otro lado también la mayoría de los usuarios considera que lo que hace a un enemigo fuerte es su patrón de movimiento.



(a) Dos alumnos de la Escuela Superior de Computo probando el juego.



(b) Grupo de alumnos de la Escuela Superior de Computo probando el juego.

Figura 4.8: Resultados de la herramienta *profiler* al analizar una cinemática.

- El Fantasma morado es considerado por muchos usuarios como el enemigo más poderoso en los niveles de plataforma, por que se si se deseará hacer niveles más difíciles este debería de ser el enemigo predominante.
- El fantasma rojo es considerado por muchos usuarios como el enemigo más débil en los niveles de plataforma, por que se si se deseará hacer niveles más fáciles este debería de ser el enemigo predominante.
- Las dos principales causas de muerte en los jugadores son el tiempo de respuesta de la *GUI* y que los enemigos eran demasiado fuertes.
- La mayoría de los usuarios consideran sus muertes como un factor de reto en el juego. Considerando que la principal causa de muerte fue el tiempo de respuesta de la *GUI*, se puede concluir que mejorando este factor se disminuiría el porcentaje de jugadores que consideran como factor de estrés su muerte.

Adicionalmente lo jugadores hicieron observaciones y peticiones que ellos consideran podrían mejorar la experiencia de juego:

- Animación que indique que un enemigo ha recibido daño.
- Barra de vida para los enemigos.
- Posibilidad de que el jugador se agache.
- Mensajes de confirmación para los botones que llevan al menú de selección y que cierran la aplicación.
- Mejorar el comportamiento de los disparos.

Si se desean consultar las gráficas se puede consultar el anexo ??.

4.2. Niveles terminados

Para que un nivel pueda ser considerado terminado debe de tener al menos la funcionalidad especificada en el documento de diseño, la cual contempla lo siguiente:

- Actualización de la barra de vida.
- Actualización de la barra de *tonalli*
- Actualización de los marcadores, en caso de que el nivel contenga objetos colecciónables.
- Enemigos, salvo por el primer nivel.
- Obstáculos.
- Ítems, salvo por el primer nivel.
- Control del personaje por medio de la *GUI*.
- Gestión de la muerte del jugador.
- Puntos de guardado.

En la figura ?? se muestra la funcionalidad con la que cuentan los niveles pares.

	Nivel 02P	Nivel 02J	Nivel 04P	Nivel 04J	Nivel 06P	Nivel 06J	Nivel 08P	Nivel 08J	Nivel 10J 01	Nivel 10J 02	Nivel 10J 03
Actualización de la Barra de tonalli.	X	X	X	X	X	X	X	X	X	X	X
Actualización de la Barra de vida.	X	X	X	X	X	X	X	X	X	X	X
Cinemáticas	X	X	X	X	X	X	X	X	X	X	X
Enemigos	X	X	X	X	X	X	X	X	X	X	X
Obstáculos			X		X	X	X				
Ítems	X	X	X	X	X	X	X	X	X	X	X
Panel de pausa	X	X	X	X	X	X	X	X	X	X	X
Panel de fin de partida	X	X	X	X	X	X	X	X	X	X	X
Panel de partida finalizada (Solo plataforma)			X		X		X				
Gestión de la muerte del jugador	X	X	X	X	X	X	X	X	X	X	X
Actualización del marcador (Solo niveles de plataforma 2 y 4)			X								
Puntos de guardado (Solo plataforma)	X		X		X		X				
Manejo del personaje por medio de la GUI	X	X	X	X	X	X	X	X	X	X	X

Figura 4.9: Funcionalidad con la que cuentan los niveles pares, la P es para los niveles de plataforma y la J para los niveles de jefes.

Capítulo 5

Conclusiones

En esta sección se presentan a manera de conclusiones las adiciones que se harían sobre la metodología *huddle* a fin de mejorarla y el trabajo a futuro a realizar sobre el juego.

5.1. Adiciones a la metodología *huddle*

En esta sección se muestran las adiciones a la metodología *huddle* como producto de analizar las estrategias desarrolladas durante todo el proyecto en la etapa de postproducción. Las modificaciones que se proponen son las siguientes:

- **Dos documentos de diseño.** Para mejorar el diseño del documento de diseño se propone realizar dos documentos de diseño: uno para los programadores y otro para los artistas. La necesidad de separar el documento de diseño en dos nace en la etapa de producción en la que se perdía de tiempo leyendo el documento de diseño la hora de programar los actores ya que este contenía tanto información de diseño de personajes como información para la programación del comportamiento de los actores; un documento especializado para programadores ahorraría tiempo en su consulta ya que incluiría únicamente información que le resulta importante al programador. Dividir el documento de diseño también permitiría una especialización del mismo; bajo este enfoque ambos documentos contendrán información relevante solo para los perfiles que lo leen pudiendo incluir en el documento de diseño del programador los diagramas de clases y parte de la documentación desde la etapa de la preproducción. Por otra parte, el documento del artista haría más énfasis en el desarrollo de la historia, los personajes, los escenarios, los diálogos, el diseño del lore, etc. El reto que vendría con la creación de dos documentos de diseño sería el lograr que ambos documentos se encuentren en sintonía y ninguno tenga información que contradiga al otro, ya que ambos deben de hablar del mismo proyecto variando únicamente el enfoque desde el que se observa dicho proyecto.
- **Orientar el desarrollo a los componentes no a los niveles.** Orientar el desarrollo a componentes permite paralelizar el desarrollo del proyecto, haciendo que cada integrante del equipo se encargue de una cantidad de componentes determinada sin que se generen conflictos de código. Reuniendo todos los componentes hasta que se construyan los niveles. El principal reto de trabajo de esta reorientación del trabajo es que requiere que todos los integrantes del equipo de desarrollo sigan las pautas de diseño al pie de la letra y en caso de modificaciones, debe existir una buena comunicación para que se realicen los cambios sin que estos impacten significativamente en el trabajo ya realizado.

- **Mantener el modelo actores-controladores para futuros proyectos.** Esto debido a que dicho modelo permite tener escalabilidad dentro del proyecto.

5.2. Trabajo a futuro

Al termino del presente trabajo termina y a fin de mejorar la experiencia de los jugadores, los siguientes puntos pueden ser mejorados o agregados a fin de enriquecer la experiencia del jugador:

- Mejorar el comportamiento del salto.
- Reducir el tiempo de respuesta del personaje.
- Aumentar indicadores numéricos a las barras de vida y de *tonalli* para que el jugador sepa que cantidad exacta le queda de dichos atributos.
- Agregar mensajes de confirmación a los botones cuya funcionalidad es abandonar el nivel que se esta jugando.
- Implementar nuevos bloques de animación en enemigos y objetos a fin de enriquecer la experiencia visual del juego.
- Agregar animación de personajes a las cinemáticas.
- Agregar audio de voces en náhuatl para los diálogos de las cinemáticas.
- Agregar una animación a los enemigos para indicar que han recibido daño.
- Agregar una barra de vida a los enemigos.
- Agregar música propia al juego.

En este capítulo se encuentran todos los anexos que se mencionaron en los capítulos anteriores.

Apéndice A

Interfaces

En este apéndice se muestran las versiones finales de las interfaces del juego.

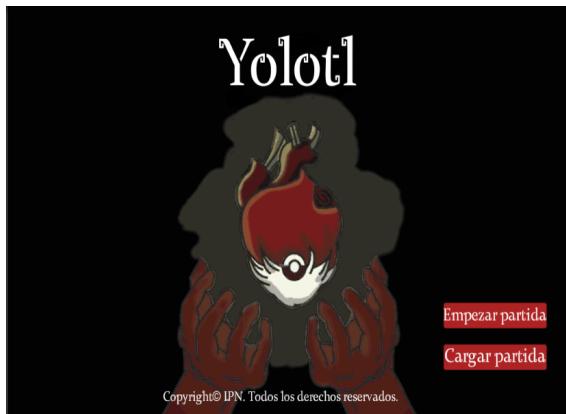


Figura A.1: Interfaz del menú principal



Figura A.2: Mensaje de advertencia en la creación de una nueva partida en la interfaz del menú principal



Figura A.3: Interfaz del menú de selección de nivel

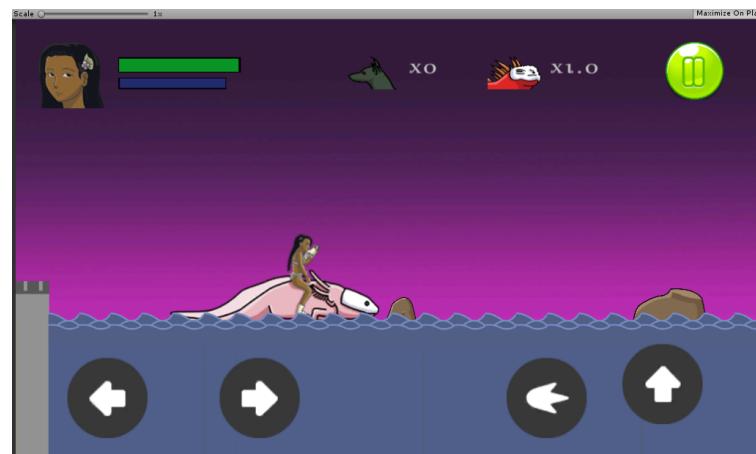


Figura A.4: Interfaz del juego de un nivel.

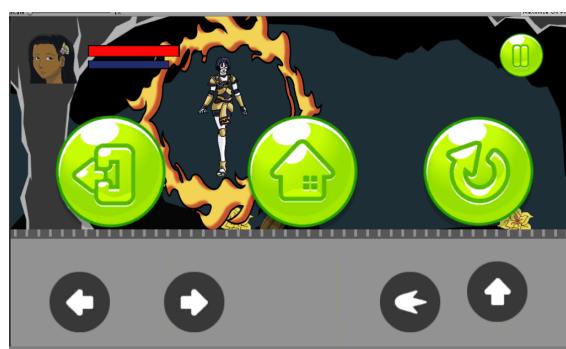


Figura A.5: Interfaz de fin de la partida.



Figura A.6: Interfaz de partida pausada.



Figura A.7: Interfaz de nivel completado.

Apéndice B

Sprites de Personajes

En este apéndice se presentan los *sprites* creados durante la duración del trabajo terminal.

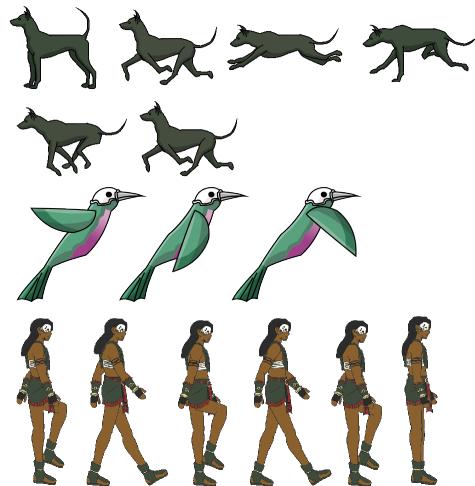


Figura B.1: Sprites para las diferentes formas que toma Xólotl a lo largo del juego.



Figura B.2: Sprites para los enemigos normales del juego.

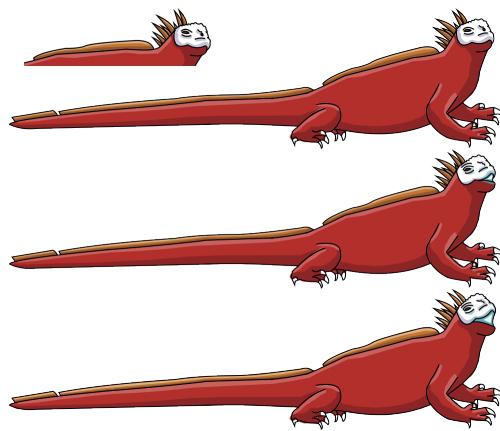
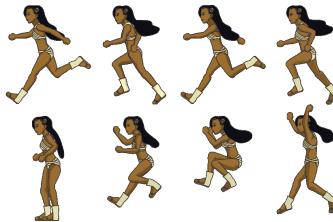
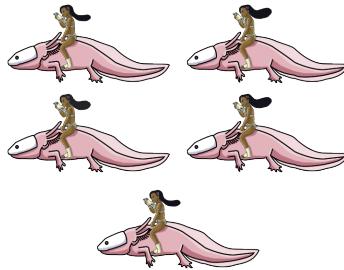


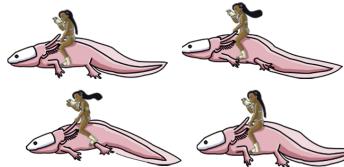
Figura B.3: Sprites de Xochitonal.



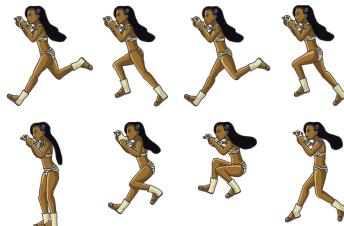
(a) Sprites Malinalli para el primer nivel.



(b) Sprites Malinalli de nado para el segundo nivel.



(c) Sprites Malinalli de salto para el segundo nivel.



(d) Sprites Malinalli para los niveles posteriores al segundo nivel.

Figura B.4: Sprites del personaje jugable (Autoria propia)

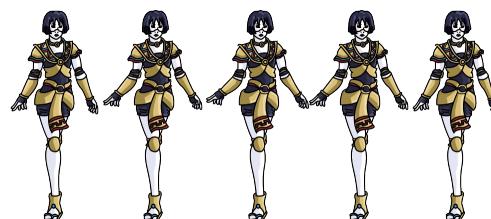


Figura B.5: Sprites de Itzpapálotl.

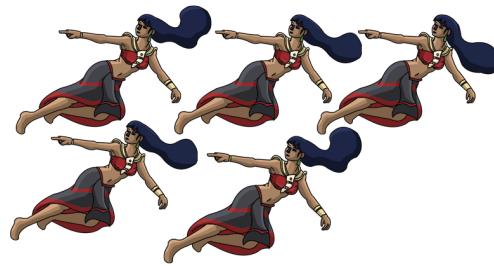


Figura B.6: Sprites de Tlazolteotl.

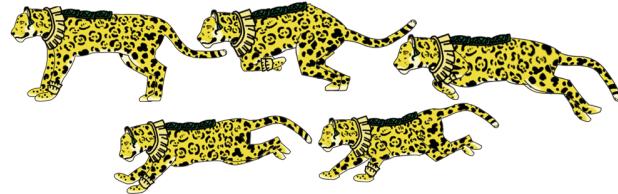


Figura B.7: Sprites de Tepeyollotl.



Figura B.8: Sprites de Mictlantecuhtli.

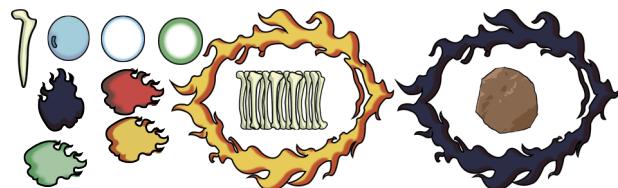


Figura B.9: Sprites de los ataques de los personajes.

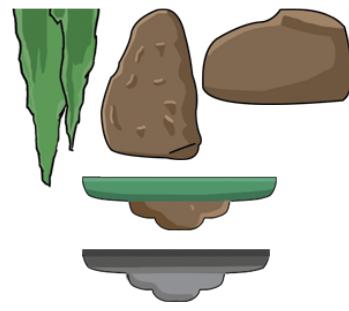


Figura B.10: Sprites de los obtáculos.

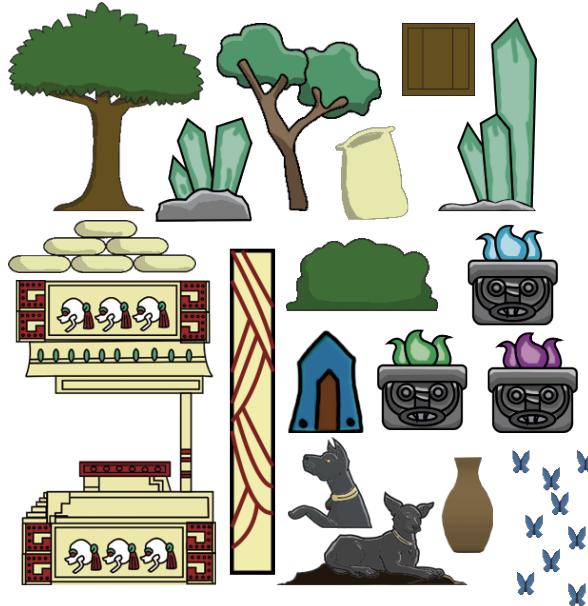


Figura B.11: Sprites de los objetos de fondo.



Figura B.12: Sprites de la animación del checkpoint.



Figura B.13: Sprites de los ciudadanos.



Figura B.14: Sprites de los iconos e ítems.

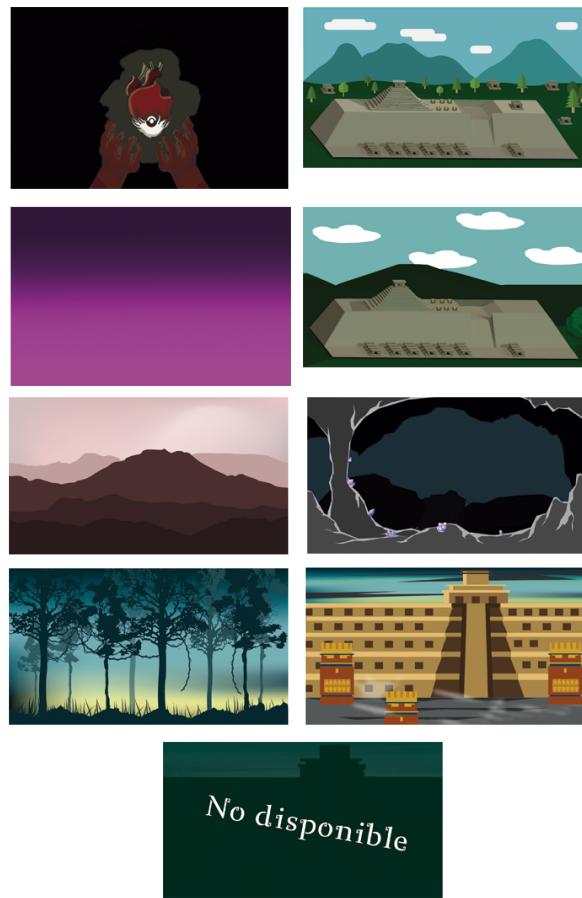
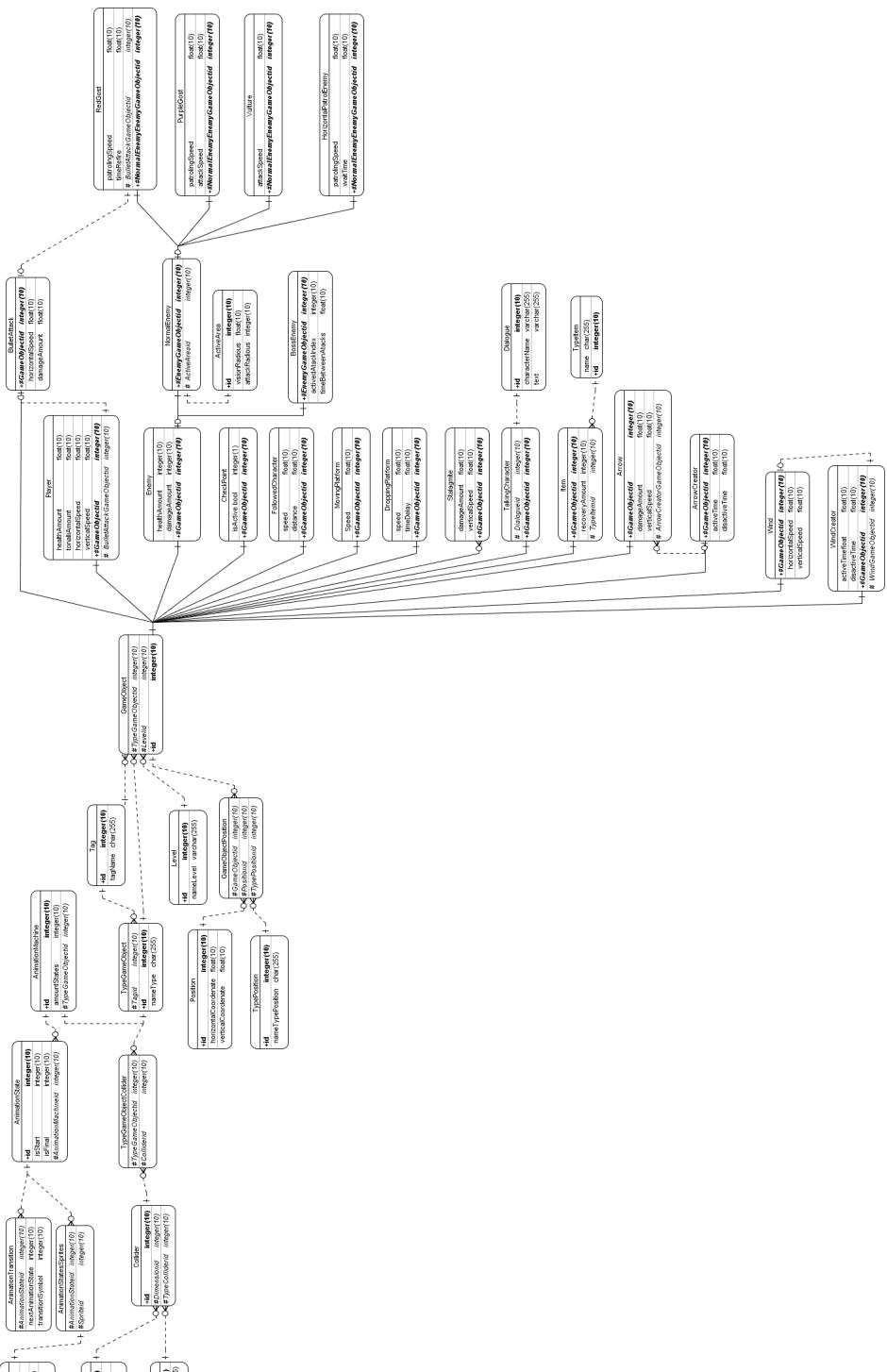


Figura B.15: Sprites de los findos de cada nivel.

Apéndice C

Modelo de Datos



Apéndice D

Control de adicción en el jugador

En este apartado se presentan las diferentes estrategias que se pueden implementar dentro del presente trabajo terminal para evitar el desorden de juego en sus jugadores. Cada estrategia se describe y en seguida se mencionan las implicaciones de su implementación.

D.1. Notificación de confirmación para continuar la partida

Esta mecánica consiste en que después de que el juego detecta que el jugador ha pasado mucho tiempo jugando (aproximadamente más de una hora); el juego se pausa a sí mismo y le comunica al jugador que ha pasado mucho tiempo jugando, dándole la opción de terminar la partida o continuar. Esta mecánica es utilizada por algunos juegos de consola como *Just Dance* o aquellos que involucran alguna actividad física; en el caso de *Just Dance*, el juego se pausa a sí mismo y le pregunta al jugador si no se siente fatigado por el largo tiempo que éste ha estado activo.

D.1.1. Implicaciones

Esta mecánica exige que se manejen módulos que contabilicen el tiempo que el jugador este dentro de la aplicación y lance mensajes después de que detecte que ha pasado una cantidad significativa de tiempo. Si bien la implementación de esta mecánica no parece involucrar modificaciones significativas dentro de los módulos ya existentes del juego; no es muy recomendable de ser llevada a la práctica ya que se vale totalmente de la buena voluntad del jugador para controlarse y no representa un limitador real del tiempo de partida.

D.2. Control paterno

En esta mecánica de control se vincula la partida del jugador con una cuenta de correo perteneciente a algún parente o tutor. La aplicación envía a esta cuenta un registro de número de veces que el jugador abre el juego y cuánto tiempo permanece en éste. Este sistema de control también se le puede implementar el envío de un formulario al correo del tutor en el cual éste pueda indicarle al juego cuantas veces al día y cuánto tiempo puede ser abierta diariamente o semanalmente.

D.2.1. Implicaciones

Esta mecánica es una de las que requiere mayor modificaciones y agregaciones al sistema pues su implementación exige un módulo de monitoreo de tiempo, envío de formularios, verificación de permisos y que el juego tenga la capacidad de bloquearse o desbloquearse con base en la información enviada por el padre o tutor. Si bien esta mecánica de control puede resultar eficiente con los jugadores más jóvenes dentro del público objetivo; no se debe de dejar a un lado que la misma puede ser considerada intrusiva para los jugadores de mayor edad, pudiendo repercutir de manera negativa en la experiencia de juego.

D.3. Sistema de vidas

Esta mecánica plantea la existencia de una cantidad de vidas que el jugador deberá de ocupar para poder avanzar dentro del juego, ya sea cuando ingrese a un nivel o para revivir dentro del mismo. La cantidad de vidas usualmente se restauran después de una determinada cantidad de tiempo, por ejemplo, en el juego de *Pokemon Shuffle* desarrollado por *Nintendo* para dispositivos móviles, las vidas con las que cuenta el jugador se van restaurando de una en una cada quince minutos. Por otro lado, existen juegos como *Kingdom Hearts Union X* en donde la mecánica que limita el tiempo de juego depende de las misiones que el jugador decida emprender; ya que para iniciar una misión el jugador debe de pagar con energía (*AP*), existiendo misiones que requieren mayor cantidad de *AP* que otras. Es importante recalcar que estas mecánicas son implementadas por las empresas no solo para limitar el tiempo que el jugador invierte dentro de la aplicación, sino que este sistema permite a las empresas implementar microtransacciones dentro de sus juegos al ofrecerle alternativas a sus jugadores para seguir jugando sin la necesidad de esperar por nuevas vidas o *AP*.

D.3.1. Implicaciones

El desarrollo de un sistema de vidas dentro del juego demanda la implementación de nuevas variables globales dentro del sistema, tal como la cantidad de vida y un contador de tiempo que restaure las mismas. De igual forma, es necesario implementar un módulo robusto que soporte las alteraciones del reloj del dispositivo móvil en caso de que el jugador decida adelantar el reloj de su dispositivo tratando de engañar al sistema y que este le restaure vidas sin que haya pasado el tiempo de espera. El uso de estas nuevas variables y la creación de estos nuevos módulos repercutiría directamente en el funcionamiento de las clases controladoras ya existentes del juego, las modificaciones a las clases controladoras repercuten directamente en el tiempo de desarrollo del juego por lo que esta estrategia no se considera viable para el cronograma que ya se tenía previsto al inicio del proyecto.

Por otra parte, el uso de un sistema de vidas no es recomendable para evitar el desorden de juego, esto debido a que existen diversos estudios en los que se ha logrado determinar que este tipo de estrategias favorecen al desorden de juego al generar ansiedad en los jugadores con los tiempos de espera entre partidas.

D.4. Observaciones

El desorden de juego es un problema que aqueja a muchos jugadores a nivel mundial sin embargo es un problema complejo y que depende de muchos factores. El principal reto que existe para desarrolladores para hacerle frente a este problema es que muchas de las estrategias que se podrían implementar, como las mecánicas de control anteriormente descritas, llegan a incrementar la complejidad del juego como sistema o repercuten en la inmersión y experiencia del usuario. De igual forma, es importante mencionar que el desorden de juego es una patología que se está estudiando recientemente y no existe, a la fecha del presente trabajo terminal, información concluyente de ningún centro de investigación sobre medidas para evitarlo o diagnosticarlo. Con lo anterior este trabajo terminal no implementara ninguna de las estrategias presentadas en este anexo; pues las implicaciones para su implementación no son posibles de llevar a cabo durante la duración del Trabajo Terminal por lo que se dejan como mera mención para quienes deseen retomar el trabajo a futuro.

Apéndice E

Cuestionario de aceptación del juego

En este apéndice se muestra el cuestionario que se le presentó a las jugadores durante la etapa de pruebas de aceptación.

Háblanos de ti:

Responde según se le pide.

1. Género:

- a) Mujer.
- b) Hombre.
- c) Prefiero no decirlo.

2. Edad:

- a) De 13 a 15 años.
- b) De 16 a 18 años.
- c) De 19 a 21 años.
- d) De 22 a 24 años.
- e) Mayor de 24 años

3. Marca del dispositivo móvil para la prueba:

- Sony.
- Samsung.
- Huawei.
- Motorola.
- ZTE.
- Xiaomi.
- Otro.

4. Versión de Android de su dispositivo

- a) Alguna distribución inferior a Android 5.
- b) Alguna distribución de Android 5.
- c) Alguna distribución de Android 6.
- d) Alguna distribución de Android 7.
- e) Alguna distribución superior a Android 7.

5. Usualmente cuantas horas de juego acostumbra en dispositivos móviles:

- a) Menos de una hora.
- b) Más de 1 pero menos de 2 horas.
- c) Más de 2 pero menos de 3 horas.
- d) Más de 3 pero menos de 4 horas.
- e) Más de 4 horas.

Háblanos del juego:

Responde según se le pide.

1. El nivel a probar es

- a) La chica y el perro (Ciudad).
- b) La chica y el perro (Selva).
- c) Nadie cruza mis dominios(Plataforma).
- d) Nadie cruza mis dominios(Jefe).
- e) La guarida del jaguar (Plataforma).
- f) La guarida del jaguar(Jefe).
- g) Alas de obsidiana (Plataforma).
- h) Alas de obsidiana (Jefe).
- i) El viento del norte (Plataforma).
- j) El viento del norte (Jefe).
- k) Sin gravedad (Plataforma).
- l) Sin gravedad (Jefe).
- m) Castigo (Plataforma).
- n) Castigo (Jefe).
- ñ) La última batalla del jaguar (Plataforma).
- o) La última batalla del jaguar (Jefe).
- p) El último caballero del rey (Plataforma A).
- q) El último caballero del rey (Plataforma B).
- r) El último caballero del rey (Jefe).
- s) El rey del Mictlán (Jefe 01).

- t) El rey del Mictlán (Jefe 02).
- u) El rey del Mictlán (Jefe 03).
2. Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras el movimiento del personaje?
- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.
3. En caso de que consideres ineficiente o pésimo el movimiento del personaje, ¿Qué es lo que hace que lo consideres de esa forma?
- a) La velocidad de movimiento es muy lenta.
- b) La velocidad de movimiento es muy rápida.
- c) El salto es inestable.
- d) Otro.
4. Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la respuesta de la GUI?
- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.
5. En caso de que consideres ineficiente o pésimo la respuesta de la GUI, ¿Qué es lo que hace que lo consideres de esa forma?
- a) El tamaño de los botones es muy pequeño.
- b) La posición del botón del salto no es intuitiva, debería estar en la posición del botón del disparo.
- c) El personaje tarda mucho en responder.
- d) Otro.
6. Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras el uso de tonalli(magia)?
- a) 1.
- b) 2.
- c) 3.

- d) 4.
- e) 5.
7. En caso de que consideres ineficiente o pésimo el uso de Tonalli, ¿Qué es lo que hace que lo consideres de esa forma?
- a) No hay suficiente cantidad de Tonalli.
- b) No hay forma de ver la cantidad de disparos que puedo hacer.
- c) La restricción de una cantidad específica de tonalli.
- d) Otro.
8. Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la actualización de la barra de vida?
- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.
9. En caso de que consideres ineficiente o pésima la actualización de la barra de vida ¿Qué es lo que hace que lo consideres de esa forma?
- a) No hay un sonido que indique cuando estoy próximo a quedarme sin vida.
- b) No hay una animación más llamativa que indique que he recibido daño.
- c) No hay un indicador numérico que me permita ver cuanta vida me queda.
- d) Otro.
10. Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la actualización de la barra de tonalli?
- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.
11. En caso de que consideres ineficiente o pésima la actualización de la barra de tonalli ¿Qué es lo que hace que lo consideres de esa forma ?
- a) No hay sonido que indique cuando me he quedado sin tonalli.
- b) No hay una animación más llamativa que indique que he gastado tonalli.
- c) No hay un indicador numérico que me permita ver cuántos disparos me quedan.
- d) Otro.

12. Del 1 al 5, considerando el 5 como muy fuerte y el 1 como muy débil, ¿Qué tan fuerte es (son) el (los) enemigo(s)?

- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.

13. En caso de que nivel sea una plataforma, seleccione el enemigo que consideres más débil:

- a) Fantasma morado.
- b) Fantasma rojo.
- c) Armadillo.
- d) Zopilote.
- e) Jaguar.

14. ¿Qué es lo que hace que consideres a ese enemigo como el más fuerte?

- a) Su patrón de movimiento.
- b) Su cantidad de daño.
- c) Su cantidad de vida.
- d) Su modo de ataque.
- e) Otro.

15. En caso de que nivel sea una plataforma, seleccione el enemigo que consideres más fuerte:

- a) Fantasma morado.
- b) Fantasma rojo.
- c) Armadillo.
- d) Zopilote.
- e) Jaguar.

16. ¿Qué es lo que hace que consideres a ese enemigo como el más débil?

- a) Su patrón de movimiento.
- b) Su cantidad de daño.
- c) Su cantidad de vida.
- d) Su modo de ataque.
- e) Otro.

17. Cuantas veces moriste antes de completar el nivel.

- a) 0 o 1 vez.

- b) 2 o 5 veces.
- c) 5 u 8 veces.
- d) Más de 8 veces.

18. ¿A que se debe que haya muerto la mayoría de las veces?

- a) El(los) enemigo(s) son demasiado fuertes.
- b) El(los) enemigo(s) son demasiados.
- c) Por un obstáculo (Rocas, viento, estalagmitas).
- d) El personaje tarda mucho en responder a los botones que oprimo.
- e) Me quede sin magia en un momento decisivo de la batalla.
- f) Falta de items para curarme.
- g) Falta de item para recuperar tonalli.

19. ¿Consideras tus muertes como un factor de estrés o como un factor de reto?:

- a) Es un factor de estrés.
- b) Es un factor de reto.

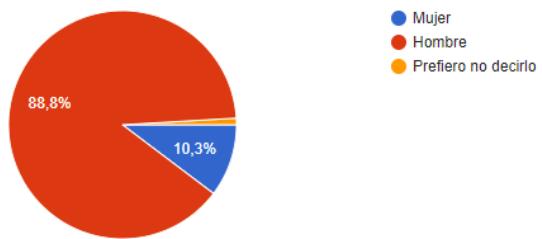
Apéndice F

Resultados de la encuesta de aceptación del juego

En este apéndice se muestran los resultados de la encuesta de aceptación a manera de gráficas.

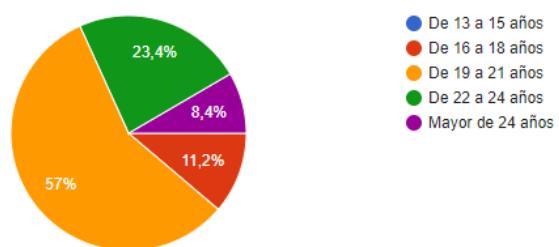
Género

107 respuestas



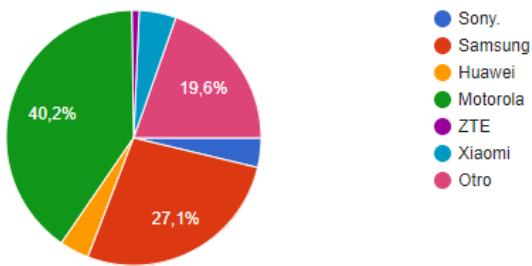
Edad

107 respuestas



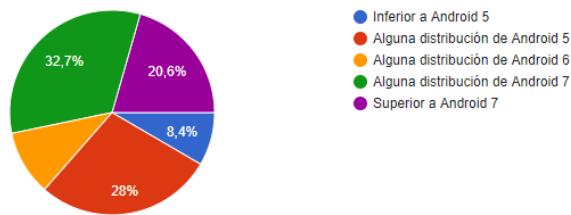
Marca del dispositivo móvil para la prueba:

107 respuestas



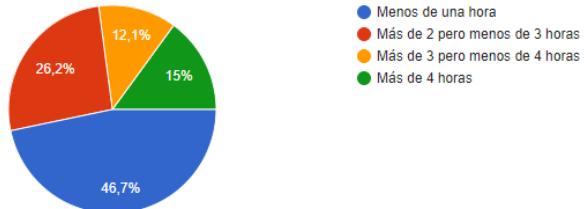
Versión de Android de su dispositivo

107 respuestas



Usualmente cuantas horas de juego acostumbra en dispositivos móviles

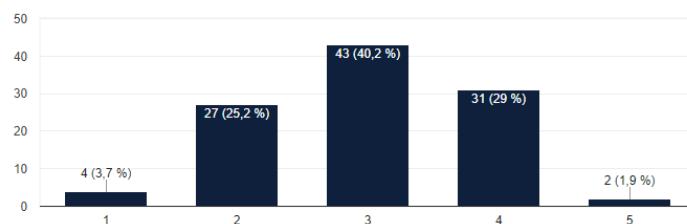
107 respuestas





Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras el movimiento del personaje?

107 respuestas



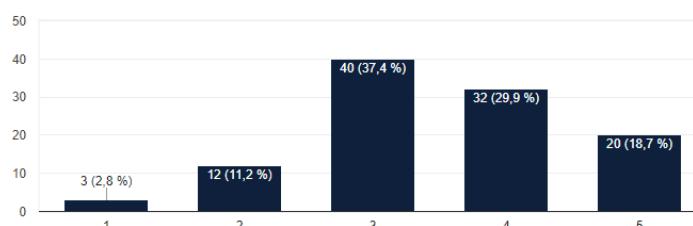
En caso de que consideres ineficiente o pésimo el movimiento del personaje, ¿Qué es lo que hace que lo consideres de esa forma?

88 respuestas



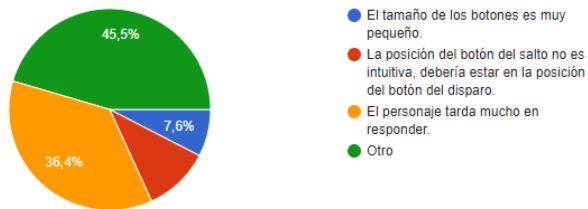
Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la respuesta de la GUI?

107 respuestas



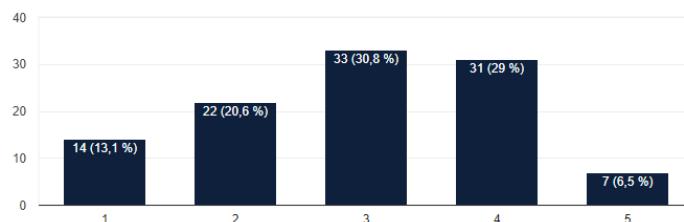
En caso de que consideres inefficiente o pésimo la respuesta de la GUI, ¿Qué es lo que hace que lo consideres de esa forma?

66 respuestas



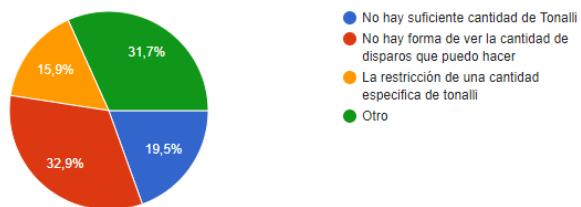
Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras el uso de tonalli(magia)?

107 respuestas



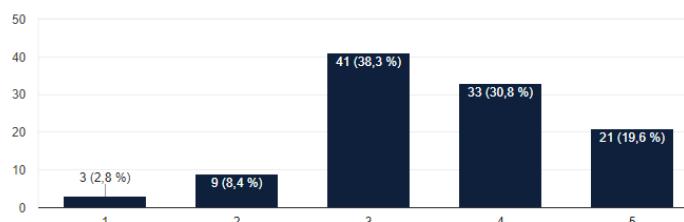
En caso de que consideres inefficiente o pésimo el uso de Tonalli, ¿Qué es lo que hace que lo consideres de esa forma?

82 respuestas



Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la actualización de la barra de vida?

107 respuestas



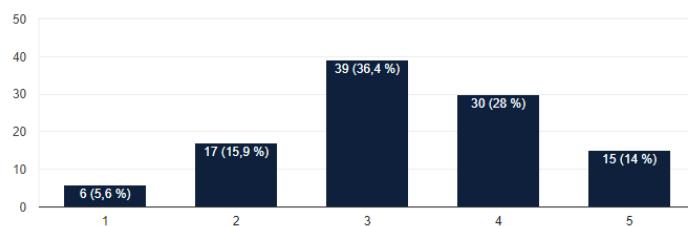
En caso de que consideres inefficiente o pésimo la barra de vida, ¿Qué es lo que hace que lo consideres de esa forma?

65 respuestas



Del 1 al 5, considerando el 5 como excelente y el 1 como pésimo, ¿Cómo consideras la actualización de la barra de tonalli?

107 respuestas



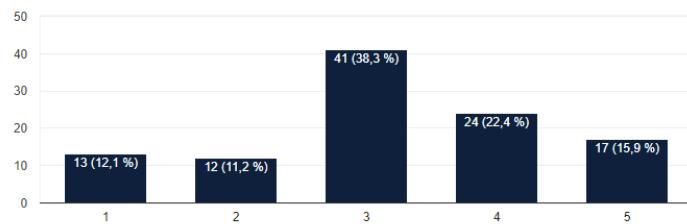
En caso de que consideres inefficiente o pésimo la barra de tonalli, ¿Qué es lo que hace que lo consideres de esa forma?

65 respuestas



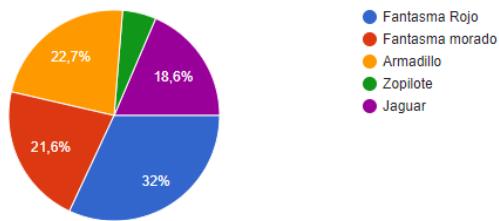
Del 1 al 5, considerando el 5 como muy fuerte y el 1 como muy débil, ¿Qué tan fuerte es (son) el (los) enemigo(s)?

107 respuestas



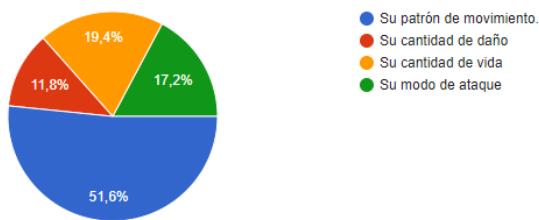
En caso de que nivel sea una plataforma, seleccione el enemigo que consideres más débil:

97 respuestas



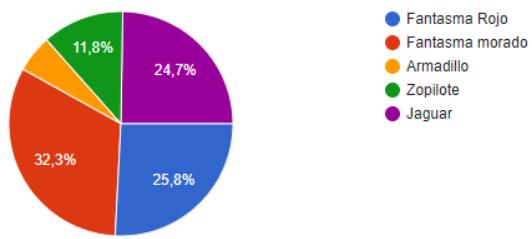
¿Qué es lo que hace que consideres a ese enemigo como el más débil?

93 respuestas



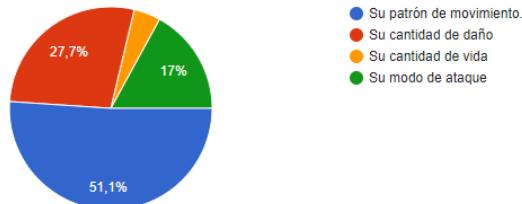
En caso de que nivel sea una plataforma, seleccione el enemigo que consideres más Fuerte:

93 respuestas



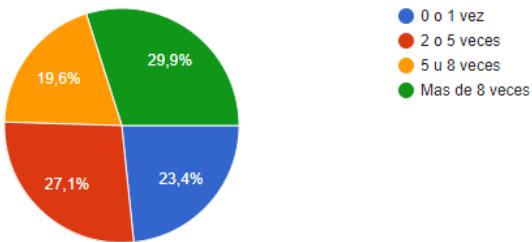
¿Qué es lo que hace que consideres a ese enemigo como el más fuerte?

94 respuestas



¿Cuántas veces moriste antes de completar el nivel?

107 respuestas



¿A qué se debe que haya muerto la mayoría de las veces?

107 respuestas



¿Consideras tus muertes como un factor de estrés o como un factor de reto?

107 respuestas

