



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.



Noviembre 2017



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



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 mexicana.

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

¹daughterofthewind10@gmail.com

²yolotl.escom@gmail.com

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

Advertencia	II
1. Introducción	1
2. Antecedentes	2
2.1. Propuesta	2
2.1.1. Planteamiento del problema	2
2.1.2. Marco Teorico	2
2.1.3. Planteamiento de la solución	2
2.2. Ajustes	2
2.2.1. Corrección	2
2.2.2. Nueva división de trabajo	2
2.2.3. Actualizando el motor de juego	2
2.3. Contribuciones	4
2.4. Trabajo realizado durante trabajo terminal 1	4
2.4.1. Etapa de Preproducción	4
3. Resultados obtenidos	7
3.1. Prueba	7
3.1.1. Objetivo de la prueba	7
3.1.2. Herramientas utilizadas durante la prueba	7
3.1.3. Aplicación de la prueba	7
3.1.4. Conclusiones de la prueba	7
4. Conclusiones	8
5. Anexos	9
5.1. Interfaces	9
5.2. Diseño de Personajes	9
5.3. Modelo de Datos	9
5.4. Control de adicción en el jugador	9
5.5. Maquetas de niveles	9

Capítulo 1

Introducción

Capítulo 2

Antecedentes

2.1. Propuesta

2.1.1. Planteamiento del problema

2.1.2. Marco Teorico

2.1.3. Planteamiento de la solución

2.2. Ajustes

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

2.2.1. Corrección del enfoque de la solución

2.2.2. Nueva división de trabajo

Antes del inicio del tercer *sprint* y teniendo como base la experiencia de desarrollo los anteriores prototipos, queda claro que se necesita diseñar una nueva estrategia que permita agilizar el desarrollo del juego sin comprometer la calidad del mismo. 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 2.1 .

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.

2.2.3. Actualizando el motor de juego

Paralelamente a la nueva asignación de tareas, fue liberada la versión 2017.3.1f de *Unity3D*. Esta versión incluye herramientas que agilizan la creación de niveles como el uso de:

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 (Jefe fase 01)
Nivel 09, el último caballero del rey (Plataforma B)	Nivel 10, el rey del Mictlán (Jefe fase 02)
Nivel 09, el último caballero del rey (Jefe A)	Nivel 10, el rey del Mictlán (Jefe fase 03)

Figura 2.1: Asignacion de tareas

- **Tilemap:** Herramienta para el mapeado de niveles. Esta herramienta facilita la creación de mapas al crear una malla sobre la que se arrastraran diferentes *Sprites* que se hayan importado previamente al tilemap (ver figura 2.2). En la sección () se profundizará su funcionamiento.
- **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 limitara el movimiento de la cámara (ver figura 2.4). *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.
- **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, ya que 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*.

Por el impacto que tendrían las nuevas herramientas de la versión de *Unity*, se propuso 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 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

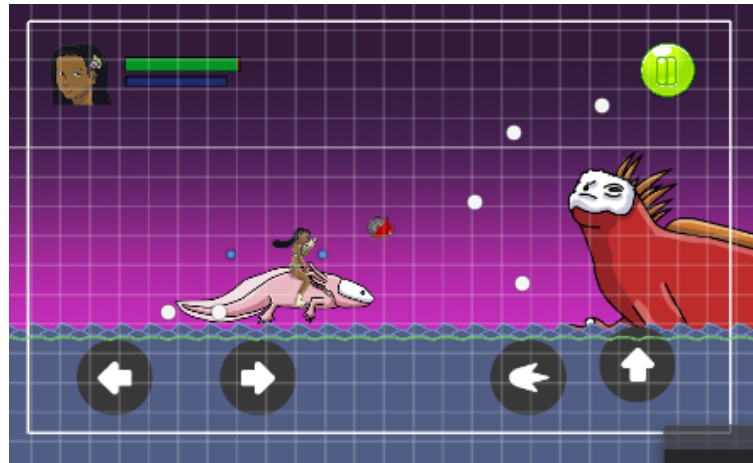


Figura 2.2: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

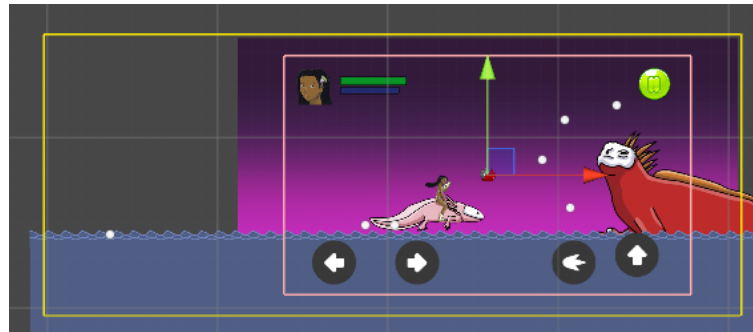


Figura 2.3: Vista de la escena cuando se tiene un *GameObject* de tipo *Tilemaps* para la construcción de niveles

de *Unity* sería la que se emplearía para el resto del desarrollo del juego.

2.3. Contribuciones

2.4. 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.

2.4.1. 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*.

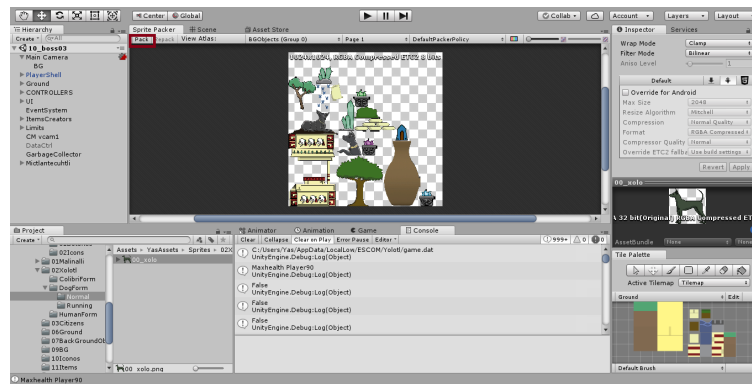


Figura 2.4: Vista de la pestaña del *Sprite Packer*.

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 mexicana:** su historia tradiciones y clases sociales.
- **Mitología mexicana:** 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, códigos, 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 mecanica, sus personajes, etc. Luego de investigar sobre los motores de juegos disponibles, se elije 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 5.1.

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 códigos. En el anexo 5.2 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.

2.4.2. 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 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 coleccionables, los checkpoints 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*. 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 animación 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.

Capítulo 3

Trabajo realizado

3.1. Etapa de producción de *Huddle*

3.1.1. Cuarto *sprint* de producción

Capítulo 4

Resultados obtenidos

4.1. Prueba

4.1.1. Objetivo de la prueba

4.1.2. Herramientas utilizadas durante la prueba

4.1.3. Aplicación de la prueba

4.1.4. Conclusiones de la prueba

Capítulo 5

Conclusiones

Capítulo 6

Anexos

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

6.1. Interfaces

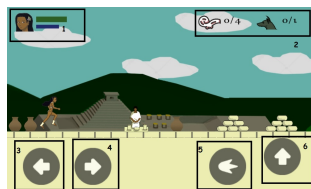


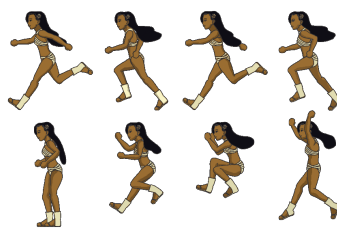
Figura 6.1: a nice plot

6.2. Diseño de Personajes

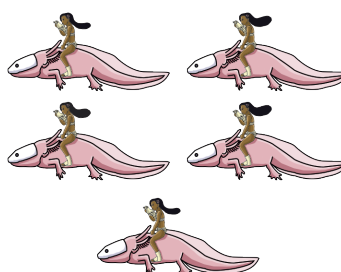
6.3. Modelo de Datos

6.4. Control de adicción en el jugador

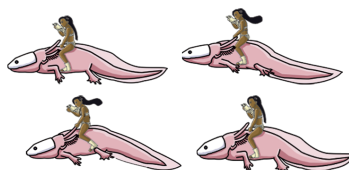
6.5. Maquetas de niveles



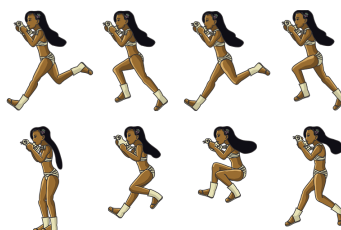
(a) Sprites Malinalli para el primer nivel.



(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 6.2: Sprites del personaje jugable (Autoria propia)

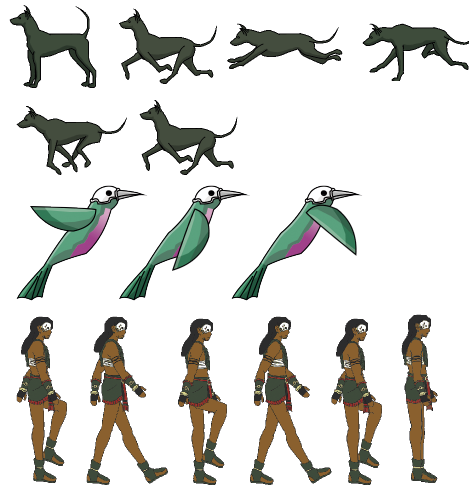


Figura 6.3: Sprites para las diferentes formas que toma Xólotl a lo largo del juego.



Figura 6.4: Sprites para los enemigos normales del juego.

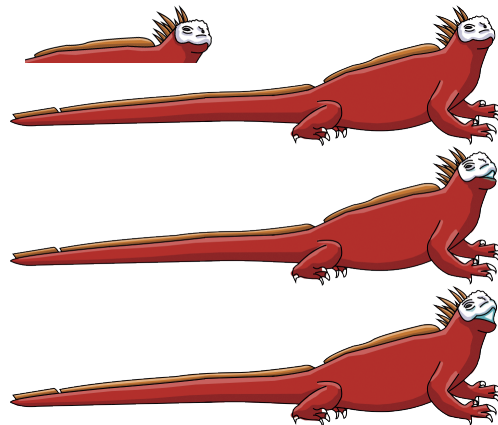


Figura 6.5: Sprites de Xochitonal.

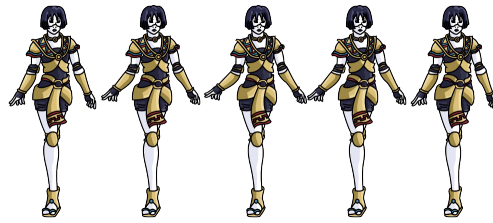


Figura 6.6: Sprites de Itzpapálotl.

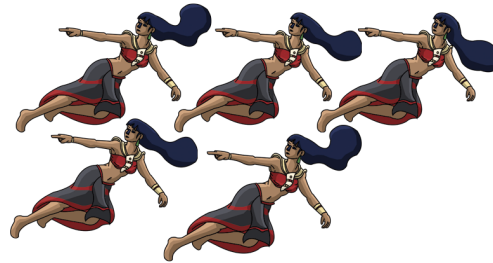


Figura 6.7: Sprites de Tlazolteotl.



Figura 6.8: Sprites de Tepeyollotl.

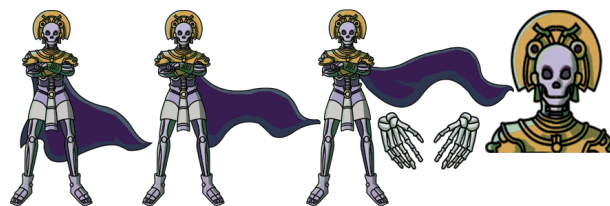


Figura 6.9: Sprites de Mictlantecuhтли.

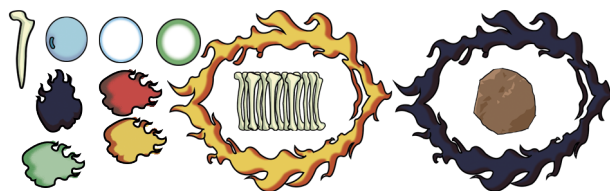


Figura 6.10: Sprites de los ataques de los personajes.