

Creación e implementación de videojuego educativo que enseña la historia de la guerra del pacífico

Ruth Margarita García López
Asesorado por Lic. Brígida Carvajal

23 de septiembre de 2015

1. Antecedentes

1.1. Juegos Comerciales

1.1.1. Age of Empires (saga)

Es un juego de estrategia en tiempo real, en el que el jugador puede recolectar recursos. Estos recursos son usados para construir edificios que sirven para construir unidades y mejorarlas. El jugador puede armar un ejército e invadir el imperio de otros jugadores, todo esto mientras avanza a través de las edades de la historia.

A parte de poder ser jugado en modo de escaramusas, también se puede jugar en modo campaña; en el modo campaña, el jugador es guiado a través de la historia medieval, peleando batallas y conociendo a los personajes más influyentes de esa época. También se muestran ocasionalmente algunos videos que explican el trasfondo de cada misión.

1.1.2. Caesar (saga)

Es un videojuego de simulación de ciudades en el que el jugador construye una ciudad tomando el papel del César en la antigua Roma.

A diferencia de Age of Empires, el aspecto educativo es muy reducido, sin embargo, la lógica de construcción de ciudades es muy acertada históricamente.

1.1.3. Valiant Hearths

Es un juego de puzzle/aventura basado en cartas enviadas durante “la gran guerra” (WWI). El jugador maneja a varios personajes en el transcurso del juego, entre ellos: Karl, uno de los muchos alemanes deportados de Francia, Karl es separado de su esposa e hijo. El jugador deberá usar sus habilidades para sobrevivir, explorar el área y resolver puzzles, todo esto mientras es parte activa de su entorno, explorando cómo era el mundo durante la gran guerra en 1914.

El juego se centra más en la vida de las personas durante la gran guerra, y aunque narra parte de la historia mediante textos informativos antes de cada nivel, esta historia no es la parte central de este juego, sino la interacción entre las personas que vivieron dicha guerra, y la forma en la que vivían. [?]

1.2. Juegos *indie*

Existen varias(Owens y Safley, s.f.)(CommonSense, s.f.) páginas web que contiene una gran colección de videojuegos educativos de historia. Actualmente existen 126 juegos de historia indexados en el portal playing history ¹(Owens y Safley, s.f.). 1 Debido a la gran cantidad, solo se tomarán en cuenta algunos de estos videojuegos.

1.2.1. 1066: The Game

Es un juego de estrategia por turnos, cuya jugabilidad se asemeja al estilo de combate del juego Heroes of Might and Magic(Ubisoft, s.f.). El jugador dirige a diferentes batallones en diversas batallas, cada batalla, representa una de las batallas en la que los ingleses lucharon defendiendo su territorio de los enemigos que los asediaban durante 1066. Antes de cada combate, una animación describe el contexto histórico de la batalla.

1.3. Edutainment

Pese a que a este proyecto le compete, más que nada, el edutainment en los videojuegos, he encontrado información muy valiosa en reseñas sobre todo tipo de edutainment, escritas por una gran comunidad entusiasta (CommonSense, s.f.) del uso de la tecnología en el salón de clases. Pese a que, gran parte de esta información proviene de reseñas sobre plataformas web basadas en texto y foros de debate, estas plataformas contienen ideas que podrían ser incluso más efectivas si se transmitieran explotando la interactividad, la popularidad y la experiencia inmersiva de un videojuego.

¹Cantidad consultada el 25 de agosto de 2015

1.3.1. Historical Scene Investigation

Es una página web en la que los estudiantes pueden explorar los archivos de varios “casos históricos”, luego de leer cartas, ver fotografías, y explorar todo tipo de “evidencias”, deberán responder a varias preguntas planteadas, respecto al contexto, sucesos y modo de vida de la época.

Las respuestas a estas preguntas no siempre serán obvias, muchas veces estarán ocultas implícitamente en detalles que podrían pasar desapercibidos por los jugadores. Por esta razón, es necesario que el jugador explore la evidencia, y preste mucha atención a todos los pequeños detalles.(Swan y Hofer, s.f.)

Debido a que el jugador concluye la respuesta a las preguntas planteadas, es mucho más probable que el aprendizaje sea significativo.(CommonSense, s.f.)

1.3.2. HistoryPin

Es una aplicación web, parecida a google maps, en la que los usuarios pueden “pinnear” fotografías a un mapa, estos pines son descritos mediante un sistema de crowdsourcing, en el que los usuarios pueden añadir información sobre algún pin en específico, sin importar si la información es histórica o personal.(brendan.knowlton@historypin.org, s.f.)

1.3.3. Idea of America

Es una página web dirigida a alumnos de pregrado. La idea es que los estudiantes accedan a esta página, para aprender los principios del debate y diálogo, con el fin de aprender historia de una forma más didáctica, y con un enfoque orientado a la crítica y la autocrítica.(history.org, s.f.)

1.4. Antecedentes Teóricos

1.4.1. Juegos *indie*

Durante la época del nacimiento de la NES, en el año 1983, desarrollar videojuegos era muy difícil, una de las causas, era que la tecnología de almacenamiento de datos no estaba tan avanzada como en la actualidad, así que los lenguajes de programación eran rudimentarios y difíciles de manejar. Los límites para los

desarrolladores y diseñadores también eran estrechos, debido a la velocidad de las computadoras de esa época y a las limitantes en el uso de sprites.

Pero actualmente, el desarrollo de videojuegos se ha simplificado bastante.

En la actualidad, niños de 13 años pueden desarrollar videojuegos sencillo (Meyer, 2013), y a causa de esto ha nacido una nueva industria: la industria de video- juegos indie.

Consiste en equipos de desarrollo de videojuegos compuestos por personas que no pertenecen a ninguna corporación “grande” como Ubisoft o Microsoft.

Estos pequeños equipos de desarrolladores crean pequeños juegos que, pese a que no pueden competir en gráficos y calidad de desarrollo contra corporaciones gigantes especializadas en el desarrollo de videojuegos, muchas veces tienen un gameplay innovador, o conceptos que los lanzan al éxito, como fue el caso de Minecraft de la ex-empresa Mojang, que fue comprada por Microsoft en 2014.

Estos pequeños desarrolladores independientes, se hacen llamar a sí mismos: indie game developers, o como abreviatura: indies.

1.4.2. Edutainment

Pese al pesimismo -justificado por investigaciones- que rodea a los videojuegos en los temas de salud, y sobre todo, en los temas de pedagogía, existen investigaciones más optimistas, que afirman que los videojuegos pueden ser usados como un recurso educativo, si se seleccionan de forma adecuada.

El término edutainment es un juego de palabras entre education, que significa “educación” en inglés, y “entertainment” que significa “entretenimiento” en inglés. Tal vez el término “edutrenimiento” sería una buena traducción.

⁶Se define como edutainment a cualquier contenido o sistema que además de desarrollar alguna habilidad o dejar una enseñanza práctica en sus jugadores, también es divertido.

El término también se aplica a cualquier contenido o sistema que no “haya sido creado” para educar, así como para cualquiera que no “haya sido creado” para entretener, y por lo tanto son entretenidos pero accidentalmente educativos, o son educativos pero accidentalmente entretenidos, o accidentalmente educativos y entretenidos.

1.4.3. Videojuegos y psicología

Existe una gran controversia respecto a los videojuegos, algunos de los conflictos son: ¿cuánto tiempo debería una persona permanecer frente a una pantalla? Los resultados de una investigación realizada durante el 2012 concluyó que los niños, durante la etapa pre-escolar, permanecían más tiempo del recomendado ² frente a una pantalla (Hinkley, Salmon, Okely, Crawford, y K., 2012)

¿hasta qué punto es un videojuego lo suficientemente real como para interferir en las decisiones, iniciativas o salud física de una persona? una investigación conducida por Mary E. Ballard y J. Rose Wiest durante el 2006 intenta determinar los efectos de jugar Mortal Kombat en las respuestas cardiovasculares en una población masculina (Ballard y Wiest, 1996).

1.4.4. Desarrollo de habilidades

Algunos videojuegos desarrollan habilidades visuales espaciales ya que en muchos se requiere que el jugador rote figuras en su mente (como téttris) e incluso, que memorice el mapa de un lugar, que identifique los puntos cardinales y se dirija a varios puntos, todo esto mientras debe prestar atención a eventos auditivos y permanece bajo cierto estrés, como en el caso de los shooters que tienen una modalidad de multijugador.

Quiroga et. al. afirman que es posible medir el factor g, un factor que mide la inteligencia, mediante juegos comerciales.(Angeles Quiroga y cols., 2015)

2. Problema

Pese a que existen muchos juegos sobre historia, no se pudo encontrar ninguno que no utilice la historia solo como “fondo”: la historia, en todos los videojuegos del marco teórico de la presente investigación (a excepción de algunos que entran en la categoría de edutainment), no utilizan a la historia como piedra angular de su jugabilidad, y por lo tanto, existe el riesgo de que el jugador se distraiga con otros detalles de la jugabilidad, y no preste atención a los detalles importantes de la historia que le están contando.

²Según recomendaciones australianas aprobadas por la APA

No existe videojuego que, mediante su jugabilidad, motive al jugador a prestar atención a los detalles correctos, de una forma divertida y basada en multimedia no-textual, sin recurrir al abuso de lectura de textos largos, o al abuso de cuestionarios de respuesta múltiple.

El jugador podría ganar el juego sin que sea necesario que entienda la historia que está sucediendo, únicamente siguiendo las reglas del juego, explotando únicamente sus habilidades motrices y estratégicas.

¿Será que un juego educativo de historia que implemente prestarle atención a los detalles correctos, a una jugabilidad con acción y agilidad, causará que los jugadores no puedan ganar el juego sin haber aprendido historia y se sientan motivados de hacerlo? Diseñar e implementar un videojuego de historia con una jugabilidad basada en acción y agilidad, que sea imposible de ganar, a menos que el jugador le preste atención, y comprenda ciertos detalles determinados de la historia, sin obligarle a permanecer pasivo durante demasiado tiempo.

3. Objetivos Específicos



- Encontrar una metodología de producción de videojuegos que se adecúe al desarrollo de videojuegos creados por un solo gamedev.
- Crear los documentos de diseño del videojuego.
- Fusionar el aprendizaje de historia con la jugabilidad, durante el diseño del juego.
- Producir el videojuego en base a los documentos de diseño creados.
- Publicar el juego en medios masivos del internet.

4. Alcances

En esta fase, y hasta finales del año 2015, el juego solamente estará completado hasta su demo.

4.1. Niveles


Los niveles del juego no serán secuenciales. El jugador puede acceder a ellos mediante un menú en el que se muestran personajes relacionados con la guerra del pacífico. Cada uno de estos personajes, tendrá asociada una *línea del tiempo*, que contendrá un conjunto de niveles que el jugador puede jugar, sin que sea necesario pasarlos en ningún orden en especial, pero que están colocados en un orden descendente, de acuerdo al tiempo en el que se desarrollaron los sucesos.


Sólamente se desarrollarán 2 niveles importantes, pero si el tiempo alcanza, se desarrollará uno más. A continuación, se describirán algunos niveles que no serán implementados, los niveles que sí serán implementados serán descritos con mayor detalle, y marcados con la etiqueta , que indica que la característica es indispensable, el resto, se marcará con la etiqueta , que indica que la característica se desarrollará sólomente si alcanza el tiempo.

Los niveles se listan a continuación:

Protagonista: Ladislao Cabrera, coronel de artillería que dirigió la defenza del puente Tópater, a pesar que su ejército de 128 voluntarios era superado en número y armas por el ejército enemigo.

Este personaje cuenta con tres niveles, de los cuales uno es indispensable:

Desventaja : Un nivel en el que se muestra el momento en el que Ladislao fue llamado a comandar la defenza del puente Tópater. En este nivel se muestra que la fuerza de defenza estaba en desventaja, y que solo contaban con 128 hombres, y algunas armas, algunas en buen estado, y otras improvisadas.

Estrategia : De cómo se planificó la defenza del puente Topater junto con Eduardo Abaroa.

4.1.1. Batalla del Topater








Este nivel es indispensable.

El jugador toma el control de Omar Quispe, un minero que trabajaba en las minas de Eduardo Abaroa.

Antecedentes La resistencia ya se enteró que el ejército chileno va a atravesar el desierto de Atacama para atacar el oasis de Calama. La batalla ya ha sido planificada, y la fuerza en la que Omar pelea se encuentra defendiendo uno de los puentes destruidos, bajo el mando del coronel Ladislao Cabrera.

Mecanismos

- **Energía**  (o stámina) Es un atributo del jugador.
 - Cada acción cuesta una cantidad de energía. 
 - Si el jugador no tiene suficiente energía, no puede realizar la acción 
 - Dentro de más baja sea la energía del jugador, todos los *cooldowns* se incrementarán

Cada acción cuesta una cantidad de energía, y si el jugador no tiene la energía suficiente, no podrá desempeñar acciones, a menos que recupere su energía. , dentro de más baja sea la energía del jugador, más lento caminará y correrá .

5. Justificación

5.1. Social

Videojuegos educativos Pese a que el internet ha cambiado la forma en la que los nativos digitales aprenden, el sistema educativo en las escuelas públicas de Bolivia, no ha cambiado mucho. Como país, necesitamos adaptarnos a este cambio, y ofrecer una formación educativa de calidad para todos: para los bolivianos principalmente, y para el mundo entero en la medida de lo posible.

Algunas ramas de la ciencia, son lo suficientemente generales como para adaptarse a cualquier estudiante, por ejemplo, un estudiante boliviano interesado en matemática, podría seguir un curso de matemática del OpenCourseWare del MIT, sin tener ningún otro obstáculo del que tendría un estudiante francés, a excepción, tal vez, del lenguaje.

Pero la historia no es una de estas ramas; cada país tiene su propia riqueza cultural e histórica, la historia de cada país, muchas veces, es conocida solamente

por los historiadores de ese país, por eso es importante que cada país difunda su propia historia.

Por eso es importante liberar la información sobre la historia de Bolivia, y difundirla, no solo entre los bolivianos, si no que a nivel mundial. Para que el mundo entero conozca nuestra historia, necesitamos, al menos, estas dos cosas:

- Hacerlo mediante medios a los que gran parte del mundo pueda acceder.
- Hacer del proceso de aprendizaje algo ameno y divertido.

Demanda marítima La historia sobre la guerra del pacífico atrajo la atención de todo el mundo, a finales del siglo XVIII, pero toda esta atención internacional se ha ido perdiendo con el paso del tiempo. Desde entonces, Bolivia es uno de los dos países de América que no cuentan con salida al mar. El 24 de abril de 2013, el gobierno de Bolivia inició formalmente una demanda marítima, sin embargo, la demanda no es internacionalmente conocida a nivel popular.

5.2. Académica

Creación de videojuegos en la UMSA Pese a que el interés por los videojuegos en Bolivia crece cada vez más, en la UMSA no existe apoyo formal para el desarrollo de videojuegos. Los videojuegos cuentan historias, y son capaces de transmitir sentimientos tan complejos, que serían extremadamente difíciles de transmitir mediante cualquier otro medio; además, pueden transmitir experiencias y conocimiento sólido mediante un aprendizaje significativo, y despertar el interés de una gran comunidad, al rededor de un tema.

Hacer que un videojuego transmita toda esta riqueza es difícil de conseguir sin un estudio sólido y formal del diseño de videojuegos, y un videojuego bien diseñado, es imposible de implementar sin el conocimiento sólido de una metodología especializada en el desarrollo de videojuegos.

6. Plataforma

El juego se realizará en javascript, usando la librería phaser(Davey, s.f.). El juego ha sido probado para Google Chrome 45.0, y no se garantiza su funcionamiento en otros navegadores.

Para que el juego funcione, todos sus archivos deberán ser puestos en un servidor, se recomienda LAMP, esto debido a que algunos recursos, como los tilemaps, son pedidos por phaser mediante un crossrequest. Una vez en el servidor, deberá accederse a su archivo index.html

7. Metodología

La creación de un videojuego implica conocimientos muy variados. Los equipos de desarrollo suelen ser heterogéneos, y la creatividad juega un papel tan importante, que muchos productores procuran proveer a sus creadores la mayor libertad creativa posible.

Todos estos factores, sumados, hacen del desarrollo de videojuegos algo peculiar. Los ciclos de vida tradicionales, como el waterfall no tienen lugar en su desarrollo (Bates y Bob, 2004), y pese a que las metodologías ágiles parecen encajar bien en este proceso, principalmente el scrum (Chandler y Maxwell, 2009), Paul Miller, un “veterano” del desarrollo de videojuegos, critica fuertemente esta metodología al ser aplicada al desarrollo de videojuegos. (Miller, 2008)

Pero el desarrollo de videojuegos es solamente una de las etapas para la creación de un videojuego. Aunque existen muchas metodologías para la creación de videojuegos, varios autores (Moore, E, Novak, y Jeannine, 2010) (Bates y Bob, 2004) concuerdan en que todas cuentan con las siguientes etapas:

7.1. Concepción

La creación de la idea del videojuego es, en sí, un proceso bastante complicado. Una jugabilidad cautivante es difícil de crear. Mike Stout, gamedev veterano, publicó la metodología que él usó para la concepción skylanders.

Aunque existen muchas metodologías, todas ellas están documentadas en libros que están detrás de murallas de pago, por esta razón, este proyecto utilizará la metodología trinity.

7.1.1. Trinity

Es una metodología de diseño de videojuegos ³ creada por Mike Stout, y, heroicamente compartida en Gamasutra ⁴

³Esto es aplicable durante la fase a la que Moore et. al. y Bates denominan: *fase de pre-producción*.

⁴Una página que contiene diversos artículos sobre videojuegos, y su desarrollo en general.

Principio 1: El juego pregunta, el jugador responde Como diseñador de juegos, tu trabajo (en un sentido muy práctico) es hacerle preguntas al jugador. El trabajo de los jugadores es responder a esas preguntas utilizando las herramientas que tú les das. (Stout, 2015)

Este enfoque podría conducir al diseñador a cometer el error de, formular una pregunta, y restringir al jugador únicamente a dos posibles tipos de respuesta: la correcta y la incorrecta, bueno y malo. El planteamiento de este tipo de preguntas no están acorde a esta metodología, ya que obliga al jugador a buscar siempre la respuesta correcta, en vez de que cada pregunta abra un mundo completo de posibilidades con cada una de las posibles respuestas. Debido a que esta dicotomía es muy difícil de evitar, a veces es necesario disfrazarla con el uso de herramientas variadas.

Preguntas Consiste en una pregunta que “el juego” le hace al jugador. Podría no ser obvia para el jugador, pero eventualmente, el jugador debería darse cuenta de lo que el juego le está “preguntando”.

El hecho de que el jugador se de cuenta de la pregunta y la rapidez con la que se de cuenta, dependerá de la habilidad del diseñador. El hecho de que el jugador plantee una respuesta correcta, dependerá de la dificultad que el diseñador ponga en el juego.

Meta preguntas Es una pregunta incompleta que integra el contexto bajo el que se pregunta, pero para ser formulada, debe ser añadida a otra pregunta.

Las meta preguntas permiten formular preguntas de una forma más específica, y añadir así una jugabilidad variada y entretenida.

Por ejemplo, si el jugador tiene varias armas, y se encuentra con un enemigo, el enemigo podría formular esta pregunta: “¿Cómo vas a pasar hacia donde estoy bloqueándote el camino?”, pero más específicamente, la pregunta podría ser: “¿Cómo vas a derrotarme?”, pero incluso más específica es la pregunta: “¿Cuál es la forma correcta de atacarme?”. Cada una de estas preguntas incorpora el contexto bajo el que el jugador está jugando, en la primera pregunta, por ejemplo, el enemigo podría estar bloqueando el camino a la salida de alguna cueva que está por derrumbarse. En la segunda, el enemigo podría tener la llave de la puerta de la prisión en la que el protagonista está preso, y en la última, el enemigo podría ser invulnerable a cierto tipo de ataques, o incluso, el jugador podría necesitar un tipo diferente de arma, en función al terreno que rodea al enemigo.

Respuestas El jugador responde a las preguntas planteadas utilizando las herramientas que le han sido dadas. Es importante diseñar estas herramientas, de tal forma, que sean reusables; sobresaturar al jugador con muchas herramientas, introduciéndolas todas de golpe, podría llevar a que el jugador se canse de explorarlas y pierda interés. También podría ocurrir que el diseñador pierda control sobre las preguntas que puede plantear, ya que el exceso de posibilidades sin explorar causarían que las preguntas no sean lo suficientemente obvias para el jugador.

Pero ninguno de los extremos es recomendable. Si las herramientas que el jugador posee no son lo suficientemente diversas, el juego podría tornarse aburrido, ya que el jugador se vería obligado a seguir una línea que ha sido estrechamente trazada por el diseñador.

7.2. Pre-producción

También conocida como fase de diseño, durante esta etapa, se diseña el juego, enfocándose en la idea y el concepto. El resultado de esta fase consiste en varios documentos iniciales de diseño, toda esta documentación describirá el itinerario de la implementación del videojuego, el diseño, el concepto, la jugabilidad, la historia y todo lo que define al juego, esta documentación también debería ser fácil de entender y lo menos ambigua posible.

A continuación, se listan los documentos iniciales resultantes de esta etapa.

7.2.1. High Concept

Una descripción corta del juego, hecha con la idea de “vender” la idea del juego a un productor, de forma que el juego pueda ser implementado con un buen presupuesto.

Debe tener de 2 a 3 páginas, y empezar con un párrafo que consista en 2 líneas o menos. El resto de la primera página, consiste en una lista de características clave, cada una, descrita en, no más de 3 oraciones. Estas características, deben ser escritas con la idea de describir cómo se verá y sentirá el juego, los detalles de cómo lograr esto no competen a esta sección. Las características más importantes del juego, deben resaltarse en negrilla, de forma que cautiven la atención del lector.

En la segunda página, va un panorama general, que contiene:

- **Motivación del jugador** Aquí se describe qué motivación tienen los jugadores para jugarlo.
- **Género**
- **Licensia**
- **Clientes** Aquí se describe qué tipo de persona estaría dispuesta a comprar el juego.
- **Competencia**
- **Hardware** Para qué hardware está hecho el juego?
- **Metas de diseño** Aquí se listan las experiencias que se espera que el juego deje en las personas.

Y por último, y si es necesario, se puede añadir un poco de información sobre el juego en sí, sobre sus personajes más únicos, o sobre sus mecanismos más alucinantes. Esto, con el objetivo de llamar la atención del lector, y hacer que se enamore del juego.(Adams, 2008)

Debido a la naturaleza del proyecto, el financiamiento no es una “pieza clave”, por lo tanto, no se le dará mucha importancia a la realización de este documento.

7.2.2. Pitch

Es un documento que describe todos los aspectos del juego que pueden hacerlo rentable. Todas las estrategias de venta, y lo que se puede hacer para ganar dinero con el videojuego van aquí.

Debido a la naturaleza del proyecto, el financiamiento no es una “pieza clave”, por lo tanto, no se le dará mucha importancia a la realización de este documento.

7.2.3. Concept

Es un documento en el que se describe el juego con más detalle. Es como el high concept, pero mucho más detallado. Antes de que el juego sea aprobado, programadores y artistas, trabajan juntos para producir este documento.

Los programadores comienzan a trabajar de forma rápida y un tanto despreocupada, para generar un prototipo. Los artistas comienzan a desarrollar el concepto de arte, y bosquejos que podrían servir como la piedra inicial de características reales del juego.

7.2.4. Game Design Document (GDD)

El reto es crear un documento de diseño que le permita a tu proyecto tolerar adaptaciones inesperadas, sin perder la integridad de su dirección original, ni sus metas.(Freeman, 1997)

Birkhead concuerda en que el documento de diseño debería ser lo suficientemente claro como para que, al finalizar de leerlo, el lector tenga totalmente clara la idea del juego.(Birkhead, 2011)

El GDD orienta a todos los que están involucrados en el proceso de creación del videojuego, de tal forma que todos tengan muy en claro lo que están haciendo, y aunque es imposible que todos tengan la misma idea, es importante que el margen de error producido por las indeterminaciones sea mínimo en las características más importantes del juego.

Tzvi Freeman cree que un buen GDD debe tener estas 10 características clave: (Freeman, 1997)

1. **Describe el alma del juego** El equipo que crea al videojuego está compuesto por humanos, creativos, testarudos, fascinantes e impredecibles humanos que aportarán sus ideas y creatividad al juego. Por eso es importante describir en este documento, no solo las características del juego, sino que también, la forma en la que debería “sentirse” al jugarlo, de esa forma, todos los que están ayudando en hacer al juego realidad, sabrán si sus sugerencias son oportunas o no.
2. **Legible**
3. **Priorizado** Esto también ayuda a que los integrantes del equipo sepan qué características del juego deben ir sí o sí, y cuáles deben obedecerse al pie de la letra, sin lugar a cambios que tergiversen su sentido. Pero también hay otras características más flexibles, que no necesariamente deben quedar como se escriben en el documento. Estas características tal vez podrían causar una mejora al juego si se cambian por otras ideas.

Una buena lista de tags podría ser:(Freeman, 1997)

- Indispensable
- Importante
- Si es posible
- Rechazado

4. **Detallado** Dentro de más detalles tenga una especificación, más difícil es que se mal entienda. Por eso es importante poner ejemplos, usar una jerga clara, y si ser redundante ayuda a que no hayan ambigüedades, entonces es necesario.
5. **Prototipos** Cuando sea necesario Cuando una característica se vuelve muy difícil de entender, a veces es necesario hacer un prototipo para que se entienda. Pero no siempre es buena idea abusar de los prototipos, hay cosas que quedan más claras en palabras.
6. **Describe el cómo y el qué (no-técnico)** A veces, la forma en la que se consigue algo, afecta al resultado final. Cuando esto ocurra, es necesario especificar cómo va a lograrse un resultado final. No debe confundirse con detalles técnicos, por ejemplo, con qué algoritmos se va a programar cierto detalle del juego, o con qué herramienta debería dibujarse; aplica únicamente al diseño. Cuando en el GDD se especifique qué es lo que se quiere que el jugador sienta en cierta parte del juego, también debería especificarse exactamente cómo causar que el jugador se sienta de esa forma.
7. **Provee alternativas** A veces, cuando una característica requiere muchos recursos, necesita alternativas más baratas. Esto podría implicar mucho tiempo y muchos párrafos dedicados al GDD, si se prioriza correctamente, solo será necesario añadir alternativas a las características indispensables.
8. **Tiene vida propia** Debido a que el equipo de desarrollo está compuesto por personas, las personas tienden a ser creativas, y a proponer cambios. De hecho, es bastante probable que el creador del juego, y escritor del GDD también aporten nuevos cambios al juego, luego de que el GDD haya sido creado. Por eso es importante que el GDD sea, hasta cierto punto, flexible. Esta flexibilidad no debe perjudicar a la idea principal del juego, no debe salirse de control, ni cambiar demasiado el rumbo que el juego estaba llevando.
9. **Nada ambiguo** Nadie debería poder decir “lo hice de esa forma porque no estaba en el GDD”. Buena presentación Si trabajas en equipo, y si tu equipo no te conoce, es probable que no lean el GDD si no se ve “lindo”. Si no se ve lo suficientemente presentable, podrían creer que no invertiste el suficiente esfuerzo en él, y podría terminar siendo ignorado.

Al ser este proyecto un one-man-project, aunque se le dará una gran importancia para definir al juego, y no comenzar a desarrollar a ciegas, no se le dará tanta importancia a desambiguar las características del juego.

7.2.5. Prototype

Cuando algo no queda lo suficientemente claro en el GDD, realizar prototipos es importante. Pero también es importante prototipar para probar características experimentales del juego.

Debido al corto tiempo, y a que el proyecto se llevará a cabo por la creadora del juego, el prototipado en la fase de producción no se usará más que en maquetas.

7.3. Producción

Esta es la etapa principal del desarrollo de un juego, y todo el equipo trabaja de forma concurrente para desarrollar el juego.

Pese a que Moore y Bates concuerdan en varios pasos que deben seguirse en esta etapa (Moore y cols., 2010) (Bates y Bob, 2004), en este proyecto se utilizará la metodología scrum, con algunos ajustes para que funcione adecuadamente en la producción de videojuegos. Bonin le llama a este tipo de scrum: scrumBut (Bonin, 2014), debido a que es scrum, “pero” con modificaciones. Incorporaré algunas de las ideas de Bonin a la metodología que usaré.

Algunos gamedevs, suelen añadir características de waterfall al scrum, sin embargo, Keith Clinton cree que no es aconsejable aplicar waterfall al desarrollo de un juego, él cree que es mejor añadirle características de lean para solucionar algunos problemas que no pueden ser afrontados correctamente mediante scrum puro. (Keith, 2008) También incorporaré algunas de sus ideas a la metodología.

Además de estos ajustes, también aplicaré algunos en base a mis experiencias de desarrollo de videojuegos. Esto es necesario porque el equipo de desarrollo consiste en una sola persona, y hay muy poca documentación sobre metodologías de desarrollo de videojuegos para una sola persona.

7.3.1. Causas para crear una nueva metodología

No existe una metodología para producir videojuegos mediante un solo *gamedev*. Los juegos desarrollados por una sola persona suelen tomar más de un año, o ser extremadamente sencillos.

Muchos de los juegos creados por un solo *gamedev* son abandonados, por lo que este proyecto ya tiene una alta probabilidad de fracasar.

Es imposible usar una metodología ya existente, debido a que todas estas metodologías han sido creadas y ajustadas por *gamedevs* expertos, y ellos trabajan en proyectos grandes, en grandes compañías con grandes y multidisciplinarios equipos. Ellos no tienen que preocuparse de que el desarrollador no quiera programar, porque tiene un ataque de creatividad y prefiere dibujar un *sprite*. Ellos se preocupan porque todos los miembros se mantengan en sus roles y terminen el trabajo que se les ha asignado.

Si algún miembro tiene un ataque de creatividad, y quiere realizar una tarea que no debería desempeñar, tendrá que esperar a que el trabajo que se le ha asignado esté terminado, o a su tiempo libre, porque, después de todo, es muy poco probable que su idea sea mejor que la de un profesional en esa área.

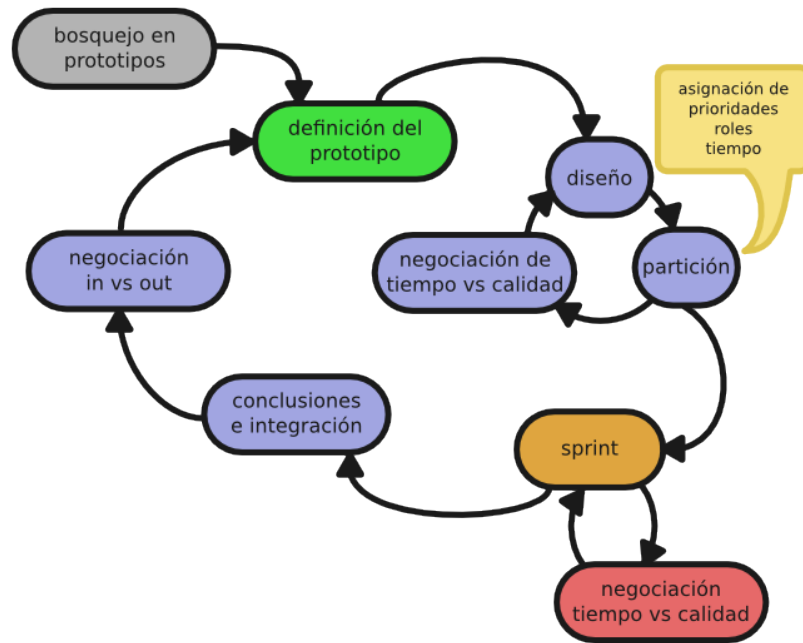
Cuando un *gamedev* trabaja solo, debe aprovechar al máximo esos ataques de creatividad, pero, si tiene un límite de tiempo y presupuesto, no debe invertir más tiempo del “necesario”, debe medir su tiempo y esfuerzo de acuerdo a la curva de satisfacción del jugador vs costo.

A esta metodología yo le llamo *lone-ranger-methodology*, o en español: metodología del llanero solitario, ya que se aprovechan las mejores características de metodologías ágiles como *lean* y *scrum*, y las adapta a un trabajo en el que un solo *gamedev* hace todo el trabajo.

7.3.2. La metodología del Llanero Solitario

Es una metodología de desarrollo de videojuegos, que aprovecha las mejores características de metodologías ágiles como *lean* y *scrum*, y las adapta al desarrollo de videojuegos llevado a cabo por un solo *gamedev*

Ciclo de vida



Bosquejo en prototipos Se define un plan de prototipos. El objetivo es tener un panorama general de la forma en la que el prototipado nos llevará a finalizar el juego. Este plan no debe ser demasiado detallado.

Definición del prototipo Se define a pleno detalle el prototipo que se espera tener funcionando al finalizar el ciclo. Es importante que los prototipos no sean ambiciosos, el flujo de entrega de los prototipos debe permanecer constante.

Diseño Se diseña, a grandes razgos el código del prototipo, y se describe el trabajo que se tiene que hacer mediante modelos. La idea de esta fase, es que el llanero solitario tenga una vaga idea de cómo va a hacer el prototipo.

Partición El modelo se parte en pequeñas particiones. Cada partición tiene asignado un rol, una prioridad dentro de una escala del 1 al 10, y por último, se calcula el tiempo estimado que cada partición tomará en finalizarse.

Negociación de tiempo vs calidad En base al tiempo que tomará el sprint en finalizarse, se revisan las particiones. Si el tiempo es demasiado grande, se deben negociar características, sacrificando calidad para que el tiempo de realización sea más corto. Si existieron negociaciones, se revisa el diseño y se particionan las partes afectadas de nuevo.

Sprint Se implementa cada una de las particiones hechas en la fase anterior.

El orden tiene muy poca importancia, el llanero solitario puede elegir las particiones en base a lo que le parezca más divertido hacer, también puede cambiar de tarea, si de pronto quiere trabajar en otra cosa.

Es importante que tenga en cuenta que las tareas marcadas con una prioridad baja, deben finalizarse lo más rápido posible, de tal forma que, si surge algún inconveniente en las tareas con prioridades altas, pueda ser compensado con el tiempo sobrante de las tareas con prioridad baja.

Cuando el llanero solitario finaliza una de las particiones, debe anotar el saldo de tiempo con el que la terminó.

Negociación tiempo vs calidad Si el llanero solitario siente que está invirtiendo demasiado tiempo en una partición, debería negociar tiempo vs calidad para completar la característica. Si el tiempo que está invirtiendo en esa partición excede del 125 % del tiempo destinado, deberá abandonarla y terminar todas las particiones que no dependan imprescindiblemente de esa partición.

Conclusiones e integración Si el prototipo construido no está integrado al resto de prototipos, se integra en esta fase. También se documenta y limpia el código, y se pone en su sitio todos los assets que podrían estar fuera de su lugar.

Se realiza una lista de las particiones que no se pudieron finalizar, y las causas por las que no se pudo hacer descritas brevemente, esto se documenta en la lista de undone. También se realiza una lista de las nuevas ideas que hayan surgido, y de las características inconclusas de las particiones. Las ideas se documentan en la lista de ideas, y las características inconclusas, en la lista de good to have.

Negociación in vs out Si existen particiones sin finalizar en la lista de undone, cada una de las particiones deberá ser negociada. El llanero solitario debe determinar qué tan importante es esa partición, y si vale la pena ponerla de nuevo en el siguiente prototipo. Esto se determina en base al tiempo y esfuerzo que tomaría finalizar la partición, y a la importancia de finalizarla.

8. Glosario

- **Acción** Ciertas actividades que el personaje que realiza el jugador realiza y tienen algún efecto en su alrededor.
- **Atributo** Cantidad con un límite máximo que no puede ser excedido, y que no puede ser negativo.
- **Cooldown** Lapso de tiempo que debe esperarse antes de volver a realizar una acción.