

Proyecto X-Kating

Grupo WASTED HORCHATA

"DOCUMENTO DE ESPECIFICACIÓN"

Hito: 0
Fecha entrega: 18-10-2017
Versión: 1

Componentes:

- Pablo López Iborra
- Laura Hernández Rielo
- Adrián Francés Lillo
- Luis González Aracil
- Alexei Jilinskiy

Contenido

Introducción	2
Propósito	2
Descripción general	2
Ámbito del sistema (Contexto e historia).	2
Funcionalidades generales.	2
Características de los personajes.	3
Escenarios.	3
Requisitos (suposiciones y dependencias).	4
Restricciones.	4
Requisitos futuros.	4
Objetos:	4
Requerimientos específicos	5
Apéndices	6

1. Introducción

1.1. Propósito

Este documento especifica la elaboración del videojuego de carreras “Wasted Racing” y sus requisitos para los clientes “Profesores del Grado Ingeniería Multimedia”. Establece en los diferentes puntos la descripción general del proyecto, los requisitos funcionales detallados y las referencias necesarias para realizarlo.

2. Descripción general

2.1. Ámbito del sistema (Contexto e historia).

Nuestro videojuego, cuyo nombre de proyecto es “Wasted Racing”, a falta de un nombre definitivo para el mismo, es un videojuego del género carreras/conducción que se basa fundamentalmente en el *Mario Kart* de *Nintendo* para la plataforma SNES. Además, también toma de referencia juegos como el *Jet Set Radio* de *SEGA* o el *Crash Team Racing* de *Naughty Dog*.

A pesar de que nuestro juego está principalmente basado a nivel mecánico en el *Mario Kart*, se caracterizará por ser un juego de patinaje y skate de carreras, en lugar de ser un juego de coches. Por lo demás, en principio, tendrá funcionalidades similares a las del juego de *Nintendo*, con recogida y uso de objetos, derrapes, etc. También nos planteamos, aunque de forma opcional, añadir mecánicas relacionadas con el mundo del patinaje o el skate.

Nuestra meta sería por tanto crear un juego competitivo de carreras de patinaje, centrándonos en conseguir una obra divertida, colorida, con un estilo retro futurista, y lo más completa posible.

Historia y contexto del juego: Año 2500 - La tierra es un lugar triste y aburrido, donde ya no merece la pena vivir... excepto porque, desde hace unos años, un grupo de consumidores de setas alucinógenas empezó a acumular fieles, con apuestas en circuitos de skate y patinaje tan rocambolescos y peligrosos que a menudo causaban la muerte de sus participantes. Con el paso de los años, el X-kating, tal y como se le conoce popularmente, se ha convertido en un deporte ilegal altamente pagado, y al que cientos de personas dedican su vida. Existen ya numerosas tribus con decenas de adeptos que compiten y apuestan, en un intento desesperado por entretener sus vida en un mundo devastado.

2.2. Funcionalidades generales.

- 2.2.1. Giro: el vehículo debe poder girar en el eje horizontal.
- 2.2.2. Aceleración del vehículo: el vehículo aumenta de velocidad.
- 2.2.3. Frenado del vehículo: el vehículo desacelera rápidamente hasta frenar, y puede ir marcha atrás.
- 2.2.4. Derrape del vehículo: el vehículo derrapa y puede obtener un aumento de velocidad puntual.
- 2.2.5. Recogida de objetos: en zonas puntuales se recogen power-ups.

- 2.2.6. Uso de objetos: todos los objetos tienen un efecto sobre el jugador, resto de jugadores o en el mapa.
 - 2.2.6.1.1. Objeto obstáculo: colocados en un punto del mapa, al colisionar con ellos los vehículos pierden velocidad.
 - 2.2.6.1.2. Aumentador de velocidad: el vehículo gana una aceleración puntual que disminuye a lo largo del tiempo.
 - 2.2.6.1.3. Invulnerabilidad: el jugador es invulnerable durante un tiempo limitado a ataques de enemigos y obstáculos básicos.
 - 2.2.6.1.4. Proyectil teledirigido al siguiente jugador: un proyectil que impacta con el siguiente jugador.
- 2.2.7. Respawn del jugador al caer de la zona de juego: el personaje es restaurado en el centro del mapa para seguir corriendo.
- 2.2.8. Enemigos con IA que compiten contra el jugador: los enemigos disponen de una IA si no son controlados, que se comporta como si fuera un jugador más.
- 2.2.9. Modo multijugador hasta 4 jugadores: Pueden jugar hasta 4 jugadores humanos online conectados en 4 máquinas distintas.
- 2.2.10. Menú principal y selección de modo, pista y vehículo: Interfaz que nos permite seleccionar un modo de juego, pista en la que correr y vehículo a utilizar; además de otras opciones futuras y salir del juego.

2.3. Características de los personajes.

Habrán 4 personajes distintos entre los que el jugador elegirá uno que será su avatar y se enfrentará a los 3 restantes. En caso de añadirse más jugadores, irían eligiendo a su vez el personaje con el que desearan correr.

Los personajes elegibles serán líderes de bandas urbanas, con uniformes de patinador personalizados y patines, que consistirán en el vehículo con el que corren las carreras.

Todos los personajes no jugables se comportarán igual, como competidores en la carrera, y sus decisiones de IA serán independientes en principio del tipo de personaje que elija el jugador. Todos ellos podrán realizar las mismas acciones que el jugador.

2.4. Escenarios.

El escenario consistirá en un circuito cerrado que los personajes recorrerán durante la carrera. Serán un entorno urbano retrofuturista y, al igual que los personajes, con estética *cartoon*.

El único entorno jugable será la calle o carretera por la que circulará la carrera. Los objetos que podrán obtener los jugadores se distribuirán a lo largo de la carretera, serán cajas que se encontrarán durante la misma que les otorgarán aleatoriamente los distintos power-ups o armas.

En principio tendremos un solo escenario elaborado, con posibilidad de añadir algún escenario básico extra en el futuro.

2.5. Restricciones.

2.5.1. Lenguajes de programación utilizados: C++: versión estandarizada actual del lenguaje de alto nivel.

2.5.2. Políticas de empresa: trabajo duro y constante, de forma organizada y colaborativa.

2.5.3. Personal: cinco empleados con tareas multidisciplinarias.

2.5.4. Tiempo:

2.5.4.1. 9 meses de desarrollo. Fecha límite: 23/5/2018.

2.5.4.2. Al menos 900 horas de trabajo invertidas en el proyecto por persona con un total de 4500 horas de proyecto.

2.5.5. Presupuesto: Contabilizamos el proyecto en función de “créditos”, que contabilizan por horas de trabajo. Cada empleado aporta 36 créditos al proyecto, siendo en total entre todo 180 créditos como máximo.

2.5.6. Planificación y comunicación: metodología SCRUM adaptada. Trabajaremos con una metodología ágil, manteniendo reuniones diarias y un sistema de control continuo. El software que utilizaremos para esto serán:

2.5.6.1. Microsoft Project: Se usará para la gestión a largo y corto plazo de la planificación global del proyecto.

2.5.6.2. Toggle: Herramienta para la medición y contabilización del tiempo de trabajo diario y/o por tareas.

2.5.6.3. Github: Control de versiones y auditoría del trabajo.

2.5.7. Limitaciones de hardware: el producto será elaborado para ser jugado en un ordenador personal de gama media-alta.

2.5.8. Requisitos de habilidad: conocimientos previos de programación con lenguajes de alto nivel, con un dominio medio/avanzado. Conocimientos de planificación y habilidades de organización y estructuración de ideas y contenidos. Capacidades artísticas y de desarrollo de personajes y entornos.

2.5.9. Operaciones paralelas:

2.5.9.1. El desarrollo de la imagen de la empresa, el marketing y el sonido de la aplicación son tareas que suceden en paralelo a lo largo del proyecto.

2.5.9.2. El desarrollo de las mecánicas del juego y la IA se suceden simultáneamente durante el primer Hito.

2.5.9.3. El desarrollo del motor gráfico, el motor de red, las interfaces e implementación de la aplicación se suceden durante el resto de etapas.

2.6. Requisitos (suposiciones y dependencias).

2.6.1. Sistemas operativos:

2.6.1.1. Linux Ubuntu 16.04 LTS - Sistema para el desarrollo

2.6.1.2. Windows 7 / 10 - Se realizará portabilidad

2.6.2. Librerías

2.6.2.1. Librería OpenGL 4.x: para el desarrollo del motor gráfico.

2.6.2.2. CEGUI: para el desarrollo de la interfaz.

2.6.2.3. RakNet 4: para el desarrollo del servidor web que dé soporte al multijugador en tiempo real.

2.6.2.4. FMOD: para el correcto funcionamiento de la librería de sonido

2.6.2.5. Irrlicht: motor inicial, se usará para el desarrollo hasta tener uno propio.

2.7. Requisitos futuros.

Los requisitos futuros se refieren a las mecánicas, sistemas de juego, objetos, mapas, etc.. qué van a ser optativos, de manera que según el tiempo sobrante durante el proyecto, se irán añadiendo.

2.7.1. Mecánicas del juego:

2.7.1.1. Salto del vehículo: El vehículo realizará un pequeño salto de manera que podrá evitar objetos que estén en la pista o coger ciertas partes con velocidad extra.

2.7.1.2. Agacharse: Permitirá al jugador agacharse, de manera que como el salto del vehículo, le permitirá esquivar ciertos obstáculos de la pista.

2.7.1.3. Skater: Además de la figura del patinador, podrá existir un personaje que controle una tabla de skate. Las mecánicas de este personaje serán similares a las del patinador estándar, pudiendo tener, no obstante, valores distintos o algún control extra.

2.7.1.4. Mecánicas de patinaje: Conjunto de mecánicas relacionadas directamente con el mundo del patinaje y que son imposibles de realizar con cualquier otro vehículo, tales como deslizamiento por una barra o saltos con trucos.

2.7.2. Objetos:

- 2.7.2.1. Proyectil teledirigido al primer jugador: Proyectil que irá de forma automática a por el jugador que esté en cabeza en la carrera, de manera que cuando lo golpee lo frenará por completo durante un tiempo.
- 2.7.2.2. Reductor de visibilidad de pantalla: Objeto que reducirá la visibilidad en pantalla de los jugadores que estén posicionados por delante del que lo lanza. Cuanto más cerca del primer puesto esté cada jugador, más duración tendrá este ítem sobre ellos.
- 2.7.2.3. Proyectil de ataque recto: Proyectil que se lanza en línea recta y detiene a los vehículos enemigos al impactar con ellos, pudiendo rebotar incluso en el propio lanzador.
- 2.7.2.4. Acelerón: Durante unos segundos, el personaje pasará a controlarse solo, será invulnerable, acelerará ampliamente, derribando a todos los enemigos a su paso.

2.7.3. Funcionalidades mejoradas:

- 2.7.3.1. Niveles de dificultad en la IA: mejorar la IA para que disponga de niveles de dificultad seleccionables. Esto dificulta la experiencia de juego y da un mayor desafío al jugador.
- 2.7.3.2. Aumento de jugadores simultáneos en multijugador: con un diseño más robusto del motor de red, poder permitir un número mayor de jugadores humanos en una misma carrera.
- 2.7.3.3. Juego en local: Poder jugar en pantalla compartida al menos dos jugadores en un mismo dispositivo.

2.7.4. Escenarios extra:

- 2.7.4.1. Diseño de nuevos escenarios: Inicialmente se propone un único escenario. Debería haber al menos dos escenarios jugables.

3. Requerimientos específicos

3.1. Requerimientos funcionales.

3.1.1. Mecánicas.

3.1.1.1. De los jugadores

- 3.1.1.1.1. Giro: Altera el movimiento del personaje horizontalmente, iniciando el giro del vehículo hacia la izquierda o derecha.
- 3.1.1.1.2. Aceleración del vehículo: El vehículo aumenta de velocidad hasta llegar a un valor máximo.

3.1.1.1.3. Frenado del vehículo: El vehículo desacelera hasta quedarse quieto. Si se sigue pulsando el botón, da marcha atrás aumentando de velocidad de retroceso hasta llegar a un máximo.

3.1.1.1.4. Derrape del vehículo: El vehículo derrapa en la dirección en la que estaba girando, acentuando este giro. Si se hace lo suficiente gana un aumento de aceleración que se aplica en terminar de derrapar.

3.1.1.1.5. Recogida de objetos: Los jugadores pueden recoger objetos del mapa de zonas específicas para ello. Esto lo realizarán cogiendo un contenedor de objeto, el cual les asignará de forma semi-aleatoria, dependiendo de su posición en la carrera y del azar, un objeto u otro.

3.1.1.2. De objetos y NPC's

3.1.1.2.1. Objeto obstáculo: Ítem que se coloca en la pista de manera que si alguno de los jugadores colisiona con este objeto, tendrá una penalización en su velocidad.

3.1.1.2.2. Acelerón: Objeto que al utilizarse servirá de acelerador, aumentando puntualmente la velocidad.

3.1.1.2.3. Invulnerabilidad: Ítem que proporciona al jugador unos segundos en los que su velocidad aumentará y, además, no puede ser afectado por colisiones de objetos, enemigos u obstáculos básicos del circuito.

3.1.1.2.4. Proyectil teledirigido al siguiente jugador: Objeto que irá de forma automática a por el jugador más próximo delante nuestra, de manera que cuando lo golpee lo frenará por completo.

3.1.1.3. Del entorno

3.1.1.3.1. Plataforma aceleradora: Plataforma presente en el suelo, en secciones concretas o determinadas del mapa, en las cuales los personajes, al pasar por encima, recibirán un acelerón temporal, similar a si utilizaran el objeto de dicho nombre.

3.1.2. Técnicas y algoritmos a desarrollar.

- 3.1.2.1.1. Árbol de escena: todos los elementos en escena estarán contenidos en un árbol jerárquico que será gestionado. Dicho árbol permitirá el control sobre los elementos en escena mediante la manipulación de sus nodos.
- 3.1.2.1.2. Sistema de control de movimiento: los personajes deberán poder tener un movimiento que parezca autónomo y realista. Para ello, deberán ser capaces de implementar técnicas de Steering Behaviors, tales como comportamientos de Wandering. A su vez, los personajes se moverán usando un sistema de Waypoints predefinidos en cada mapa.
- 3.1.2.1.3. Sistema de percepción sensorial: la inteligencia artificial deberá recibir los mismos estímulos que recibe el jugador y ser capaz de reaccionar ante ellos de forma realista y consistente.
- 3.1.2.1.4. Sistema de predicción de movimiento para multijugador: Durante el juego en red, estará activo un sistema de predicción de movimiento mediante interpolación predictiva.
- 3.1.2.1.5. Arquitectura basada en componentes: Se utilizará una arquitectura basada en componentes, paradigma de programación que añade una capa nueva capa de abstracción, la del componente funcional. Mediante la unión y acople de los mismos, se consigue la creación de elementos (u objetos) de juego.
- 3.1.2.1.6. Level of Detail (IA): Se utilizará un sistema de nivel de detalle por el que aquellos jugadores más cercanos al jugador usarán lógica difusa, árboles de decisión y percepción sensorial mientras que los jugadores más alejados tendrán un comportamiento más básico y arbitrario.
- 3.1.2.1.7. Lógica difusa: Se utilizará un sistema de lógica difusa para el control de los vehículos de los NPC, con el objetivo de simular una conducción realista, similar a la realizada por el jugador humano.
- 3.1.2.1.8. Behaviour trees: Se implementará un árbol de comportamiento para la inteligencia artificial de los personajes en el sistema de batalla. Este se utilizará para gestionar los distintos objetos, así como la decisión y los momentos apropiados para su utilización.
- 3.1.2.1.9. Event Manager: Para la gestión de los eventos durante la carrera se utilizará un sistema de gestión de eventos, desde el cual se gestionan los distintos cambios y sucesos que ocurran, así como la respuesta que se provoque en el entorno o el resto de jugadores.

3.2. Requerimientos no funcionales.

- 3.2.1. Eficiencia: Se requerirá que el juego funcione a una tasa de fotogramas fluida, en torno a los 30 FPS idealmente, permitiendo el juego en tiempo real.
- 3.2.2. Requerimientos de red: La velocidad del funcionamiento del juego en red debe ser tal que permita un juego fluido entre varias personas, una vez la conexión se haya establecido y la partida haya sido iniciada.
- 3.2.3. Estética cartoon: Se desarrollará una estética cell-shading en el motor gráfico, para darle una estética cartoon al mismo. Los personajes y elementos del entorno se verán afectados por esta estética.
- 3.2.4. Música: Se desarrollará una banda sonora para el juego, que aporte los componentes necesarios tanto de música, como de voces y sfx. Se requerirá la inclusión del sonido mediante un motor sonoro, en nuestro caso FMOD, en el juego de forma apropiada.
- 3.2.5. Formato de datos: Formato propio de datos atendiendo a las posibles consideraciones de las estructuras de datos. Por su flexibilidad y facilidad de uso se considera la utilización de XML.
- 3.2.6. Motor gráfico: En la implementación del motor gráfico, se implementarán técnicas gráficas como:
 - 3.2.6.1. Técnicas de diseño:
 - 3.2.6.1.1. Shader cartoon.
 - 3.2.6.1.2. Luces y sombras volumétricas.
 - 3.2.6.1.3. Algoritmo de radiosidad.
 - 3.2.6.1.4. Sistema de partículas.
 - 3.2.6.1.5. Objetos emisores de luz.
 - 3.2.6.1.6. Antialiasing.
 - 3.2.6.2. Técnicas de rendimiento y optimización:
 - 3.2.6.2.1. Oclusión ambiental.
 - 3.2.6.2.2. Distintos tipos de culling.
 - 3.2.6.2.3. Level of Detail.

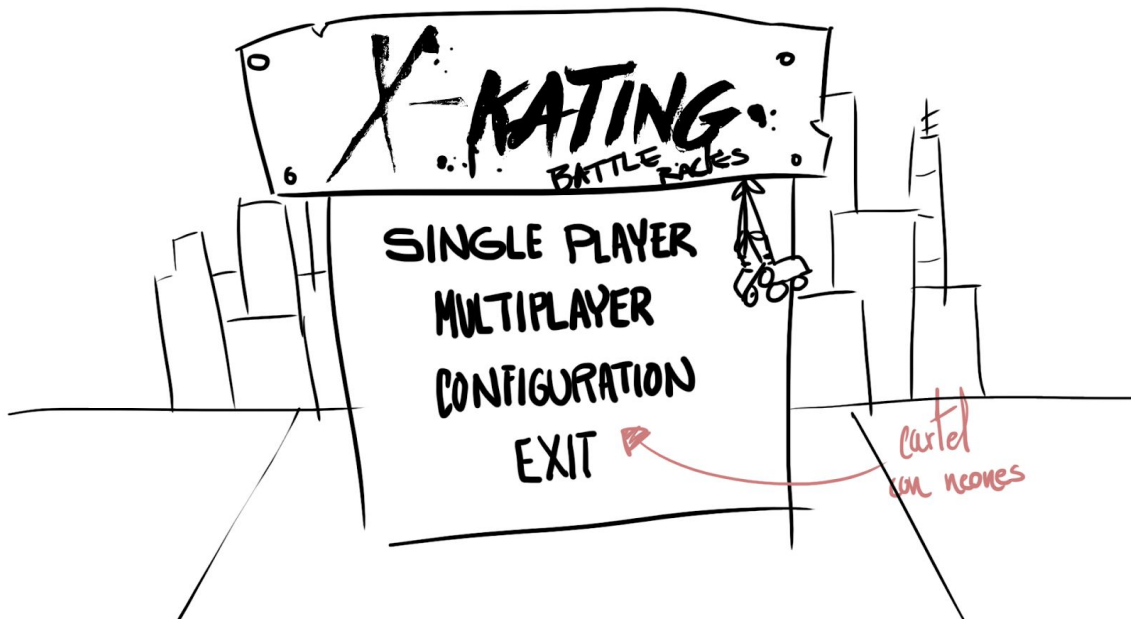
4. Apéndices

4.1. Referencias

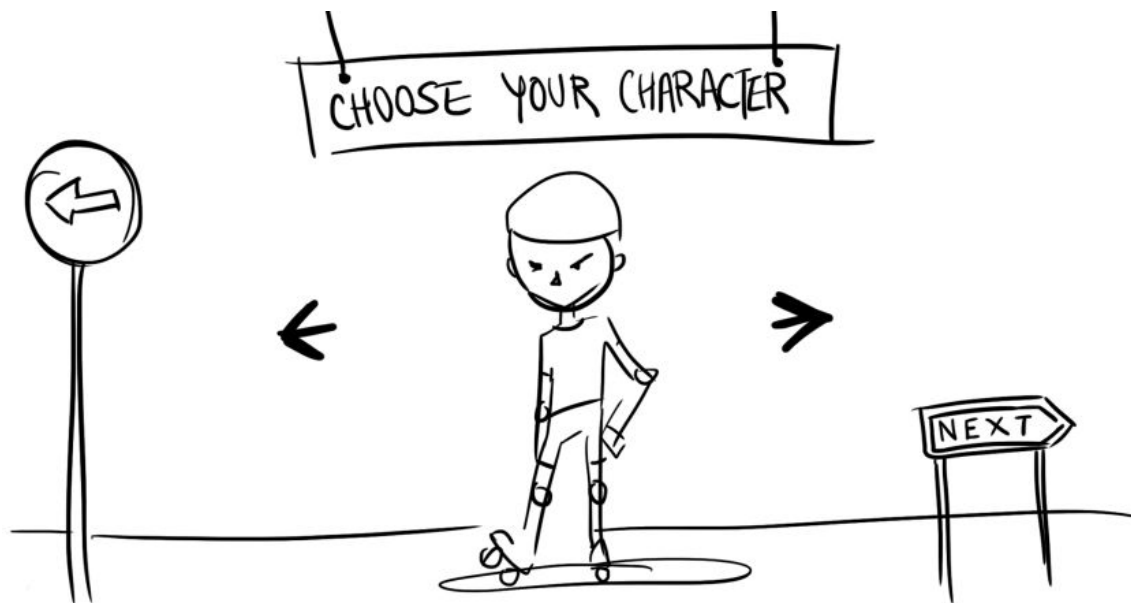
- 4.1.1. Mario Kart NES (1992), así como parcialmente en las demás versiones existentes de Mario Kart.
 - 4.1.1.1. Link: <https://www.youtube.com/watch?v=AlAmXXNz5ac>
- 4.1.2. Jet Set Radio SEGA (2000):
 - 4.1.2.1. Link: <https://www.youtube.com/watch?v=4XWzulQmy64>

4.2. Bocetos

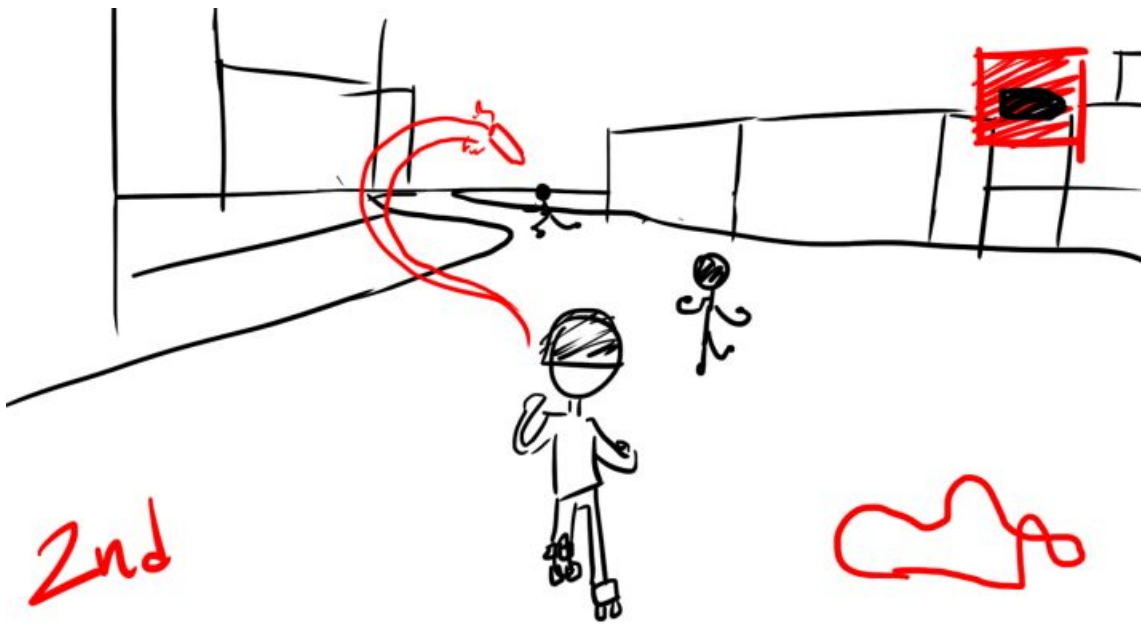
- 4.2.1. Visuales:



Menú inicial

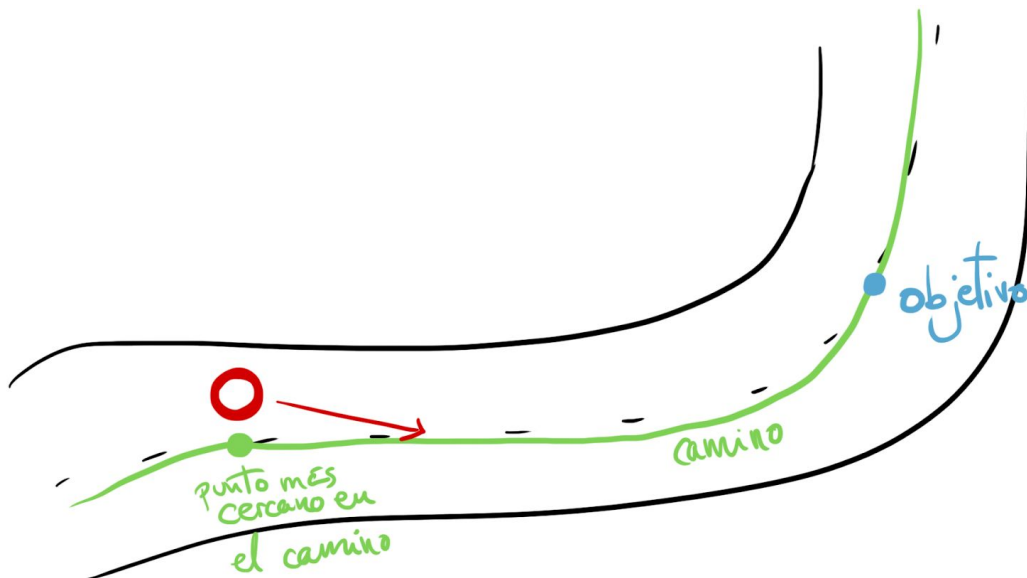


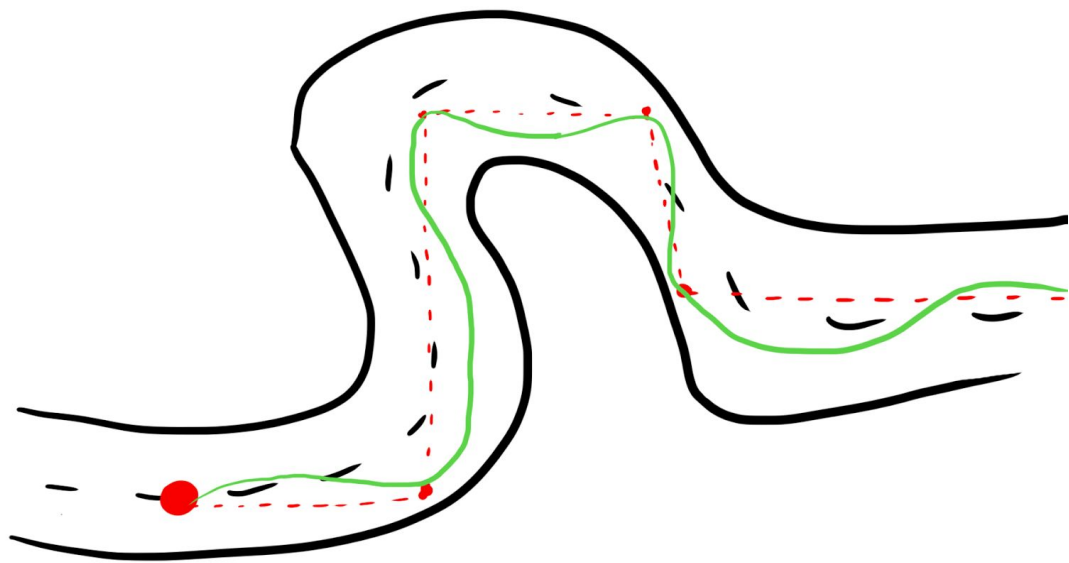
Menú de selección de personaje



Posible vista in-game usando un ítem

4.2.2. Mecánicas y tecnologías:





Steering Behaviour y Waypoints