

10 DE DICIEMBRE DE 2023



GAME DESIGN DOCUMENT

RALLY TEAM TACTICS

BARRACUDA INTERACTIVE

ADRIÁN RUBIO GARRIDO
ALEJANDRO ASENSIO PÉREZ
ÁLVARO MARTÍN HITA
DANIEL HERNÁNDEZ TAMAYO
ERIC MARTÍNEZ GAMERO
SERGIO MONTES VEREDAS



Universidad
Rey Juan Carlos

TABLA DE CONTENIDO

Índice de figuras.....	5
Índice de ilustraciones.....	5
Índice de tablas.....	10
1 Introducción	11
1.1 Introducción al juego y su enfoque al aprendizaje pensamiento computacional	11
1.2 Contenido del documento	11
2 Referencias y concepto de juego	12
2.1 Principales referencias.....	12
2.2 Concepto de juego y bases de la jugabilidad	16
2.2.1 Rol del jugador	16
2.2.2 Objetivo del jugador, jugabilidad y gestión de recursos.....	16
3 Enseñanza del Pensamiento Computacional a niños pequeños como objetivo.....	17
3.1 Los videojuegos en el aprendizaje del Pensamiento Computacional.....	17
3.2 Público objetivo.....	17
3.3 Destrezas del Pensamiento Computacional en el juego.....	17
3.4 Plataformas	18
4 Trasfondo	19
4.1 Ambientación.....	19
4.2 Narrativa	20
5 Cámara y perspectiva	22
6 Gameplay loop	23
6.1 Fase de reconocimiento.....	24
6.1.1 Puntos de control	24
6.1.2 Colocación de instrucciones en la pista	26
6.2 Fase de ejecución.....	27
6.2.1 Posibles eventos y sus consecuencias	28
6.2.2 Cantado de instrucciones durante la carrera	29
7 Control y comportamiento del vehículo	30
7.1 Relación entre tipos de sección de pista e instrucciones	30
7.1.1 Ejemplos de posibles tipos de sección	31
7.2 Obstáculos	34
7.2.1 Obstáculos de relieve	34
7.2.2 Obstáculos del entorno	37
7.2.3 Concatenación de obstáculos	38
8 Gestión de recursos.....	39

8.1 Consumo de combustible	39
8.2 Sistema de daños	40
8.2.1 Daño por colisión con obstáculos	40
8.2.2 Daño por caída	41
8.2.3 Umbral mínimo para sufrir daños.....	42
8.3 Agotamiento de recursos	43
8.4 Recursos entre pruebas	43
9 Progresión.....	44
9.1 Objetivos del jugador	44
9.2 Estructura del juego.....	44
9.3 Métricas: guardado de datos y clasificación en línea.....	47
9.3.1 Base de datos.....	47
10 Diseño de niveles	49
10.1 Diseño de secciones	49
10.2 Diseño de etapas	49
10.2.1 Primer evento.....	50
10.2.2 Segundo evento.....	53
10.2.3 Tercera etapa.....	55
10.3 Diseño de etapas: obstáculos	57
11 Controles.....	59
11.1 Ratón y teclado (PC y Web).....	59
11.2 Táctiles (Android)	60
12 Interfaz de usuario	61
12.1 Requisitos de la interfaz.....	61
12.2 Diagramas de flujo de navegación	62
12.3 Diseño esquemático de la UI	64
12.3.1 Diseño esquemático de las pantallas de menú	64
12.3.2 Diseño esquemático de las pantallas in-game.....	70
13 Arte	73
13.1 Estética del juego	73
13.1.1 Referencias artísticas	73
13.1.2 Diseño visual del juego	74
13.2 Concept	83
13.2.1 Concepts del coche	83
13.2.2 Concepts de props.....	84
14 Producción.....	89
14.1 Modelo de negocio del juego y plan de financiación	89

14.1.1 Información sobre el usuario	89
14.1.2 Mapa de Empatía	90
14.1.3 Caja de Herramientas.....	91
14.1.4 Modelo de Lienzo	92
14.2 Plan de marketing	93
14.2.1 Campaña Meet The Team	93
14.2.2 Planes para cada plataforma	95
14.2.3 Crowdfunding a través de Kickstarter	98
15 Pruebas y análisis.....	99
15.1 Análisis MDA	99
16 Post-mortem	101
16.1 Post-mortem de la Beta	101
16.1.1 ¿Qué fue bien?	101
16.1.2 ¿Qué iba mal y se corrigió a tiempo?	102
16.1.3 ¿Qué fue mal?	104
17 Capturas	106
17.1 Beta	106
17.2 Gold Master.....	112
17.2.1 Capturas sin UI.....	112
17.2.2 Capturas con UI.....	120
Glosario.....	127
Terminología del juego	127
Acciones dentro del juego.....	129
Historial de versiones.....	130
Versión 0.1 (26/09/23)	130
Versión 0.2 (28/09/23)	130
Versión 0.3 (05/10/23)	130
Versión 0.4 (08/10/23)	130
Versión 0.6 (12/10/23)	131
Versión 0.7 (17/10/23)	131
Versión 0.8 (20/10/23)	131
Versión 1.0 (22/10/23)	132
Versión 1.1 (03/11/23)	132
Versión 1.2 (07/11/23)	132
Versión 1.3 (09/11/23)	132
Versión 1.4 (15/11/23)	133
Versión 1.5 (18/11/23)	133

Versión 1.6 (19/11/23)	133
Versión 1.7 (30/11/23)	133
Versión 1.8 (03/12/23)	134
Versión 1.9 (10/12/23)	134

ÍNDICE DE FIGURAS

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Colin McRae Rally 3 (PS2)	12
Ilustración 2. DiRT Rally (PS4).....	12
Ilustración 3. WRC 10 (PC)	13
Ilustración 4. Virtual Racing (Arcade)	13
Ilustración 5. Ridge Racer (Arcade)	14
Ilustración 6. Daytona USA (Arcade).....	14
Ilustración 7. Art of Rally (PC)	15
Ilustración 8. You Suck at Parking (Xbox One)	15
Ilustración 9. WRC: Audi Quattro de Walter Röhrl en 1985	19
Ilustración 10. WRC: Toyota Celica GT-Four de Carlos Sainz en 1990	20
Ilustración 11. Perspectiva y movimiento de la cámara	22
Ilustración 12. Gameplay loop de un tramo	23
Ilustración 13. Gameplay loop de una etapa completa	23
Ilustración 14. Leyenda para diagramas de flujo sobre el gameplay	24
Ilustración 15. Gameplay loop de la fase de reconocimiento.....	25
Ilustración 16. Gameplay loop de la fase de ejecución.....	28
Ilustración 17. Leyenda para el comportamiento del vehículo	30
Ilustración 18. Iconos de las instrucciones de tipo de trazado	31
Ilustración 19. Comportamiento del coche en recta preliminar.....	31
Ilustración 20. Comportamiento del coche en curva poco cerrada preliminar	32
Ilustración 21. Comportamiento del coche en curva cerrada preliminar	32
Ilustración 22. Comportamiento del vehículo en curva muy cerrada preliminar	33
Ilustración 23. Comportamiento del coche en horquilla preliminar	33
Ilustración 24. Sketch de una rampa preliminar	34
Ilustración 25. Sketch de una meseta preliminar.....	35
Ilustración 26. Sketch de un valle preliminar	35
Ilustración 27. Sketch de un hundimiento preliminar.....	36
Ilustración 28. Sketch de un charco preliminar	36
Ilustración 29. Sketch de un bache seguido de un charco en una curva poco cerrada	38
Ilustración 30. Sketch de las barras de combustible y daño	39
Ilustración 31. Vectores de impacto y dirección.....	40
Ilustración 32. Sketch de las barras recursos: sin combustible	43
Ilustración 33. Sketch de las barras recursos: coche destrozado.....	43

Ilustración 34. Estructura de un tramo cronometrado (días/pruebas 1 a 3).....	45
Ilustración 35. Estructura de una etapa completa cronometrada (último día/prueba)	45
Ilustración 36. Estructura del juego	46
Ilustración 37. Estructura de un evento	46
Ilustración 38. Diseño final de las secciones de trazado	49
Ilustración 39. Esquema de diseño de la etapa completa del primer evento	50
Ilustración 40. Esquema de diseño del tramo 1 del primer evento	51
Ilustración 41. Esquema de diseño del tramo 2 del primer evento	51
Ilustración 42. Esquema de diseño del tramo 3 del primer evento	52
Ilustración 43. Esquema de diseño de la etapa completa del segundo evento.....	53
Ilustración 44. Esquema de diseño del tramo 1 del segundo evento	53
Ilustración 45. Esquema de diseño del tramo 2 del segundo evento	54
Ilustración 46. Esquema de diseño del tramo 3 del segundo evento	54
Ilustración 47. Esquema de diseño de la etapa completa del tercer evento	55
Ilustración 48. Esquema de diseño del tramo 1 del tercer evento	55
Ilustración 49. Esquema de diseño del tramo 2 del tercer evento	56
Ilustración 50. Esquema de diseño del tramo 3 del tercer evento	56
Ilustración 51. Esquema de diseño de la etapa completa del primer evento con obstáculos	57
Ilustración 52. Esquema de diseño de la etapa completa del segundo evento con obstáculos	58
Ilustración 53. Esquema de diseño de la etapa completa del tercer evento con obstáculos	58
Ilustración 54. Controles con teclado y ratón.....	59
Ilustración 55. Controles pantalla táctil	60
Ilustración 56. Leyenda de colores para diagramas de flujo de la UI.....	62
Ilustración 57. Diagrama de navegación de inicio de la aplicación	62
Ilustración 58. Diagrama de navegación por los menús.....	63
Ilustración 59. Boceto esquema de la Splash Screen con el logo de la empresa	64
Ilustración 60. Boceto esquema de la selección de idioma (primera vez que se inicia el juego)	64
Ilustración 61. Boceto esquema de la pantalla de título	65
Ilustración 62. Boceto esquema del pop-up de inicio de sesión, si no se ha iniciado previamente en el dispositivo	65
Ilustración 63. Boceto esquema del pop-up de registro	66
Ilustración 64. Boceto esquema del pop-up de inicio de sesión	66

Ilustración 65. Boceto esquema del menú principal	67
Ilustración 66. Boceto esquema del pop-up de ajustes.....	67
Ilustración 67. Boceto esquema del pop-up de la tienda.....	68
Ilustración 68. Boceto esquema del pop-up del garaje	68
Ilustración 69. Boceto esquema de la pantalla de selección de evento.....	69
Ilustración 70. Boceto esquema de la pantalla de selección de prueba.....	69
Ilustración 71. Boceto esquema de la HUD durante la fase de reconocimiento.....	70
Ilustración 72. Boceto esquema del pop-up del menú de ajustes.....	70
Ilustración 73. Boceto esquema del pop-up de mensaje de error.....	71
Ilustración 74. Boceto esquema del HUD durante la fase de ejecución.....	71
Ilustración 75. Boceto esquema del pop-up de ranking	72
Ilustración 76. Boceto esquema del pop-up de resultado	72
Ilustración 77. Ridge Racer (Namco, 1993).....	73
Ilustración 78. Art of Rally (Funselektor Labs, 2020)	73
Ilustración 79. Color primario escogido	74
Ilustración 80. Paleta de sombras del color primario	74
Ilustración 81. Color primario y su complementario	74
Ilustración 82. Color secundario escogido	75
Ilustración 83. Paleta de sombras del color secundario	75
Ilustración 84. Color secundario y su complementario	75
Ilustración 85. Negro no puro escogido	76
Ilustración 86. Blanco no puro escogido	76
Ilustración 87. Paleta base de RTT	76
Ilustración 88. Muestra de Mubenk Regular	77
Ilustración 89. Mapa de caracteres de Cooper Hewitt Semibold	78
Ilustración 90. Logo de RTT.....	79
Ilustración 91. Ejemplo de estilo del marco de un menú emergente.....	80
Ilustración 92. Ejemplo de estilo de un botón normal	80
Ilustración 93. Ejemplo de estilo de botones especiales superpuestos a un marco de ventana emergente	81
Ilustración 94. Ejemplo de estilo de pantalla de carga.....	81
Ilustración 95. Ejemplo de estilo de elementos del HUD.....	82
Ilustración 96. Concept de coche en perspectiva	83
Ilustración 97. Concept de coche de perfil.....	84
Ilustración 98. Concept de un punto de control	84
Ilustración 99. Concept de la salida	85

Ilustración 100. Concept de diferentes variantes de árbol.....	85
Ilustración 101. Concept de diferentes rocas	86
Ilustración 102. Concept de un barril	86
Ilustración 103. Concept de una pila de ruedas	87
Ilustración 104. Concept de una valla de obra.....	87
Ilustración 105. Concept de un cono de tráfico	88
Ilustración 106. Mapa de Empatía del cliente.....	90
Ilustración 107. Caja de Herramientas.....	91
Ilustración 108. Canvas del modelo de negocio.....	92
Ilustración 109. Captura sin interfaz de un escenario y el coche I.....	106
Ilustración 110. Captura sin interfaz de un escenario y el coche II.....	106
Ilustración 111. Captura sin interfaz de un escenario y el coche III	107
Ilustración 112. Captura sin interfaz de un escenario y el coche IV	107
Ilustración 113. Captura sin interfaz de un escenario y el coche V.....	108
Ilustración 114. Captura sin interfaz de un escenario y el coche VI	108
Ilustración 115. Captura con HUD de un punto de control I	109
Ilustración 116. Captura con HUD de un punto de control II.....	109
Ilustración 117. Captura del HUD al modificar la instrucción de un punto de control	110
Ilustración 118. Captura con HUD durante fase de reconocimiento	110
Ilustración 119. Captura del HUD durante fase de ejecución.....	111
Ilustración 120. Captura sin HUD de la primera prueba I.....	112
Ilustración 121. Captura sin HUD de la primera prueba II.....	112
Ilustración 122. Captura sin HUD de la primera prueba III	113
Ilustración 123. Captura sin HUD de la primera prueba IV	113
Ilustración 124. Captura sin HUD de la primera prueba V.....	114
Ilustración 125. Captura sin HUD de la primera prueba VI	114
Ilustración 126. Captura sin HUD de la segunda prueba I	115
Ilustración 127. Captura sin HUD de la segunda prueba II.....	115
Ilustración 128. Captura sin HUD de la segunda prueba III	116
Ilustración 129. Captura sin HUD de la segunda prueba IV.....	116
Ilustración 130. Captura sin HUD de la segunda prueba V	117
Ilustración 131. Captura sin HUD de la tercera prueba I	117
Ilustración 132. Captura sin HUD de la tercera prueba II	118
Ilustración 133. Captura sin HUD de la tercera prueba III.....	118
Ilustración 134. Captura sin HUD de la tercera prueba IV.....	119

Ilustración 135. Captura sin HUD de la tercera prueba V	119
Ilustración 136. Captura HUD durante la fase de reconocimiento I.....	120
Ilustración 137. Captura HUD durante la fase de reconocimiento II	120
Ilustración 138. Captura HUD durante la fase de reconocimiento III: selección de instrucción	121
Ilustración 139. Captura HUD durante la fase de reconocimiento IV: selección de instrucción	121
Ilustración 140. Captura HUD durante la fase de reconocimiento V: selección de instrucción	122
Ilustración 141. Captura HUD durante la fase de reconocimiento VI: instrucciones seleccionadas	122
Ilustración 142. Captura HUD durante la fase de reconocimiento VII: todas las instrucciones seleccionadas	123
Ilustración 143. Captura HUD durante la fase de ejecución I.....	123
Ilustración 144. Captura HUD durante la fase de ejecución II	124
Ilustración 145. Captura HUD durante la fase de ejecución III	124
Ilustración 146. Captura HUD durante la fase de ejecución IV	125
Ilustración 147. Captura HUD durante la fase de ejecución V: checkpoint superado con éxito.....	125
Ilustración 148. Captura HUD durante la fase de ejecución VI: prueba finalizada ..	126

ÍNDICE DE TABLAS

Tabla 1. Destrezas del Pensamiento Computacional en el juego	18
Tabla 2. Consumo de combustible en función del comportamiento del coche.....	39
Tabla 3. Fechas campaña MeetTheTeam	93
Tabla 4. Campaña de marketing en Instagram.....	95
Tabla 5. Campaña de marketing en X	96
Tabla 6. Campaña de marketing en Youtube.....	97
Tabla 7. Glosario de terminología del juego	128
Tabla 8. Glosario de acciones dentro del juego	129

1 INTRODUCCIÓN

Este documento cubre todos los aspectos referentes al diseño del juego *Rally Team Tactics (RTT)*, desarrollado por *Barracuda Interactive* para PC, navegadores web y dispositivos Android y cuyo lanzamiento está previsto para finales de 2023.

1.1 INTRODUCCIÓN AL JUEGO Y SU ENFOQUE AL APRENDIZAJE PENSAMIENTO COMPUTACIONAL

Como breve introducción al juego, *Rally Team Tactics (RTT)* es un *arcade racer* donde el jugador asume el rol del copiloto de un equipo de rally. Como tal, este debe analizar el trazado del circuito para dar las instrucciones adecuadas al piloto antes de que comience la prueba.

El juego pone gran énfasis en el ensayo y error gracias a la posibilidad de volver al inicio. El jugador tiene así la posibilidad de resolver el problema basándose en lo acontecido anteriormente. La identificación de patrones (respuesta de las físicas del vehículo al relieve y el itinerario) le permiten mejorar la elección de instrucciones y hacer así una gestión de los recursos (gasolina y daño recibido) más eficiente en su siguiente intento.

1.2 CONTENIDO DEL DOCUMENTO

A lo largo de los siguientes capítulos, se cubre en este documento:

- Principales referencias y jugabilidad
- Propósito del proyecto: enseñanza del Pensamiento Computacional a niños pequeños
- Proceso de diseño del juego en todos sus ámbitos:
 - Trasfondo
 - Mecánicas
 - Progresión
 - Interfaces y experiencia de usuario (UI y UX)
 - Arte
 - Música y sonido
- Producción y modelo de negocio

2 REFERENCIAS Y CONCEPTO DE JUEGO

2.1 PRINCIPALES REFERENCIAS

Rally Team Tactics (RTT) es un *arcade racer* inspirado por otros juegos de simulación de carreras o de conducción *arcade*, con ejemplos provenientes desde los 90 hasta la actualidad. Entre estas referencias, encontramos:

- Franquicias de rally y carreras off-road realistas:
 - **Colin McRae Rally**, conocida actualmente como **DiRT** y **DiRT Rally** (*Codemasters*, EA, 1998-2020). Ver Ilustración 1 e Ilustración 2.
 - **WRC** (múltiples desarrolladores, EA, 2001-2023). Ver Ilustración 3.
- De estas se toman su énfasis en la simulación de las físicas de los vehículos sobre diferentes terrenos.



Ilustración 1. Colin McRae Rally 3 (PS2)



Ilustración 2. DiRT Rally (PS4)



Ilustración 3. WRC 10 (PC)

- Juegos de carreras para recreativas de los años 90:
 - *Virtua Racing* (Sega AM2, 1992). Ver Ilustración 4.
 - *Ridge Racer* (Namco, 1993). Ver Ilustración 5.
 - *Daytona USA* (Sega AM2, 1993). Ver Ilustración 6.

Estos juegos sirven como inspiración tanto para el estilo artístico colorido y retro de RTT como para mecánicas arcade como las pruebas contrarreloj, el uso del turbo o *leaderboards*.



Ilustración 4. Virtual Racing (Arcade)



Ilustración 5. Ridge Racer (Arcade)



Ilustración 6. Daytona USA (Arcade)

- Juegos indie de conducción *arcade* contemporáneos:
 - *Art of Rally* (Funselektor Labs, 2020): las principales influencias provenientes de este título son su *setting* en la época dorada del rally y su arte 3D *low-poly* de gran calidad y su iluminación. Ver Ilustración 7. Otros aspectos por destacar:
 - Su perspectiva de cámara aérea.
 - Sus físicas que no llegan al realismo de los títulos del primer punto pero que encuentran un buen balance entre el control arcade y la simulación centrada en las características del terreno (relieve, superficie) y las condiciones climatológicas.
 - *You Suck at Parking* (Happy Volcano, 2022): otra gran inspiración, especialmente por su naturaleza ensayo-error (reinicio rápido). El juego introduce además elementos de puzzles que ponen a prueba la memoria

y el reconocimiento de patrones del jugador. También cuenta con una cámara aérea, con la posibilidad de visualizar el nivel libremente antes de empezar un intento. Este último elemento es una fuerte influencia para RTT. Ver Ilustración 8.



Ilustración 7. Art of Rally (PC)



Ilustración 8. You Suck at Parking (Xbox One)

2.2 CONCEPTO DE JUEGO Y BASES DE LA JUGABILIDAD

Rally Team Tactics (RTT) se concibe como un *arcade racer* en 3D ambientado en la época dorada del rally.

2.2.1 ROL DEL JUGADOR

El jugador toma el control de un **copiloto de rally** en lugar del piloto. Este giro le da una nueva perspectiva al género, pues en lugar de tener agencia directa sobre el vehículo, **se deben “cantar” las instrucciones, que describen el trazado (notas) y aconsejan un temperamento concreto**. Además, como a los copilotos en la realidad, al jugador se le otorga la **responsabilidad de supervisar aspectos del coche** como el nivel de combustible restante o los daños sufridos.

Como no controlamos el coche directamente, qué instrucciones “cantemos” determinarán el comportamiento de este. Para situar las instrucciones del copiloto en el trayecto, se permite en todo momento tener una “vista de pájaro” con una **cámara libre para navegar por toda la etapa**, como si tuviésemos acceso a las cámaras del helicóptero de televisión.

2.2.2 OBJETIVO DEL JUGADOR, JUGABILIDAD Y GESTIÓN DE RECURSOS

El objetivo es lograr el **mejor tiempo en cada etapa de rally**, que se divide en **tramos cronometrados**. Durante los tres primeros días de un evento, **el jugador se debe enfrentar a pruebas cronometradas de cada tramo por separado**, refinando su estrategia de cara al día de la prueba de etapa.

Solo podemos **situar las instrucciones mientras el vehículo se encuentre parado**, por lo que para cambiarlas se debe reiniciar el tramo o etapa mediante **un reinicio que conserva las instrucciones seleccionadas**.

La experimentación puede llevar al jugador de chocar en la primera curva a conseguir su mejor tiempo en el tramo o etapa a base de ensayo y error. **Si el coche choca o sufre demasiados daños, es obligatorio regresar al inicio** para idear una nueva estrategia.

Siempre que se finaliza un tramo o etapa se muestra el **tiempo conseguido**, y si se ha completado el tramo o etapa en el pasado también si se ha superado al histórico y por cuánto.

La gestión de recursos juega un papel importante. Tanto el daño sufrido por el vehículo como el combustible restante deben tenerse en cuenta. Al inicio del tramo o etapa, el coche tiene 0% de daños y 100% de combustible.

- Si el coche sufre una caída o impacta con algún objeto su integridad se ve comprometida, y el **medidor de daño (damage)** incrementa en diferentes cantidades.
- El tanque de combustible siempre se va consumiendo, pero utilizar el turbo o acelerar tras una frenada importante producen un mayor consumo del **medidor de gasolina (fuel)**.

3 ENSEÑANZA DEL PENSAMIENTO COMPUTACIONAL A NIÑOS PEQUEÑOS COMO OBJETIVO

3.1 LOS VIDEOJUEGOS EN EL APRENDIZAJE DEL PENSAMIENTO COMPUTACIONAL

Llamamos Pensamiento Computacional (PC) a la habilidad cognitiva relacionada con el uso de estrategias computacionales para la resolución de problemas. Su aprendizaje desde una edad temprana puede proporcionar amplios beneficios en la vida cotidiana. Por este motivo, la enseñanza del PC es parte del currículo escolar en España y otros países desde la etapa de educación infantil.

Los videojuegos orientados a potenciar el PC pueden tener una mayor efectividad que otras actividades como la programación o los juegos de mesa. Los niños y niñas en la actualidad tienen una relación estrecha con el mundo de los videojuegos, y las posibilidades que estos ofrecen en la enseñanza apenas se han explotado.

Por los motivos citados anteriormente, el juego pretende explorar ese nicho, creando una experiencia lúdica que impulse el uso del PC dentro de un entorno de entretenimiento.

3.2 PÚBLICO OBJETIVO

El *target* del videojuego son niños y niñas desde 8 hasta 12 años. Esta demográfica juega con frecuencia a títulos de cierta complejidad como *Fortnite* o *Roblox*, por lo que se tiene en cuenta este factor a la hora de diseñar el juego: pese a tratarse de un producto educativo, no subestima las capacidades de los niños y evita las características habituales de los *serious games*. Se busca reforzar el PC sin hacerlo explícito, comprometer la diversión ni la posibilidad de que el juego pueda ser disfrutado por audiencias de mayor edad.

3.3 DESTREZAS DEL PENSAMIENTO COMPUTACIONAL EN EL JUEGO

Desde la concepción del juego se busca potenciar múltiples facultades propias del PC. Las destrezas particulares que se pretenden ejercitan jugando RTT son las siguientes:

Destreza del PC	Cómo se pone en práctica en el juego
Abstracción	Esta habilidad se instruye gracias a la decisión de diseño que pone al jugador en los pies del copiloto. El jugador debe abstraerse de la reacción inmediata de la conducción, situándose en una capa superior de planificación y toma de decisiones . La “vista de pájaro” pone énfasis en esta capacidad de ver el problema en su totalidad y reconocer así sus elementos clave .
Pensamiento algorítmico	En RTT se pone a disposición del jugador un conjunto de notas y un <i>slider</i> de temperamento que determinan el comportamiento del coche para cada sección de una prueba. Se debe crear un algoritmo , al fin y al cabo, una secuencia de acciones concretas que da lugar al mejor resultado posible.
Descomposición	En base a la respuesta del vehículo en un determinado tramo con una determinada serie de instrucciones, el jugador puede adquirir

	la capacidad de deducir cómo se va a comportar el coche en el mismo escenario u otro similar en función de sus decisiones en la fase de reconocimiento.
Evaluación	<p>El <i>RTT</i> se presta especial atención a la gestión de recursos, es este caso, el combustible y el daño sufrido. Se deben considerar a corto y medio plazo la consumición de combustible y el desgaste del coche. Esto implica no solo tener en mente el tramo inmediatamente posterior, sino la etapa en su totalidad. Reservar combustible para un momento concreto o arriesgar la integridad del vehículo en el punto adecuado de la pista son ejemplos de decisiones que pueden dar lugar a mejores tiempos.</p> <p>Además, mediante el ensayo y error, el jugador asimila la detección de fallos, y toma decisiones ajustadas al objetivo. Si el coche responde de una determinada forma a una secuencia de instrucciones que producen un consumo de recursos correspondiente, puede usar ese conocimiento para siguientes intentos.</p>
Generalización	<p><i>RTT</i> pone el foco en la repetición para encontrar soluciones mejores. Resolver el problema en base a soluciones anteriores es parte fundamental del diseño. Cuando el jugador encuentra patrones y similitudes en la reacción del vehículo ante distintos obstáculos o trazados, aprende a utilizar las estrategias que llevan a un resultado más satisfactorio.</p> <p>Aunque el jugador no conoce al detalle las físicas del coche (ni se pretende que lo haga), a medida que se va familiarizando con su comportamiento, toma decisiones que le resultan más favorables.</p>

Tabla 1. Destrezas del Pensamiento Computacional en el juego

3.4 PLATAFORMAS

Como los niños suelen disponer de tabletas en los centros educativos o en su hogar, es fundamental que el juego se desarrolle no solo para **escritorio y navegadores Web**, sino también para dispositivos táctiles Android (en este caso **tablets Android**).

4 TRASFONDO

4.1 AMBIENTACIÓN

RTT está ambientado en la actualidad, y el jugador viajará alrededor del mundo compitiendo cara a cara con los mejores pilotos.

El mundo de *Rally Team Tactics* es una realidad alternativa donde la época dorada de los *rallies* nunca cesó y esta competición automovilística mantiene una tremenda popularidad.

Se suele considerar que la época dorada del rally fue la década de los 80, cuando se creó la categoría de Grupo B, que permitía a los fabricantes desarrollar coches extremadamente potentes y veloces, y las restricciones de velocidad y técnicas eran mínimas. Las carreras eran espectaculares y peligrosas: los pilotos se ponían en riesgo su vida en cada etapa y las masas de espectadores se agolpaban al borde de las carreteras para ver a sus ídolos de cerca. Algunos de los modelos más emblemáticos de esta época fueron el Audi Quattro, el Lancia Delta S4, el Peugeot 205 T16 o el Ford RS200.



Ilustración 9. WRC: Audi Quattro de Walter Röhrl en 1985



Ilustración 10. WRC: Toyota Celica GT-Four de Carlos Sainz en 1990

4.2 NARRATIVA

El varias veces campeón del mundo de rally **Erik Ström** estaba en el pico de su carrera. Su destreza y agresividad en la carretera lo hacían casi imparable.

Pese a haber crecido en una familia humilde, el talento de Ström lo había llevado de la pobreza a ser una de las figuras más admiradas y reconocibles del mundo del automovilismo.

No obstante, su escalada al zenit no fue sin consecuencias para su salud mental. Tras una serie de derrotas en etapas donde era el favorito, Ström no era capaz de concentrarse en la pista. Jóvenes pilotos estaban despuntando, y Ström no tenía un coche competitivo y estaba harto de seguir las instrucciones de los ingenieros, a los que consideraba ineptos. Las averías eran demasiado frecuentes, y la actuación cautelosa del equipo estaba acabando con su paciencia. Desesperado, firmó con un constructor dispuesto a dejarle desobedecer las órdenes del equipo, incluido su copiloto, si lo veía necesario. Además, sabía que su nuevo coche no cumplía con la normativa, y que las piezas prohibidas habían sido camufladas y para pasar irregularmente las comprobaciones de seguridad.

La temporada de rally de 2013 comenzó bien para Ström, posicionándose entre los tres primeros justo antes de una de las etapas más temidas del circuito anual: la *Acropolis Rally of Greece*. Terrenos montañosos, acantilados rocosos y calor sofocante sobre un asfalto antiguo y unas secciones campo a través desafiantes. No era la etapa para la audacia ni el riesgo, pero Ström no dudó en jugarse el campeonato en tierras griegas.

El día de la carrera, el piloto sueco estaba codo a codo con los mejores tiempos marcados por sus rivales más cercanos en la clasificación, y decidió acelerar en uno de los tramos más infames de la etapa: la curva del abismo. Fue aquí donde Ström, cegado por la ambición, no frenó a tiempo y cayó por el barranco. Cuando los equipos médicos

le encontraron, Erik, inconsciente, no respondía. Pero la suerte le había sonreído, pues no había sido un accidente fatal.

Sin embargo, esto le supondría su ingreso hospitalario, donde estaría años en rehabilitación, sometiéndose a numerosas operaciones, hasta que pudo recuperar parte de la movilidad. Su tren inferior no era funcional, nunca podría volver a competir. Su tiempo de recuperación le llevó a replantearse su vida y su carrera profesional: sabía que no podría volver a competir, y no encontraba su propósito para seguir adelante. Fueron años oscuros, pero Ström superó la depresión y, tras finalizar su rehabilitación, regresó al mundo de los *rallies*. Su discapacidad le impedía conducir, pero sabía que tenía mucho conocimiento que aportar a las nuevas generaciones.

En 2019, fundó una escuela en su ciudad natal y hizo de mentor para decenas de aspirantes a futuros pilotos. Se aseguraba también de que los jóvenes pusiesen su salud mental y física por encima de cualquier objetivo en la pista.

Ström era una figura respetada en el automovilismo y los rumores de su vuelta a los *rallies* se multiplicaban, pero nadie sabía qué rol tomaría de ser así. A sus 56 años, en 2022, se asoció con algunos de sus amigos y antiguos conocidos de la industria para montar un equipo y firmar un contrato con una reputada constructora. Sus coches llevaban los colores de su país natal en su carrocería.

Erik tenía claro cuál era su sitio ahora: el de copiloto. Quería redimirse por todos sus errores y llevar el talento de su escuela a lo más alto a base de aprendizaje y constancia.

Dado el pistoletazo de salida a la temporada 2023, Ström acaparaba todas las portadas: ¿será este el verdadero regreso del rey de los *rallies*?

5 CÁMARA Y PERSPECTIVA

La cámara es **aérea**, emula la vista desde un helicóptero de televisión. El entorno tridimensional se puede navegar tanto en fase de reconocimiento como en fase de ejecución.

Las limitaciones de movimiento de la cámara son:

- Imposibilidad de hacer *zoom*
- Imposibilidad de regular la altitud
- La inclinación es fija, de tal manera que se observen con suficiente claridad las características del terreno y el trayecto del circuito

El jugador tiene control, por tanto, sobre:

- La posición en el plano horizontal definido a la altitud predeterminada
- El giro de la cámara, 360 grados de rotación

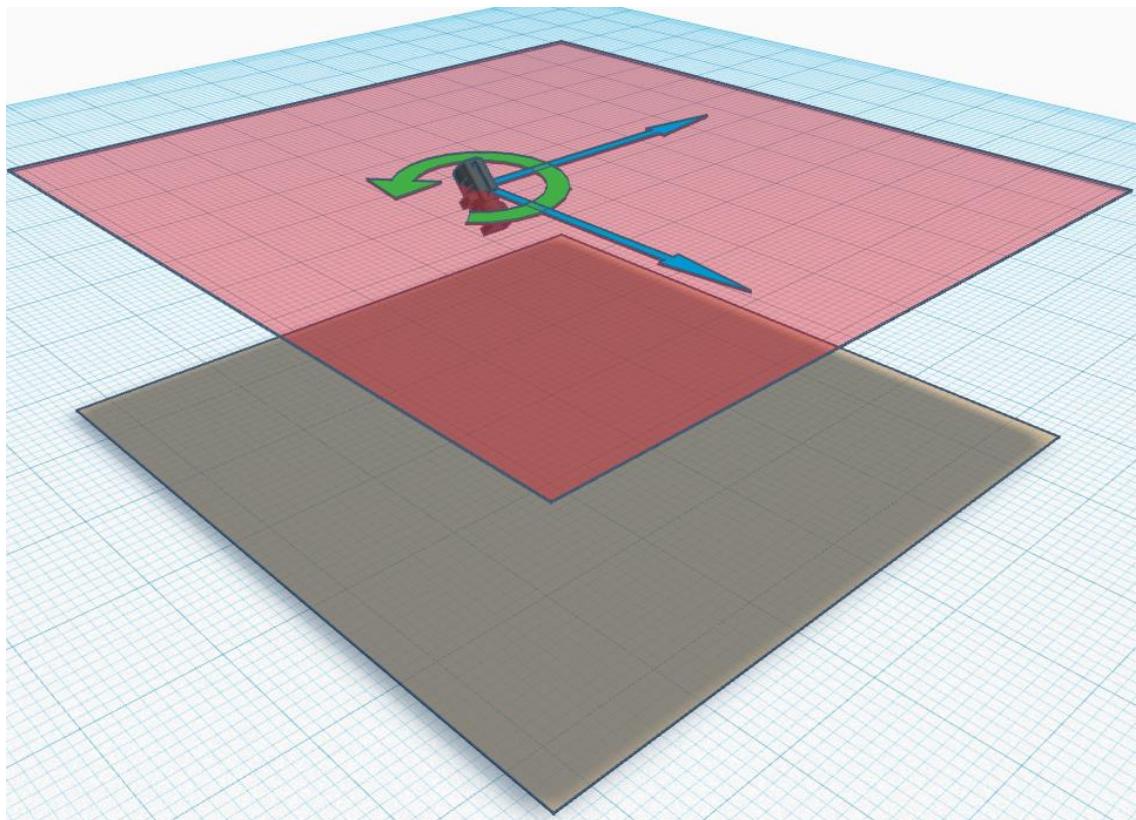


Ilustración 11. Perspectiva y movimiento de la cámara

Las flechas azules y verdes representan el movimiento horizontal y giro de la cámara, respectivamente, mientras que el plano rojo indica la altitud fija, y como se puede observar, la inclinación de la cámara también está bloqueada (ver Ilustración 11).

6 GAMEPLAY LOOP

En este apartado se van a describir con detenimiento las fases que componen el *gameplay loop* del juego.

La jugabilidad de RTT se divide en dos fases o turnos: reconocimiento y ejecución. Esto es igual en los días de pruebas preparación (los distintos tramos) y en el día de prueba final (la etapa completa). Los diagramas de alto nivel para tramos y etapas completas se presentan a continuación:

Gameplay Loop de un tramo

En un tramo los jugadores solo pueden elegir instrucciones para diez secciones de pista, que determinan el tramo. Pueden buscar mejorar su tiempo en este tramo para prepararse para la etapa final

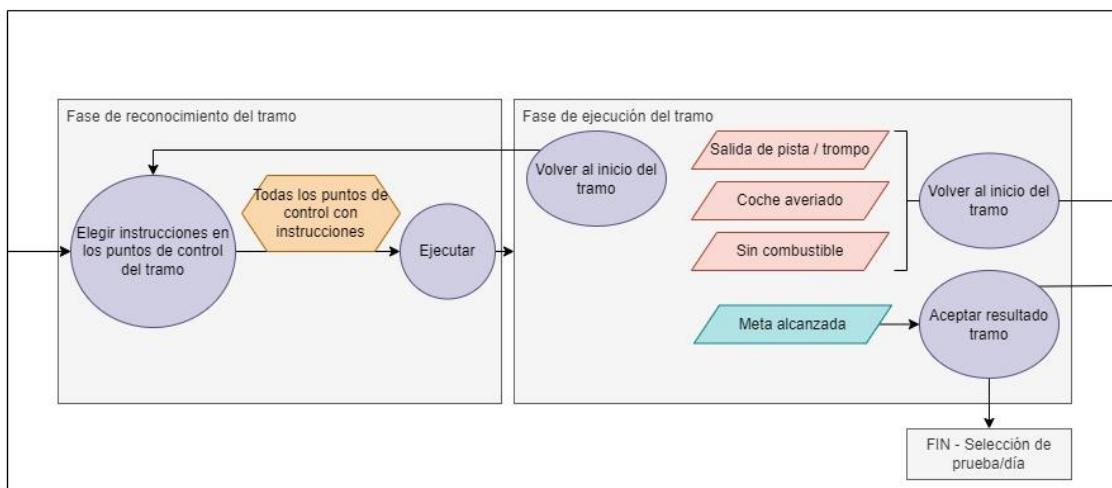


Ilustración 12. Gameplay loop de un tramo

Gameplay Loop de una etapa completa

En la etapa completa los jugadores comienzan con las instrucciones que eligieron en los tramos de los días anteriores, ahora pueden variarlas para enlazar dichos tramos de manera óptima

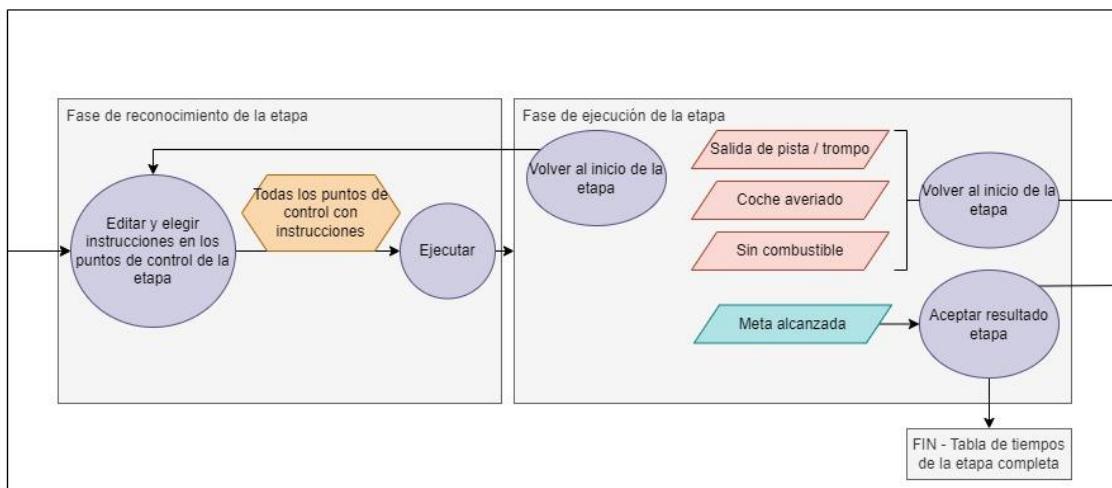


Ilustración 13. Gameplay loop de una etapa completa

En la leyenda que se muestra en la ilustración inferior se explica el significado de las formas en los diagramas de flujo referentes a *gameplay loop*:



Ilustración 14. Leyenda para diagramas de flujo sobre el gameplay

6.1 FASE DE RECONOCIMIENTO

Como se plasma en el diagrama Ilustración 15, el jugador puede elegir instrucciones durante la **fase de reconocimiento**. Una vez se hayan seleccionado las instrucciones deseadas para todos los puntos de control, se puede ejecutar.

Durante el reconocimiento, el jugador puede navegar con libertad tal como se ha descrito en [5 Cámara y perspectiva](#), y, además:



6.1.1 PUNTOS DE CONTROL

Los **puntos de control** son puntos del circuito donde se debe especificar la **instrucción** que se va a “cantar” al piloto cuando llegue a cada uno de ellos. Al seleccionar un punto de control, el jugador tiene las siguientes opciones:

- Seleccionar la nota de código nemotécnico** que ve más conveniente en función de el trazado inmediatamente posterior al punto de control. Entre estas notas encontramos, por ejemplo, “curva muy cerrada” o “recta”.
- Elegir el temperamento** con el que realizar la acción. Esto es posible gracias a un *slider continuo* que permite determinar si se quiere un acercamiento **más cauteloso o más agresivo**.
 - Por defecto, cuando se elige una nota por primera vez en un punto, el *slider* se encuentra en la posición intermedia (temperamento completamente moderado).
 - Para ilustrar esta idea, supongamos que tenemos la nota “curva muy cerrada”: cuanto más agresivos seamos, el frenado se producirá más tarde y con mayor fuerza, mientras que, si optamos por un acercamiento más cauteloso, el frenado será más largo y comenzará mucho antes.

En el caso de una nota de tipo “recta”, los dos extremos se corresponderían con reducir la velocidad y acelerar, siendo el punto medio del *slider* mantener la velocidad.

Todos los casos se explican al detalle más adelante (ver [7 Control y comportamiento del vehículo](#)).

C. Deseleccionar un punto de control

que es tan sencillo como pulsar o clicar en otro lugar del circuito.

Por otro lado, en todo momento se puede **limpiar el circuito**, que sería equivalente a comenzar de cero el nivel: todas las instrucciones situadas se eliminarían.

La acción que daría paso a la fase de ejecución es **comenzar la prueba**, eso sí, esto solo se puede hacer si se han elegido instrucciones en todos los puntos de control.

Además de todo lo anterior, siempre es posible **entrar en el menú de ajustes**, pausando el juego. Aquí, entre otras cosas (configuración), se permite salir de la prueba.

El siguiente diagrama muestra el *gameplay loop* correspondiente a la fase de reconocimiento en mayor detalle:

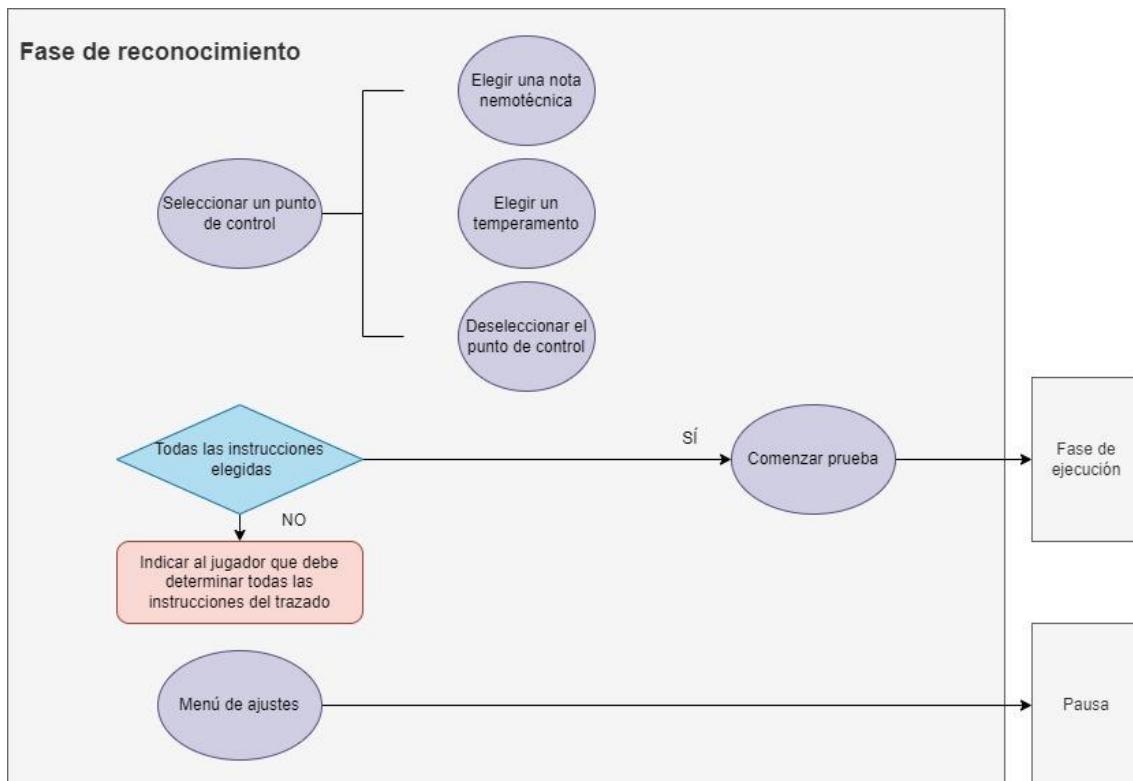


Ilustración 15. Gameplay loop de la fase de reconocimiento

Ver la leyenda de la Ilustración 14 para el significado de las diferentes formas.

6.1.2 COLOCACIÓN DE INSTRUCCIONES EN LA PISTA

En este apartado se hace énfasis en el sistema de selección de puntos de control. Los puntos de control deben quedar claramente señalizados en la pista, y deben proveerse opciones de configuración a los jugadores para que los puedan distinguir con facilidad. Se explican a continuación cómo se plantea esta cuestión en función de la fase de juego en la que se encuentre el jugador:

- El uso de banderas a los laterales de la pista es un marcador diegético de la posición de estos puntos de control, que recordemos se encuentran entre secciones de pista cuya separación el jugador no puede discernir. **Esta señalización se mantiene entre fases de reconocimiento y ejecución**, ya que es parte del mundo de juego.
- De forma extradiegética, se puede mostrar un muro luminoso perpendicular al trazado en esos puntos para hacerlos más notables. Además, el color puede variar si este ha sido elegido ya o si está vacío aún. **Este muro solo es visible en fase de reconocimiento**.
- También de forma extradiegética, aquellos puntos de control donde se ha seleccionado instrucción (nota y temperamento) pueden mostrar la elección de manera flotante o sobre la pista. El temperamento recordemos está controlado por un *slider*, pero el ícono puede variar únicamente al superar un determinado valor en dicho *slider*. Por ejemplo, la barra se dividiría en tres partes, y el ícono solo cambiaría cuando se pasase a un tercio distinto. Como el muro luminoso, **estos iconos son únicamente visibles en fase de reconocimiento**.

6.2 FASE DE EJECUCIÓN

El paso a la **fase de ejecución** arranca el vehículo desde el punto de salida. Ahora podemos ver el vehículo realizar el trayecto de acuerdo con las instrucciones situadas en reconocimiento que “canta” el piloto. Durante la ejecución seguimos teniendo el control de la cámara y podemos reiniciar en cualquier momento, volviendo a la fase de reconocimiento.

Durante la fase de ejecución, se simula la prueba de *rally* de acuerdo con las notas nemotécnicas y temperamento elegidos a lo largo del trazado. El vehículo recorre el trayecto automáticamente, reaccionando según sus físicas con el relieve y las curvas de la carretera. Además, el copiloto “canta” las instrucciones cuando el coche pasa por cada una de ellas (*voiceover*) para dotar de mayor realismo a la carrera.

Además de mover libremente la cámara, hay dos acciones que se pueden realizar siempre en esta fase:

- A. **Acceder al menú de ajustes** supone lo mismo que en la fase de reconocimiento
- B. **Reiniciar** implica que el coche volverá a la salida del tramo/etapa, conservándose todas las instrucciones elegidas previamente para cada punto de control

Si el coche choca quedando inmóvil, se queda sin combustible o sufre daños irreparables, estas son las únicas opciones disponibles.

No obstante, si se completa el tramo o etapa de principio a fin, también se nos da la posibilidad de **aceptar el resultado**, que se muestra en el temporizador junto con la diferencia de tiempo respecto al histórico. En caso de que el jugador vea el resultado satisfactorio, entonces se le reconducirá a una pantalla donde podrá ver una clasificación con otros jugadores.

Veamos un diagrama al detalle de esta fase:

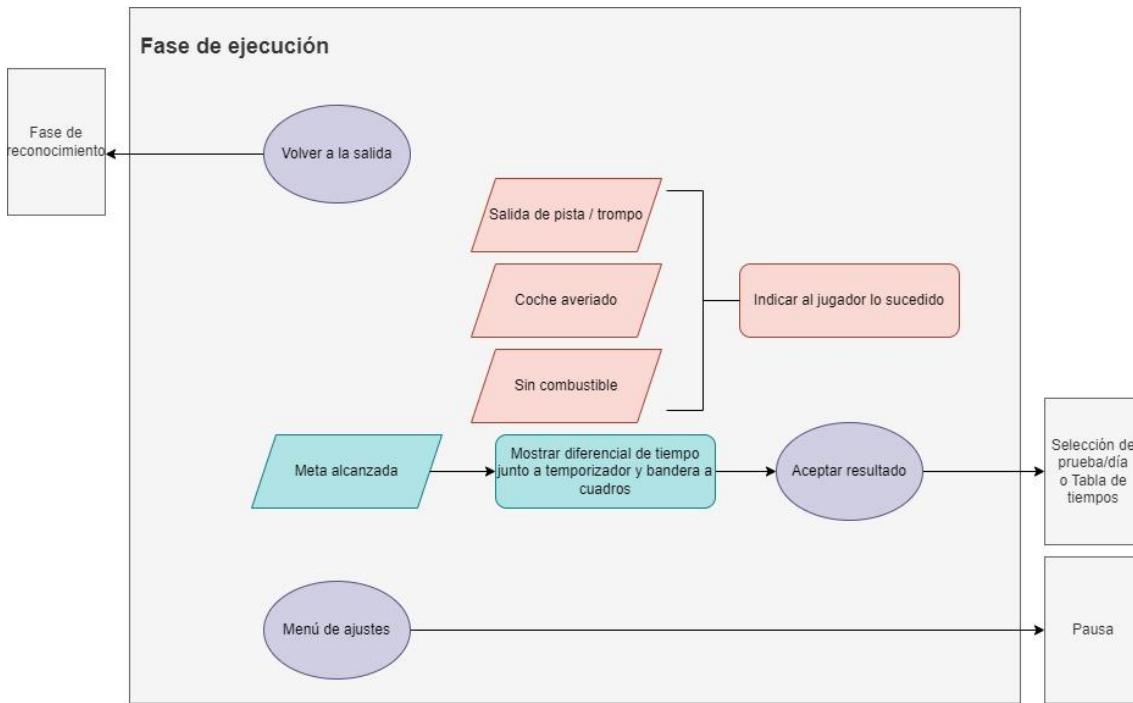


Ilustración 16. Gameplay loop de la fase de ejecución

Ver la leyenda de la Ilustración 14 para el significado de las diferentes formas.

6.2.1 POSIBLES EVENTOS Y SUS CONSECUENCIAS

Se pueden producir una serie de eventos negativos, que nos forzarán a reiniciar, como una salida de pista completa o el consumo del tanque de gasolina.

Por otro lado, los eventos positivos posibles son alcanzar el final del tramo o la meta. La meta es el final solo en el último tramo y en la etapa completa. Si el jugador acepta el resultado (no reinicia) se daría por finalizado el tramo o etapa. Además. De tratarse de una etapa completa, se mostraría la **pantalla de clasificación (leaderboards)** correspondiente; de lo contrario, se pasaría directamente a la selección de prueba del evento que se está disputando.

6.2.2 CANTADO DE INSTRUCCIONES DURANTE LA CARRERA

Al igual que para el reconocimiento, este apartado trata cómo se hacen visibles los puntos de control al jugador, pero esta vez durante la fase de ejecución.

- Las banderas físicas que marcan los puntos de control siguen presentes en esta etapa, son elementos diegéticos.
- Los muros luminosos ya no están presentes, ni tampoco los iconos de nota y temperamento, es decir, ningún elemento extradiegético flotante se ve.
- Para que el jugador tenga un refuerzo sobre las decisiones tomadas en la fase de reconocimiento, cada vez que el vehículo pase por un punto de control, **el copiloto (voiceover) “canta” las instrucciones**, no solo diciendo el nombre de esta, sino también el grado de agresividad determinado por el temperamento. Al ser el temperamento continuo y no discreto, como en la fase de reconocimiento, se tiene que discretizar en varios estados. Estos estados corresponden con los tercios del *slider*, siendo estos equivalentes a acompañar la nota en el “cantado” de la instrucción de expresiones, por ejemplo: “curva cerrada, tomar con cuidado”, “recta, recorrer con agresividad” o “curva muy cerrada, acometer con moderación”.

Otra forma de recordar al jugador durante la fase de ejecución las instrucciones elegidas sería **mostrarlas además en el HUD como iconos** al tiempo que son “cantadas” (esto podría ser configurable).

7 CONTROL Y COMPORTAMIENTO DEL VEHÍCULO

Se van a explicar a continuación el control y el comportamiento del vehículo ante las diferentes curvas y obstáculos preliminares si se elige su nota de código nemotécnico correcta, así como el efecto del temperamento en cada caso. En caso de no elegirse la nota correspondiente, el coche seguirá las directrices de la seleccionada, dando lugar a resultados que normalmente serán negativos, saliendo de la pista o chocando, pero que en raras ocasiones pueden optimizar el recorrido. La clave al fin y al cabo es la experimentación.

RTT presenta una serie de secciones de carretera predefinidas por su curvatura, todas ellas de la misma longitud. Es justo entre estas secciones en el trazado donde se eligen las notas y el temperamento. También se elige instrucción en el punto de inicio para la primera sección, naturalmente.

El diseño final de secciones y pruebas se encuentra en [10 Diseño de niveles](#).

7.1 RELACIÓN ENTRE TIPOS DE SECCIÓN DE PISTA E INSTRUCCIONES

A continuación, se muestran bocetos de todos los tipos de sección preliminares y se explica cómo se comportaría el vehículo de ser seleccionada la nota correcta, en tres casos concretos en el *slider* de temperamento: cauteloso, moderado y agresivo. La interpolación entre estos casos daría lugar a los comportamientos intermedios.

Las secciones preliminares son ejemplos que sirven para ilustrar el comportamiento deseado en base al temperamento, para ver el diseño de las secciones utilizadas en el juego, acudir a [10.1 Diseño de secciones](#).

La siguiente leyenda indica el significado del código de colores de las flechas situadas en el trazado:

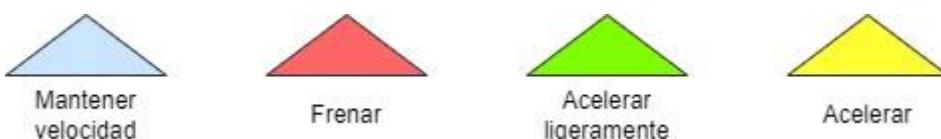


Ilustración 17. Leyenda para el comportamiento del vehículo

A diferencia de la aceleración, que tiene dos niveles de intensidad, no existen variantes de frenado, solo un tipo que supone una reducción gradual de la velocidad.

La imagen inferior muestra los iconos de las notas del código nemotécnico creadas para los bocetos:

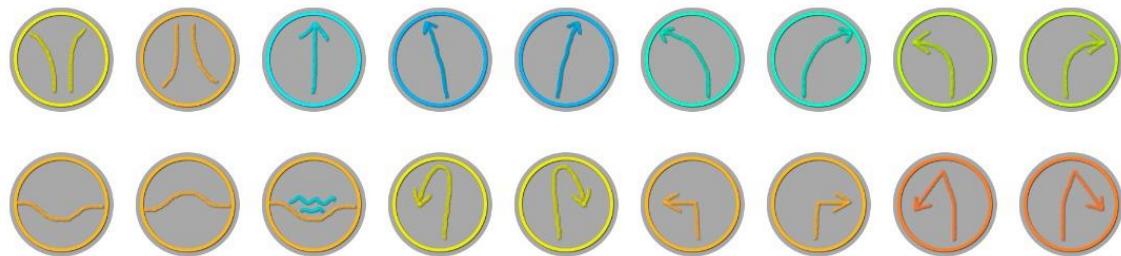


Ilustración 18. Iconos de las instrucciones de tipo de trazado

7.1.1 EJEMPLOS DE POSIBLES TIPOS DE SECCIÓN

En este primer caso se van a mostrar ejemplos más graduales, para que se vea cómo se interpola entre un extremo del *slider* de temperamento y el opuesto.



Ilustración 19. Comportamiento del coche en recta preliminar

Como se observa, a medida que se toma un perfil más agresivo, la aceleración es más fuerte y prolongada.

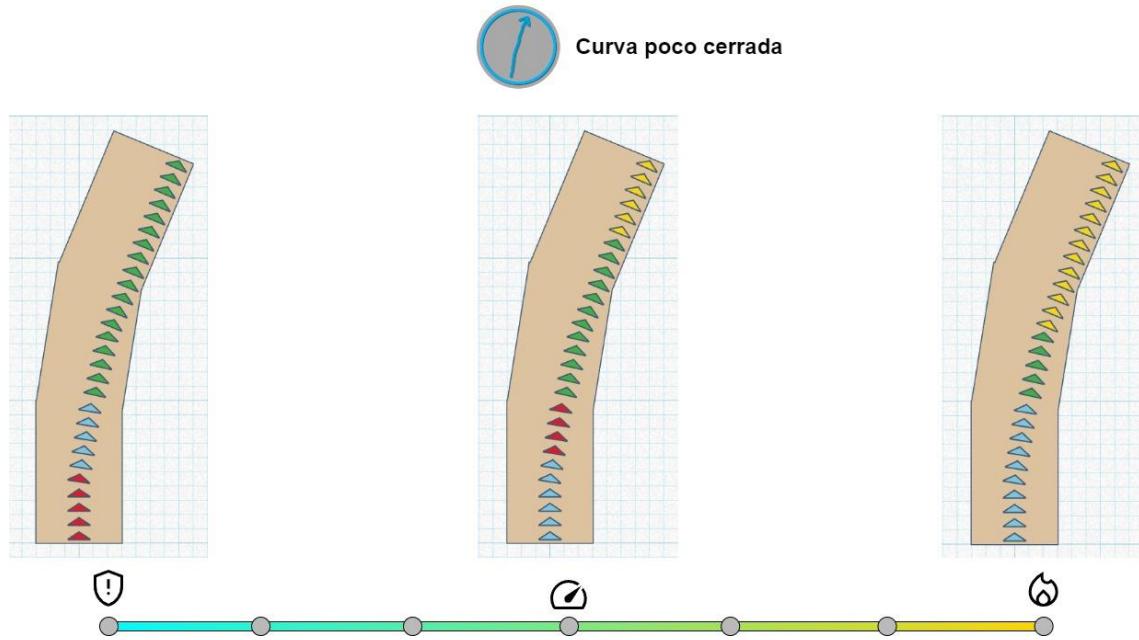


Ilustración 20. Comportamiento del coche en curva poco cerrada preliminar

Aquí se observa por primera vez la presencia de tramos de frenado.

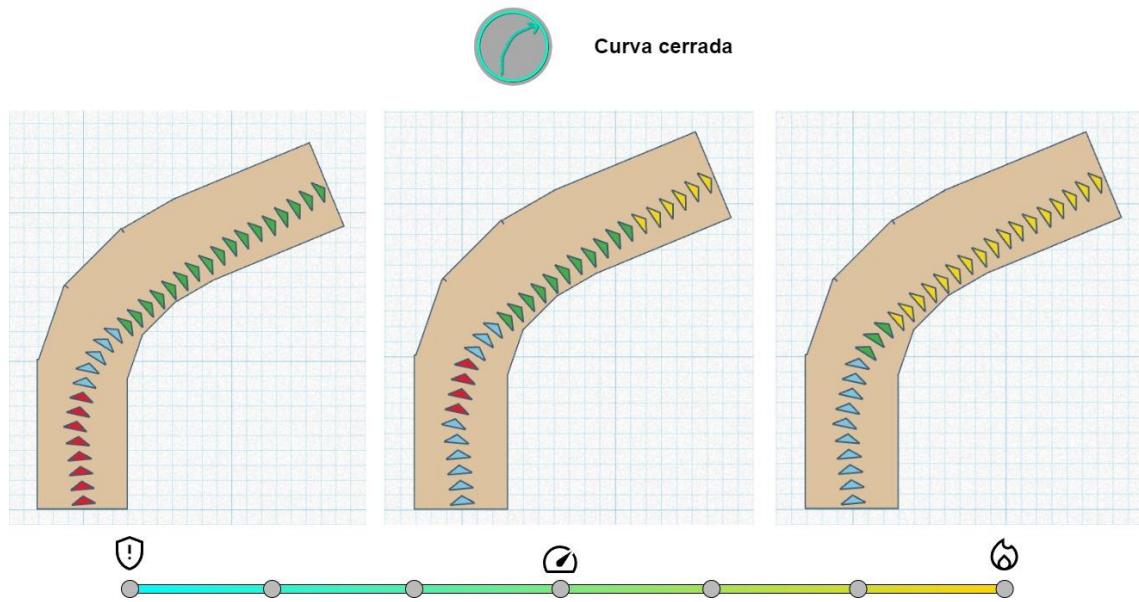
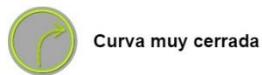


Ilustración 21. Comportamiento del coche en curva cerrada preliminar



Curva muy cerrada

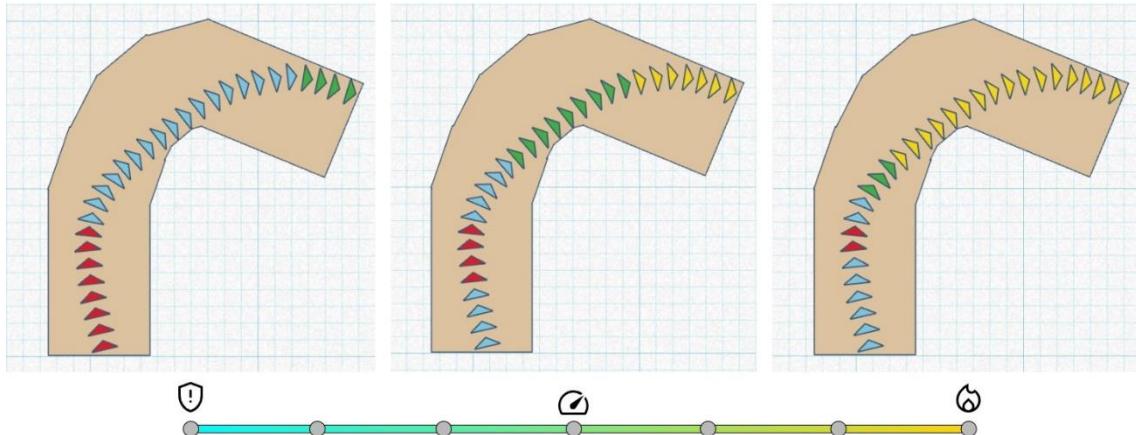


Ilustración 22. Comportamiento del vehículo en curva muy cerrada preliminar



Horquilla

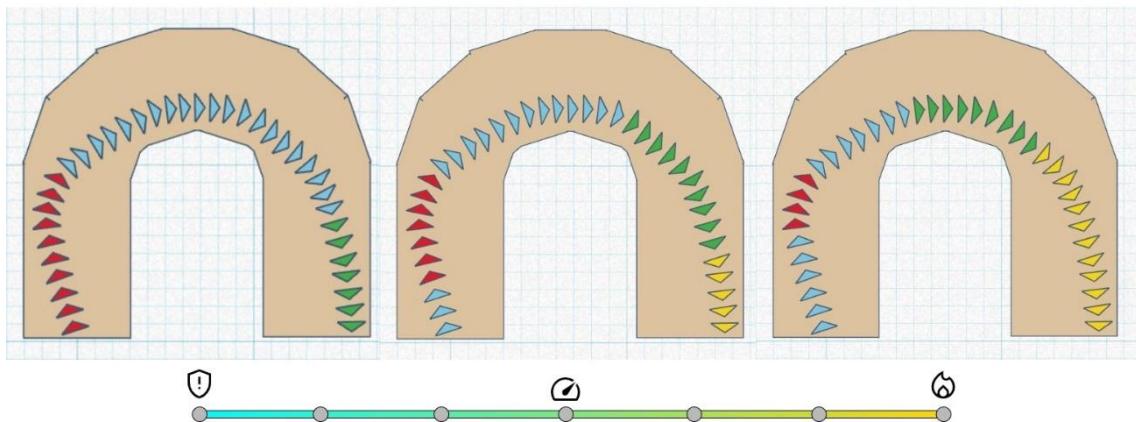


Ilustración 23. Comportamiento del coche en horquilla preliminar

7.2 OBSTÁCULOS

7.2.1 OBSTÁCULOS DE RELIEVE

Para dar mayor diversidad a las etapas y plantear diferentes escenarios aún con secciones repetidas, se distribuyen a lo largo del itinerario obstáculos en forma de variaciones del relieve. Estas variaciones no afectan a cómo afronta el coche cada sección, pues eso está completamente determinado por la nota elegida y el temperamento seleccionado. Lo que hacen estos obstáculos es generar reacciones en las físicas del vehículo que le dificultan seguir el trayecto objetivo o provocan daños.

Es aquí donde entra en juego un buen manejo del temperamento: considerar el relieve de las secciones siguientes para determinar con qué nivel de cuidado o agresividad queremos entrar en cada una. Por ejemplo, si tenemos una recta con un bache a la mitad quizás sea más inteligente entrar con cautela, mientras que si se presenta una pendiente al inicio quizás sea mejor salir de la curva anterior con mayor agresividad para no perder mucha velocidad. Además, las colisiones del vehículo y caídas al dar saltos suponen daños, otro factor a considerar a la hora de enfrentarse a obstáculos en el circuito.

Los obstáculos se distribuyen en tres categorías: baches, depresiones y charcos. En principio estos ocupan todo el ancho de la pista, variando solo en altitud y longitud:

- Los **baches** serán porciones de una sección con una elevación seguida de un descenso. Pueden ser más o menos agudos en función del diseño de nivel elegido. Además, existen dos variantes: **rampas** y **mesetas**.

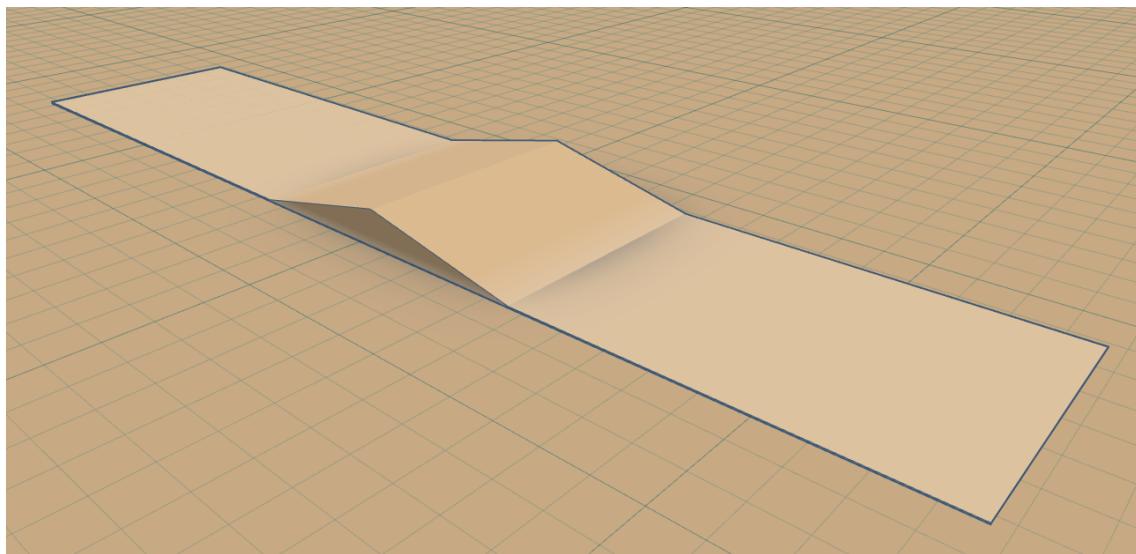


Ilustración 24. Sketch de una rampa preliminar

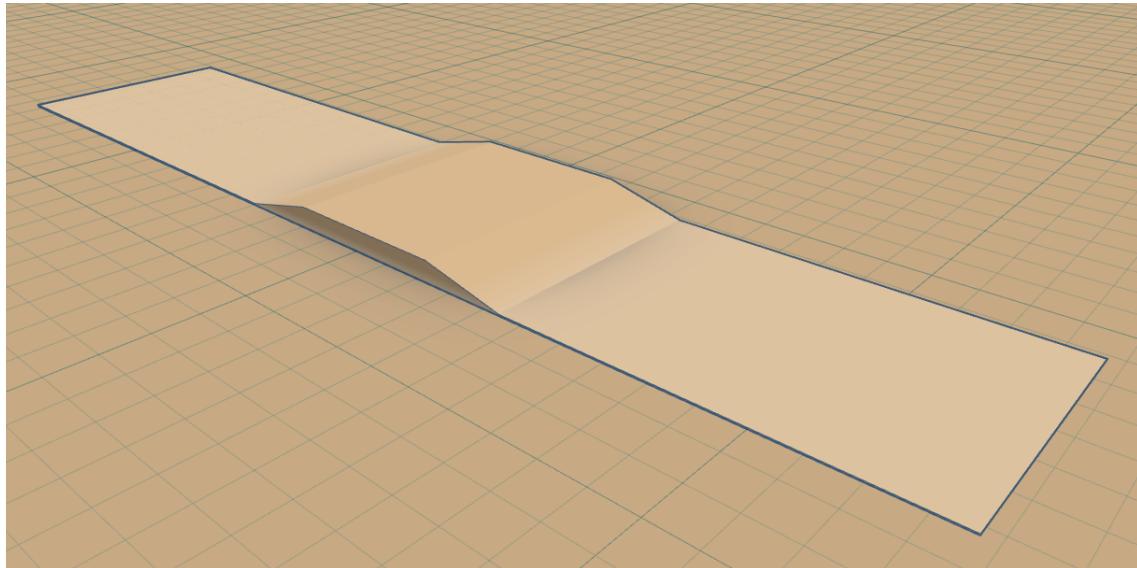


Ilustración 25. Sketch de una meseta preliminar

- Las **depresiones** son justo lo opuesto a los baches, es decir, hundimientos en el terreno de mayor o menor tamaño y longitud, a elección del diseñador de niveles. Las depresiones tienen dos variantes: **valles** y **hundimientos**.

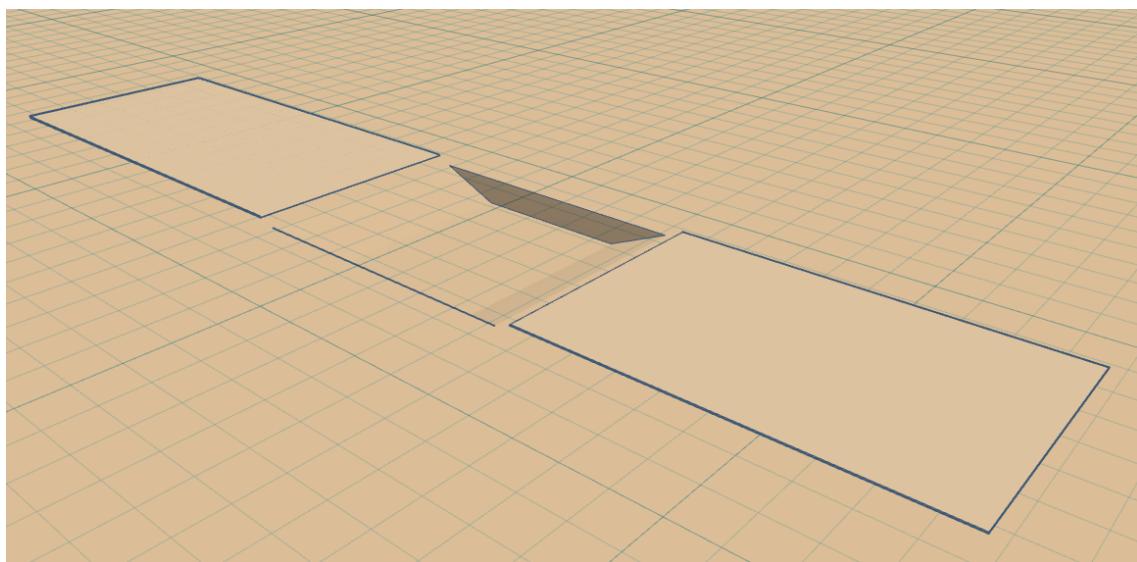


Ilustración 26. Sketch de un valle preliminar

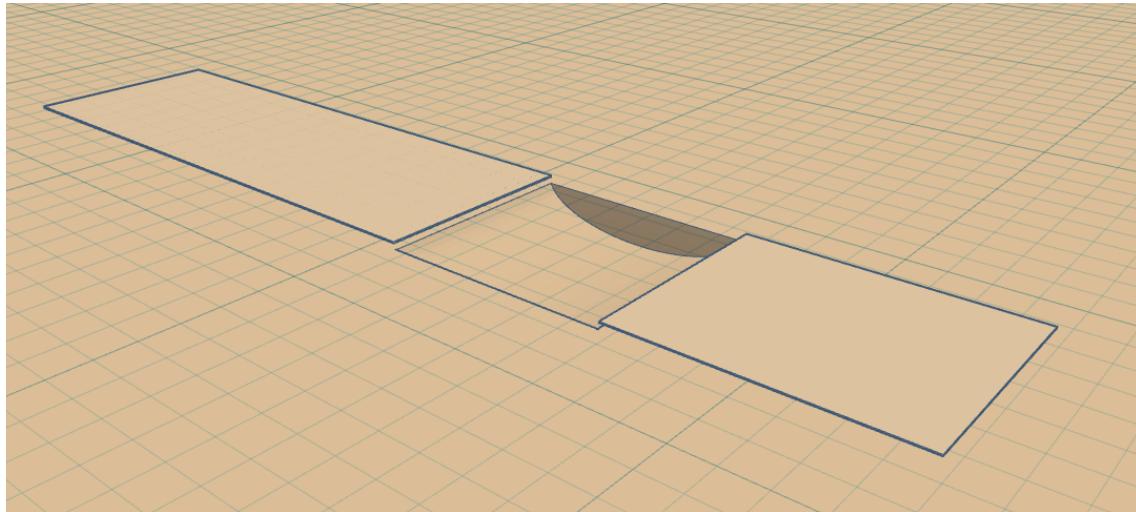


Ilustración 27. Sketch de un hundimiento preliminar

- Los **charcos** son zonas de una sección cubiertas por una fina capa de agua que ralentizan el avance del coche o incluso hacerlo resbalar y perder la dirección en curvas. Realmente se trata de valles de muy poca profundidad que están llenos de agua.

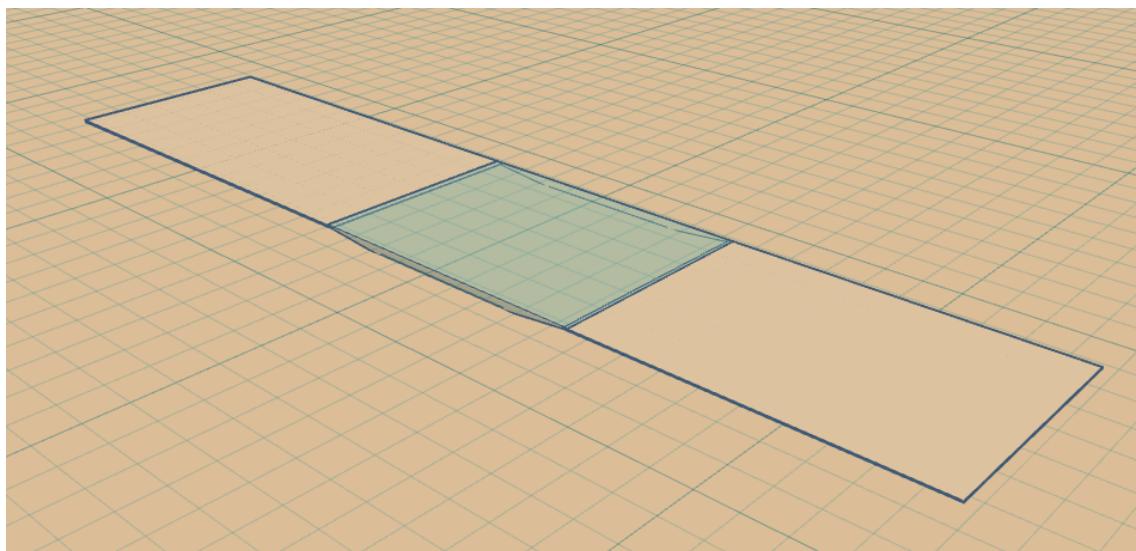


Ilustración 28. Sketch de un charco preliminar

Los diseños finales de estos obstáculos se presentan en [10.3 Diseño de obstáculos](#).

7.2.2 OBSTÁCULOS DEL ENTORNO

Más allá de los obstáculos de tipo relieve, se ha planteado el diseño de obstáculos más tradicionales como piedras, muros o vallas. Este tipo de obstáculos se pueden situar en cualquier punto del trazado, y pueden desestabilizar, ralentizar o causar daños al vehículo. Como se explica más adelante en [8.2 Sistema de daños](#), son los que provocarán la mayor parte de los daños al coche, debido a que generan colisiones horizontales con este.

Es importante mencionar que también se pretende que estos obstáculos en ocasiones incentiven el riesgo o incluso den seguridad al jugador (por ejemplo, con muros no destructibles).

Los diseños finales de estos obstáculos se presentan en [10.3 Diseño de obstáculos](#).

7.2.3 CONCATENACIÓN DE OBSTÁCULOS

Los obstáculos se pueden concatenar.

Pongamos un ejemplo y cómo este supondría una toma de decisión para el jugador: en el *sketch* mostrado a continuación se observa un bache seguido de un charco en una curva poco cerrada:

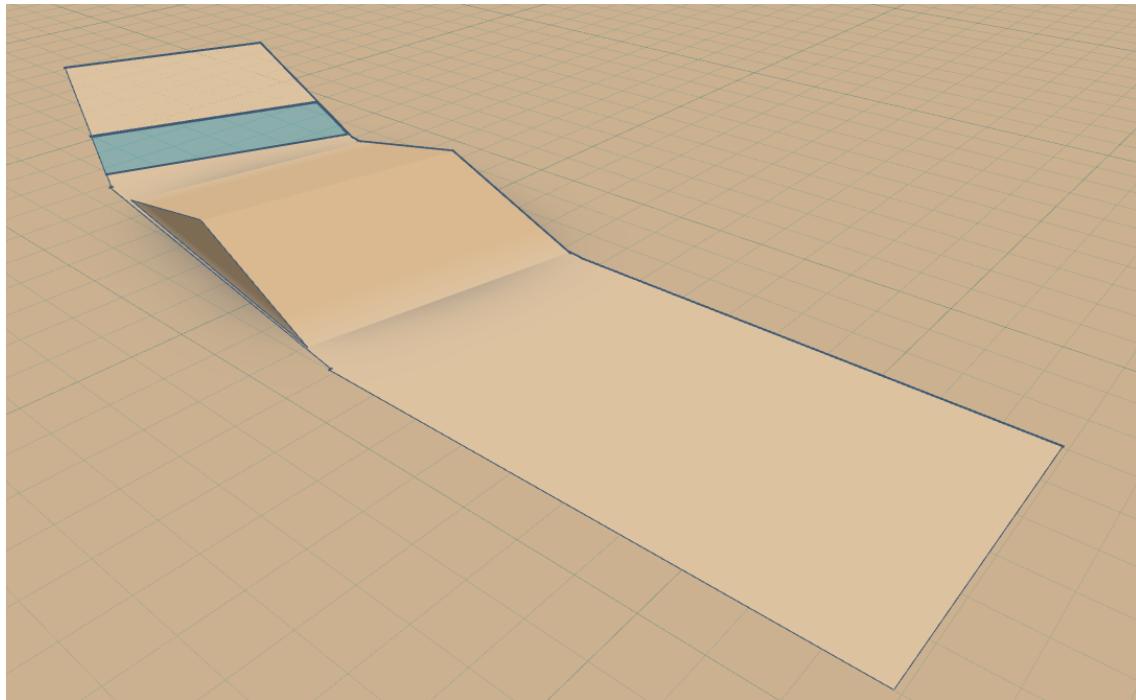


Ilustración 29. Sketch de un bache seguido de un charco en una curva poco cerrada

En este caso, salvo que el coche esté altamente dañado, quizás lo más interesante sería ejecutar la curva de manera agresiva, aprovechando el salto tras la rampa inicial del bache para evitar el charco y así no perder velocidad. Ahora bien, esto puede aplicar solo en caso de que realmente estemos dispuestos a intercambiar daño por un mejor tiempo en esta sección: quizás más adelante hay otro obstáculo que va a deteriorar el coche significativamente y se debe reconsiderar por tanto la estrategia aquí.

8 GESTIÓN DE RECURSOS

Uno de los pilares fundamentales del *gameplay* de *Rally Team Tactics* es el uso adecuado de los recursos provistos al inicio de cada prueba al jugador. Estos son el **combustible en el depósito** y los **daños sufridos**.

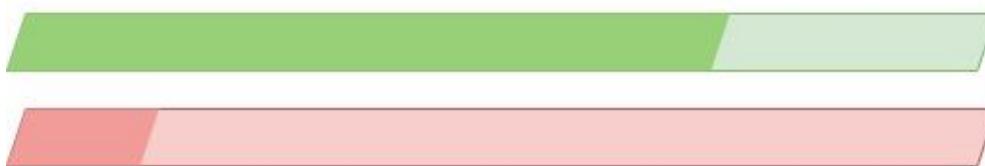


Ilustración 30. Sketch de las barras de combustible y daño

Son representados en el **HUD** como **barras horizontales o verticales**, según la preferencia del diseñador de UI y UX.

Al comenzar o reiniciar una prueba (sea tramo o etapa completa) el combustible del vehículo está siempre al 100%, mientras que el daño comienza al 0%.

Se va a realizar seguidamente un análisis del diseño de la consumición de los recursos descritos: combustible y daño.

8.1 CONSUMO DE COMBUSTIBLE

El combustible se gasta de manera constante si se mantiene la velocidad, a un ritmo constante `FUEL_CONSUMPTION_RATE` y que se mide en unidades de combustible por segundo:

$$\text{FUEL_CONSUMPTION_RATE} = 1 \text{ uds/s}$$

Ahora se presentan los diferentes casos de consumo que usan esta constante:

<i>Cuando...</i>	<i>Ritmo de consumo (uds/s)</i>
<i>Se mantiene la velocidad</i>	<code>FUEL_CONSUMPTION_RATE</code>
<i>Se está frenando</i>	<code>1.5*FUEL_CONSUMPTION_RATE</code>
<i>Se está acelerando poco</i>	<code>1.5*FUEL_CONSUMPTION_RATE</code>
<i>Se está acelerando mucho</i>	<code>2*FUEL_CONSUMPTION_RATE</code>

Tabla 2. Consumo de combustible en función del comportamiento del coche

8.2 SISTEMA DE DAÑOS

8.2.1 DAÑO POR COLISIÓN CON OBSTÁCULOS

Este tipo de daño se produce siempre que el coche colisione con obstáculos, como pequeños muros de piedra o rocas.

Para especificar el daño que sufre el cuando el vehículo se estrella, se define de una constante denominada `BASE_COLLISION_DAMAGE` en unidades de daño:

$$\text{BASE_COLLISION_DAMAGE} = 50 \text{ uds}$$

Se va a explicar paso a paso cómo se calcula el daño para una colisión. Cuando el coche impacta con un obstáculo, se recogen dos parámetros, siendo estos `collisionAngle` y `horizontalSpeed`. La velocidad horizontal se obtiene con facilidad, ya que es el componente x del vector de velocidad, pero el proceso de cálculo del ángulo de colisión es algo más enrevesado:

1. Se extrae el vector que va desde el centro de gravedad del coche hasta el centro del obstáculo.
2. Se computa **el ángulo absoluto entre los componentes x de este vector y el vector de dirección del vehículo**. Si el ángulo resultante fuese de 0 grados, el coche estaría impactando frontalmente, mientras que si supera los 90 grados, se trata de un roce. Eso sí, el ángulo nunca podrá ser mayor a 180 grados.

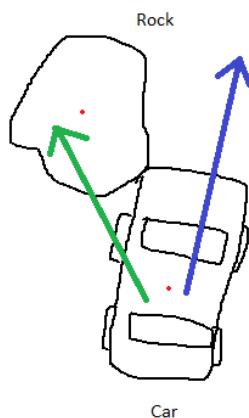


Ilustración 31. Vectores de impacto y dirección

Una vez se cuente con ambas variables, se puede proceder al cómputo del daño recibido. La función debe reflejar que, a mayor sea el ángulo de entrada, el daño debe ser menor. Por ello, debe ser una función inversamente proporcional a este parámetro:

$$\text{damageReceived} = \frac{\text{BASE_COLLISION_DAMAGE}}{\text{collisionAngle}}$$

No obstante, se da ahora el problema de que, si el ángulo de entrada en 0, el daño no se puede calcular (tendería a ∞). La solución escogida consiste en sumar 1 al ángulo de entrada:

$$damageReceived = \frac{BASE_COLLISION_DAMAGE}{collisionAngle + 1}$$

Otro problema que encontrariamos sería que el daño base debería ser muy alto para que el daño recibido fuese considerable. Como se prefiere trabajar con números pequeños, una solución posible es dividir el ángulo de entrada por 22.5:

$$damageReceived = \frac{BASE_COLLISION_DAMAGE}{\frac{collisionAngle}{22.5} + 1}$$

Con esto se consigue que, cuando el coche impacte de frente con un obstáculo, reciba la totalidad del daño base, mientras que cuando lo haga lateralmente, sufra una quinta parte de este.

Por último, es crucial considerar la velocidad en la ecuación. La velocidad debe aumentar o reducir proporcionalmente el daño recibido:

$$damageReceived = \frac{horizontalSpeed}{60} \cdot \frac{BASE_COLLISION_DAMAGE}{\frac{collisionAngle}{22.5} + 1}$$

8.2.2 DAÑO POR CAÍDA

Este tipo de daño se produce cuando el vehículo impacta con el terreno después de un salto al entrar o salir de una variación del relieve de la pista.

Para especificar el daño que sufre el cuando el vehículo choca con el suelo, se define de una constante denominada `BASE_FALL_DAMAGE` en unidades de daño:

$$BASE_FALL_DAMAGE = 30 \text{ uds}$$

Se tiene en cuenta en este caso el ángulo de caída, `fallAngle`, ángulo absoluto entre el vector de dirección del coche y el vector global hacia abajo, $(0, -1, 0)$. La velocidad que se necesita es también distinta, pues se trata del componente y del vector de velocidad del coche, que llamaremos `verticalSpeed`.

La fórmula de cálculo del daño muy diferente. Si cayésemos con un ángulo de 180 grados el coche estaría impactando de manera perpendicular con el terreno, lo cual debería causar significativamente más daño. Debería pasar lo mismo si caemos con la parte frontal, es decir, con un ángulo de 0 grados. La función debe alcanzar su máximo en estos dos casos. Por ello, se utiliza un exponente y se centra la función a 90 grados:

$$damageReceived = BASE_FALL_DAMAGE \cdot \left(\frac{fallAngle - 90}{90} \right)^2$$

Sin embargo, también se debe considerar que el vehículo sufrirá daño incluso cuando caiga de manera paralela al terreno, pues la amortiguación no absorberá todo el impacto:

damageReceived

$$= \text{BASE_FALL_DAMAGE} \cdot \left(\frac{\text{fallAngle} - 90}{90} \right)^2 + \frac{\text{BASE_FALL_DAMAGE}}{6}$$

Finalmente, se debe tener en cuenta la velocidad vertical para escalar el daño:

damageReceived

$$= \frac{\text{verticalSpeed}}{5} \cdot (\text{BASE_FALL_DAMAGE} \cdot \left(\frac{\text{fallAngle} - 90}{90} \right)^2 + \frac{\text{BASE_FALL_DAMAGE}}{6})$$

8.2.3 UMBRAL MÍNIMO PARA SUFRIR DAÑOS

Hasta cierto punto el coche debe ser capaz de absorber el daño. Para evitar que sufra daños en situaciones donde apenas se levanta o simplemente roza un obstáculo, se ha decidido establecer un daño mínimo que *damageReceived* debe superar o igualar para poder sumarse al daño total sufrido. Este daño umbral lo denominaremos *MIN_DAMAGE*:

$$\text{MIN_DAMAGE} = 5$$

IF (damageReceived ≥ 5) THEN (damage += damageReceived)

8.3 AGOTAMIENTO DE RECURSOS

El consumo de todo el combustible o sufrir un daño irreparable supondría (alcanzar el 0% o el 100%, respectivamente) la **detención del coche y forzarían a reiniciar la prueba o salir de esta.**

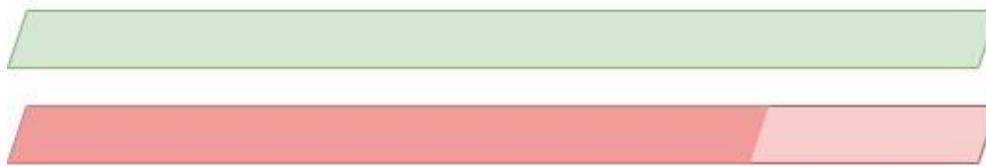


Ilustración 32. Sketch de las barras recursos: sin combustible

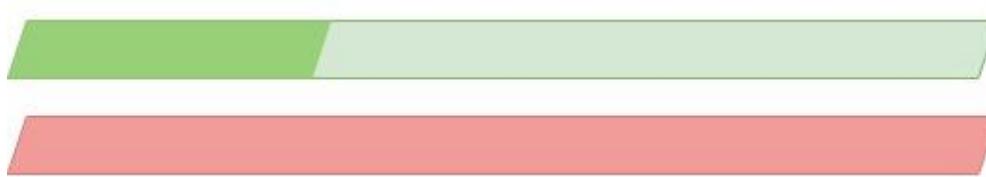


Ilustración 33. Sketch de las barras recursos: coche destrozado

8.4 RECURSOS ENTRE PRUEBAS

Como el coche se repara y recarga gasolina siempre que terminamos una prueba, se debe mostrar al jugador encima de cada una cuánto daño sufrió y cuánto combustible consumió en su último intento guardado.

Esto permite al jugador considerar, por ejemplo, si su desempeño en un tramo es apropiado para enfrentarse a la etapa completa. Recordemos que la etapa completa incluye todos los tramos de los días/pruebas anteriores, y como se va a disponer de la misma cantidad de combustible y la misma resistencia que en los tramos separados, el jugador debe aprender no solo obtener buenos tiempos en los tramos, sino a hacerlo con un consumo de combustible y daños que sea proporcional a su dificultad y que lleve a una gestión óptima de recursos en la prueba final.

9 PROGRESIÓN

9.1 OBJETIVOS DEL JUGADOR

El objetivo del jugador es conseguir el mejor tiempo en cada etapa de *rally*, que presenta un trazado distinto y una ambientación única. Como se ha explicado con detenimiento en apartados anteriores, para lograr dicho tiempo, el jugador deberá ser capaz de seleccionar las instrucciones y el temperamento en cada sección que produzcan los mejores resultados.

Gestionar los recursos adecuadamente es crucial para llegar a la meta en primer lugar, por tanto, un manejo inteligente de estos puede dar mejores resultados.

9.2 ESTRUCTURA DEL JUEGO

El juego se divide en varios **eventos**, pero el progreso no es lineal, es decir, se pueden completar en el orden que se desee. La etapa de cada evento presenta su propio itinerario y características, no necesariamente suponiendo un incremento o decremento de dificultad. Son simplemente distintos niveles que se pueden jugar en cualquier orden en función de la preferencia del jugador.

Para preparar al jugador de cara a una **etapa completa**, que se compone de **tres tramos cronometrados**, cada evento se divide **cuatro días**, cada uno albergando **una prueba**:

- Los tres primeros días ponen al jugador en cada uno de los tramos en orden de aparición en la etapa. Una vez que se completa un día correspondiente a un tramo, se guarda el resultado en tiempo y recursos y se desbloquea el siguiente día si no ha sido desbloqueado ya. Se pueden intentar estos tramos repetidas veces antes o después de la etapa final, son pruebas preparatorias. Los tramos se componen de **diez secciones de pista**.
- Cuando se comienza prueba de la etapa completa el último día, los puntos de control estarán ya por defecto elegidos tal y como quedaron en los tramos por separado (cuando se intentaron por última vez).

Los siguientes esquemas muestran la estructura de cada día de prueba cronometrada de tramo y de un día de prueba cronometrada de etapa completa:

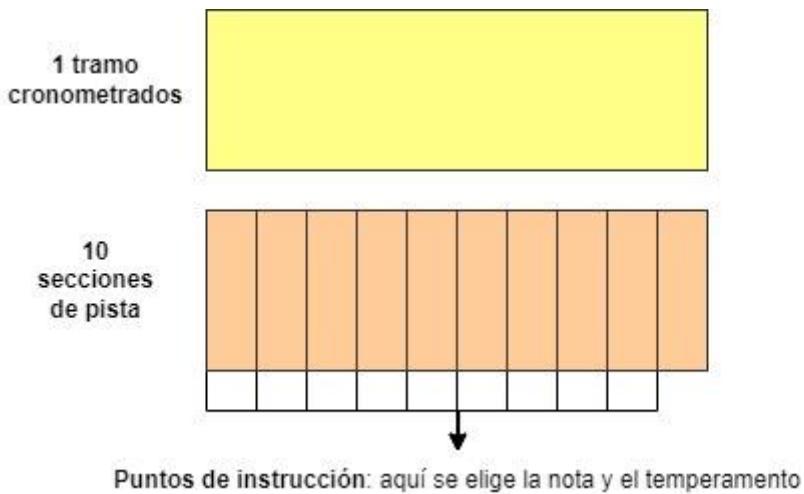


Ilustración 34. Estructura de un tramo cronometrado (días/pruebas 1 a 3)

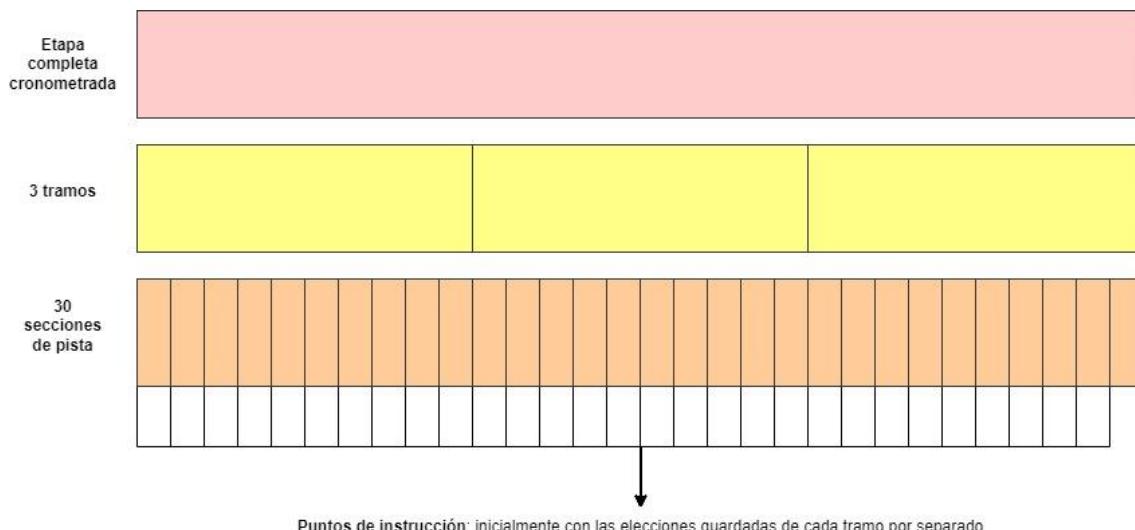


Ilustración 35. Estructura de una etapa completa cronometrada (último día/prueba)

A continuación, se presentan varios esquemas que ilustran la estructura del juego y de cada evento:

Estructura del juego

Supongamos que se ha completado C, se debe hacer visible al jugador. En todos los casos, se muestra el porcentaje de completitud, se podría reforzar el número con el borde del selector.



Ilustración 36. Estructura del juego

Estructura de un evento

Solo se puede acceder al día siguiente si se ha completado el anterior. Una vez completado un día, se guarda el tiempo, combustible gastado y daño sufrido en este. Se mostrará entonces, y cambiará cada vez que se vuelva a intentar, con el último resultado. En el caso de la etapa completa, se mostrará siempre el mejor resultado junto al último

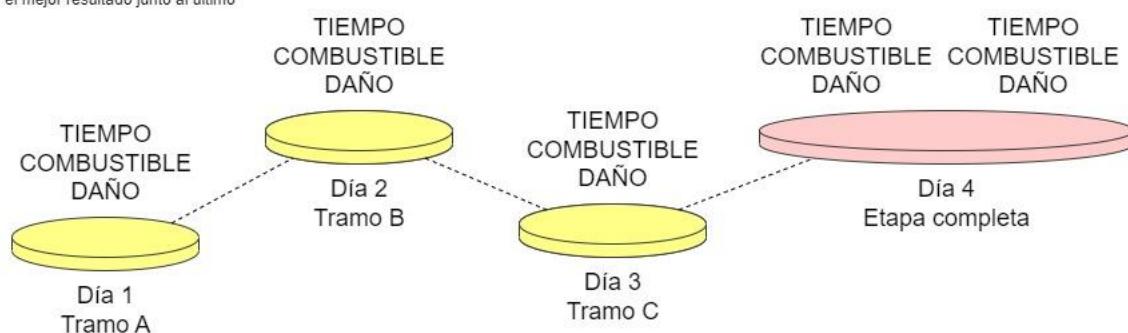


Ilustración 37. Estructura de un evento

9.3 MÉTRICAS: GUARDADO DE DATOS Y CLASIFICACIÓN EN LÍNEA

Una de las funcionalidades más importantes del juego es la posibilidad de realizar analíticas mediante la recogida de métricas. Se debe conocer información sobre el usuario, como su nombre, edad, sexo y contraseña para darlo de alta y que así pueda iniciar sesión en próximas ocasiones. Una vez el usuario está registrado, este tiene la posibilidad de optar (si así lo requiere el centro educativo) a la recolección de información sobre sus resultados en el juego. Estos datos, que permiten medir las destrezas del jugador, son el mejor tiempo, el combustible consumido y el daño sufrido en cada prueba.

Para poder almacenar y actualizar toda la información de sesión (usuario, contraseña, sexo, etc.) y todos los resultados, se ha hecho uso de una base de datos. Además, gracias a esto, con un servidor, se podría tener guardada la clasificación en función del tiempo que se mostraría al jugador al terminar cada prueba.

9.3.1 BASE DE DATOS

Para dar funcionalidad a la BD, se han utilizado clases dedicadas, *MySQLManager* y archivos .php. Cabe recalcar que la BD se ha implementado con *MySQL*.

TABLAS DE LA BD

La base de datos tiene dos tablas:

- *User*: guarda toda la información del usuario, es decir, nombre, contraseña (cifrada), edad y sexo. Tiene como clave primaria el nombre de usuario (único, no es necesario un id).
- *Classification*: es una tabla que recoge el registro de un usuario en cada prueba que completa. Consta de un nombre de usuario asociado a un evento, una prueba dentro de este y las métricas a guardar, siendo estas el tiempo (mejor histórico), el daño y el combustible restante. Esta tabla almacena el mejor resultado en base al tiempo, por lo que, de repetir una determinada prueba, solo se actualizaría la tabla si se consigue un mejor tiempo.

CLASES DEDICADAS EN UNITY

En cuanto a las clases dedicadas, se explica a continuación su función en la BD:

- *UserData*: clase que guarda el nombre de usuario activo. Este nombre debe ser accesible desde cualquier escena, por ello, esta clase debe ser adjuntada a un objeto (aunque esté vacío) no destructible entre escenas. Además, esta clase también contiene funciones para comunicarse con la BD, como iniciar sesión, registrarse o devolver todos los nombres de los usuarios registrados.
- *Classification*: por un lado, esta clase copia la estructura de la tabla correspondiente en la base de datos (no hace falta que esté adjuntada a ningún objeto). Por otro, posee llamadas a la BD para modificarla. La clase se crea al inicio de una prueba y guarda las métricas al completarla (si precisa). También sirve para mostrar el tiempo en la pantalla de selección de prueba y la clasificación en la escena de *leaderboards*.

MYSQLMANAGER

Cuando en el apartado anterior se mencionaba que una determinada clase se comunica con la BD, en realidad lo que haría esta sería realizar llamadas a funciones asíncronas de propias de *MySQLManager*. Estas funciones mandan información a los archivos de comunicación .php.

La clase envía un diccionario con información a un archivo PHP, esperando después una respuesta. Si es necesario, puede devolver una lista de *String* o de *Classification*, siendo las funciones públicas las encargadas de retornar solo lo imprescindible.

Además, esta clase tiene implementada la conexión con la BD mediante una variable *String*, la URL de esta última. De esta forma, si la base de datos se muda, solo es necesario cambiar esta cadena de caracteres.

ARCHIVOS PHP

Los archivos escritos en PHP son quienes conectan y envían *queries* a la base de datos. Se sitúan dentro de la BD, aunque se pueden escribir primero en local para su posterior subida. Aunque cada archivo es diferente, hay varias características en común:

- En primer lugar PHP define el nombre de la BD, el usuario y la contraseña (como si una persona estuviera accediendo a ella a través de un navegador), y la tabla a la que va a mandar información. Se conecta entonces a la BD.
- Despues, el archivo detecta los datos recibidos, es decir, el diccionario de enviado por Unity. Extrae la información que requiere y la guarda en variables para crear la *query*.
- Posteriormente, redacta la *query* correspondiente, para insertar, borrar, actualizar u obtener valores. La *query* se guarda en una variable de texto, pero PHP permite detectar las palabras clave para MySQL. Pasa así el contenido de la variable como argumento.
- Finalmente, comprueba que la *query* ha sido exitosa (valor devuelto por la BD) y devuelve la información a Unity para que la procese. En la mayoría de *queries*, se devuelve una pequeña frase que el motor de juegos no procesa (se puede mostrar por consola para *debugging*). Ahora bien, si se seleccionan una o más líneas, entonces el archivo PHP crea una variable de texto que separa los atributos con espacios y las líneas con tabulaciones. Cuando la información solicitada es devuelta a Unity, es *MySQLManager* quien se encarga de gestionarla.
- También cabe mencionar que los archivos PHP implementan una sentencia para terminar la conexión devolviendo un texto. La función *die()* devuelve códigos de error como 2 - error de conexión a la BD, entre otros. Estos códigos son leídos únicamente por *MySQLManager*, y se pueden mostrar por consola.

10 DISEÑO DE NIVELES

Se describe en este capítulo con imágenes el diseño de las piezas utilizadas en el juego final, es decir, las secciones modeladas y que componen las diferentes pruebas de RTT. Posteriormente, se muestran los diseños de los diferentes eventos, desglosados en tramos (tres primeras pruebas), seguidos de la etapa completa (última prueba del evento).

10.1 DISEÑO DE SECCIONES

Cada circuito del juego se compone por secciones predefinidas que abarcan una variedad de formas, desde rectas hasta horquillas. Cada una de las secciones con curvatura cuenta con dos versiones, una hacia cada dirección, y por tanto se corresponde con dos notas de código nemotécnico (izquierda y derecha). La estrategia a la hora de abordar estas secciones no depende únicamente de elegir la nota correcta, sino también de la gran variabilidad determinada por el sistema de temperamento explicado en [7 Control y comportamiento del vehículo](#).

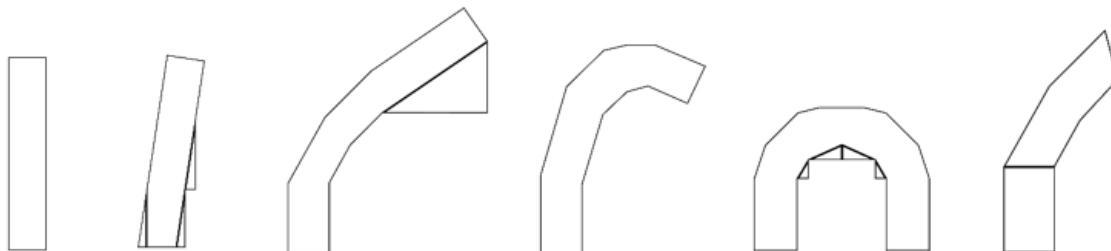


Ilustración 38. Diseño final de las secciones de trazado

10.2 DISEÑO DE ETAPAS

En cuanto al diseño de las etapas, se ha buscado una distribución de secciones y división en tramos que proporcione una progresión lo más natural y desafiante posible. Como se explicó en [9.2 Estructura del juego](#), cada etapa se compone de tres tramos, y a su vez, cada tramo se compone de diez secciones. Cada sección está vinculada a una nota de código nemotécnico concreta, asociación que se pretende que el jugador aprenda. La experimentación más profunda viene de la variabilidad dada por la elección del temperamento.

Tras cada conjunto de secciones se ha integrado una sección marcada en amarillo en los esquemas inferiores, que representa una **sección punto de control**. Esta sección está predeterminada, reconduce el coche hacia el centro de manera automática y sirve como respiro al jugador, pero el papel que juega en la fase de reconocimiento es lo más relevante: es en esta sección donde el usuario elige las instrucciones de manera conjunta y premeditada para todas las secciones hasta el siguiente punto de control. Este enfoque fomenta la exploración de soluciones de medio a largo plazo, incrementando así el desafío del juego y dando la posibilidad de poner a prueba el Pensamiento Computacional del jugador en situaciones cada vez más complejas.

10.2.1 PRIMER EVENTO

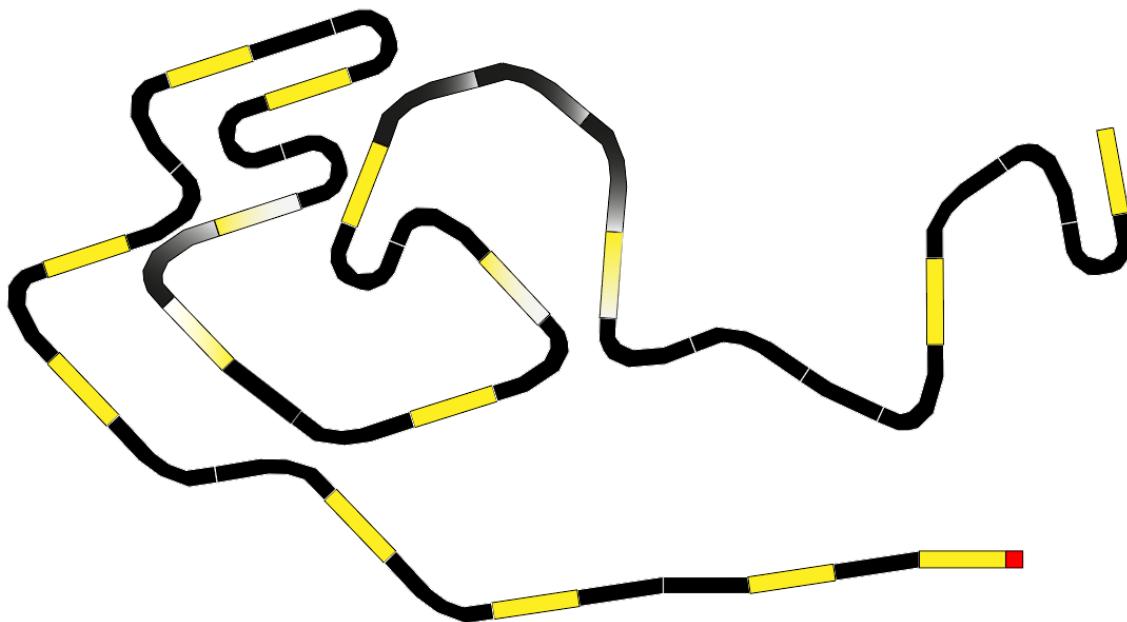


Ilustración 39. Esquema de diseño de la etapa completa del primer evento

La representación esquemática diferencia secciones en negro, las cuales son susceptibles a recibir las instrucciones del jugador. Estas están situadas en secuencias más o menos grandes después de puntos de control (secciones amarillas). El hecho de que el jugador no tenga que seleccionar instrucción para todas las secciones sino que lo haga para las que siguen a cada punto de control de manera conjunta tiene como objetivo aliviar el potencial tedio que supondría la primera aproximación.

Como se puede apreciar, las secciones amarillas son rectas. Estas secciones, designadas como **zonas de aceleración**, están estratégicamente ubicadas para controlar la velocidad máxima que el jugador puede alcanzar al comienzo de cada punto de control. Esta implementación de secciones de aceleración no solo contribuye a incentivar la gestión táctica de la velocidad, sino que también introduce un elemento estratégico adicional que permite optimizar el rendimiento del vehículo al comenzar cada punto de control.

A continuación se presentan de manera individual los tramos que componen la etapa completa y que se deben acometer los tres primeros días del evento, explicando sus particularidades si procede:

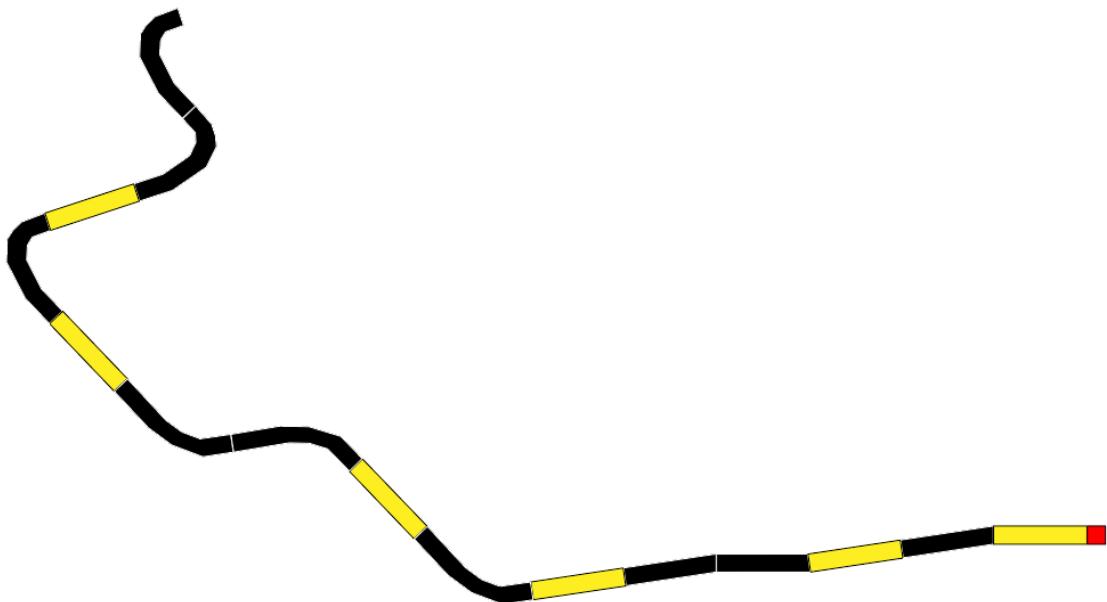


Ilustración 40. Esquema de diseño del tramo 1 del primer evento

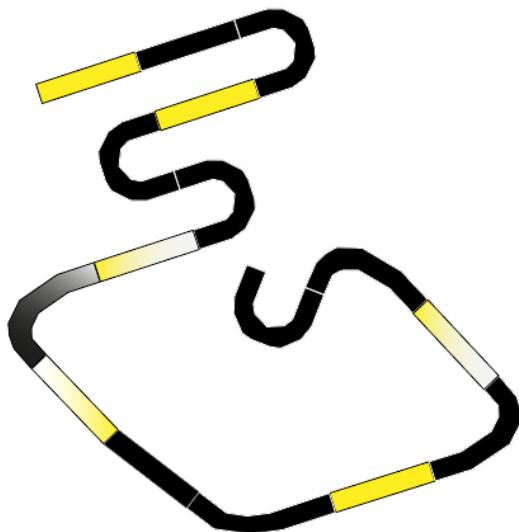


Ilustración 41. Esquema de diseño del tramo 2 del primer evento

A partir del segundo tramo del primer evento, se incorporan cambios de altitud. Esto plantea un desafío añadido al jugador, quien deberá regular la velocidad de manera proporcional al desnivel. Este ajuste es crucial si se quieren evitar salidas o colisiones.

Como se puede ver en el diseño, algunas secciones están marcadas con un degradado: esto simboliza cambios en el nivel del terreno. De claro a oscuro indica subida, el inverso bajada. Esta representación visual proporciona información intuitiva sobre la topografía del circuito que da al jugador la posibilidad de anticipar y ajustar su estrategia de manera acorde.

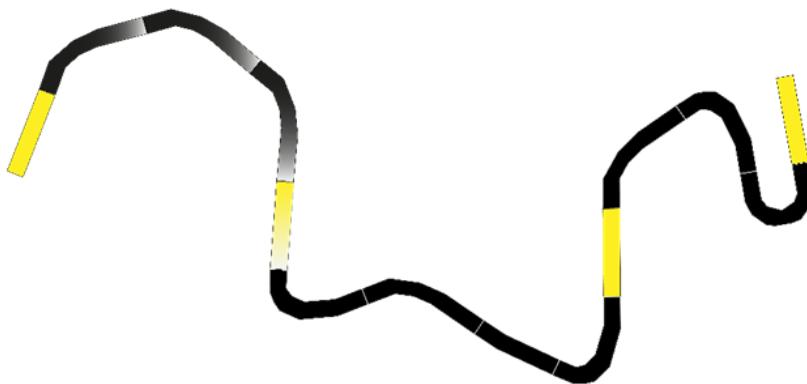


Ilustración 42. Esquema de diseño del tramo 3 del primer evento

Como se puede observar en el diseño de esta primera etapa, se ha concebido como tutorial para introducir progresivamente al jugador a los distintos tipos de sección y distintas secuencias. Las combinaciones son más largas a medida que se progresó en el evento, pero la dificultad se mantiene baja, permitiendo así la incorporación gradual de los primeros obstáculos en el recorrido. Se busca de esta manera una curva de aprendizaje razonable y cuidada que facilite la asimilación de las mecánicas de juego y la preparación efectiva para los posteriores desafíos, de dificultad superior.

10.2.2 SEGUNDO EVENTO

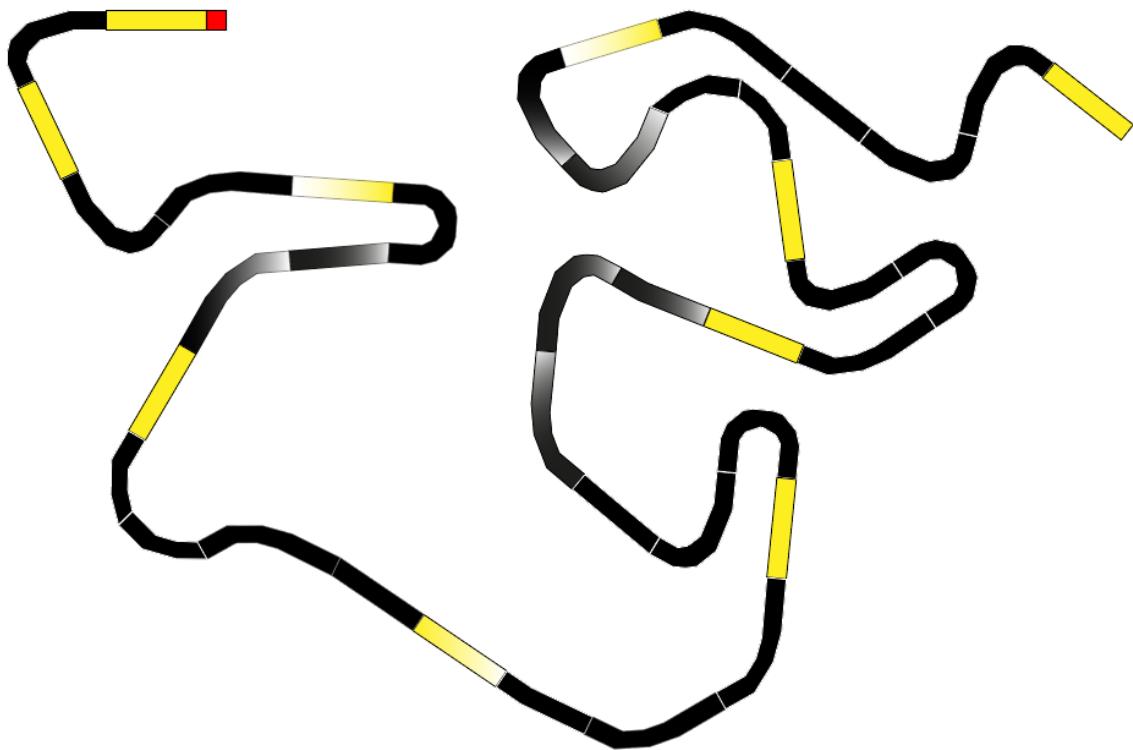


Ilustración 43. Esquema de diseño de la etapa completa del segundo evento

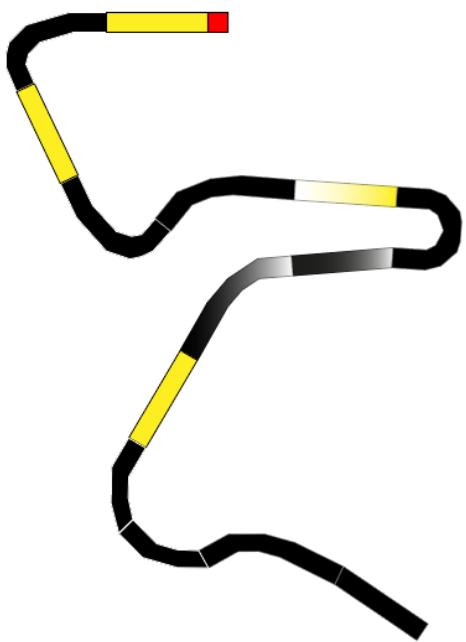


Ilustración 44. Esquema de diseño del tramo 1 del segundo evento

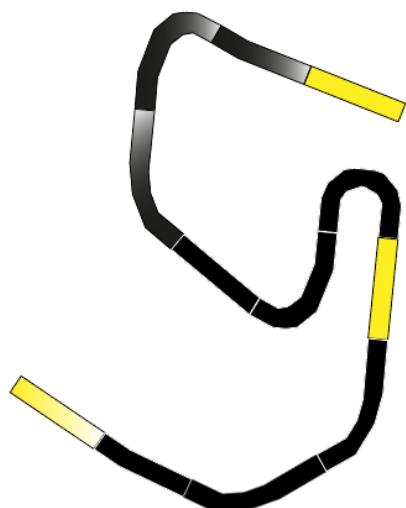


Ilustración 45. Esquema de diseño del tramo 2 del segundo evento

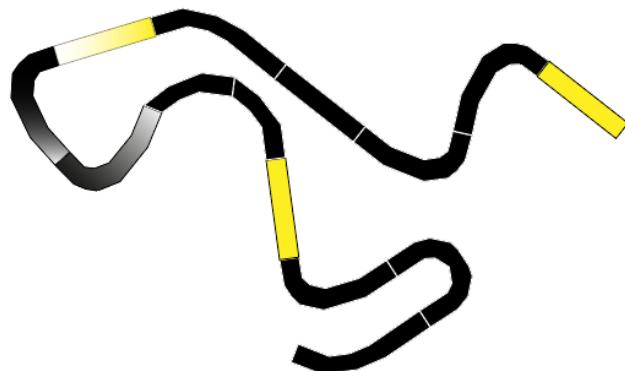


Ilustración 46. Esquema de diseño del tramo 3 del segundo evento

Como se ve en los esquemas superiores, la segunda etapa está un escalón por encima de la primera en términos de dificultad. Ahora se incorporan más cambios de altitud en el trazado y más alternancia. Esto plantea al jugador un desafío adicional al tener que considerar con mayor atención el temperamento y el gasto de combustible, regulando la velocidad adecuadamente de acuerdo con los desniveles. Este ajuste es crucial para evitar salirse de la pista o entrar en colisión con obstáculos.

10.2.3 TERCERA ETAPA

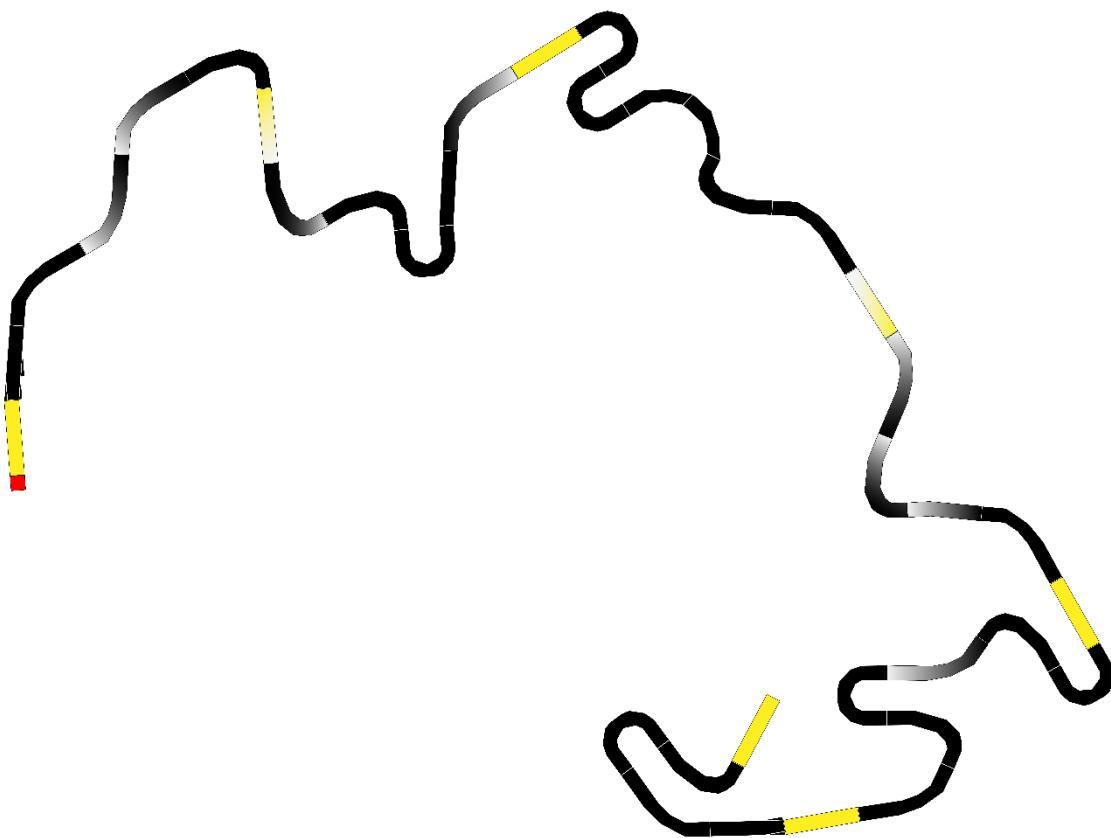


Ilustración 47. Esquema de diseño de la etapa completa del tercer evento

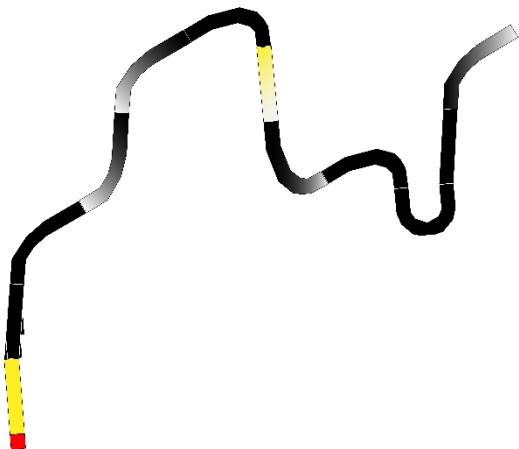


Ilustración 48. Esquema de diseño del tramo 1 del tercer evento

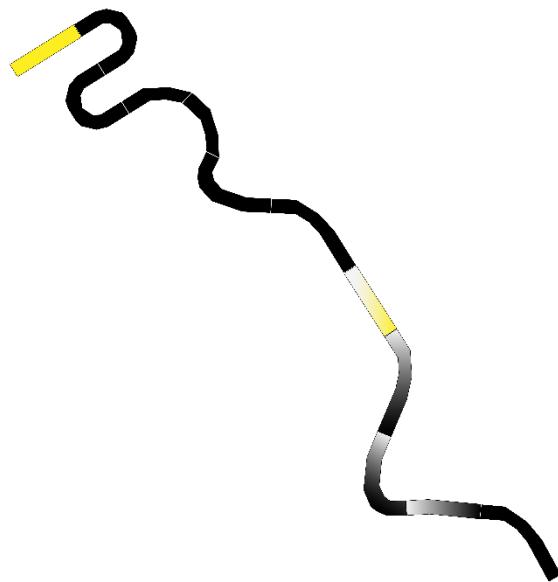


Ilustración 49. Esquema de diseño del tramo 2 del tercer evento

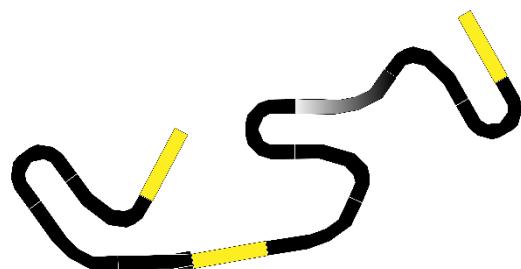


Ilustración 50. Esquema de diseño del tramo 3 del tercer evento

10.3 DISEÑO DE ETAPAS: OBSTÁCULOS

En relación a los obstáculos, se plantea la incorporación de elementos que modifiquen la vía, desde elevaciones y depresiones como las descritas en [7.2.1 Obstáculos de relieve](#) hasta otros elementos como vallas (brevemente explicados en [7.2.2 Obstáculos del entorno](#)).

La introducción a los obstáculos de diferentes tipos y clases debe ser gradual para que los jugadores se familiaricen primero con las mecánicas de conducción básicas. Por ello, en los primeros eventos, se limita la presencia de obstáculos a vallas ubicadas en el borde de algunas curvas. No obstante, según el jugador vaya avanzando, las vallas aparecen de manera más ajustada a los límites de la pista. De esta manera, si el coche va a salir de los límites de la pista, sufre daños pero tiene la oportunidad de corregir su trayectoria. Este enfoque pretende ofrecer una transición suave hacia decisiones de mayor riesgo, dando al jugador la posibilidad de experimentar bajo menos presión durante sus primeras horas en *RTT*.

A continuación se muestran las etapas incorporando las vallas en los límites de la pista (líneas rojas):

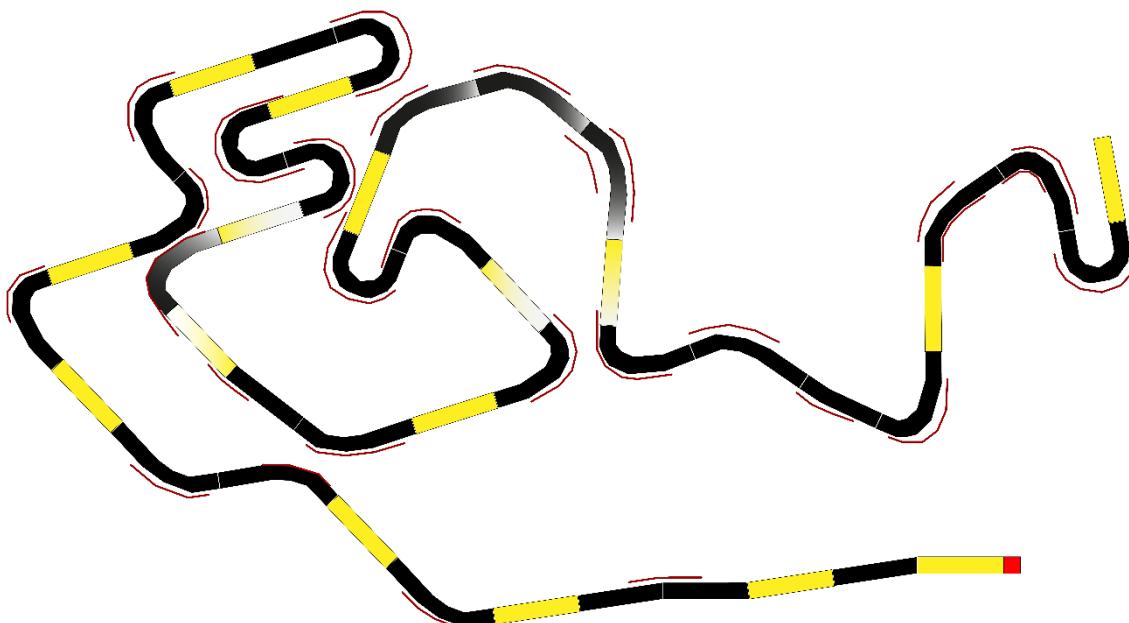


Ilustración 51. Esquema de diseño de la etapa completa del primer evento con obstáculos

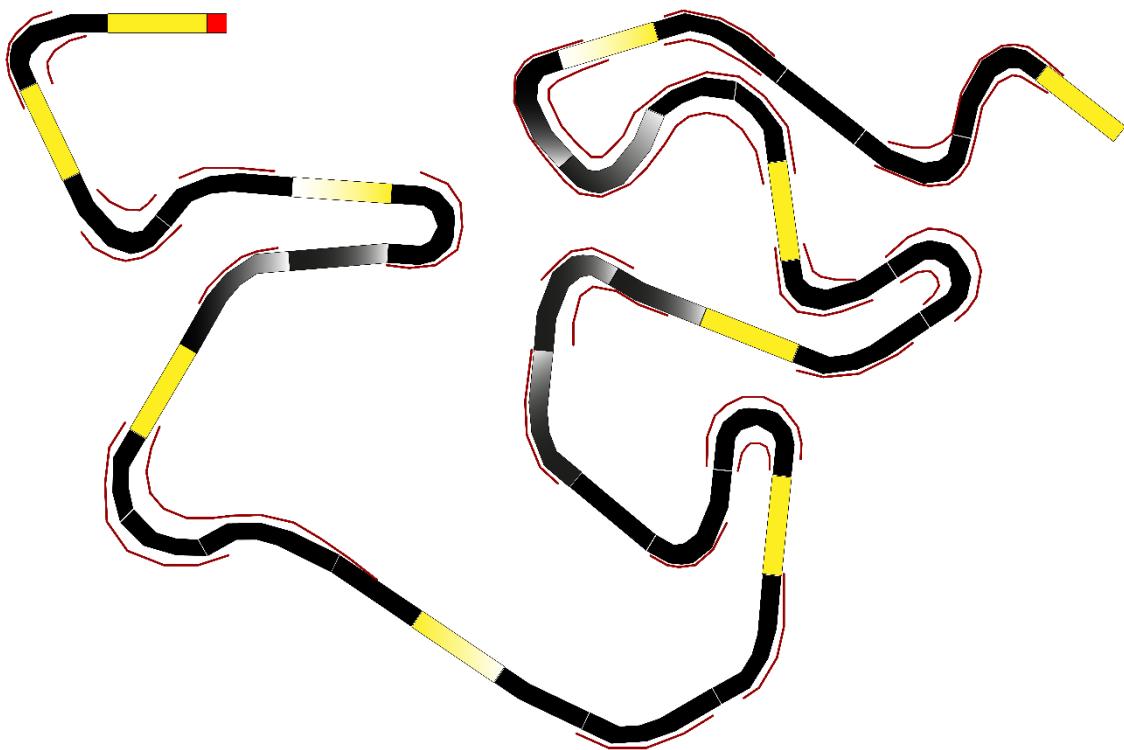


Ilustración 52. Esquema de diseño de la etapa completa del segundo evento con obstáculos

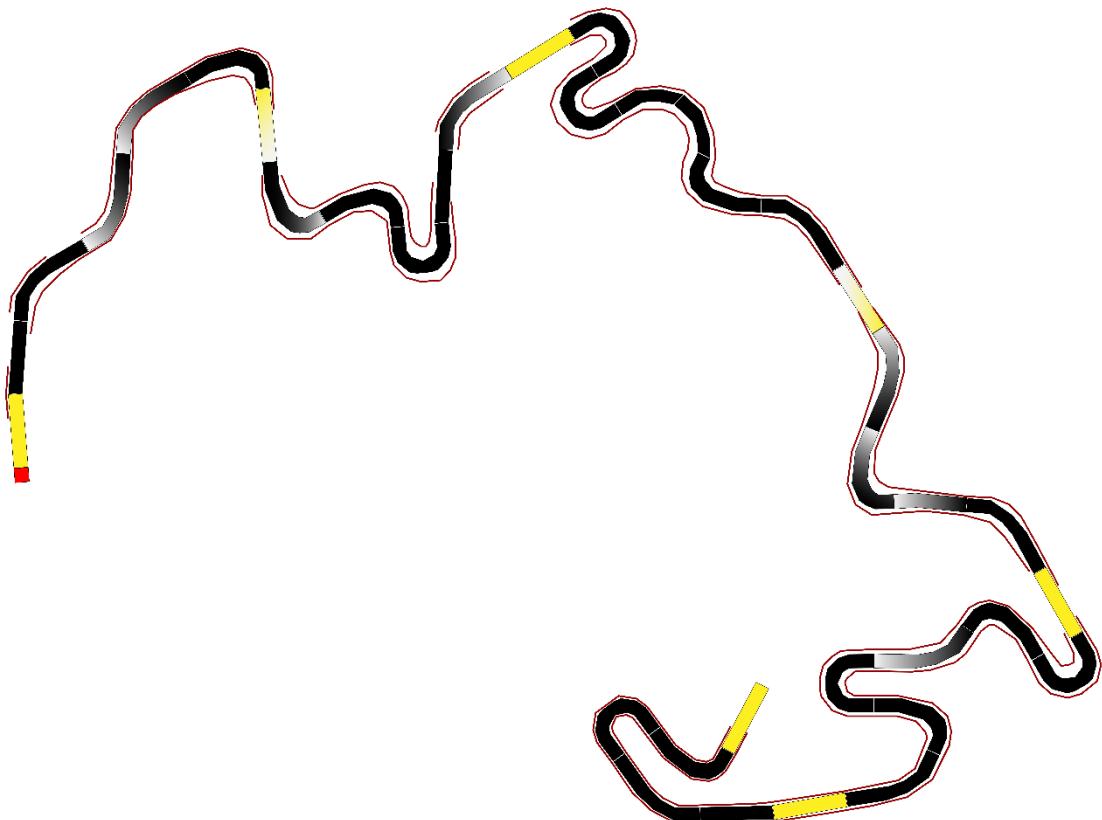


Ilustración 53. Esquema de diseño de la etapa completa del tercer evento con obstáculos

11 CONTROLES

11.1 RATÓN Y TECLADO (PC Y WEB)

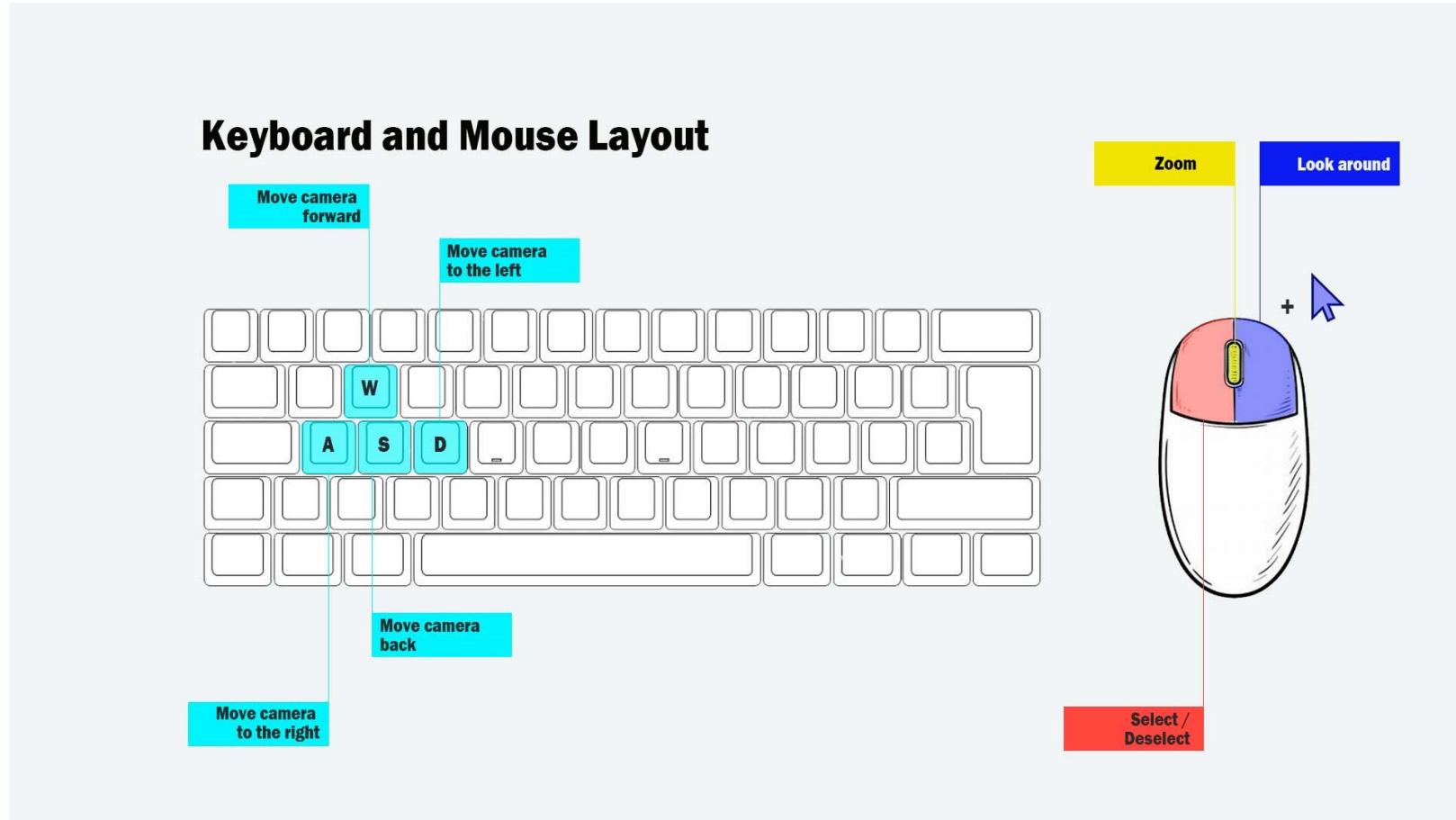


Ilustración 54. Controles con teclado y ratón

11.2 TÁCTILES (ANDROID)

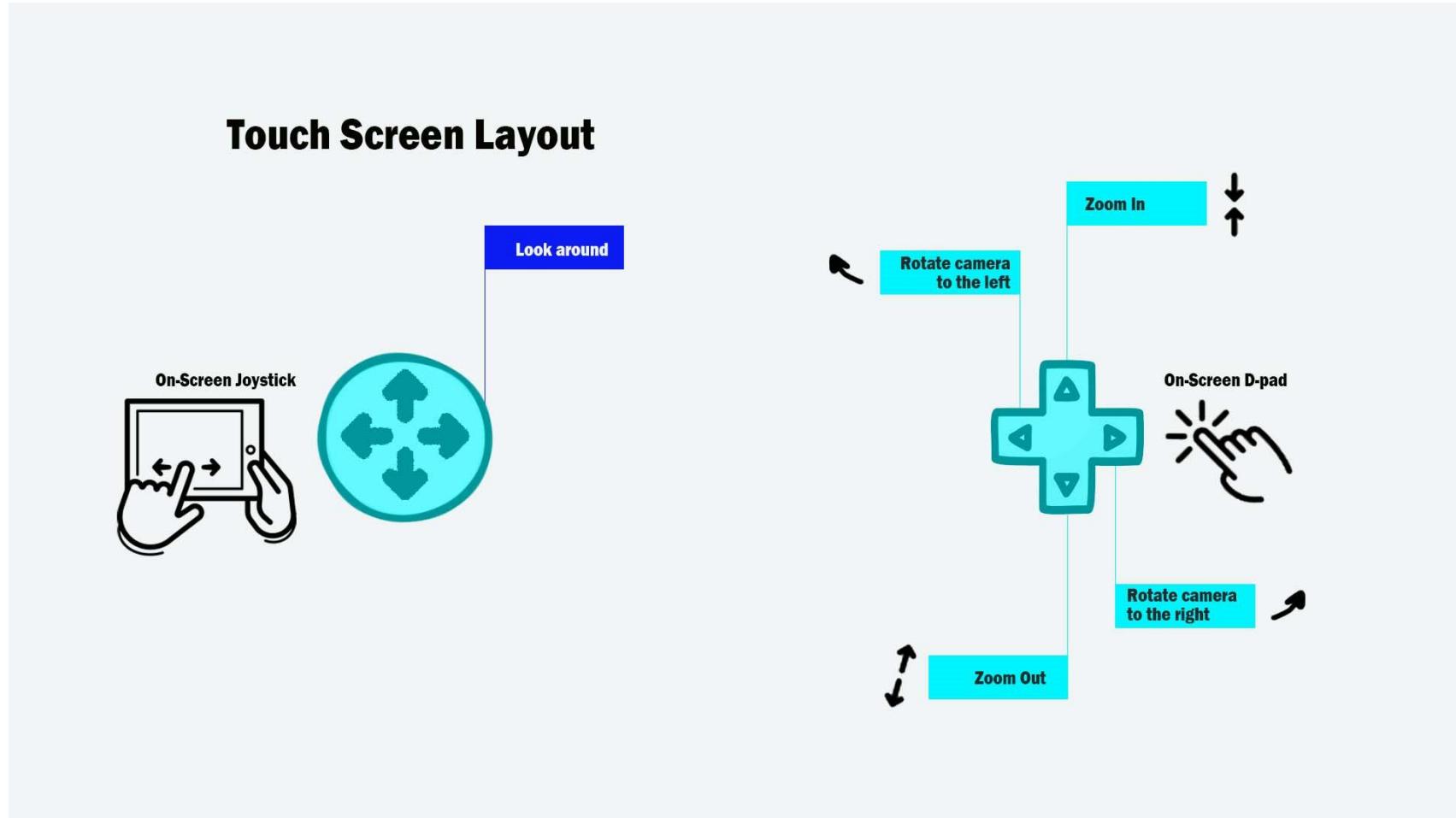


Ilustración 55. Controles pantalla táctil

12 INTERFAZ DE USUARIO

12.1 REQUISITOS DE LA INTERFAZ

El siguiente esquema muestra los requisitos que se han considerado para el diseño y flujo de la interfaz de usuario:

Interfaz Gráfica Amigable

- La interfaz debe ser colorida, amigable y atractiva para niños, con personajes y elementos visuales que les resulten llamativos

Tutorial Interactivo

- Incluir un tutorial intuitivo que enseñe los conceptos básicos

Feedback Inmediato

- Proporcionar retroalimentación instantánea cuando los niños tomen decisiones o completen tareas, reforzando positivamente el aprendizaje

Indicadores de Progreso

- Mostrar un medidor de progreso o puntuación para que los niños puedan ver su avance en el juego

Incentivos y Recompensas:

- Ofrecer recompensas, medallas o logros por completar tareas o niveles, fomentando la motivación y el compromiso

Sistema de Pistas

- Proporcionar pistas o sugerencias cuando los niños se enfrenten a desafíos difíciles, sin revelar la solución completa

Interfaz Táctil o Controlador Simple

- El juego debe ser fácil de controlar con un toque en pantalla o con controles simples que los niños puedan entender fácilmente

Instrucciones Visuales y de Voz

- Si es posible, incluir instrucciones visuales y de voz para ayudar a los niños en la comprensión de las tareas y conceptos

Registro de Actividad

- Mantener un registro de las acciones y decisiones de los niños para que los padres o maestros puedan revisar el progreso y el aprendizaje

Ajustes de Dificultad

- Permitir a los padres o tutores ajustar la dificultad del juego según la edad y habilidades del niño

Compatibilidad Multiplataforma

- Hacer que el juego esté disponible en varias plataformas (Web, PC, tabletas) para una mayor accesibilidad. Hay que tener en cuenta el aspect ratio y las características de cada una

Música y Sonidos Atractivos

- Incluir música y efectos de sonido que sean agradables y reconocibles, que permitan reconocer patrones

12.2 DIAGRAMAS DE FLUJO DE NAVEGACIÓN

A continuación, se mostrarán los diagramas de flujo de navegación por la interfaz de usuario de *Rally Team Tactics*.

La siguiente leyenda indica el significado del código de colores de los elementos y botones situados en la interfaz:



Ilustración 56. Leyenda de colores para diagramas de flujo de la UI

En todos los diagramas se incluyen números para relacionar los bloques con las correspondientes pantallas cuyos bocetos se presentan en el siguiente apartado.

El diagrama inferior indica el recorrido desde que se entra al juego hasta llegar al menú principal. Se diferencia entre si es la primera vez que el usuario accede al juego o si ya tiene una sesión activa:

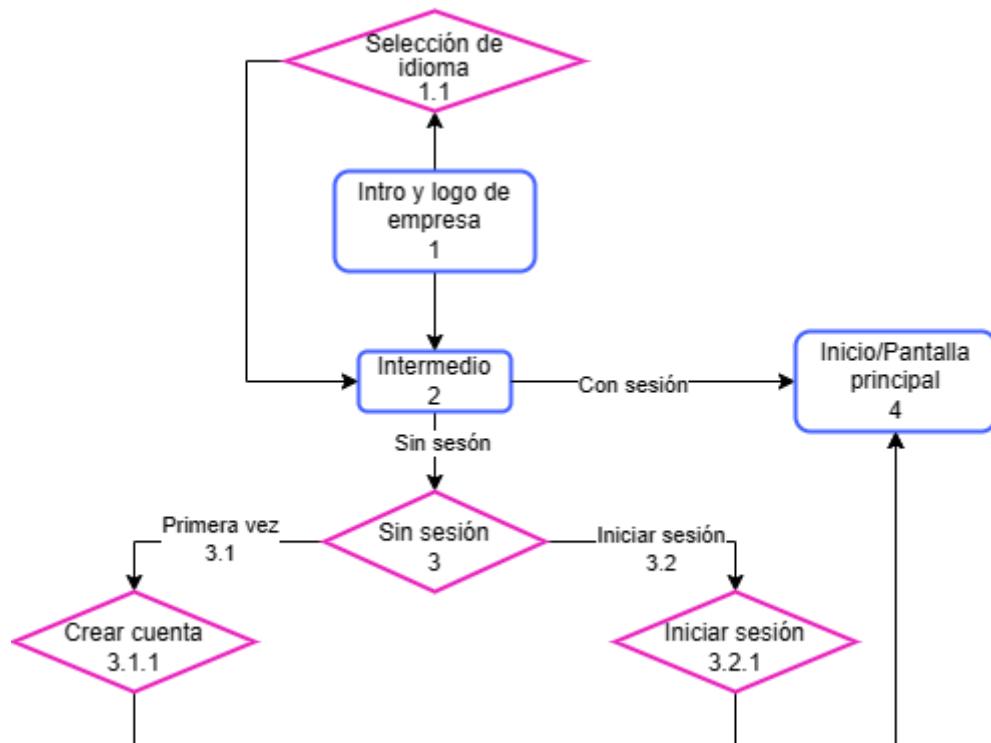


Ilustración 57. Diagrama de navegación de inicio de la aplicación

El diagrama que se dispone a continuación refleja el flujo de navegación de los menús una vez se ha iniciado sesión:

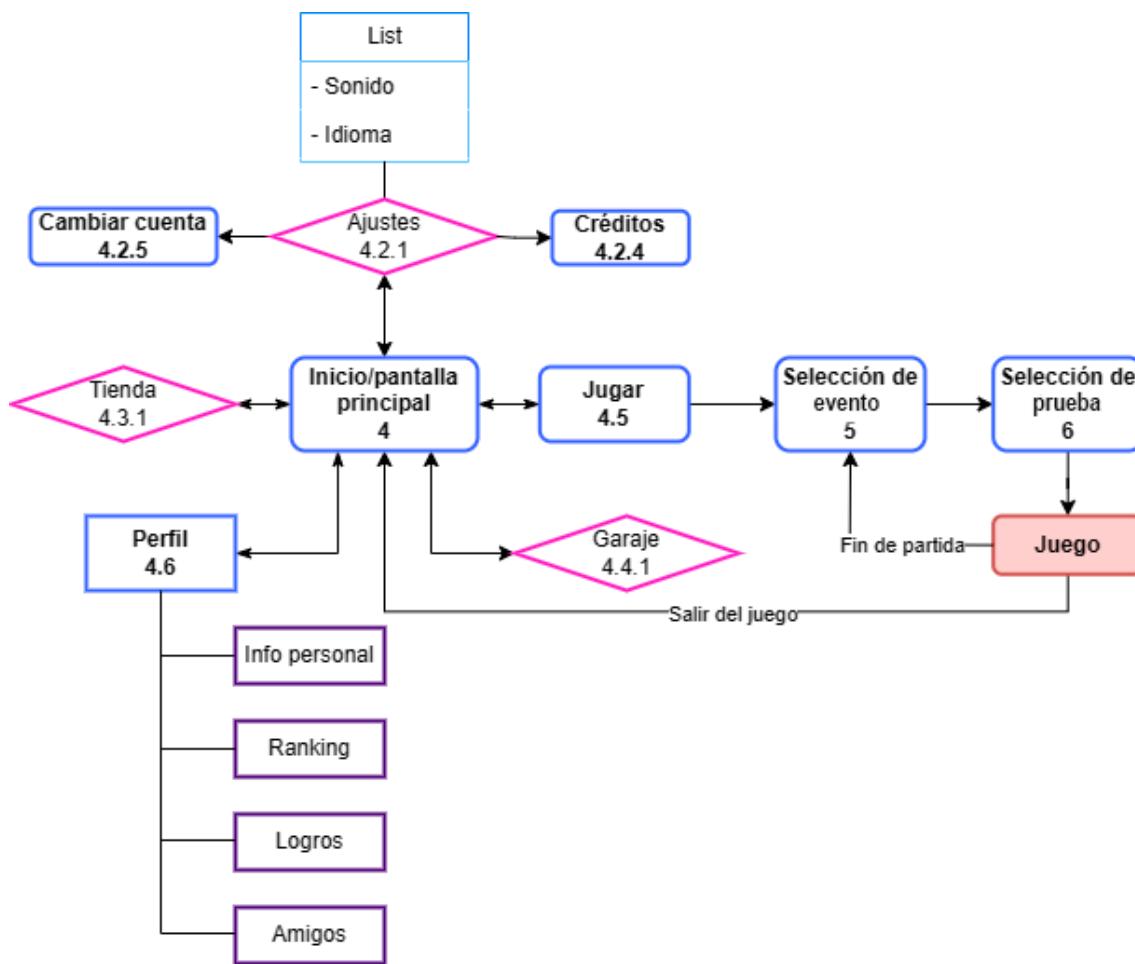


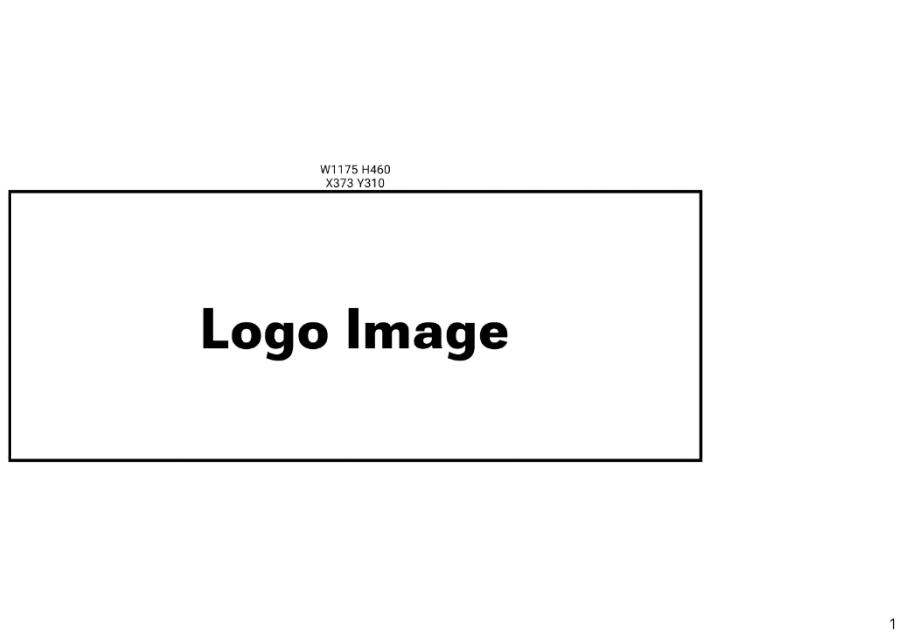
Ilustración 58. Diagrama de navegación por los menús

12.3 DISEÑO ESQUEMÁTICO DE LA UI

En este apartado se muestran *mockups* de las pantallas, menús y pop-ups que se han diseñado para el prototipo. El formato de estas es para juego en navegador web.

12.3.1 DISEÑO ESQUEMÁTICO DE LAS PANTALLAS DE MENÚ

Se recopilan seguidamente los bocetos esquema de las pantallas de menú para el formato web:



1

Ilustración 59. Boceto esquema de la Splash Screen con el logo de la empresa



1.1

Ilustración 60. Boceto esquema de la selección de idioma (primera vez que se inicia el juego)

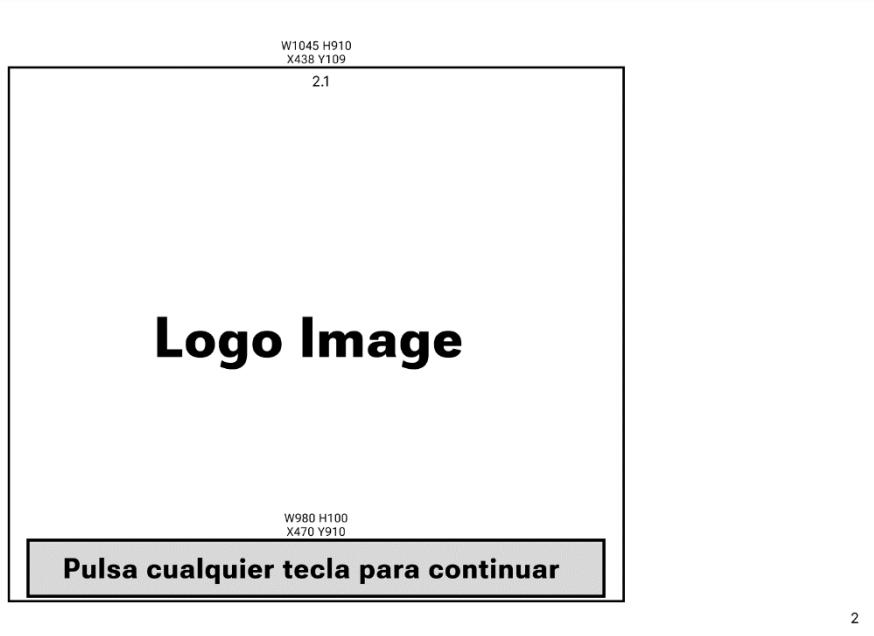


Ilustración 61. Boceto esquema de la pantalla de título

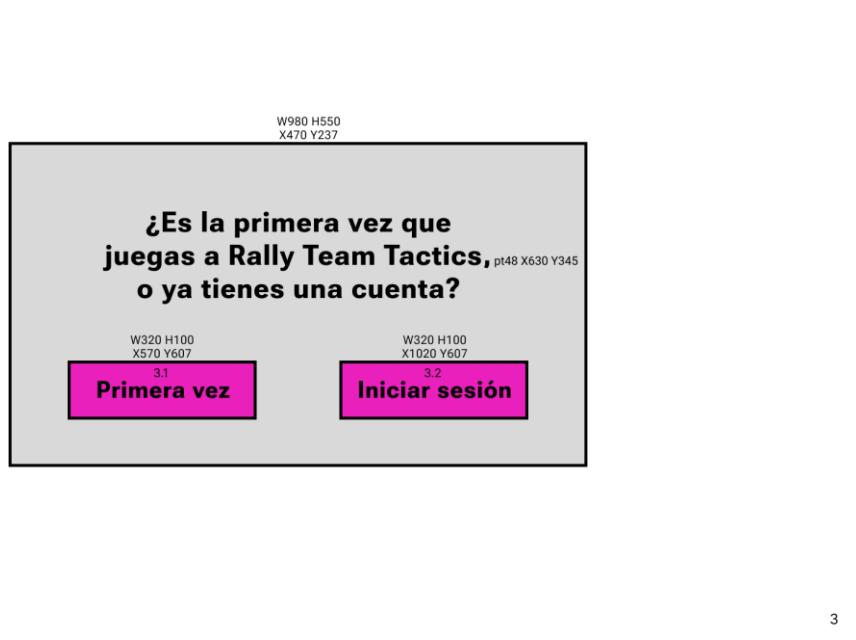


Ilustración 62. Boceto esquema del pop-up de inicio de sesión, si no se ha iniciado previamente en el dispositivo



Ilustración 63. Boceto esquema del pop-up de registro



Ilustración 64. Boceto esquema del pop-up de inicio de sesión

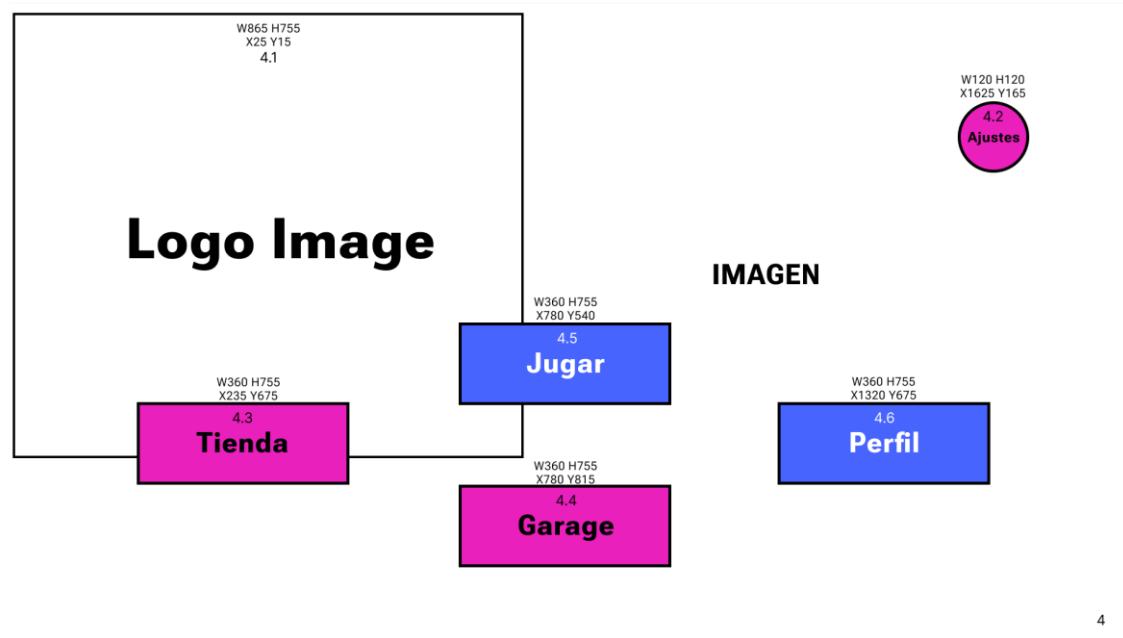


Ilustración 65. Boceto esquema del menú principal

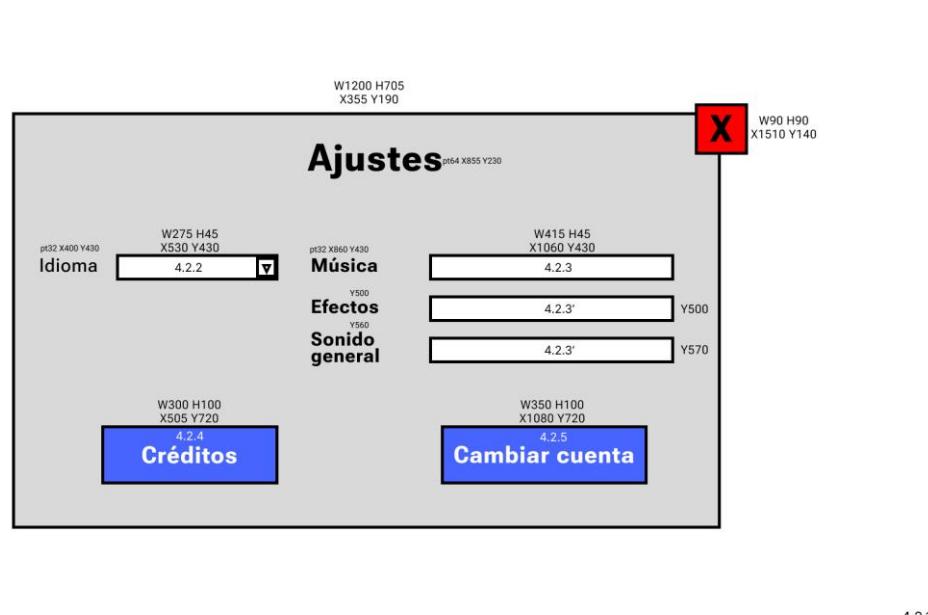


Ilustración 66. Boceto esquema del pop-up de ajustes

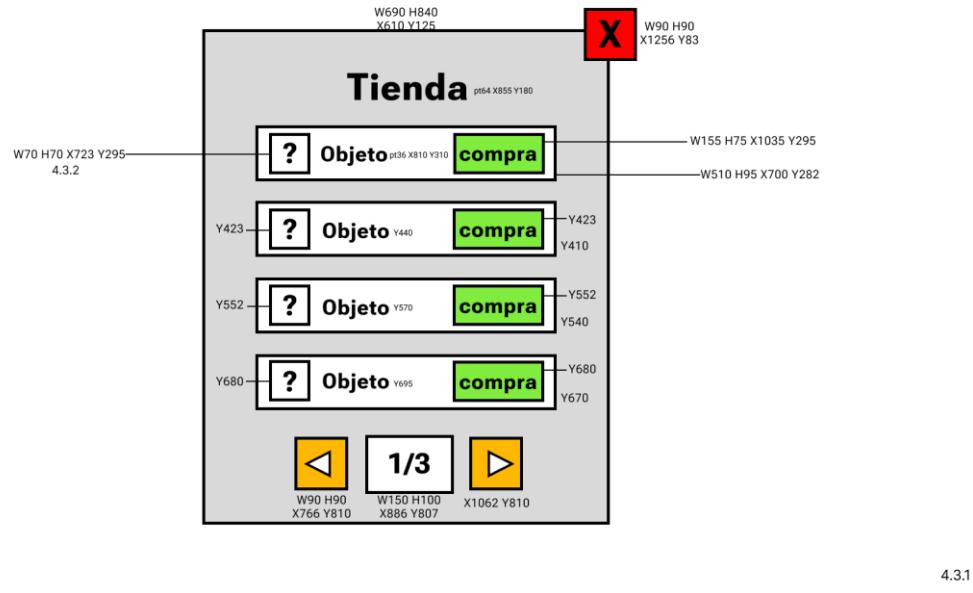


Ilustración 67. Boceto esquema del pop-up de la tienda

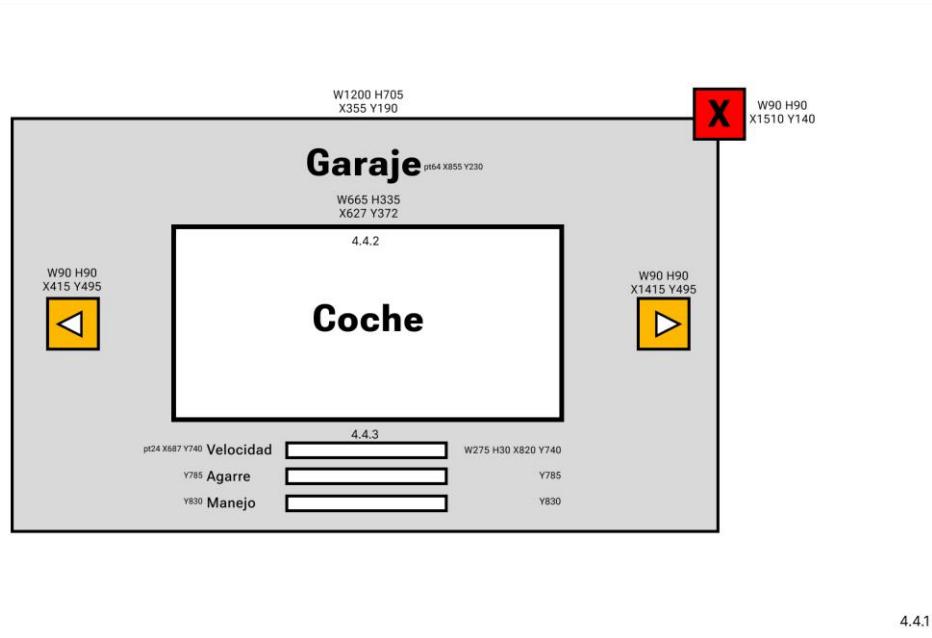


Ilustración 68. Boceto esquema del pop-up del garaje

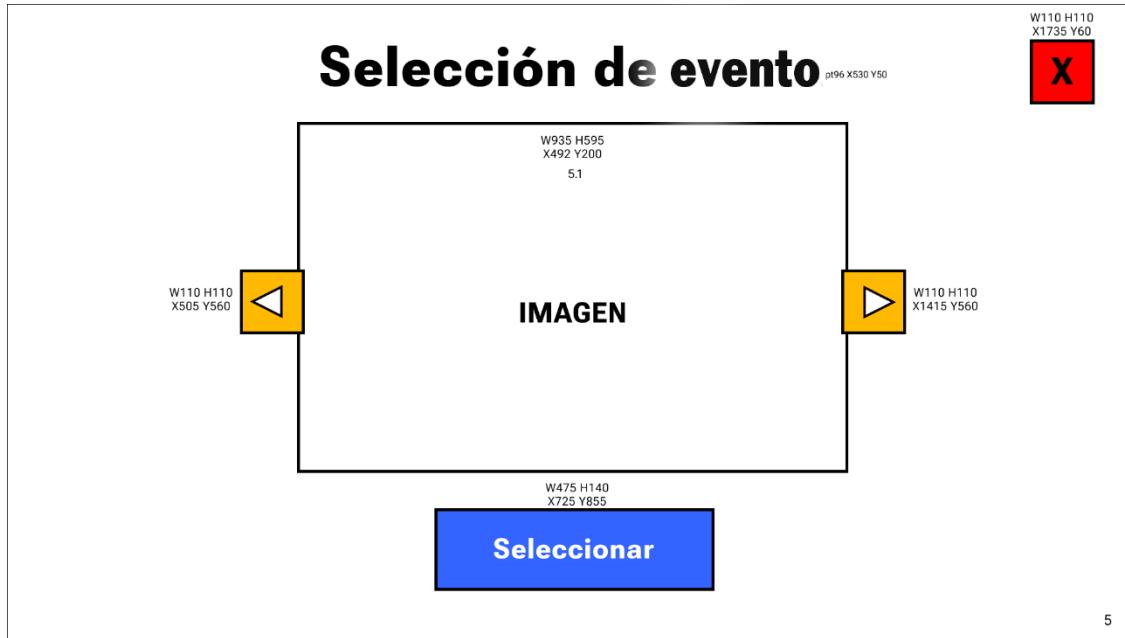


Ilustración 69. Boceto esquema de la pantalla de selección de evento

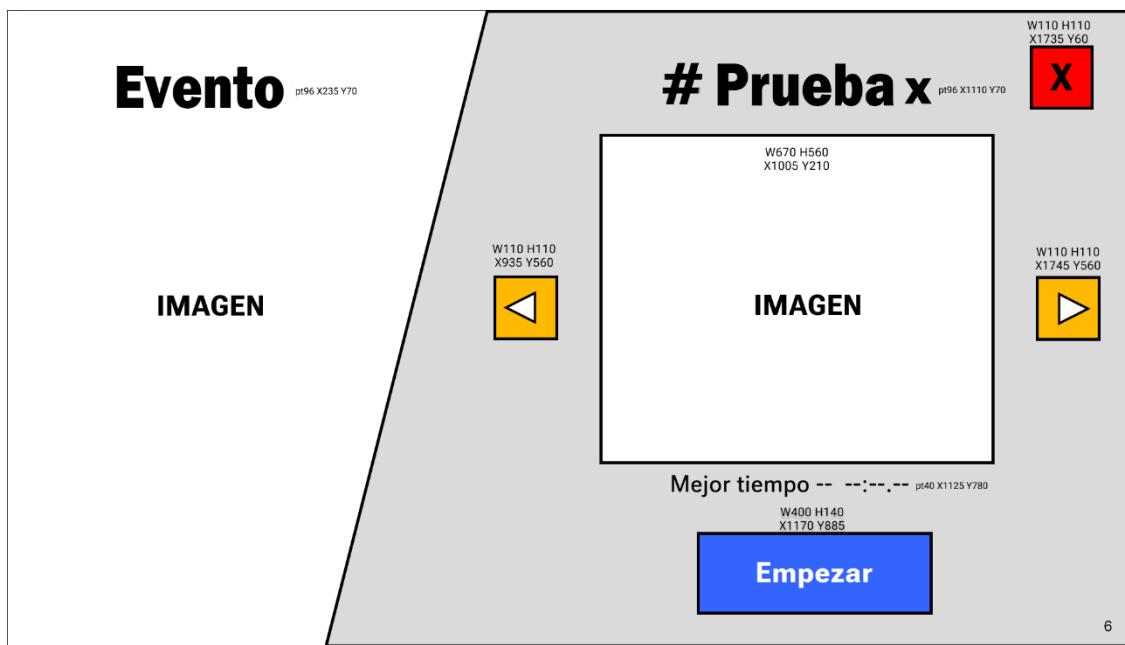
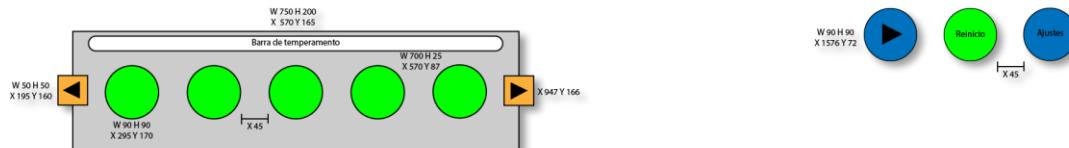


Ilustración 70. Boceto esquema de la pantalla de selección de prueba

12.3.2 DISEÑO ESQUEMÁTICO DE LAS PANTALLAS IN-GAME

Por último, se muestran los *mockups* de las de la HUD y de los pop-ups *in-game*:



Vista cenital del circuito



Ilustración 71. Boceto esquema de la HUD durante la fase de reconocimiento

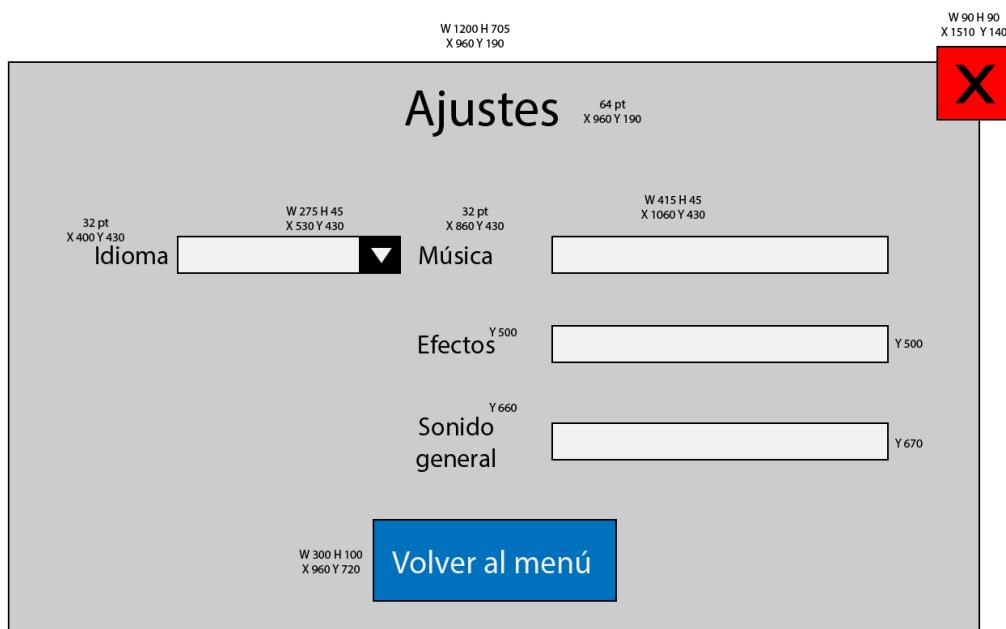


Ilustración 72. Boceto esquema del pop-up del menú de ajustes

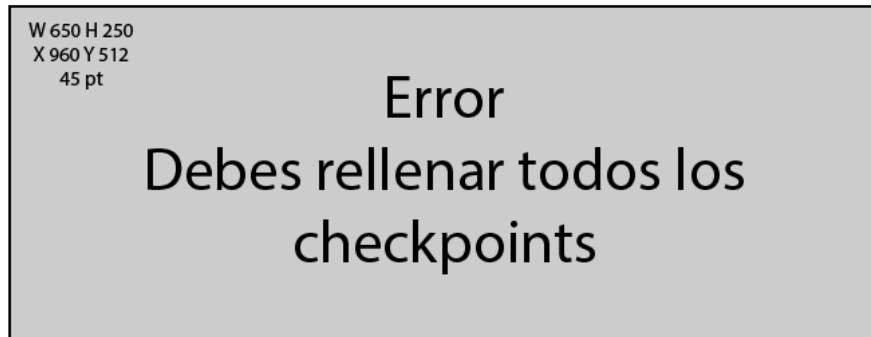


Ilustración 73. Boceto esquema del pop-up de mensaje de error



Vista cenital del circuito

Ilustración 74. Boceto esquema del HUD durante la fase de ejecución



Ilustración 75. Boceto esquema del pop-up de ranking

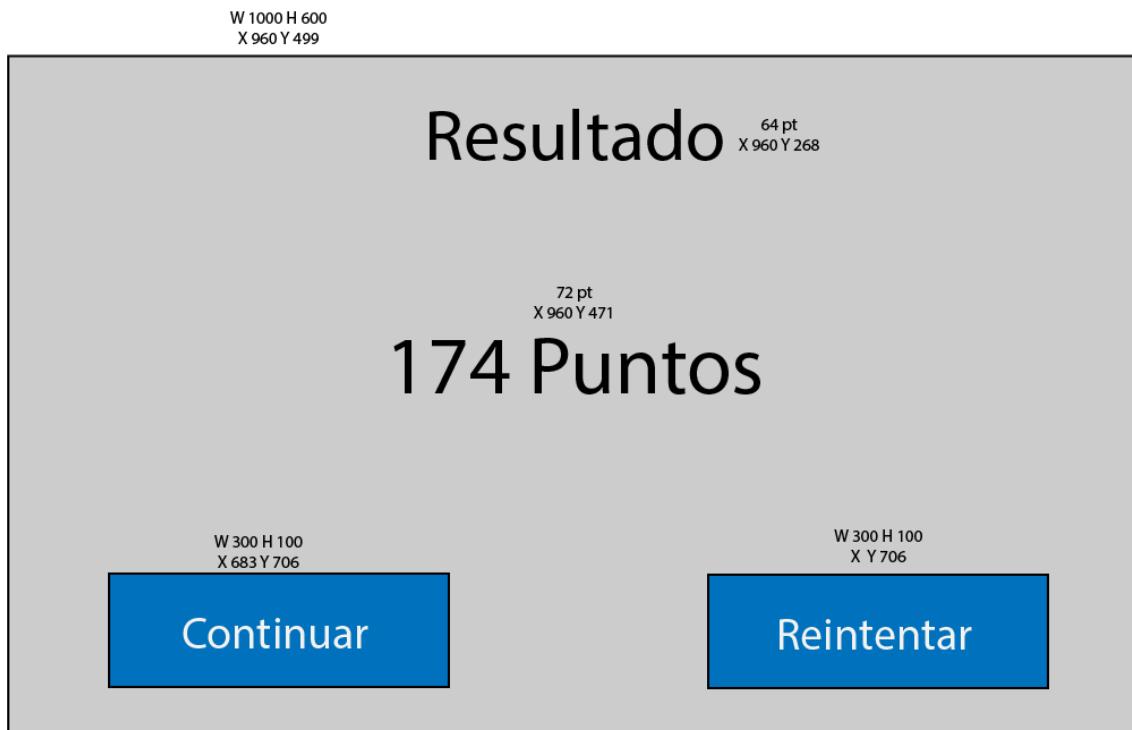


Ilustración 76. Boceto esquema del pop-up de resultado

13 ARTE

13.1 ESTÉTICA DEL JUEGO

13.1.1 REFERENCIAS ARTÍSTICAS

Como se explicó en el apartado [2.1 Principales referencias](#), el juego presentará una estética *low-poly* colorida, inspirada en especial por el *arcade racer Ridge Racer* (Namco, 1993) y el *indie* de *rallies* contemporáneo *Art of Rally* (Funselektor Labs, 2020).



Ilustración 77. Ridge Racer (Namco, 1993)



Ilustración 78. Art of Rally (Funselektor Labs, 2020)

13.1.2 DISEÑO VISUAL DEL JUEGO

Para que todo el contenido relacionado con *RTT*, tanto jugable como promocional (y su documentación), siguiesen el mismo estilo visual, se elaboró un documento de estilo de juego.

Este apartado del GDD recogen los aspectos más importantes del documento.

PALETAS DE COLORES

RTT tiene una paleta base que incluye el color principal, secundario, sus respectivos complementarios, y colores para texto y otros elementos sobre oscuro o claro.

El color principal es un amarillo que se usa principalmente en todos los elementos de la UI:



Ilustración 79. Color primario escogido

Para añadir efectos de relieve a aquellos elementos de la interfaz que lo requieran se ha definido una paleta de sombras. Además, de esta también se extrae el color del fondo del menú:



Ilustración 80. Paleta de sombras del color primario

El color complementario a este amarillo es un violeta intenso que se utiliza para contrastar ciertos detalles en elementos que usen el color primario:

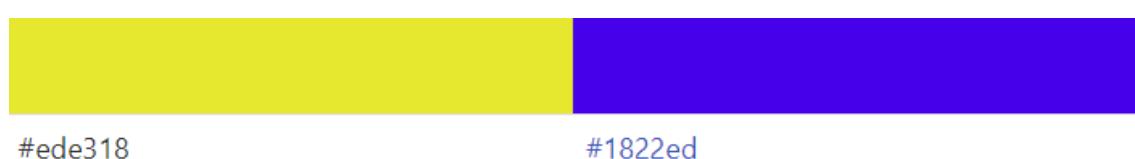


Ilustración 81. Color primario y su complementario

Por otro lado, el color secundario es un cian neón cuyo propósito es destacar elementos al interactuar con la interfaz:



Ilustración 82. Color secundario escogido

Igualmente para este se define una paleta de sombras para regular efectos de iluminación en la UI:



Ilustración 83. Paleta de sombras del color secundario

El color complementario sirve para añadir detalles sobre elementos de la UI del color secundario cian:

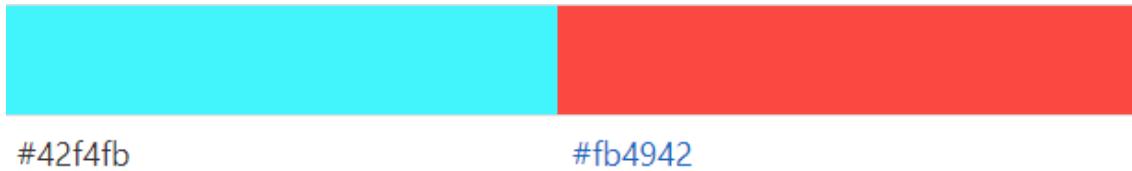


Ilustración 84. Color secundario y su complementario

El negro no puro definido para *RTT* se usa para texto y elementos situados sobre fondos claros o para añadir detalles sobre el fondo base:

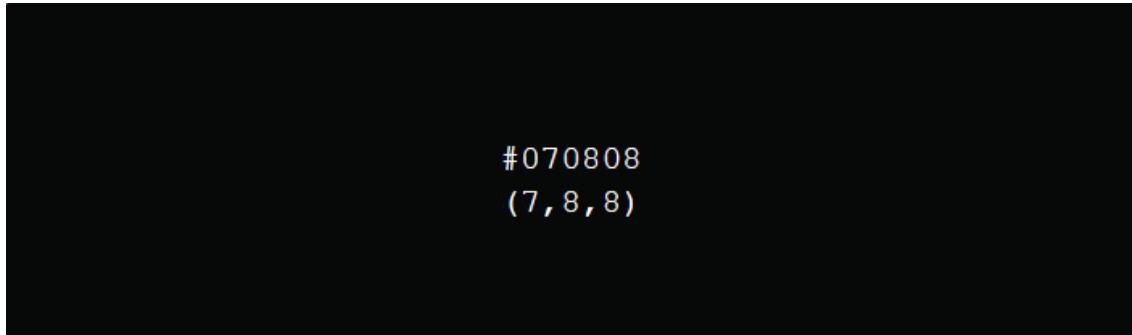


Ilustración 85. Negro no puro escogido

Por el contrario, el blanco no puro elegido se usa para textos y elementos sobre oscuro:



Ilustración 86. Blanco no puro escogido

La paleta base resultante sería la siguiente:

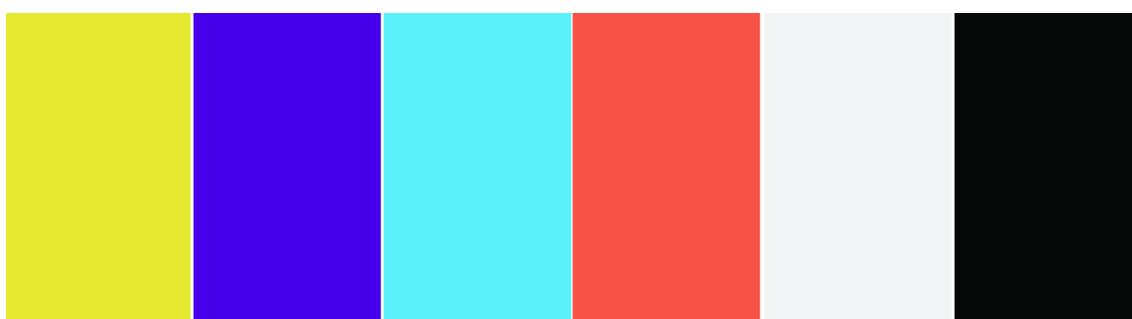


Ilustración 87. Paleta base de RTT

TIPOGRAFÍAS

MUBENK REGULAR

Esta tipografía se utiliza exclusivamente para el título del juego y en los créditos, si se incluyen encabezados en ellos:



Ilustración 88. Muestra de Mubenk Regular

COOPER HEWITT SEMIBOLD

Esta tipografía con mayor legibilidad a diferentes tamaños sirve como fuente para todo el texto del juego que no sean el título ni los encabezados de los créditos.



Ilustración 89. Mapa de caracteres de Cooper Hewitt Semibold

ESTILO DE LA UI

Se procede a explicar el estilo de todos los elementos gráficos que componen todas las pantallas menú y la interfaz *in-game*.

Todas las pantallas mostrarán un fondo cuyo color será un degradado entre #80880e y #d5cc15. No obstante, la pantalla de selección de evento tendrá una particularidad, pues su fondo estará dividido entre este color y la imagen del evento.

TÍTULO

La pantalla de título usa la tipografía *Mubenk*, pues presenta el logo del juego, caracterizado por los colores primario y el negro escogidos.



Ilustración 90. Logo de RTT

MARCOS DE LOS MENÚS EMERGENTES

Aquellas pantallas que requieren de un marco, como el pop-up de *login*, utilizan el color negro para este.

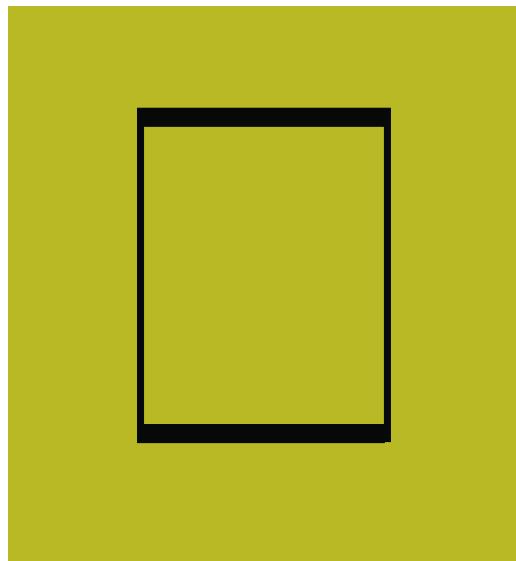


Ilustración 91. Ejemplo de estilo del marco de un menú emergente

BOTONES EN LOS MENÚS

La mayor parte de los botones de la interfaz se presentan en el color primario con detalles de sombreado #d5cc15. El interior puede contener iconos o texto en el negro escogido, #070808. El texto, como en el resto de la UI (a excepción de algunos elementos como el título), está en fuente *Thecnic* y *ThecnicBold*. Si el jugador se sitúa sobre un botón, el texto o ícono cambia al color secundario (cian).

Algunos botones especiales que contienen iconos, como los que se superponen al marco de un pop-up, siguen un código de colores distinto. Los botones para “salir” usan el naranja complementario al cian definido como color secundario, mientras que los botones de “avance” o “retroceso” utilizan el violeta complementario al amarillo definido como color primario.

Si el jugador se sitúa sobre un botón, el texto o ícono cambia al color secundario (cian).



Ilustración 92. Ejemplo de estilo de un botón normal

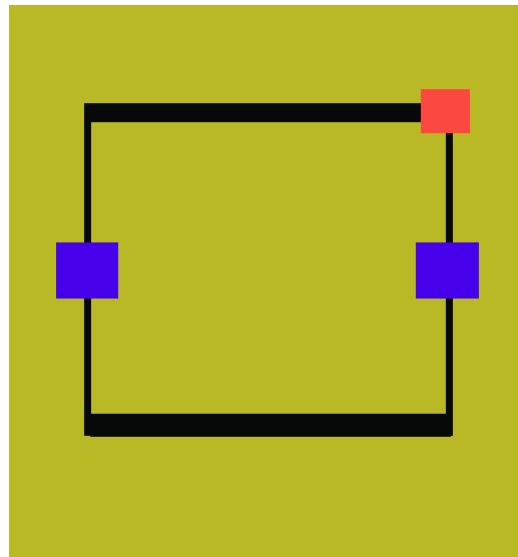


Ilustración 93. Ejemplo de estilo de botones especiales superpuestos a un marco de ventana emergente

PANTALLA DE CARGA

La pantalla de carga sigue la tonalidad del fondo base, pero cuentan con elementos extras en el color negro escogido que están relacionados con el mundo del rally.

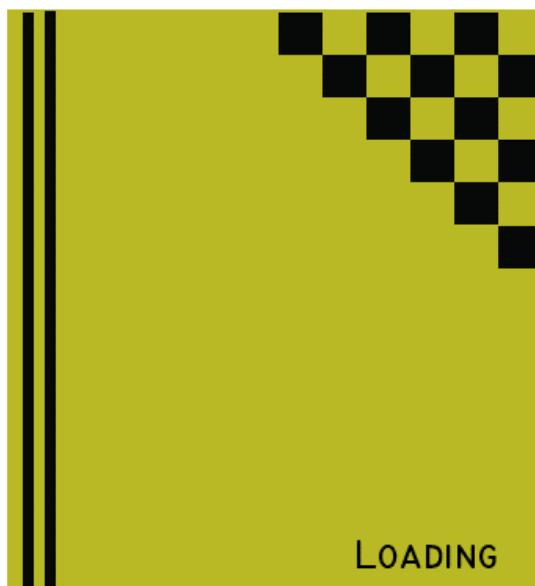


Ilustración 94. Ejemplo de estilo de pantalla de carga

HEADS-UP DISPLAY (INTERFAZ IN-GAME)

Dentro de una partida encontramos seis elementos principales a tratar en el HUD: los medidores de daño y combustible, los botones de acceso a menús o de acción, los selectores de nota, el *slider* de temperamento y los indicadores del estado de cada punto de control.

- Los botones de acceso a menús emergentes o de acción siguen el mismo estilo que los botones estándar fuera de la partida.

- Los selectores de nota son similares en estilo, y son redondeados y contienen el ícono correspondiente a la instrucción. Eso sí, no se iluminan ni resaltan al ponerse sobre ellos.
- El *slider* de temperamento usa principalmente el gris #919191.
- La barra de daño usa el rojo #db0f0f para el relleno y una tonalidad más suave del mismo para la parte sin llenar (este no se muestra en la previsualización inferior). La barra de combustible es igual, pero con el verde #21db21. En ambos casos la parte que contiene el ícono o texto referente al recurso está en el amarillo primario.
- El muro luminoso que se muestra sobre los puntos de control adquiere un color verdoso (#21db21) cuando se ha seleccionado instrucción, negro #070808 en caso contrario.

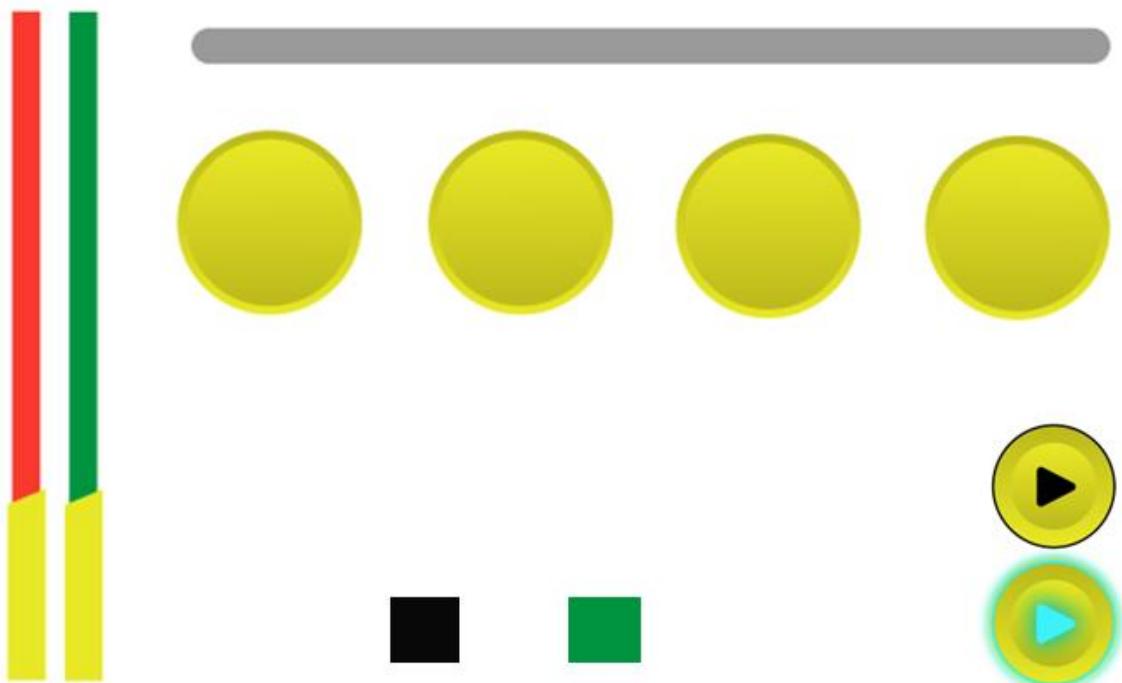


Ilustración 95. Ejemplo de estilo de elementos del HUD

13.2 CONCEPT

Se presentan en este apartado una serie de *concepts* de elementos del juego, como el coche o los *props* de los escenarios.

13.2.1 CONCEPTS DEL COCHE

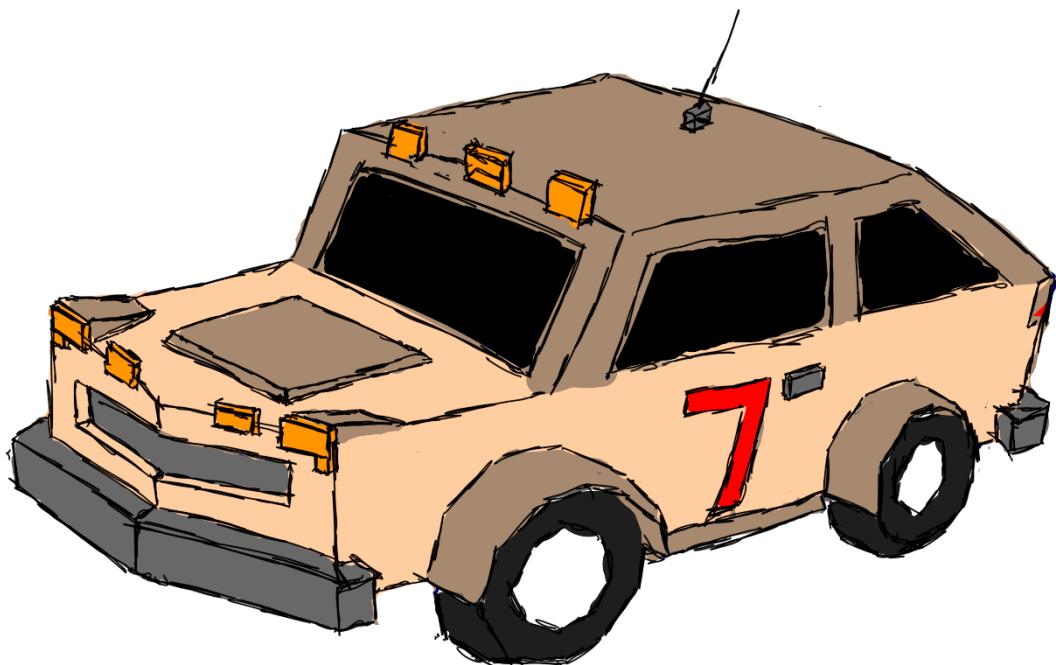


Ilustración 96. Concept de coche en perspectiva



Ilustración 97. Concept de coche de perfil

13.2.2 CONCEPTS DE PROPS



Ilustración 98. Concept de un punto de control

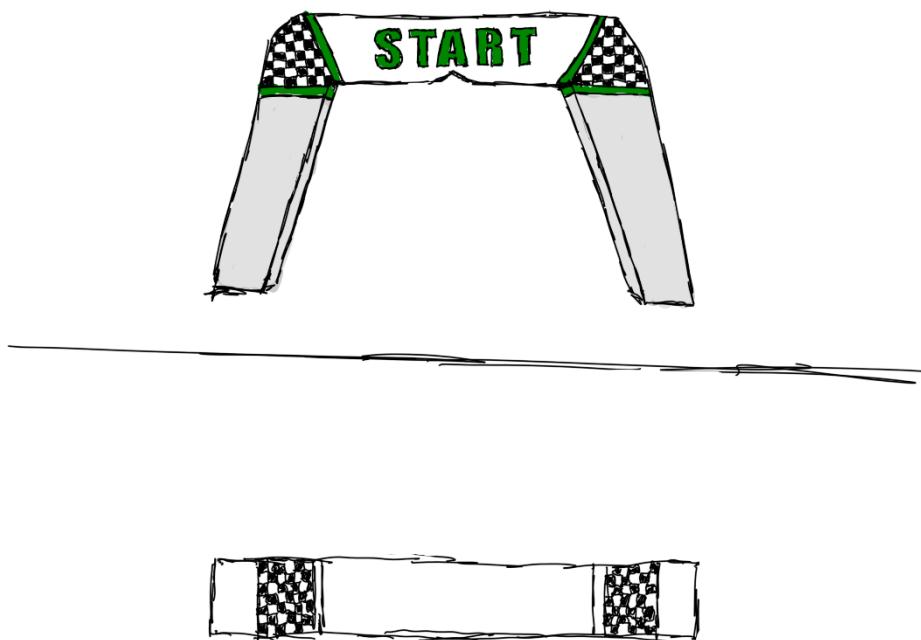


Ilustración 99. Concept de la salida

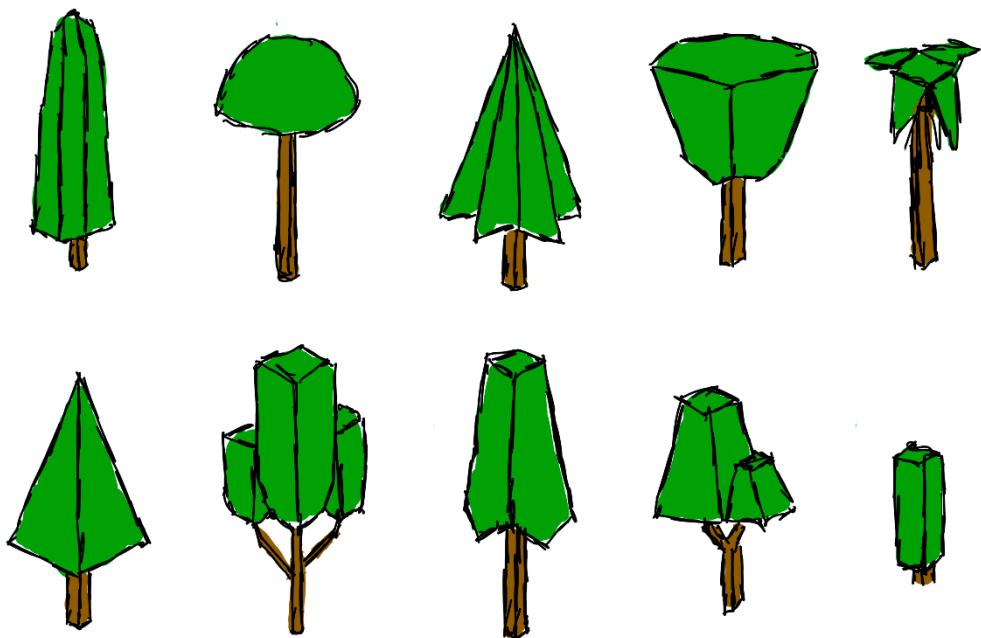


Ilustración 100. Concept de diferentes variantes de árbol

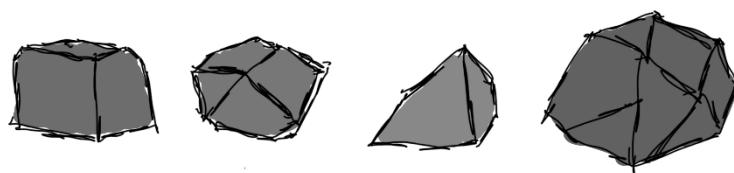
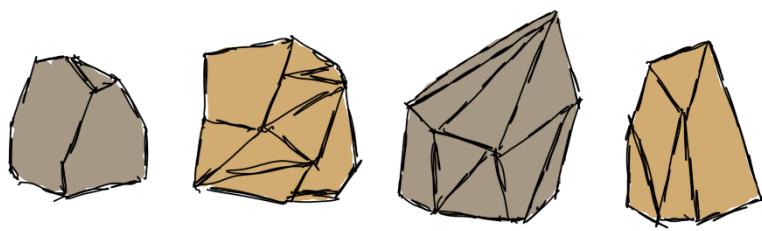


Ilustración 101. Concept de diferentes rocas

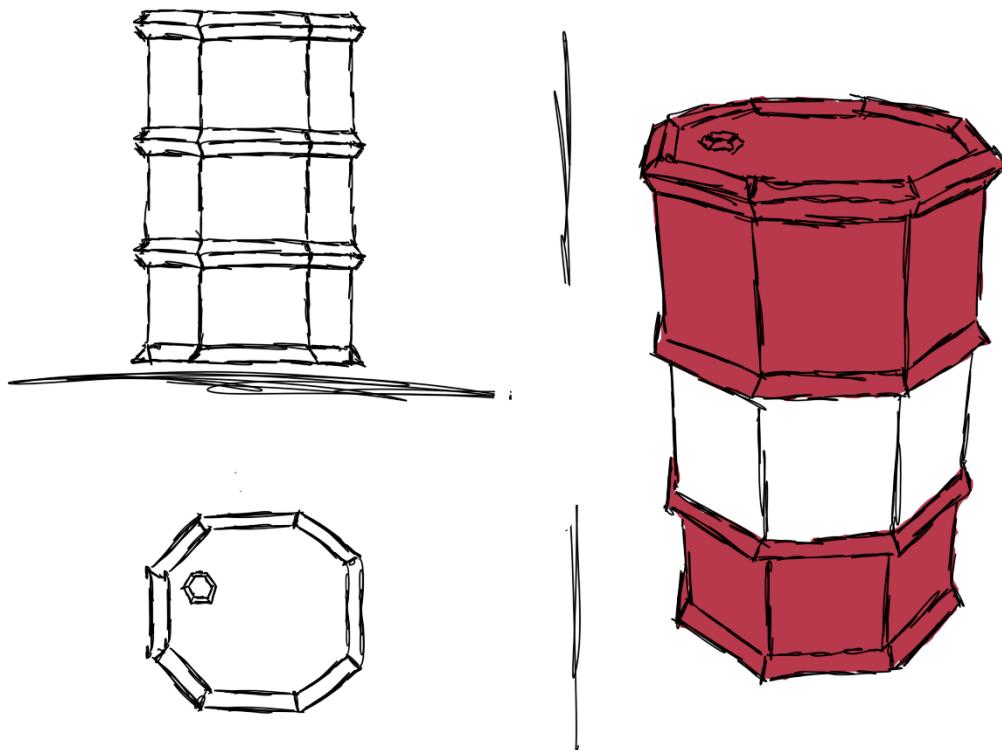


Ilustración 102. Concept de un barril

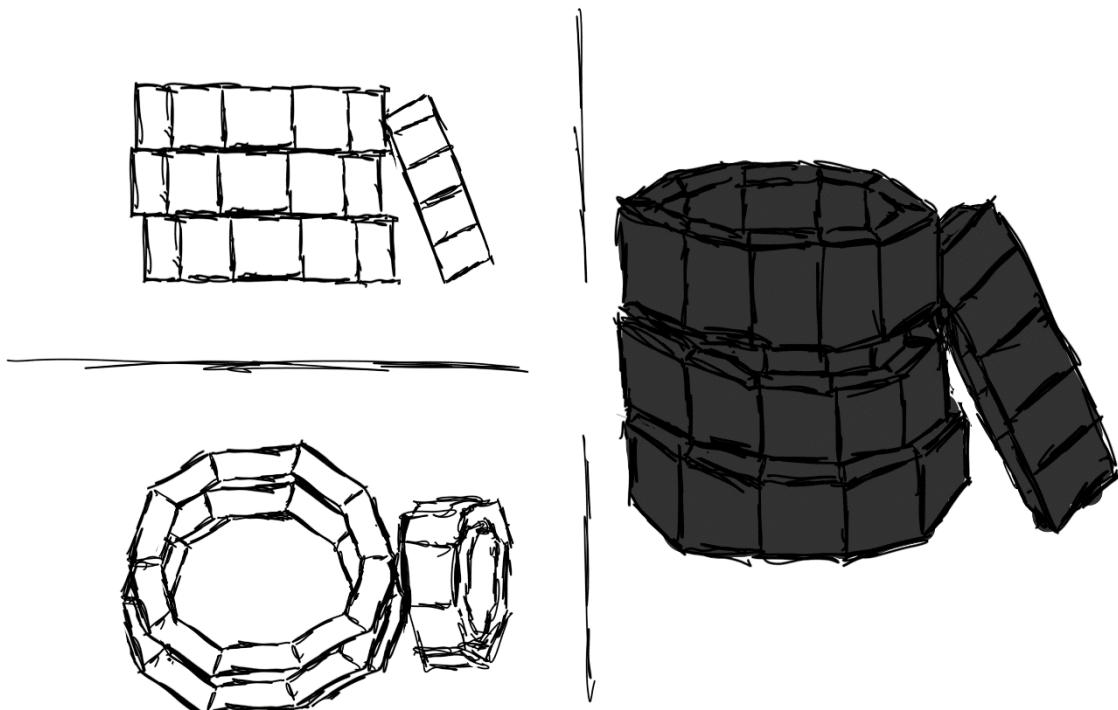


Ilustración 103. Concept de una pila de ruedas

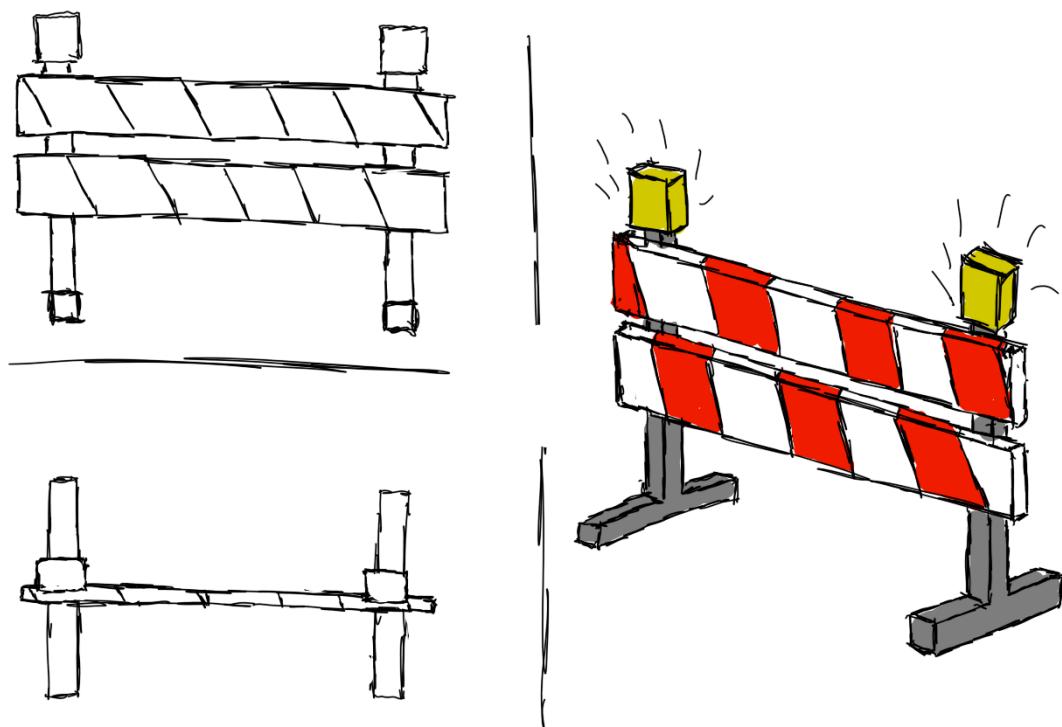


Ilustración 104. Concept de una valla de obra

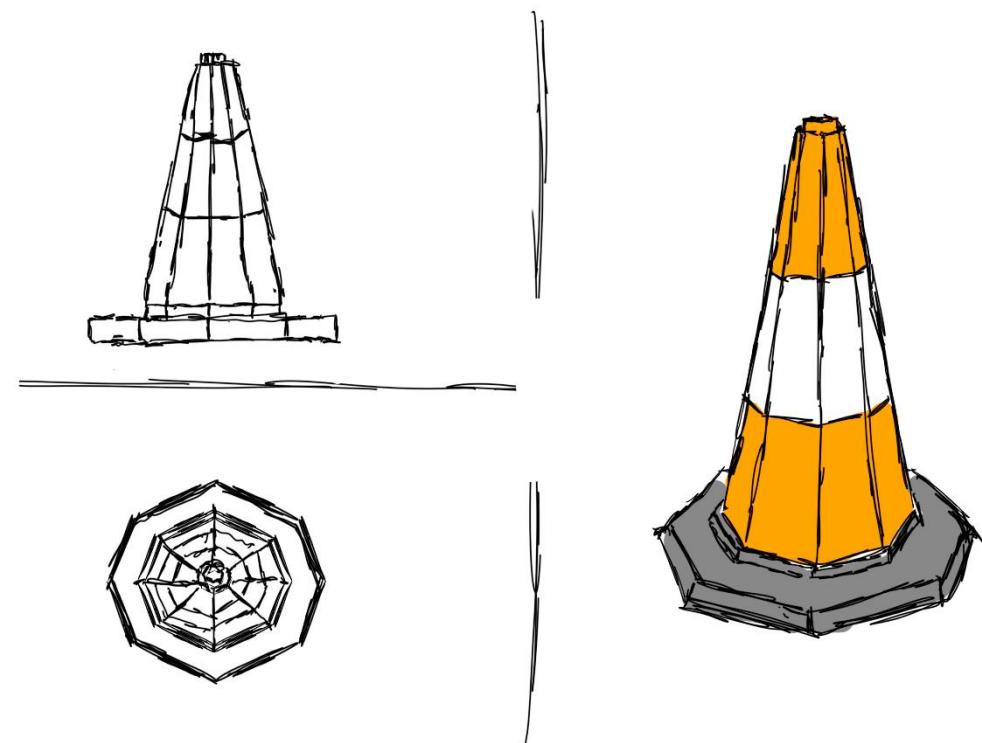


Ilustración 105. Concept de un cono de tráfico

14 PRODUCCIÓN

14.1 MODELO DE NEGOCIO DEL JUEGO Y PLAN DE FINANCIACIÓN

14.1.1 INFORMACIÓN SOBRE EL USUARIO

¿Quién es?	<ul style="list-style-type: none">• Existen dos tipos de usuarios, los padres, interesados en que sus hij@s estén más motivados y mejoren sus habilidades computacionales, y un usuario final, el/la niñ@ que juega al videojuego y genera las métricas.
¿Qué quiere?	<ul style="list-style-type: none">• El cliente quiere que la escuela cambie su modelo educativo para incentivar la creatividad y la resolución de problemas en los niñ@s• El usuario final quiere diversión y entretenimiento sin tener la sensación de estar haciendo una actividad académica
¿Aficiones?	<ul style="list-style-type: none">• El usuario final es un/a niñ@ de entre 8 y 12 años, que frecuenta videojuegos contemporáneos, de diferentes géneros o tiene al menos nociones básicas sobre la interacción con videojuegos y sus mecánicas.
¿Situación?	<ul style="list-style-type: none">• El cliente pone su dinero y confianza en las instituciones educativas con la esperanza de que el desentusiasmo que ve en sus hij@s y el desequilibrio de nivel en las clases se minimicen o desaparezcan. Está descontento con los defectos del sistema educativo y el impacto que está teniendo en los niñ@s.
¿Actividad?	<ul style="list-style-type: none">• El cliente sostiene a la familia y comparte su preocupación con otros padres por el trato que su hij@ recibe en la escuela, que considera ampliamente mejorable.• El usuario final asiste y participa en las clases como alumn@ del centro educativo.
¿Cómo es?	<ul style="list-style-type: none">• El cliente es un parent/madre proactivo, que no se da por vencido ante la poca predisposición de las instituciones y escuelas a cambiar el modelo educativo.
¿Qué necesita?	<ul style="list-style-type: none">• El cliente quiere conocer las capacidades de su hij@, y asegurarse de que se le da una educación a medida, que le genere además interés e impulse sus habilidades de resolución de problemas. Necesita un cambio en el sistema educativo, que incorpore programas para la mejora del Pensamiento Computacional a través de actividades lúdicas.

14.1.2 MAPA DE EMPATÍA

Se muestra a continuación el Mapa de Empatía de los clientes, padres de niños y niñas entre 8 y 12 años:

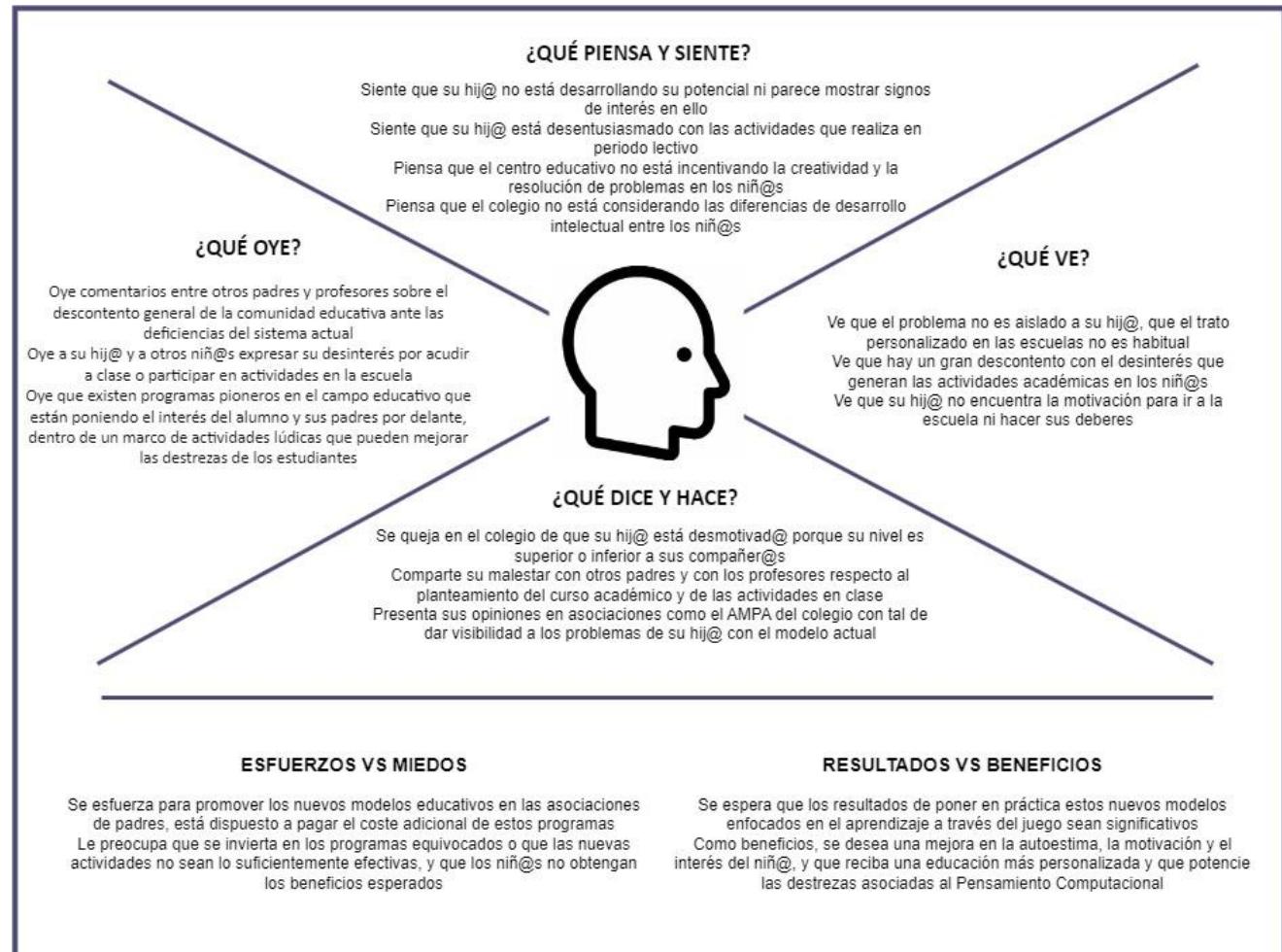


Ilustración 106. Mapa de Empatía del cliente

14.1.3 CAJA DE HERRAMIENTAS

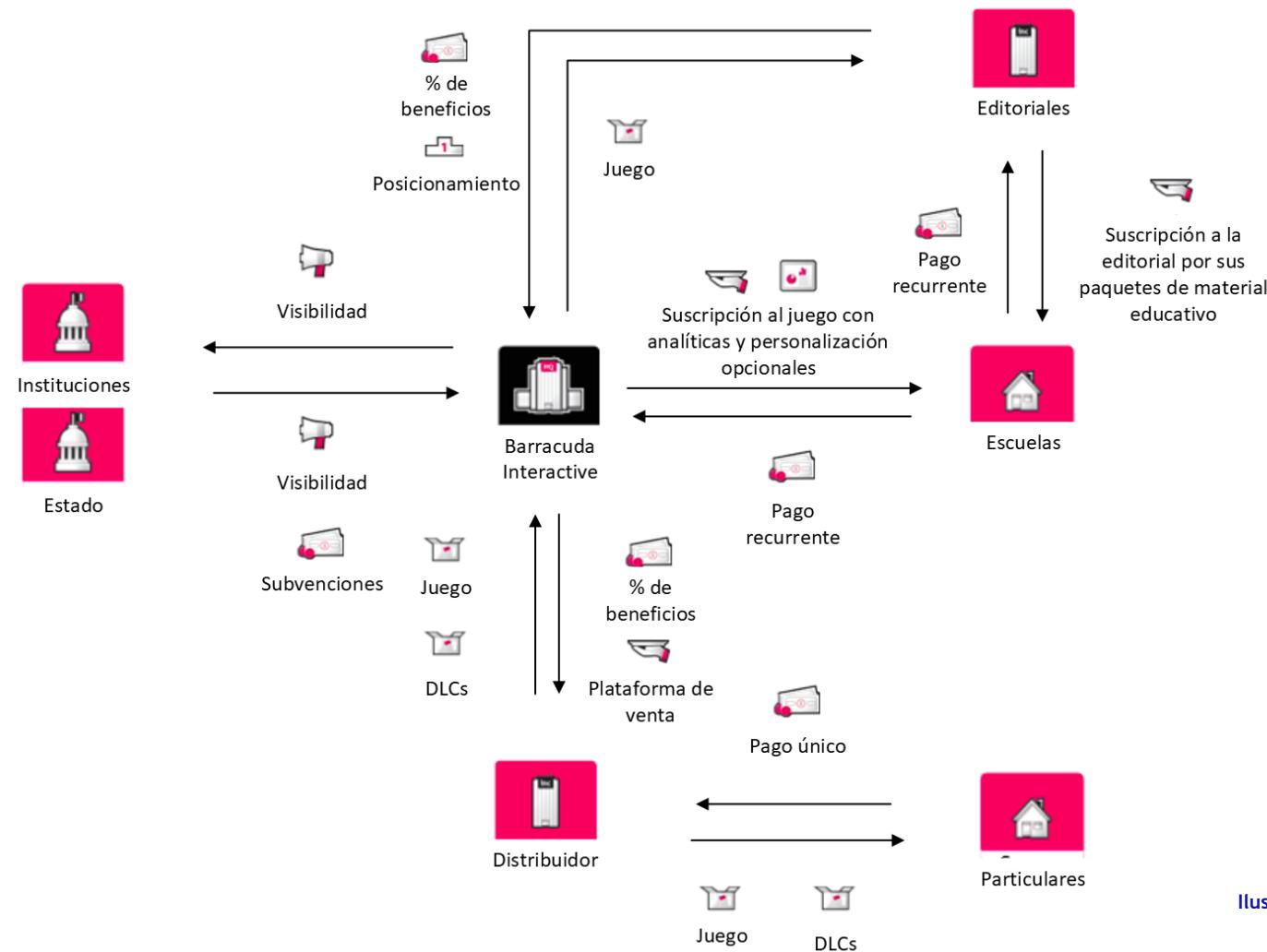
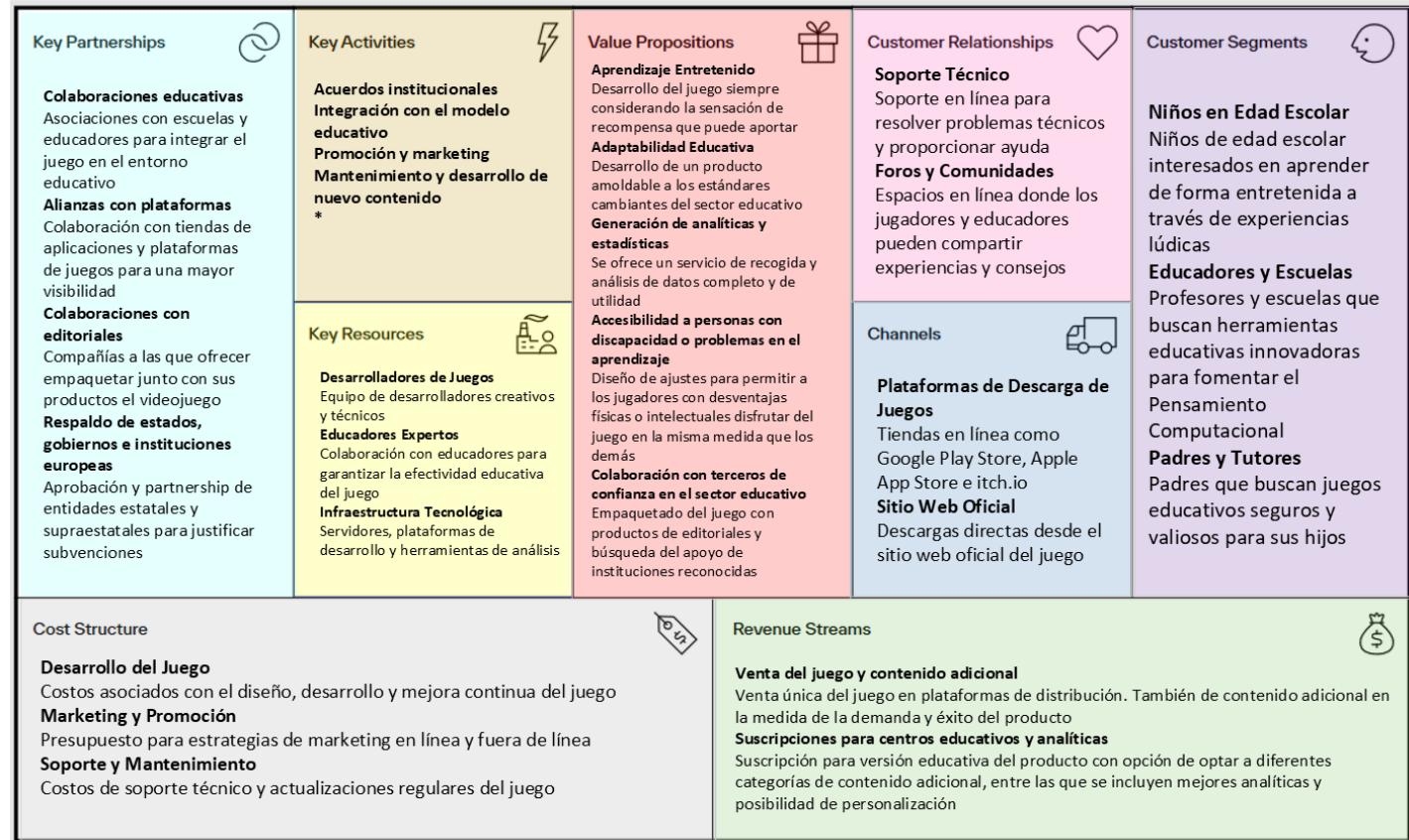


Ilustración 107. Caja de Herramientas

14.1.4 MODELO DE LIENZO

Se presenta seguidamente el Canvas elaborado para *Rally Team Tactics*:



*

Acuerdos institucionales: mantener conversaciones y llegar a acuerdos con instituciones nacionales e internacionales y el Estado para financiar nuestro videojuego y promover su uso en las aulas.

Integración con el modelo educativo: asegurarse de que el producto sea adecuado a los currículos escolares actuales.

Promoción y marketing: promoción del videojuego en ferias del sector educativo, en colaboración con las editoriales, en las plataformas de distribución y en redes sociales.

Mantenimiento y desarrollo de nuevo contenido: supervisión de la recepción al juego tras el lanzamiento para el despliegue de parches y el potencial desarrollo de contenido adicional.

Ilustración 108. Canvas del modelo de negocio

14.2 PLAN DE MARKETING

Para la exposición de *Rally Team Tactics* al mercado, se han considerado las redes sociales como una de las vías principales de generar interés entre el público y captar posibles *leads*. Como las publicaciones y la interacción con la audiencia a través de las redes es fundamental para el marketing en Barracuda Interactive, se han planteado una serie de estrategias a seguir en las diferentes plataformas.

14.2.1 CAMPAÑA MEET THE TEAM

Como *kick-off* de la campaña de marketing digital de *RTT*, se realizará una presentación de los integrantes del equipo de desarrollo en las semanas posteriores al lanzamiento de la *beta*. Se ha planeado el lanzamiento de la campaña en Instagram y X, comenzando con un **anuncio el lunes 20 de noviembre de 2023**. A partir de esa fecha, de manera regular y en horarios donde se pueda alcanzar al mayor número de clientes objetivo, se irán realizando publicaciones para cada uno de los miembros del equipo. Teniendo en cuenta que los clientes principales de *RTT* son los padres y los profesionales de la educación, estas publicaciones tienen prevista su salida en horarios donde estos suelen haber terminado su jornada laboral y consultan las redes sociales, entre las 20hrs y las 23hrs. Se probarán diferentes horas de publicación con tal de determinar con mayor certidumbre el momento de mayor tráfico de cara a futuras campañas.

Publicación	Día	Hora	Redes
Anuncio de la campaña	20/11	20hrs	 
Sobre el integrante 1	22/11	21hrs	 
Sobre el integrante 2	25/11	22hrs	 
Sobre el integrante 3	29/11	23hrs	 
Sobre el integrante 4	02/12	20hrs	 
Sobre el integrante 5	06/12	21hrs	 
Sobre el integrante 6	09/12	22hrs	 

Tabla 3. Fechas campaña MeetTheTeam

La presentación de cada miembro consistirá en una foto con su nombre, su ocupación en el equipo y un pequeño vídeo donde responderá una serie de preguntas relacionadas con el mundo de los videojuegos y la cultura popular. En el caso de X solo se publicará el vídeo, con el nombre y la ocupación de la persona dentro del propio *post*. La lista de preguntas provisional es la siguiente:

1. ¿Cuál es tu videojuego favorito de la infancia y por qué te gustaba tanto?
2. ¿Cuál es el logro en un videojuego del que estás más orgulloso y por qué?
3. Si pudieras intercambiar habilidades con otro miembro del equipo por un día, ¿a quién elegirías y por qué?
4. Si fueras un súper héroe con un alter ego común, ¿cuál sería tu empleo diurno menos sospechoso y más divertido?
5. Si tu vida tuviera una banda sonora, ¿cuál sería la canción principal y por qué?

14.2.2 PLANES PARA CADA PLATAFORMA

PLAN DE MARKETING EN INSTAGRAM

Además de los *posts* programados dentro de la campaña antes descrita, se explica en este apartado, de manera detallada, el plan de marketing en la red social Instagram, donde se tiene acceso a una enorme cantidad de clientes potenciales debido a su tamaño. Será una plataforma de exposición, más enfocada en la publicidad que en la conversación con la audiencia.



Semana	Publicaciones
20 - 26 de noviembre	Anuncio de la campaña <i>MeetTheTeam</i> , presentaciones de los dos primeros integrantes Capturas de pantalla del juego
27 de noviembre - 3 de diciembre	Presentaciones de los dos siguientes integrantes <i>Clips</i> de <i>gameplay</i> del juego <i>Cover reveal</i> <i>Teaser trailer</i>
4 - 10 de diciembre	Presentaciones de los dos últimos integrantes
11 - 17 de diciembre	<i>Reel</i> de una partida destacada en el juego Vídeos promocionales que explican los objetivos del juego y su relación con la jugabilidad <i>Launch Trailer</i>

Tabla 4. Campaña de marketing en Instagram

PLAN DE MARKETING EN X

En X, por su enfoque más conversacional y de discusión e interactividad, se pretenderá crear una comunidad más leal que se sienta involucrada. Por ello, además de las publicaciones de *MeetTheTeam*, en X se subirán encuestas, se responderá a comentarios, se hará mención a publicaciones ajenas relevantes, etc.



Semana	Publicaciones
20 - 26 de noviembre	Anuncio de la campaña <i>MeetTheTeam</i> , presentaciones de los dos primeros integrantes Capturas de pantalla del juego GIFs divertidos relacionados con el juego <i>Posts</i> que plantean la colaboración activa de la comunidad como encuestas de opinión
27 de noviembre - 3 de diciembre	Presentaciones de los dos siguientes integrantes <i>Clips</i> de <i>gameplay</i> del juego <i>Cover reveal</i> <i>Teaser trailer</i> <i>Posts</i> que plantean la colaboración activa de la comunidad como encuestas de opinión
4 - 10 de diciembre	Presentaciones de los dos últimos integrantes Vídeo <i>behind the scenes</i> del desarrollo
11 - 17 de diciembre	<i>Reel</i> de una partida destacada en el juego Vídeos promocionales que explican los objetivos del juego y su relación con la jugabilidad <i>Launch Trailer</i>

Tabla 5. Campaña de marketing en X

PLAN DE MARKETING EN YOUTUBE

Youtube servirá para la publicación de vídeos de gran relevancia, especialmente aquellos que tengan mayor longitud o más calidad de edición. En principio se subirán de uno o a tres videos a la semana:



Semana	Publicaciones
20 - 26 de noviembre	<i>Preview</i> con comentarios del equipo de desarrollo
27 de noviembre - 3 de diciembre	<i>Teaser trailer</i>
4 - 10 de diciembre	Vídeo <i>behind the scenes</i> del desarrollo Compilación de los vídeos de <i>MeetTheTeam</i>
11 - 17 de diciembre	Vídeos promocionales que explican los objetivos del juego y su relación con la jugabilidad <i>Launch Trailer</i>

Tabla 6. Campaña de marketing en Youtube

PLAN DE MARKETING EN GAMEJOLT Y REDDIT

Estas dos últimas plataformas se caracterizan por el fuerte énfasis en la creación de comunidades. Se busca por tanto el compromiso de una audiencia dedicada, específicamente interesada en videojuegos independientes y en *racers*. Se participará activamente en ambas redes con tal de retener a los seguidores, preguntando por su *feedback*.



En GameJolt se harán publicaciones como:

- Anuncio del juego con *Teaser Trailer* y descripción breve de las mecánicas
- Capturas de juego y GIFs animados que muestren la jugabilidad acompañados de pequeñas explicaciones
- Actualizaciones del desarrollo hasta el *Launch Trailer*

En Reddit se planea:

- Participar en subreddits relacionados con videojuegos, desarrollo, juegos independientes, juegos de carreras, etc. Se compartirá contenido atractivo del juego, ya sean *trailers*, *clips*, *screenshots*, GIFs animados, la portada, etc.
- Celebrar un AMA (*Ask Me Anything*), una sesión de preguntas y respuestas entre el equipo de desarrollo y la comunidad
- Participar en discusiones sobre el juego, haciendo referencia al *feedback* recibido, y en discusiones sobre otros juegos o temas relevantes de la industria

PLAN DE MARKETING EN LINKEDIN

Dado que una gran número de profesionales de la industria de los videojuegos y de la educación son activos en LinkedIn, se pretende crear un perfil de empresa en esta plataforma.

El enfoque de las publicaciones será de corte más profesional, poniendo especial énfasis en mostrar las capacidades del equipo, los objetivos de la compañía con el juego *Rally Team Tactics* y los resultados del desarrollo (el producto final). De esta manera se buscará captar la atención de instituciones educativas o *publishers* de videojuegos para financiar este y futuros proyectos.

14.2.3 CROWDFUNDING A TRAVÉS DE KICKSTARTER



A raíz de que no hay certeza de que se pueda conseguir apoyo financiero por parte de instituciones o *publishers*, se ha planteado una campaña de Kickstarter con los siguientes objetivos:

- Recaudación de fondos poniendo inicialmente un listón relativamente bajo que incrementaría considerablemente de superarse, prometiendo más contenido y una experiencia más completa y accesible (*ports*, DLCs, etc.)
- Construcción de una comunidad de *backers*, entusiastas de los videojuegos independientes o de carreras, que ayuden económicamente al equipo, participen en el *testing* y actúen como promotores del juego

15 PRUEBAS Y ANÁLISIS

15.1 ANÁLISIS MDA

A través del *framework MDA (Mechanics, Dynamics, Aesthetics)*, se pretende en este apartado formalizar la experiencia del videojuego. Recordemos que este marco requiere de la definición de reglas (mecánicas), estilos y estrategias de juego emergentes (dinámicas) y emociones y estados de ánimo (estéticas). Para lograr el estado emocional deseado, como diseñadores debemos comprender y prever cómo va a interactuar el jugador con las reglas descritas.

En este análisis, se describirán las respuestas emocionales que se desean evocar en el jugador, acompañadas de los comportamientos que las producirían, es decir, las estrategias que se espera que surjan de las mecánicas de RTT. Repasemos las mecánicas de juego antes de abordar las estéticas.

De manera resumida, *Rally Team Tactics* gira en torno a la elección de instrucciones en un trazado de *rallies*. Los jugadores toman el papel de copilotos, y disponen de una serie de notas de código nemotécnico y una barra de control del temperamento para determinar el comportamiento del vehículo (controlado por una IA) en diferentes partes de cada prueba. Deben considerar además el consumo de recursos como el combustible o la integridad del coche (inversamente proporcional al daño sufrido).

Los jugadores tienen una visión aérea con una cámara que pueden mover libremente a una determinada altura siempre que estén dentro de una prueba. Toman decisiones durante la fase de reconocimiento, y ven el resultado de estas cuando pasan a la fase de ejecución al seleccionar todas las instrucciones en todos los puntos de control de la prueba y pulsar al botón de ejecutar. Durante la ejecución, siempre pueden reiniciar, y solo se verán obligados a hacerlo si se quedan sin algún recurso o se salen de la pista. Si se completa la prueba, se puede registrar el tiempo obtenido, junto con las métricas de los recursos, y visualizar una tabla de clasificación en línea.

Una vez recapituladas las mecánicas, continuamos el análisis determinando qué es lo que hace “divertido” a *Rally Team Tactics*, indicando y justificando los tipos de “diversión” que se quieren conseguir por orden de relevancia en el juego. Al mismo tiempo se explica la jugabilidad prevista derivada de las mecánicas de juego que, potencialmente, daría lugar a cada estética:

- **DESAFÍO:** RTT es un juego de puzzles en realidad camuflado como juego de conducción. Se plantea un reto al jugador que debe superar mediante el uso del Pensamiento Computacional. Se ponen a prueba las destrezas descritas en [3.3 Destrezas del Pensamiento Computacional en el juego](#).

La presencia de unos límites estrictos sobre cuándo podemos dar órdenes al piloto y de qué manera fomenta el ejercicio de las destrezas del Pensamiento Computacional por parte de los jugadores. Además, la gestión de recursos sube el nivel de reto a superar, y es otra variable a considerar al tomar decisiones durante la partida.

- **FANTASÍA:** ponemos al jugador en la piel de un copiloto de *rally*. El público objetivo del juego, los niños y niñas, pueden dirigir al piloto, estar en control

de un equipo de *rallies*, con la responsabilidad que eso supone pero también el orgullo que produce. Los niños tienden a sentirse orgullosos cuando viven esa fantasía de que pueden hacer cosas que solo los adultos hacen, como conducir o dirigir un equipo profesional en alguna disciplina.

Para hacer sentir responsabilidad y poder de decisión, se refuerzan el peso de la carga que el jugador asume y las implicaciones de sus acciones. Esto se hace tanto de manera positiva como negativa. Por ejemplo, si el coche choca, los daños deben ser visibles y notables para el jugador, de tal manera que se sienta culpable por lo sucedido y se vea incentivado a adoptar una estrategia más segura la próxima vez. Conseguir completar una etapa o un tramo con un buen tiempo busca todo lo contrario a través del sonido y las clasificaciones: crear un sentimiento de orgullo, de recompensa por la buena toma de decisiones.

- **COMPETICIÓN:** la comparación de tiempos frente a otros usuarios pone al jugador de *RTT* en competencia con ellos. Lograr mejores tiempos va más allá de la superación personal por la presencia de clasificaciones. Se busca incentivar la rejugabilidad a través de la satisfacción de superar a los demás. La implicación emocional de los jugadores en el marco competitivo se pretende alcanzar mediante la muestra de tablas de clasificación el límite siempre que se finaliza una prueba. Así puede ver claramente a quienes tiene que batir o ha batido. El tiempo del jugador siempre aparece en la tabla junto a su posición, aunque no esté en los primeros puestos, para que sepa cuál es el listón que el resto de usuarios están poniendo y cómo de lejos (o cerca) está de alcanzarlo.

- **SENSACIÓN:** se pretende evocar alegría en el jugador al completar etapas o tramos, viendo cómo su estrategia da resultados. El estilo artístico colorido y los sonidos al cruzar puntos de control o la meta buscan reforzar este sentimiento de éxito, de logro. El jugador ve como su habilidad le ha llevado a la consecución del objetivo y se siente entusiasmado.

Los sonidos de paso por puntos de control y la meta tienen como objetivo hacer sentir entusiasmado y alegre al jugador. Llegar a estos puntos de control supone un esfuerzo previo al jugador, quien durante la fase de reconocimiento decide las instrucciones. Todo ese trabajo da sus frutos en la fase de ejecución, cuando el jugador puede observar, simplemente con pulsar un botón, cómo sus decisiones le dan resultados.

- **DESCUBRIMIENTO:** el comportamiento físico del vehículo y su reactividad ante las decisiones del jugador buscan crear esa chispa de curiosidad por explorar y experimentar.

El ensayo y error que están en el núcleo del diseño de *RTT* son los potenciadores de esta estética. La experimentación la permite la elección de instrucciones en el trazado, que posibilita una infinidad de combinaciones, y que resulta en comportamientos físicos que el jugador podría no esperar una vez se pasa a la fase de ejecución. Cuando el jugador descubre inesperadamente una estrategia que da un gran resultado, se ve incentivado a continuar probando nuevas tácticas.

16 POST-MORTEM

16.1 POST-MORTEM DE LA BETA

En este apartado se presenta la retrospectiva del equipo de desarrollo tras la entrega de la versión beta del juego. Para esta tarea, se recogieron primero documentos individuales de cada uno de los componentes. Después de una reunión donde se discutieron dichos documentos, se consensuó este *post-mortem* conjunto:

16.1.1 ¿QUÉ FUE BIEN?

EL RESPETO ENTRE COMPAÑEROS

A excepción de un par de momentos tensos, la mayor parte del desarrollo se caracterizó por una conversación respetuosa y de crítica constructiva. Quienes quedaron rezagados en algún punto o fallaron en comunicar sus avances o problemas asumieron su responsabilidad y demostraron voluntad de cambio. A pesar del enfado que el programador principal (y Scrum Master) y yo (como director y principal diseñador), sentíamos ante la falta de interés que el resto mostraba en ciertas ocasiones, no se crearon brechas en el grupo, se fue duro pero no injusto, se midió la crítica para impulsar el cambio de manera natural y positiva.

DESCRIPCIONES DETALLADAS DE LAS TAREAS

Gracias al esfuerzo del Project Manager y del Scrum Master, las tareas definidas en Trello contaron con descripciones extensas y detalladas. Esto facilitó en gran medida el trabajo al resto del equipo, poniendo plazos y condiciones. Todos los miembros tenían acceso a todos los tableros, y aunque algunos no consultaban la plataforma con frecuencia, se utilizó con asiduidad para guiar al grupo.

GESTIÓN ESPACIAL DE DOCUMENTACIÓN Y ORGANIZACIÓN

Durante el desarrollo hubo múltiples cambios de localización espacial y en sintáctica de los documentos, imágenes e indexación, y todos ellos comunicaron por el grupo, siguiéndose las directrices. Esto resultó en facilidad de navegación por los directorios, y de subida de *assets*, documentos y otros. Cada miembro podía trabajar en mayor o menor medida asíncrona e independientemente porque no se necesitaba una comunicación explícita entre distintas personas o equipos.

Por ejemplo, si programación necesitaba *assets* en un momento dado, no incurría en un *deadlock* porque sabía dónde encontrar esos recursos y qué tareas no eran bloqueantes. Es decir, sabía qué estaban haciendo los demás para ajustar el flujo de trabajo acorde con el *sprint* actual.

CORRECTO USO DE GIT

Algunos componentes del grupo estaban usando por primera vez Git y GitHub y pudieron adaptarse con rapidez al uso de la herramienta en el día a día. Hubo momentos donde se cometieron errores en el manejo de ramas, pero en general se hizo un uso responsable y fructífero de los repositorios.

ENTREGAS PUNTUALES Y DIRECCIÓN EFECTIVA

La mayoría de fechas y plazos se cumplieron puntualmente, evidenciando un compromiso sólido con los tiempos establecidos. La dirección firme del *Project Manager* garantizó la claridad en las tareas y una presión adecuada para cumplir con las entregas a tiempo.

16.1.2 ¿QUÉ IBA MAL Y SE CORRIGIÓ A TIEMPO?

PROBLEMAS CON EL DISEÑO INICIAL

Al comienzo, programación especialmente no comprendió adecuadamente o no le fue expuesta con claridad la descripción mecánica del juego, así como su terminología, jerarquía de niveles, y demás.

Sin embargo, se captó el problema rápidamente en la entrega de la versión Alpha, y consecuentemente el equipo se reunió para acordar todas las definiciones que no eran conforme a la intención de la visión original del *Game Designer*. También, de cara a la Beta, y con el objetivo de conseguir un diseño en la práctica más orgánico de los niveles, se optó por modificar con prontitud la esquematización antes de modelar o implementar algún aspecto.

MECÁNICAS AMBICIOSAS

Cierta mecánica original del juego dictaba que el coche se detendría en cada punto de control para la selección posterior de instrucciones. Este modelo resultó ser muy difícil de implementar y se actuó rápidamente desde el equipo de diseño para plantear un modelo más simple donde todas las instrucciones debían ser seleccionadas antes de ejecutar y no se podría rebobinar sino reiniciar.

REVISIÓN Y EVOLUCIÓN DE LA JUGABILIDAD

Se identificó inicialmente un enfoque poco atractivo en la jugabilidad, que consistía en asignar orden tras orden. Se realizó un cambio significativo, optando por un sistema donde se enlazan varias órdenes entre puntos de control. Esta modificación transformó la jugabilidad, haciéndola más divertida y dinámica.

DISEÑO DE UI MÁS COMPLEJO DE LO ESPERADO

Se diseñaron funcionalidades para los menús que finalmente no se implementaron porque eran inabordables para el equipo de programación. No obstante, este equipo entendió que se podían tomar ciertas libertades y consiguieron sacar adelante la mayor parte de la UI sin ceñirse de manera estricta al diseño inicial.

DIFICULTADES AL ALBERGAR LA BASE DE DATOS

A pesar de que uno de los programadores puso gran esfuerzo para sacar adelante una BD en MySQL para almacenar información de usuarios y métricas, no se encontró un servidor externo gratuito para albergarla. Esto no detuvo el almacenamiento de datos, se cambió de cara a la Beta a archivos de texto modificables en tiempo de ejecución.

Se consiguió implementar este modelo provisional antes de la entrega, dejándose la BD con servidor externo para más adelante.

TRANSPARENCIA Y COMPROMISO CON LAS TAREAS

El uso de Trello de manera más exhaustiva permitió que gran parte del equipo, que no comunicaba sus avances con claridad ni se comprometía a los objetivos con determinación, cambiase de modus operandi. La monitorización y consulta frecuente de los tableros dio lugar a un mayor grado de transparencia y la comunicación entre los miembros del equipo mejoró: ahora cada uno sabía lo que tenía que hacer, se sujetaba a un plazo y avisaba de retrasos inesperados.

COMUNICACIÓN DEL EQUIPO

Durante la primera entrega, hubo muchos problemas en la comunicación. Algunos miembros solo sabían en qué estaban trabajando el resto a través de las reuniones semanales.

A lo largo de la segunda fase se desarrolló un mayor compromiso de comunicar qué se finalizaba, qué necesitaba más tiempo y qué era inviable.

Ahora bien, la comunicación, a pesar de que mejoró con respecto a la primera entrega, aún tiene un largo camino por recorrer para evitar problemas como los que surgieron durante la entrega de la beta.

EQUILIBRIO DE TRABAJO

Aunque no se ha resuelto por completo, tras el lanzamiento de la versión Alpha, la reunión donde se pusieron en común ideas para hacer más equitativo el reparto de trabajo de cara a la Beta surtió efecto. Miembros del equipo que habían estado poco activos comenzaron a esforzarse por sacar adelante su parte, e incluso cambiaron ocupaciones de manera voluntaria para ayudar en un campo donde se estaban teniendo dificultades.

VOLUNTAD DE SER GENEROSOS CON LOS DEMÁS Y APRENDER DE ELLOS

Uno de los factores más importantes para la entrega exitosa de la Beta fue que varios miembros del equipo se aventuraron a entrar en disciplinas poco conocidas, abriendose a aprender de otros que conocían mejor el flujo de trabajo en cada campo.

El responsable de marketing y redes sociales, por ejemplo, es la representación más clara de este cambio. Debido a su poca carga de trabajo durante la primera entrega, de cara a la segunda entró en programación, con conocimiento prácticamente nulo de Unity, y puso mucho de su parte por aprender a utilizarlo correctamente, escuchando a aquellos más experimentados. No solo eso, este compañero, en cuanto vio que se quedaba sin tarea y a pocas horas de la entrega de la Beta, inmediatamente se ofreció a ayudar al responsable de producción, que tenía mucho trabajo. No solo demostró humildad al cambiar su tendencia defensiva respecto a las críticas constructivas por la escucha atenta de las opiniones otros miembros, también lo hizo al no conformarse con hacer las tareas hechas de cualquier manera, aspirando a dar lo mejor de sí mismo.

16.1.3 ¿QUÉ FUE MAL?

CONFUSIÓN RESPECTO AL DISEÑO DE JUEGO

Cierta mecánica original del juego generó problemas debido a discrepancias entre el diseño inicial y final, que fueron realizados por dos miembros diferentes del equipo. Esta mecánica es la que abarca la selección instrucciones en los puntos de control. Se tenían dos modelos distintos, adoptándose finalmente con dificultades el último de estos, donde un único punto de control permitía la selección de instrucciones para varios que lo seguían en lugar de únicamente tener gobierno sobre sí mismo. Esto dio origen a confusión también entorno al concepto de “punto de control” en el juego, y embarró las reuniones entre programadores y diseñadores.

DOCUMENTACIÓN INNECESARIA

La cantidad de documentación que se ha desarrollado para la entrega Beta ha sido excesiva. Esto ha provocado que ciertas tareas tuvieran que rehacerse o aplazarse, en vez de poder trabajar en otros aspectos e ir menos ajustados de tiempo. Además, esta documentación que supuestamente aliviaria problemas de comunicación entre distintos equipos, no los evitó, en parte porque no se utilizó o era demasiado ambiciosa.

Antes de establecer tantas especificaciones, se debería hablar del alcance del equipo. Deben quedar claras las capacidades y conocimientos de los programadores, para juzgar qué es posible desarrollar en el tiempo establecido.

GESTIÓN DE TIEMPO

Uno de los principales problemas fue no establecer un plazo para todas las tareas. Algunas ocuparon todo el desarrollo de la beta, limitando significativamente el número de personas en las que podían recaer nuevas responsabilidades. Esto ocasionó que otras tareas de importancia se aplazasen o incluso descartasen. Se debería establecer un sistema de prioridades que evite sobrecargas de trabajo en plazos de tiempo muy justos.

PLANIFICACIÓN DEL PROYECTO A MEDIO Y LARGO PLAZO

La gran cantidad de dependencias entre las tareas y la definición difusa de objetivos a medio-largo plazo ha llevado a que el producto entregado en la Beta no sea el juego completo. La mayor parte del trabajo ha tendido a hacerse tarde y deprisa, se ha fallado en juzgar el peso de las tareas de principio a fin. El responsable de arte del equipo no entró con fuerza hasta bien tarde, e igual pasó con los programadores. Se enfocó mucho esfuerzo a definir el diseño con claridad, y se pospuso la implementación. El detallismo por mi parte no fue de ayuda, pues estar refinando constantemente el diseño hace que este tarde más en llegar a quienes deben evaluar su viabilidad e implementarlo. Hay que buscar una vía más iterativa, ponerse objetivos a corto, medio y largo plazo razonables desde el comienzo. No se puede estar planificando cada semana, agota a quienes se encargan de repartir tareas (como yo), quienes escriben descripciones detalladas de estas y tienen que asegurarse de que

todos entienden y están al tanto de lo que hay que hacer, lo cual lleva mucho tiempo y esfuerzo.

ASIGNACIÓN DE ROLES

Cuando se asignaron roles al comienzo del proyecto, especialmente los que se pusieron al frente de programación sobreestimaron sus capacidades. Yo mismo, como diseñador, sentía frustración porque tenía claro que podía aportar en ese campo, pero mi compromiso con el diseño era fundamental para cuando se vio que el equipo de programación no estaba trabajando con fluidez. Yo por mi parte considero que tengo que ser menos exigente con el resto de componentes del grupo cuando realizan tareas de diseño, pues no todos tienen los mismos estándares que yo tengo, y creo que establecer una asignación de roles más dinámica podría dar resultados positivos.

AUSENCIA DE PROGRAMADORES CON EXPERIENCIA EN EL EQUIPO DE PROGRAMACIÓN

En el videojuego la programación presenta cierta dificultad, y con la falta de experiencia del equipo de programación en Unity aparecieron numerosos problemas en diversos puntos del desarrollo.

- En primer lugar, la dificultad de incrustar la mecánica de control por PID, que, por un lado, era muy compleja, y por otro, una vez escogida la parametrización supuso mucha dedicación en horas de trabajo. Además, se continúa sin resolver algunas problemáticas como el giro en curvas con los pesos escogidos.
- Después, la falta de homogeneidad en cuanto a la implementación y el código derivó en el desperdicio de código que no había sido correctamente escrito.

La asignación de roles podría haber ayudado. Si en un inicio se hubiese evaluado con más seguridad quienes estaban más capacitados para cada tarea, se podría haber mitigado el problema, pues personas que quedaron fuera del equipo de programación mostraron su frustración en ciertos momentos al ver que podrían haber aportado mucho en ese campo.

CARENCIA DE TESTING

Apenas se dedicó tiempo a realizar pruebas y *testing* del juego. Ni siquiera internamente. Algunos miembros no habían probado el juego antes de la entrega.

El *testing* se debe priorizar y debe realizarse constantemente, pues como se está viendo en este proyecto, su ausencia da lugar a decenas de errores de implementación y de diseño.

PLAN DE MARKETING POCO TRABAJADO Y TARDÍO

El plan de marketing quedó relegado a los últimos días antes de la entrega, y no se pudo por tanto conseguir visibilidad en redes sociales y en las esferas de desarrollo indie. Por este motivo, se ha tenido que comenzar a promocionar el juego muy tarde, y se están teniendo dificultades a la hora de captar la atención de las audiencias objetivo.

17 CAPTURAS

17.1 BETA

Se muestran a continuación algunas capturas tomadas de la versión *beta* (19 de noviembre de 2023) de *Rally Team Tactics*:



Ilustración 109. Captura sin interfaz de un escenario y el coche I



Ilustración 110. Captura sin interfaz de un escenario y el coche II



Ilustración 111. Captura sin interfaz de un escenario y el coche III

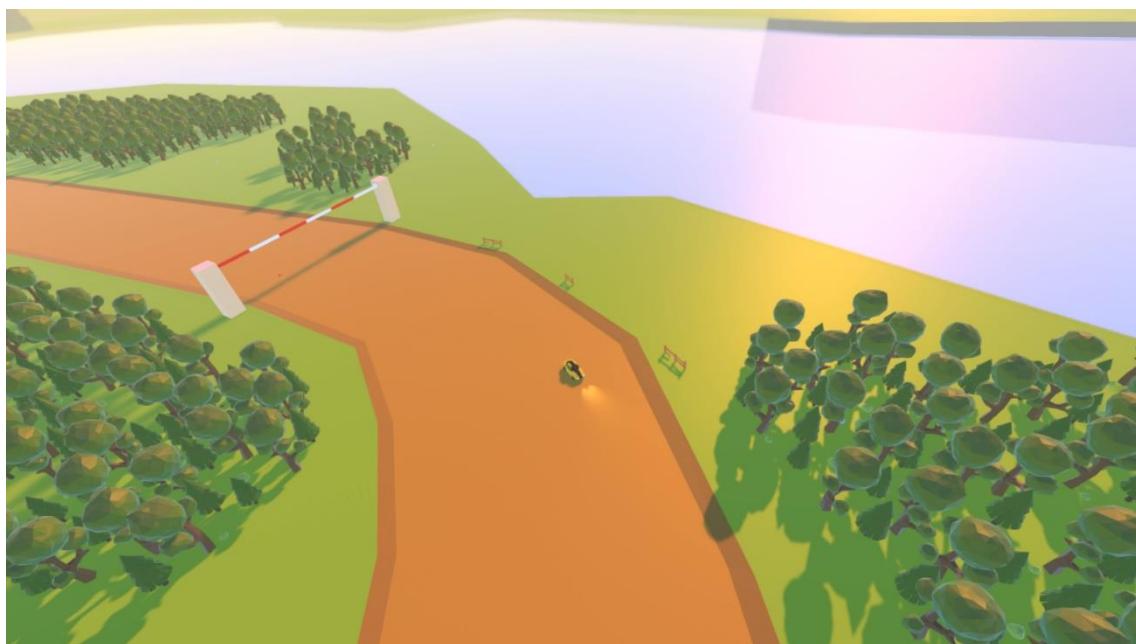


Ilustración 112. Captura sin interfaz de un escenario y el coche IV

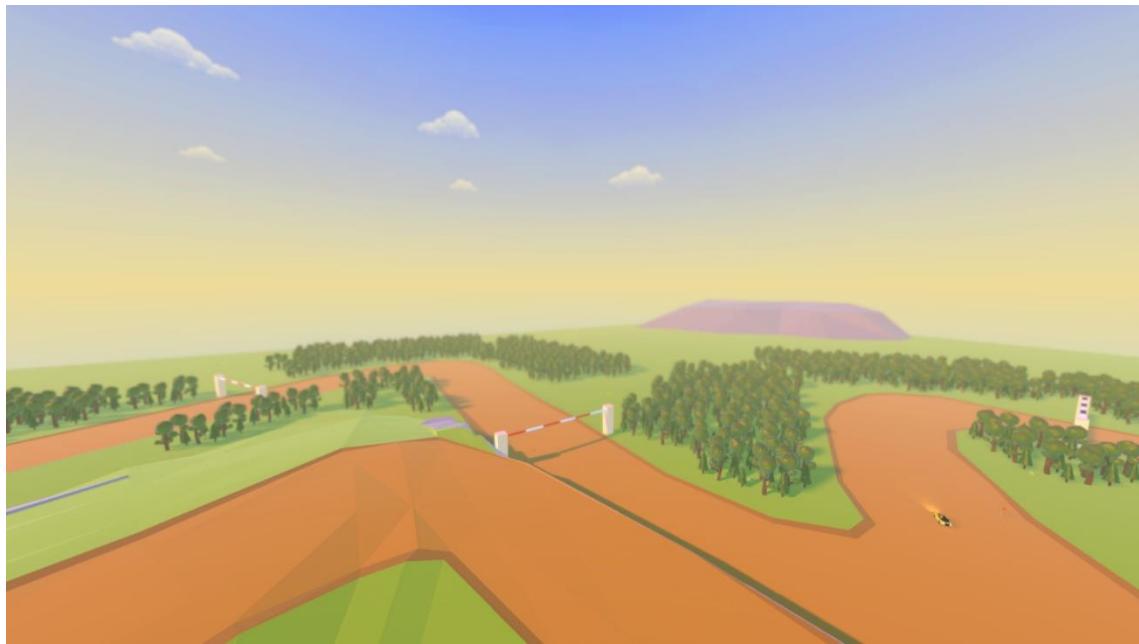


Ilustración 113. Captura sin interfaz de un escenario y el coche V

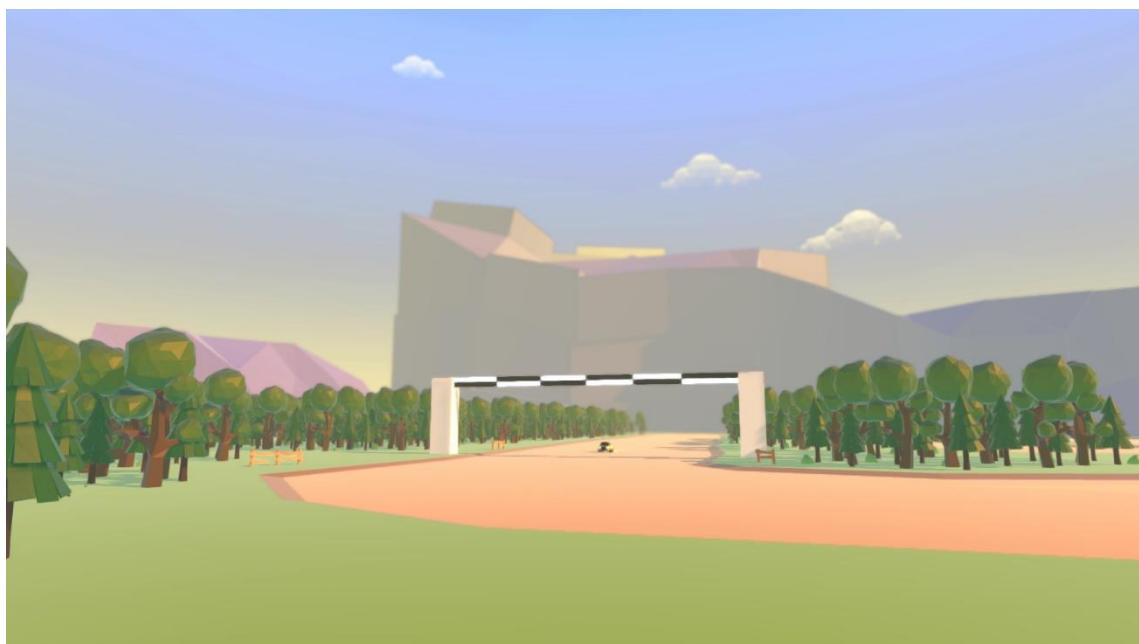


Ilustración 114. Captura sin interfaz de un escenario y el coche VI

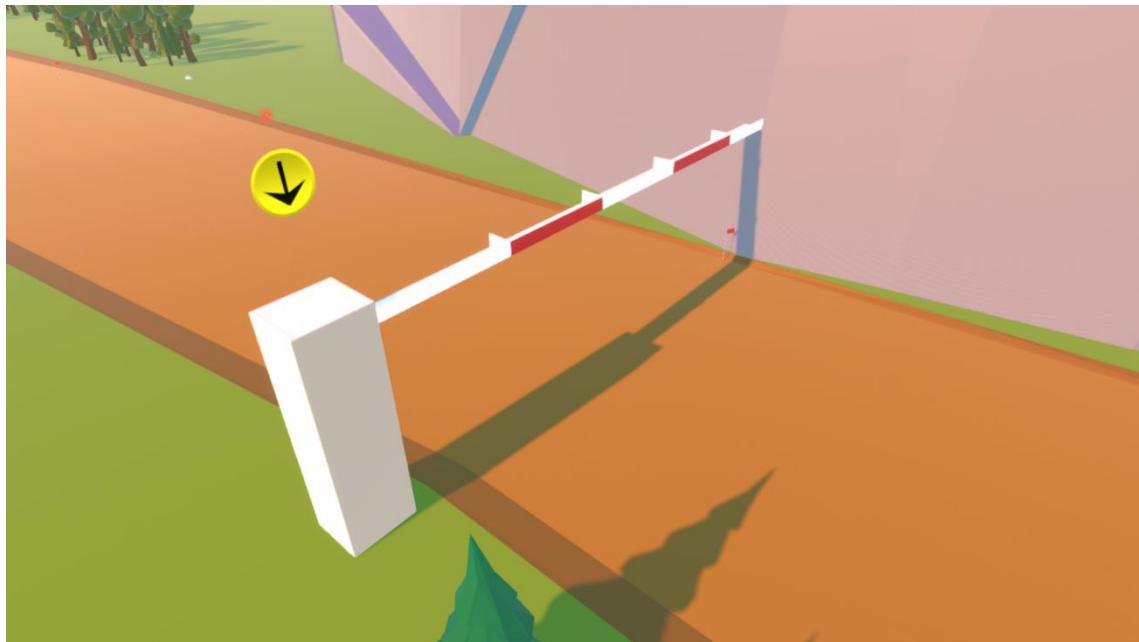


Ilustración 115. Captura con HUD de un punto de control I

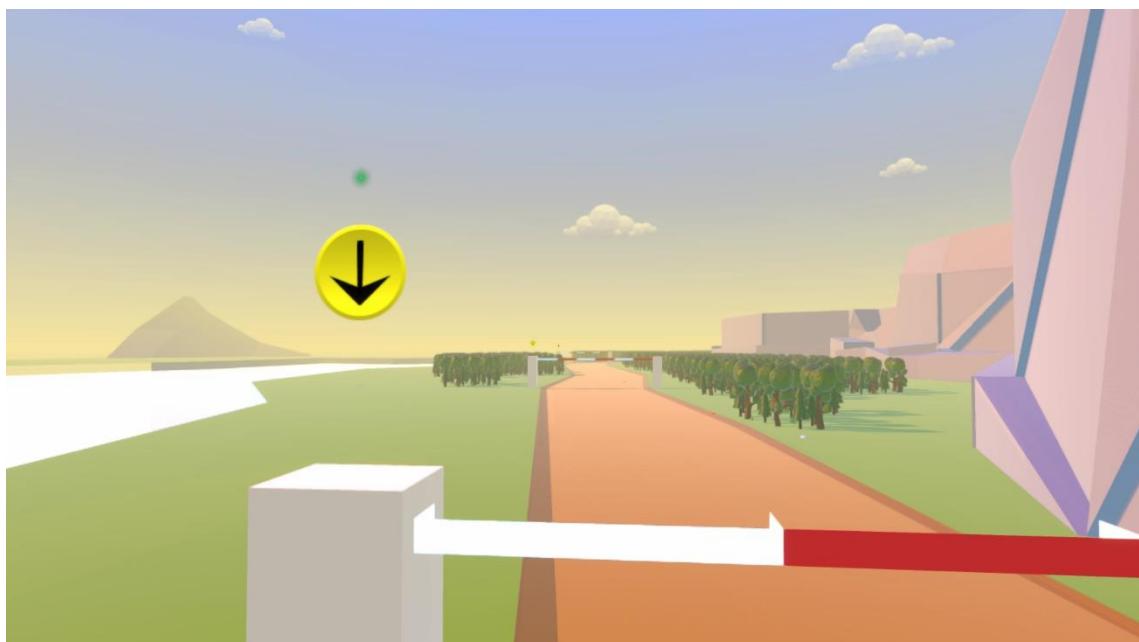


Ilustración 116. Captura con HUD de un punto de control II

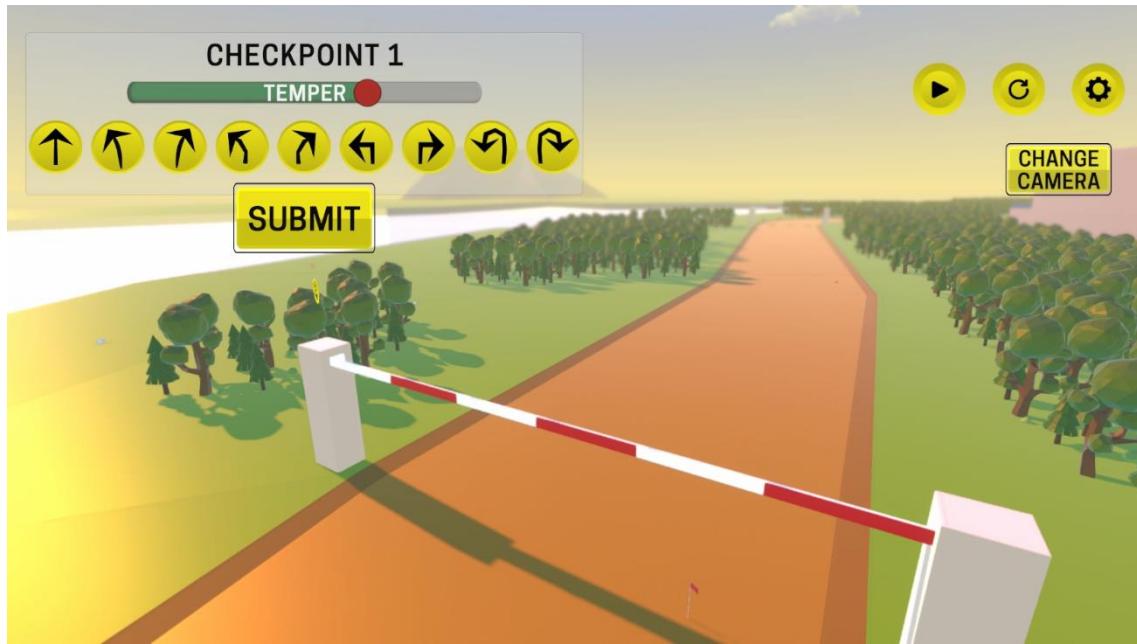


Ilustración 117. Captura del HUD al modificar la instrucción de un punto de control



Ilustración 118. Captura con HUD durante fase de reconocimiento



Ilustración 119. Captura del HUD durante fase de ejecución

17.2 GOLD MASTER

17.2.1 CAPTURAS SIN UI

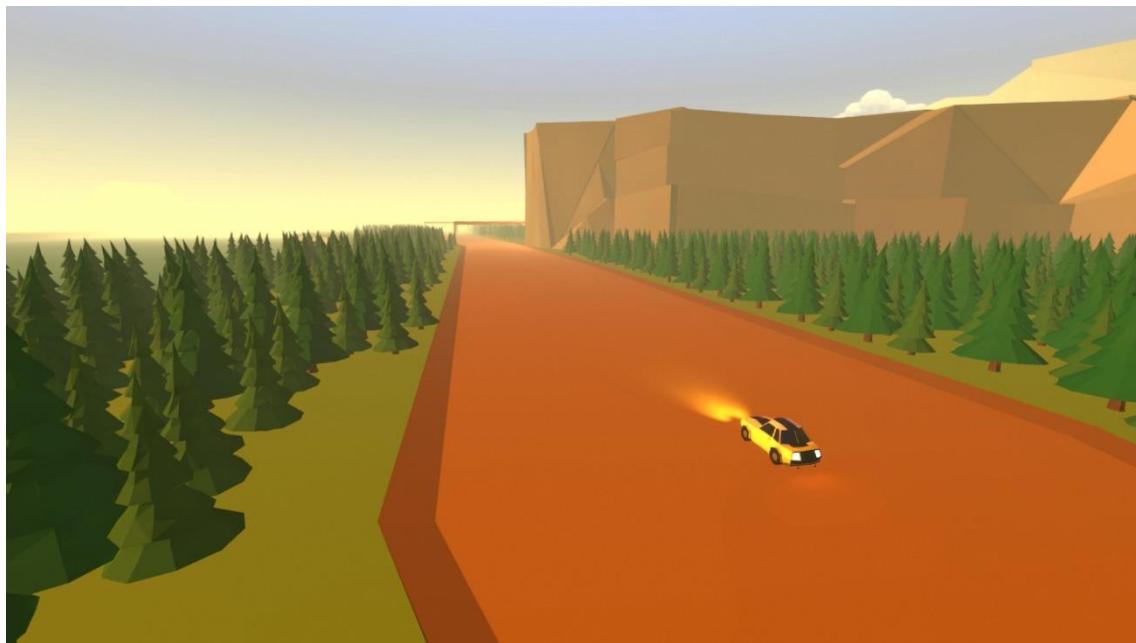


Ilustración 120. Captura sin HUD de la primera prueba I



Ilustración 121. Captura sin HUD de la primera prueba II



Ilustración 122. Captura sin HUD de la primera prueba III

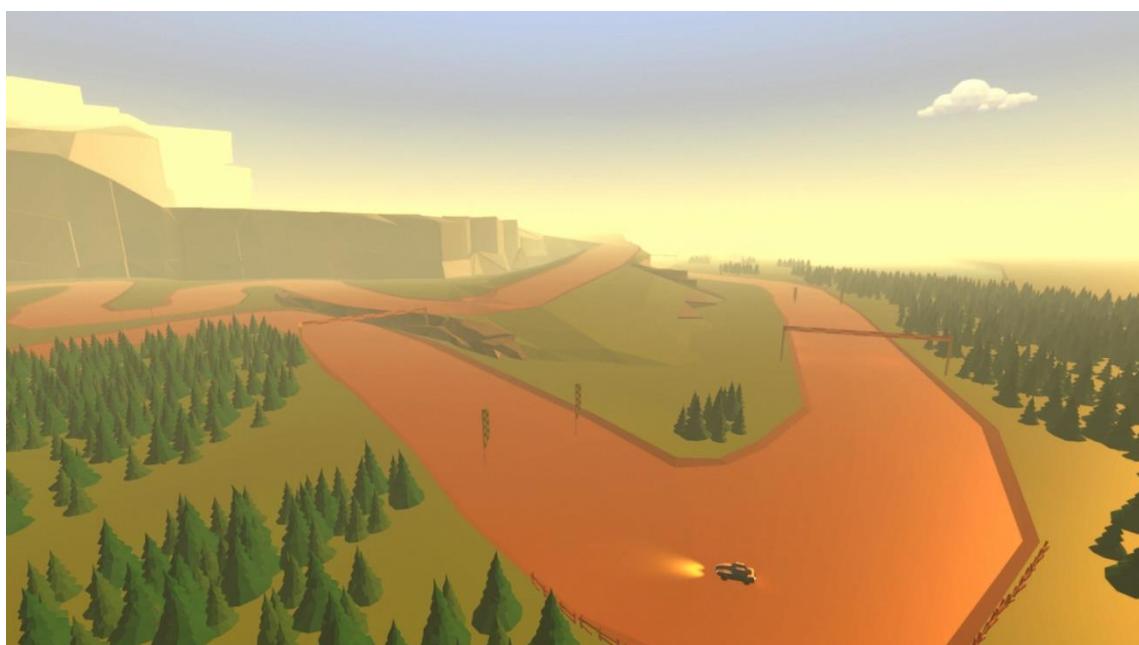


Ilustración 123. Captura sin HUD de la primera prueba IV



Ilustración 124. Captura sin HUD de la primera prueba V

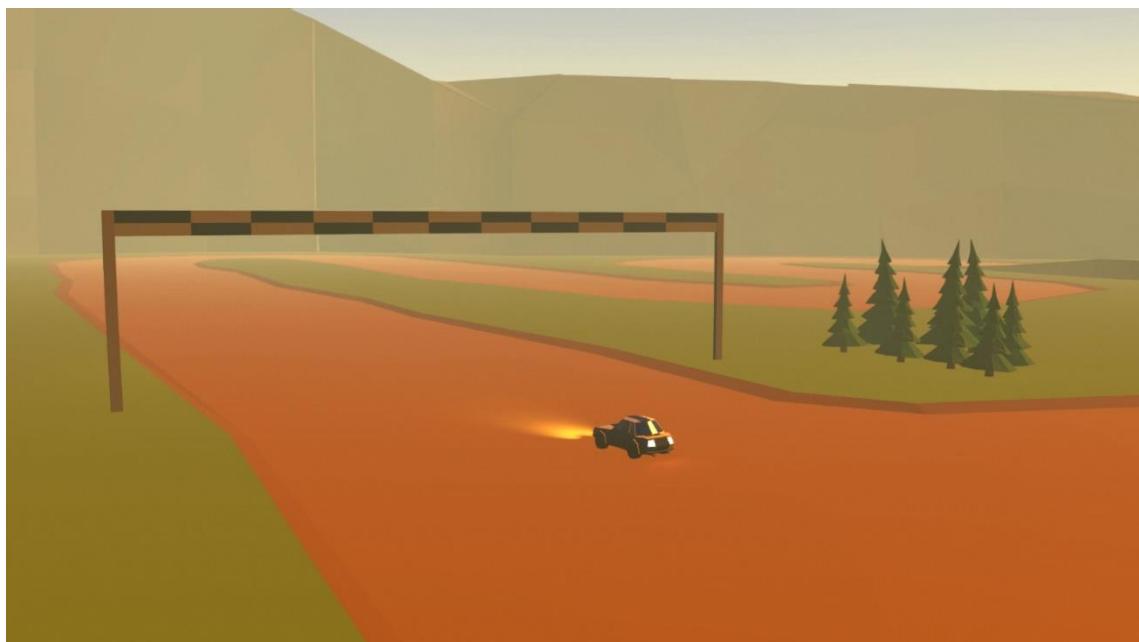


Ilustración 125. Captura sin HUD de la primera prueba VI



Ilustración 126. Captura sin HUD de la segunda prueba I



Ilustración 127. Captura sin HUD de la segunda prueba II



Ilustración 128. Captura sin HUD de la segunda prueba III



Ilustración 129. Captura sin HUD de la segunda prueba IV



Ilustración 130. Captura sin HUD de la segunda prueba V

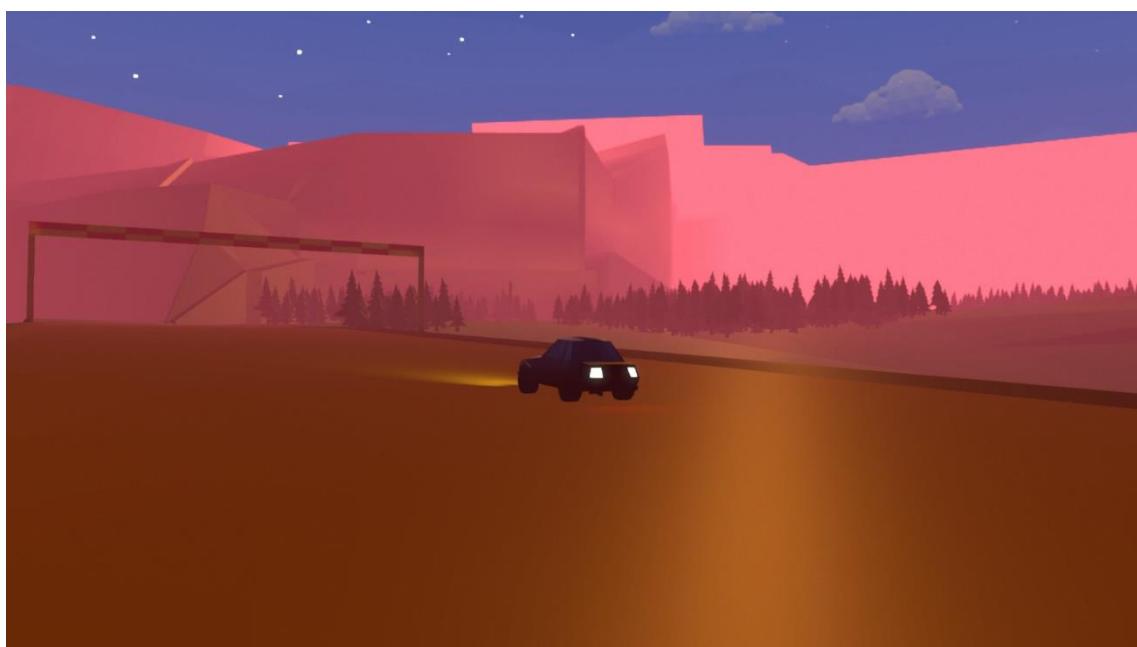


Ilustración 131. Captura sin HUD de la tercera prueba I



Ilustración 132. Captura sin HUD de la tercera prueba II



Ilustración 133. Captura sin HUD de la tercera prueba III



Ilustración 134. Captura sin HUD de la tercera prueba IV

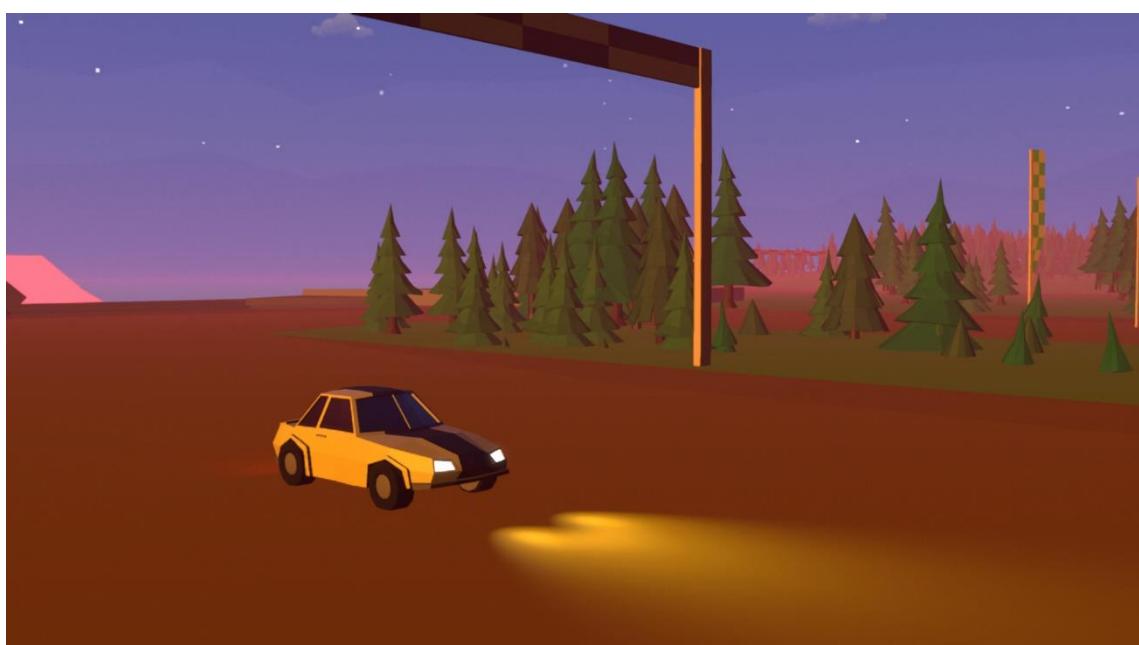


Ilustración 135. Captura sin HUD de la tercera prueba V

17.2.2 CAPTURAS CON UI

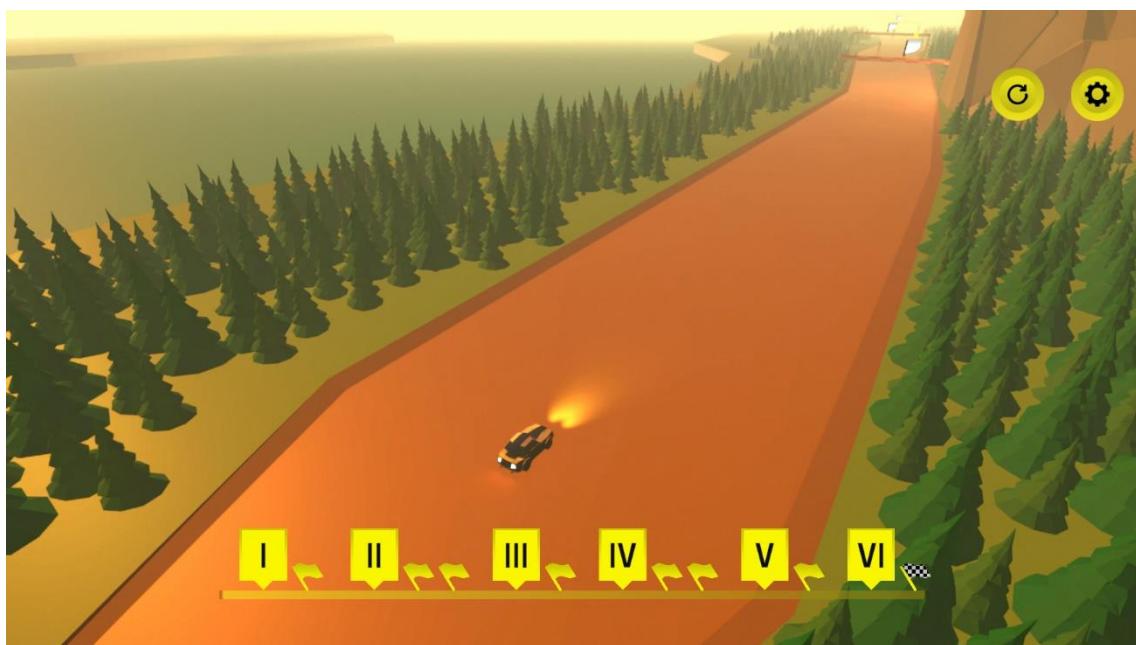


Ilustración 136. Captura HUD durante la fase de reconocimiento I

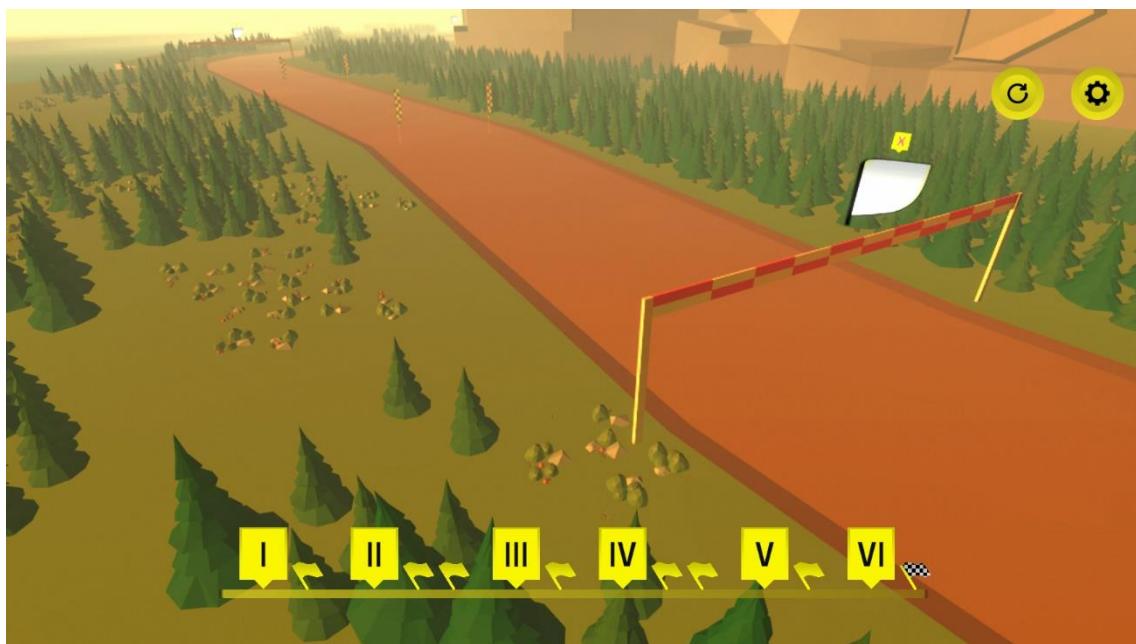


Ilustración 137. Captura HUD durante la fase de reconocimiento II

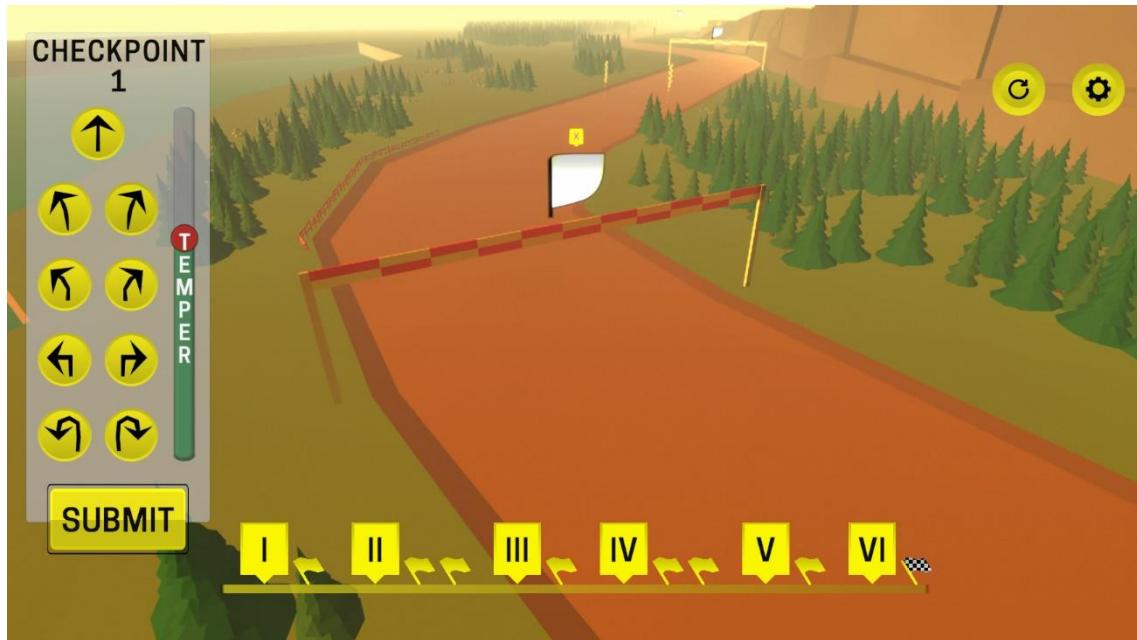


Ilustración 138. Captura HUD durante la fase de reconocimiento III: selección de instrucción

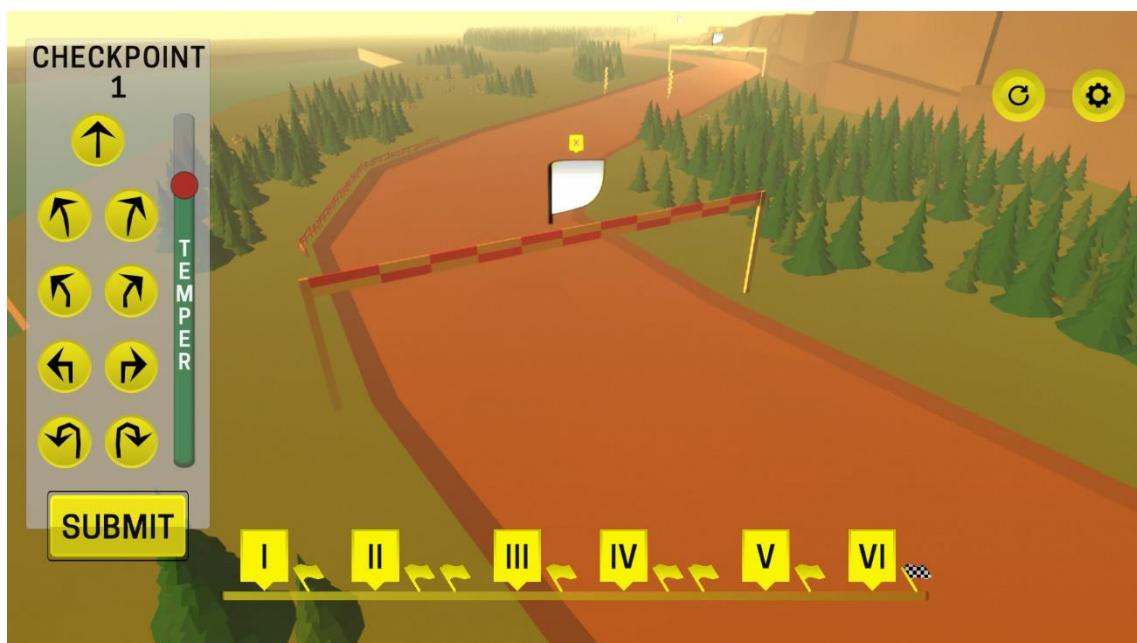


Ilustración 139. Captura HUD durante la fase de reconocimiento IV: selección de instrucción

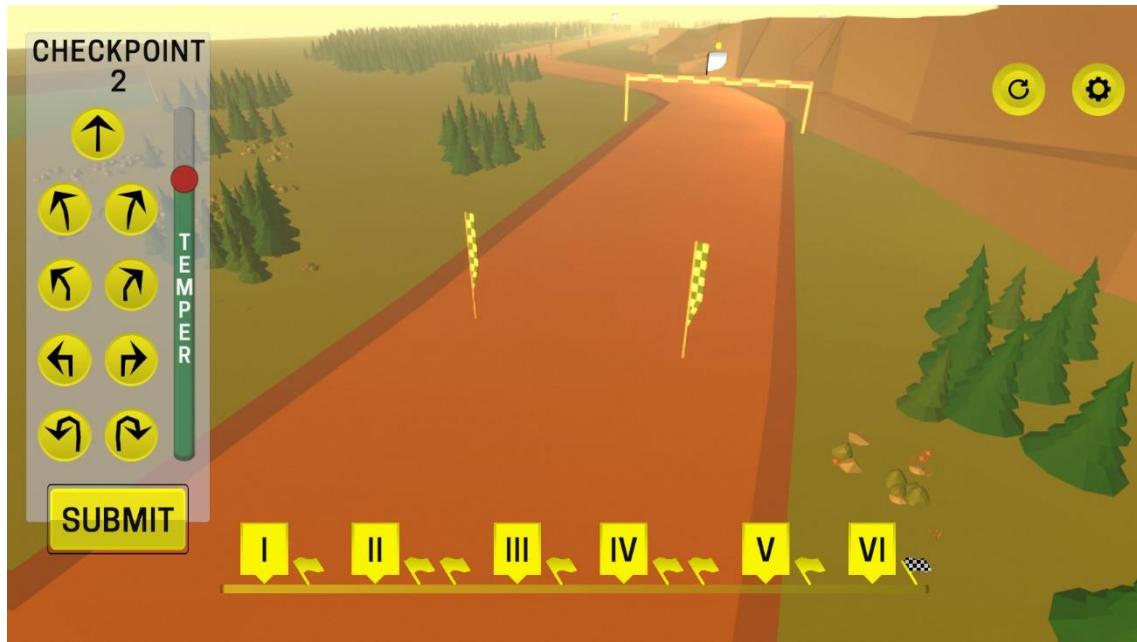


Ilustración 140. Captura HUD durante la fase de reconocimiento V: selección de instrucción

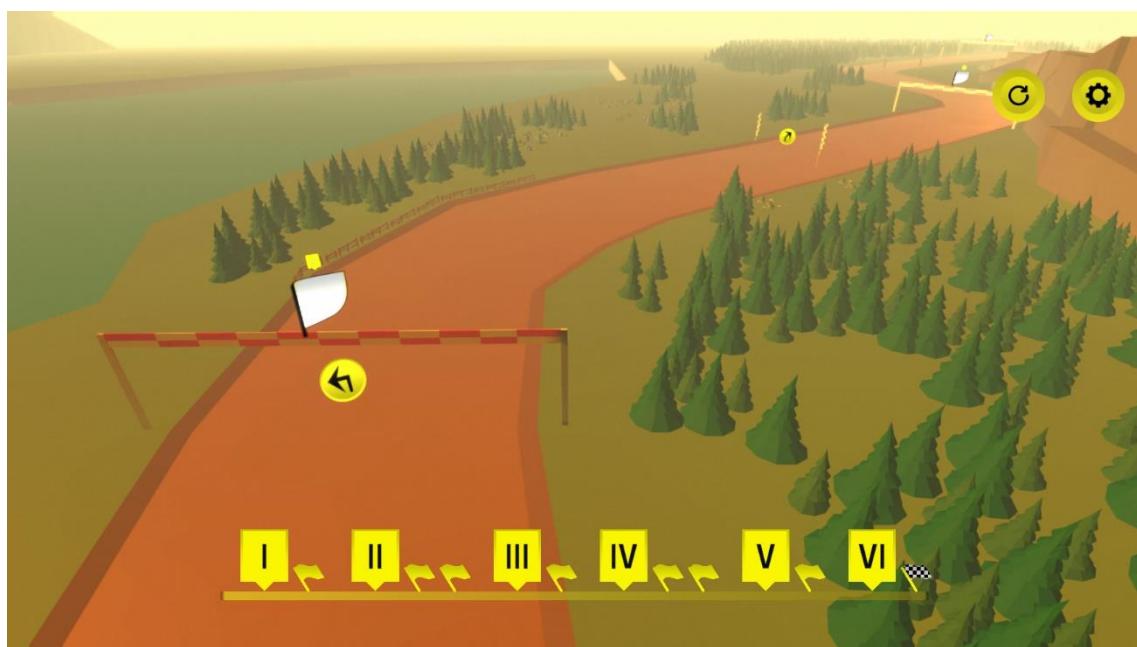


Ilustración 141. Captura HUD durante la fase de reconocimiento VI: instrucciones seleccionadas

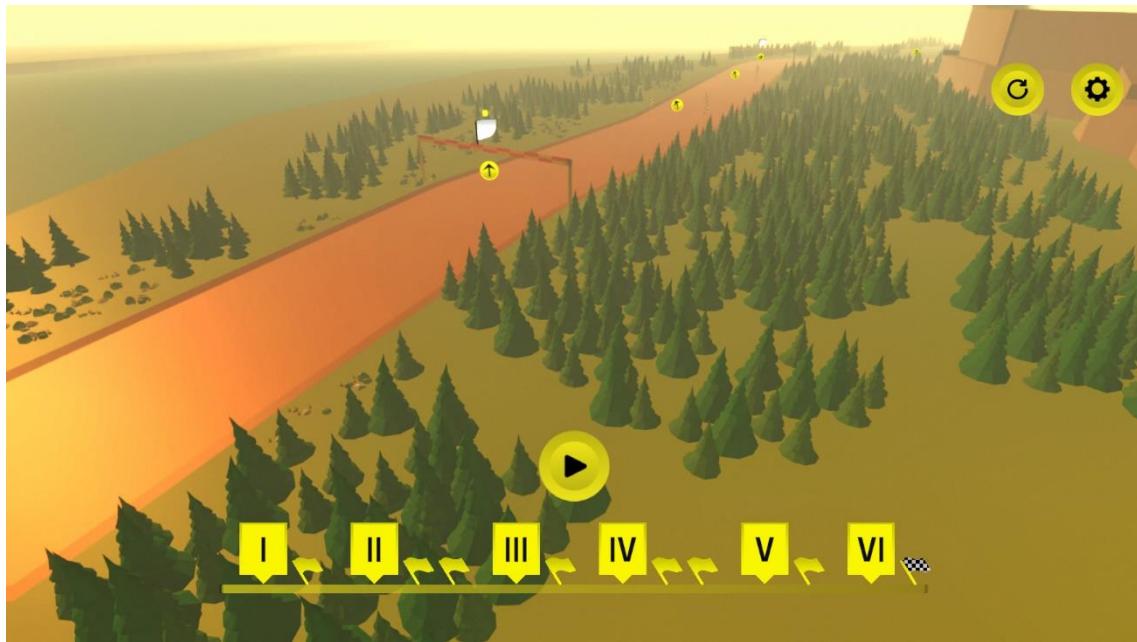


Ilustración 142. Captura HUD durante la fase de reconocimiento VII: todas las instrucciones seleccionadas



Ilustración 143. Captura HUD durante la fase de ejecución I



Ilustración 144. Captura HUD durante la fase de ejecución II



Ilustración 145. Captura HUD durante la fase de ejecución III



Ilustración 146. Captura HUD durante la fase de ejecución IV



Ilustración 147. Captura HUD durante la fase de ejecución V: checkpoint superado con éxito

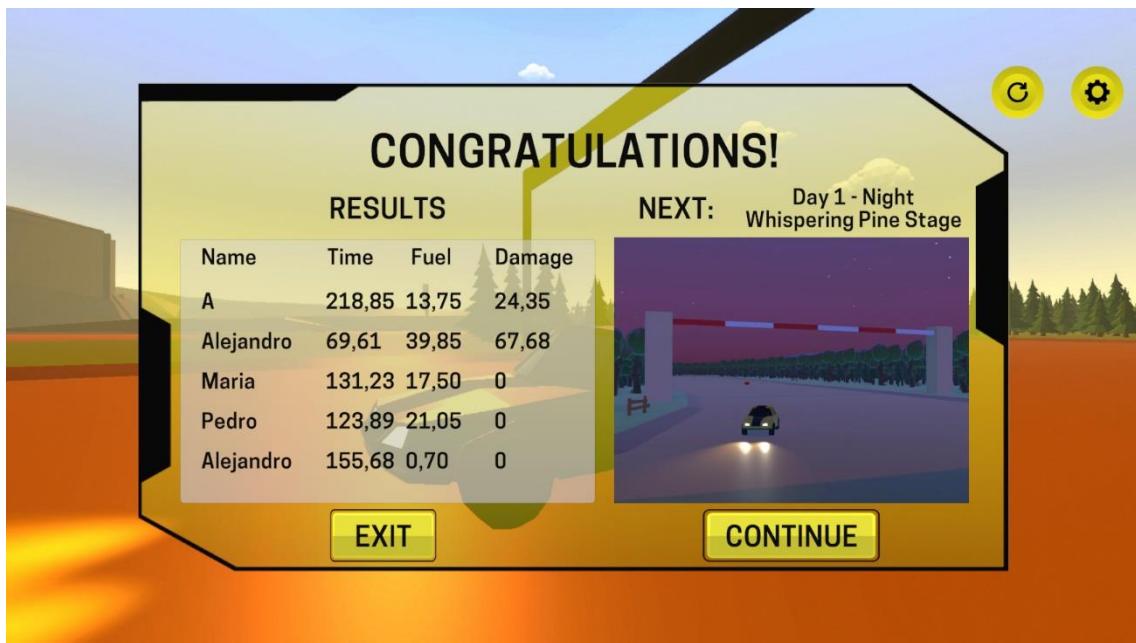


Ilustración 148. Captura HUD durante la fase de ejecución VI: prueba finalizada

GLOSARIO

TERMINOLOGÍA DEL JUEGO

Término en español	Término en inglés	Significado
Evento	Event	Conjunto de pruebas de cuatro días de duración: los tres primeros para pruebas cronometradas en los tramos de la etapa, el último para la etapa al completo.
Prueba	Time Trial	Competición contrarreloj celebrada cada día de un evento, puede ser un tramo cronometrado o la etapa completa cronometrada.
Día	Day	Uno de los cuatro días durante los cuales se celebra un evento. A cada día le corresponde una prueba, ya sea un tramo o la etapa completa cronometrada.
Etapa	Leg	Círculo punto a punto cronometrado completo correspondiente a un determinado evento. Está compuesto por tres tramos que se han probado los días anteriores.
Tramo	Stage	Una tercera parte de la etapa del evento en progreso. Se dedica un día para realizar una prueba punto a punto exclusiva de tramo, para luego conformar una de las tres partes de la etapa completa. Está compuesto por diez secciones de pista.
Sección	Section	Trozo de pista de una longitud y curvatura concretas. Se conectan para formar tramos o etapas.
Punto de Control	Checkpoint	Puntos de la prueba que el jugador puede y debe seleccionar para elegir instrucciones antes de ejecutar. Estos se sitúan al inicio de la prueba y en todas las intersecciones entre secciones. El punto de control final no es seleccionable, pues es la meta del tramo o etapa y no tiene sentido asignarle instrucción, únicamente sirve para monitorear el tiempo del jugador.
Línea de Meta	Finish Line	Último punto de control de un tramo o de una etapa (es decir, de una prueba). No se puede seleccionar para elegir instrucción, solo tiene como propósito marcar el final y pedir la muestra del tiempo y del diferencial.
Instrucción	Instruction	Conjunto de nota y temperamento asignado a un determinado punto de control por el jugador y que va a determinar el comportamiento del vehículo en la sección siguiente.
Nota	Pacenote	Código nemotécnico representado en forma de ícono en el juego que indica al piloto las características de la sección siguiente. Determina el trayecto que va a intentar seguir el coche en la sección.
Temperamento	Temper	Valor continuo que indica al piloto con qué nivel de cautela o agresividad debe acometer la sección siguiente. Determina cuando se frena, acelera o se mantiene la velocidad en la sección.

Rampa	Ramp	Obstáculo sólido de relieve que implica una inclinación de terreno en subida seguida de una bajada de la inclinación opuesta.
Meseta	Plateau	Obstáculo sólido de relieve que implica una inclinación de terreno en subida seguida de un tramo liso en altura que termina en una bajada de la inclinación opuesta a la de entrada.
Depresión	Dip	Obstáculo sólido de relieve que implica una inclinación de terreno en bajada seguida de un tramo liso en profundidad que termina en una subida de la inclinación opuesta a la de entrada.
Charco	Pool	Depresión que contiene agua y que reduce significativamente la velocidad del coche.
Reconocimiento	Reconnaissance	Fase de <i>gameplay</i> donde el jugador debe seleccionar todos los puntos de control de la prueba y asignarles instrucciones. Entonces podrá ejecutar, pasando a la fase de ejecución.
Ejecución	Execution	Fase de <i>gameplay</i> donde el coche recorre el trazado de la prueba de acuerdo a las instrucciones seleccionadas por el jugador durante el reconocimiento. En cualquier momento el jugador puede reiniciar la prueba, volviendo a ejecución. Es obligatorio reiniciar si el coche sale por completo de la pista o se queda sin recursos. En caso de que se complete la ejecución exitosamente (el coche cruce la meta), se da la opción de aceptar el resultado también.
Combustible	Fuel	Recurso de juego que se consume progresivamente de 100 a 0 en función de la velocidad, aceleración y frenado. El agotamiento de este supone que el coche quedaría varado, lo que fuerza al reinicio.
Daño	Damage	Recurso del juego que consume progresivamente la integridad del vehículo cuando se producen colisiones con obstáculos del entorno. Va de 0 a 100, llegar a 100 supone daños irreparables en el coche y fuerza al reinicio.

Tabla 7. Glosario de terminología del juego

ACCIONES DENTRO DEL JUEGO

Acción en español	Acción en inglés	Significado
Seleccionar Punto de Control	Select Checkpoint	Selección de un punto de control de la prueba para poder elegir instrucción para este. Acción solo disponible durante el reconocimiento.
Deseleccionar Punto de Control	Deselect Checkpoint	Deselección de un punto de control de la prueba para guardar la instrucción elegida en este, si es que se ha elegido alguna. Acción solo disponible durante el reconocimiento.
Elegir Instrucción	Choose Instruction	Elección de una nota de código nemotécnico y un temperamento (que por defecto se pondrá en 0.5 al elegir nota) para el punto de control que se tiene seleccionado. Acción solo disponible durante el reconocimiento cuando se ha seleccionado un punto de control.
Limpiar El Circuito	Clear Track	Eliminación de todas las instrucciones asignadas a cada punto de control de la prueba para empezar de cero. Acción solo disponible durante el reconocimiento.
Ejecutar	Execute	Supone el comienzo de la ejecución, es decir, el coche arranca y sigue las instrucciones en cada punto de control. Acción solo disponible durante el reconocimiento cuando se han seleccionado instrucciones para todos los puntos de control de la prueba.
Reiniciar	Restart	Supone el regreso del coche, con todos los recursos reestablecidos, al inicio de la prueba. No elimina ninguna instrucción seleccionada. Acción solo disponible durante la ejecución.
Aceptar Resultado	Accept Result	Aceptación del tiempo obtenido en una prueba junto con el diferencial que lo compara con el histórico. La otra opción sería reiniciar. Tras la aceptación, el tiempo quedaría registrado y se daría la opción de pasar a la siguiente prueba si es que la hay (si no, mostrar la clasificación, pues se acaba de completar la etapa) o de volver al menú de selección de prueba. Acción solo disponible durante la ejecución si se ha cruzado la línea de meta.

Tabla 8. Glosario de acciones dentro del juego

HISTORIAL DE VERSIONES

VERSIÓN 0.1 (26/09/23)

Borrador del GDD, versión de presentación de la idea inicial de acuerdo con el tema:

- Escritura de *Introducción*.
- Escritura de *Referencias y concepto de juego*. Se incluyen diagramas de flujo para el *gameplay loop*.
- Escritura de *Objetivo del proyecto: enseñanza del Pensamiento Computacional a niños pequeños*. Se incluye una tabla sobre las destrezas.

VERSIÓN 0.2 (28/09/23)

Inclusión de la plantilla con todos los capítulos y apartados que se consideran necesarios en este momento.

VERSIÓN 0.3 (05/10/23)

Modificación del concepto de juego y explicación de mecánicas inicial:

- Cambio de *Concepto de juego y bases de la jugabilidad* para adaptarse mejor a la temática y simplificar el gameplay. Se incluyen nuevas ilustraciones y diagramas.
- Escritura de los dos apartados *Cámara y perspectiva* y *Gameplay loop*. Se incluyen ilustraciones y diagramas.

VERSIÓN 0.4 (08/10/23)

Ampliación de las mecánicas, ahora describiendo obstáculos y gestión de recursos:

- Escritura del apartado *Mecánicas de juego en detalle*. Se incluyen ilustraciones y diagramas explicativos.

Explicación de la progresión en el juego:

- Escritura del apartado *Objetivos del jugador* y *Estructura del juego*. Se incluyen ilustraciones y diagramas explicativos.

VERSIÓN 0.6 (12/10/23)

Cambios en la elección de vocabulario para los diferentes elementos de juego, de tal manera que este sea consistente en todo el documento.

Modificación del *gameplay loop* general y adición de secciones para la explicación detallada de los puntos de control en las fases de juego:

- Retoque en los diagramas de flujo de *Gameplay loop* para hacerlos consistentes en vocabulario con el resto del documento y reflejar cambios en el diseño del juego.
- Escritura de los apartados *Colocación de notas en la pista* y *Cantado de notas durante la carrera*.

VERSIÓN 0.7 (17/10/23)

Reorganización de los capítulos referentes a UI/UX y arte:

- Ahora el diseño de UI va antes del capítulo de arte y se cambia el último apartado a *Diseño esquemático de la UI*, para incluir en una versión futura los bocetos o esquemas de las pantallas para todas las plataformas.
- El apartado de arte ahora se desglosa en más apartados, aún vacíos. Se incluye uno para el diseño visual de la UI y otro para *concept art*.

Integración de parte del contenido del apartado de UI/UX que ya está listo:

- Integración de los apartados *Requisitos de la interfaz* y *Diagrama de flujo de navegación*. Se incluyen diagramas.

VERSIÓN 0.8 (20/10/23)

Escritura de *Trasfondo del juego*, con una ambientación con imágenes de referencia y una narrativa donde se cuenta la historia del copiloto al que el jugador encarna en el juego.

Integración del modelo de negocio, revisado, en el GDD, concretamente en el apartado *Modelo de negocio del juego y plan de financiación*.

Paleta de colores del documento de diseño visual del juego, logo del juego, del estudio y Splash Art añadidos.

VERSIÓN 1.0 (22/10/23)

Versión del GDD para la entrega del prototipo del videojuego. En esta iteración se incorpora contenido a:

- Apartado *Diseño esquemático de la UI*, con mockups de las pantallas diseñados hasta la fecha (integración del documento de diseño de interfaces revisado).
- Apartados *Estética del juego* (integración del documento de diseño visual del juego revisado) y *Concept*, con imágenes y bocetos.

VERSIÓN 1.1 (03/11/23)

Reorganización de los capítulos del documento, en especial de la parte que cubre las mecánicas y la jugabilidad, para poder expandir en ellas sin sobrecargar ningún capítulo en particular.

Cambio de vocabulario y terminología de acuerdo con el glosario que se incluye ahora al final del documento en *Glosario*. Este se ha elaborado con el propósito de mantener un lenguaje común respecto a los elementos y acciones más importantes del juego, tanto en la documentación como en el diseño y desarrollo de este.

Integración de nuevos documentos de diseño en el GDD:

- Adición y revisión del *Sistema de daños*
- Adición del *Glosario*

Cambio del logo y de elementos de la portada y cambio de estilo del documento.

VERSIÓN 1.2 (07/11/23)

Modificación del vocabulario usado a lo largo de todo el documento referente a los elementos del juego de acuerdo con el glosario. También se han modificado esquemas y diagramas que contenían vocabulario inconsistente en los capítulos de *Gameplay loop* y *Progresión*.

Reorganización de apartados y corrección de la numeración.

Integración y corrección de documentos:

- Documento de Diseño Visual final para el apartado *Diseño visual del juego*

VERSIÓN 1.3 (09/11/23)

Escritura del apartado *Análisis MDA* del nuevo capítulo *Análisis y Pruebas*.

Integración y corrección de documentos:

- Documento de UI/UX completo, para modificar y ampliar el apartado *Interfaz de usuario*.

VERSIÓN 1.4 (15/11/23)

Integración de documento provisional de diseño de niveles, revisado y corregido, en el capítulo *Diseño de Niveles*.

Creación e incorporación de los *layouts* de controles para teclado y ratón y para pantallas táctiles al documento en el capítulo de *Controles*.

Modificación de *Información del Usuario* y del *Mapa de Empatía* para reflejar la perspectiva de los padres, quienes son realmente los clientes que van a pagar por el producto.

Reestructuración del documento para incluir nuevos capítulos, y correcciones generales.

VERSIÓN 1.5 (18/11/23)

Integración de nuevas imágenes y texto del documento provisional de diseño de niveles, revisado y corregido, en el capítulo *Diseño de Niveles*.

Ecritura de apartado *Métricas: guardado de datos y clasificación en línea* con toda la información sobre el funcionamiento de la base de datos.

Cambio de la fuente para texto en general del juego, siendo ahora *Cooper Hewitt Semibold* (apartado *Diseño Visual del Juego* modificado).

VERSIÓN 1.6 (19/11/23)

Integración del documento de Plan de Marketing para el apartado del mismo nombre, revisado y corregido.

Añadido de capturas del juego en versión Beta en el GDD (apartado *Capturas*).

Correcciones finales y revisión.

VERSIÓN 1.7 (30/11/23)

Nuevo formato de los sub-apartados del *Plan de Marketing*.

Nuevos apartados y capítulos:

- Se añade un apartado en *Capturas* para incluir imágenes del juego en versión Gold Master.
- Se incorpora el post-mortem de la fase de desarrollo de la Beta en el capítulo *Post-mortem*.

VERSIÓN 1.8 (03/12/23)

Actualización del capítulo *Diseño de Niveles* con los cambios realizados en el documento de diseño de niveles. Destaca la sustitución de los esquemas de diseño antiguo y el añadido de esquemas de diseño con obstáculos para cada etapa completa.

Redacción del *Análisis MDA* en un único apartado, sin separar estéticas, dinámicas y mecánicas.

VERSIÓN 1.9 (10/12/23)

Actualización del capítulo de Controles para sustituir los antiguos esquemas por los nuevos.

Incorporación de capturas de la Gold en el apartado correspondiente.