

Manual Técnico

Lalín



David Fernández Reboredo
Desarrollo de Aplicaciones Multiplataforma



Indice

1. Definición del proyecto
 - a. Introducción y objetivos
 - b. Descripción de procesos
2. Requisitos del programa
3. Tecnologías, frameworks, estándares, paquetes y librerías utilizadas.
4. Bocetos y esquemas
5. Diseño y estilos
 - a. Fuentes utilizadas
 - b. Tipo de diseño
6. Organización del proyecto
 - a. Carpetas y contenidos



Definición del proyecto

Introducción y objetivos

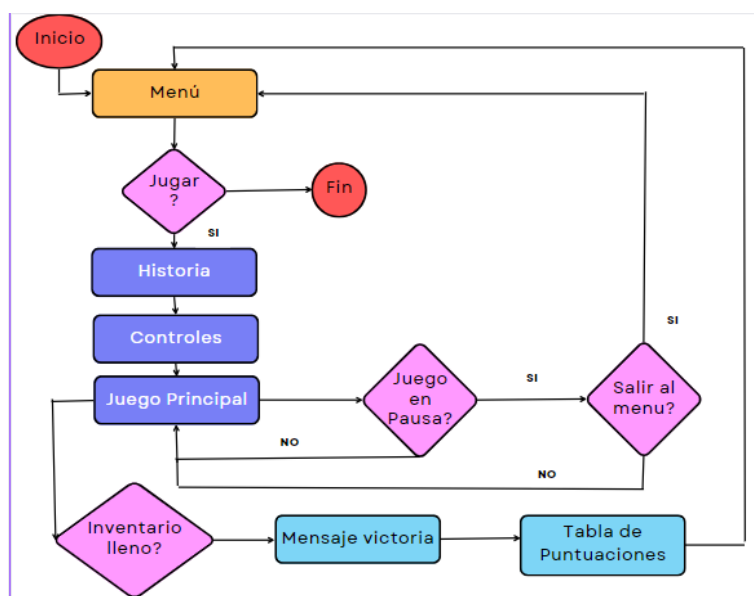
Lalín Run nos presenta un juego en tercera persona del estilo Top-Down ambientado en la villa de Lalín en el que buscaremos recuperar una serie de alimentos que nos han sido robados por nuestros enemigos en el menor tiempo posible, ya que a su vez competiremos contra otros jugadores para ver quien es capaz de recoger los alimentos en el menor tiempo posible.

Este proyecto nace con la intención de acercar la cultura y tradiciones de Lalín además de una sátira social de una forma amena y divertida. Este juego estará disponible para todo tipo de públicos y edades.

El proyecto además busca una intención competitiva que sumerja más al jugador y le proporcione más horas de juego que la historia en sí.

Descripción de procesos

El juego se desarrollará partiendo desde el menú principal, si deseamos jugar entrará en la historia, si por el contrario preferimos salir se nos cerrará el juego. Una vez dentro, nos moverá por las distintas escenas de historia y ajustes, hasta llegar finalmente al juego principal. Allí si presionamos “Esc” se nos dará la opción de volver al menú o seguir con la escena vigente. Si perdemos se reinicia la escena y si ganamos pasamos a una escena de victoria y una tabla de puntuaciones. Finalmente de la tabla de puntuaciones pasaremos de nuevo al menú principal.





Requisitos del programa

Para un excelente funcionamiento del programa debemos tener los siguientes requisitos:

- Sistema Operativo: Windows 7 o posteriores a este, así como la versión de Windows Server 2008 o versiones posteriores.
- Procesador: se recomienda mínimo un procesador de 1Ghz
- Pantalla con resolución: 1920x1080.
- Capacidad: 2 GB de almacenamiento en tu disco duro.
- Capacidad de RAM: al menos 2GB serían recomendables tenerlos libres.

Tecnologías, frameworks, estándares, paquetes y librerías utilizadas.

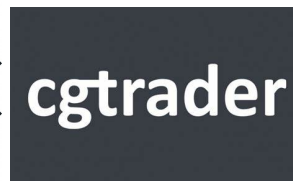
Podemos hacer una lista variada de tecnologías, frameworks, estándares, paquetes y librerías utilizadas en nuestro proyecto de Unity y que son propias de este:



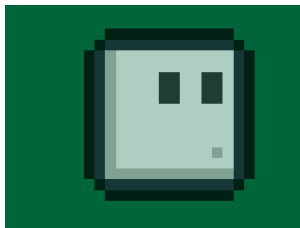
C# es un lenguaje de programación orientado a objetos que permite desarrollar juegos y aplicaciones en Unity.



Framework propio de Unity como serían: Unity Editor, Unity Asset Store, Unity UI ,Physics Engine, Unity Networking, Unity Analytics, Unity Physics, Universal Render Pipeline (URP) y High Definition Render Pipeline (HDRP),Cinemachine,NavMesh,Unity Collaborate...



La biblioteca de Assets de Unity, Itch.io, cgtrader o GameDev Market de la que salieron los diseños de personajes.



Proyectos de scoreboard como el de Danial Jumagaliyev junto con WebGL que permite el funcionamiento de nuestro scoreboard.



Bocetos y esquemas

Tal y como hemos mostrado anteriormente las escenas están esquematizadas de una forma no lineal y su lectura puede ser un tanto confusa pero sería la mostrada anteriormente.

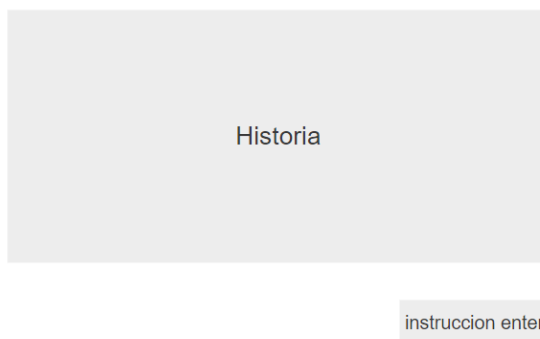
Por otra parte en cuanto a los bocetos y esquemas realizados podemos observar:

El menú:

TITULO DEL JUEGO



La historia o Escena1:

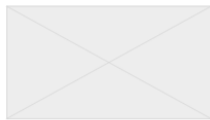




La escena de controles o Escena2:



Dirigir Personaje



Boton de accion

instruccion enter

La escena del Juego:





En la escena de Perder:



En la escena de Ganar:



Y finalmente la escena de la tabla de puntuaciones:





Diseños y estilos

Fuentes utilizadas

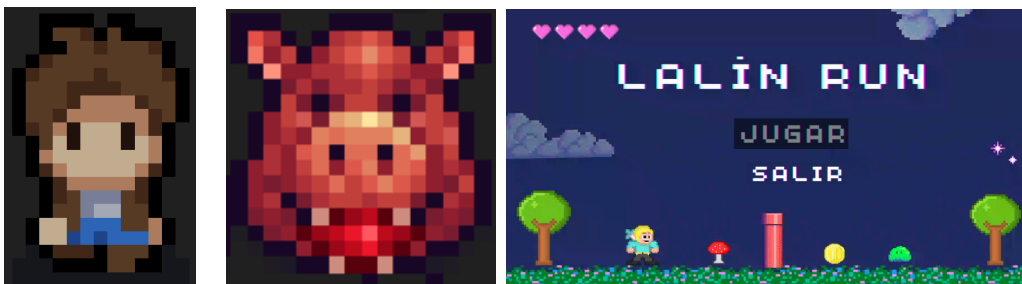
El juego usa nos presenta una fuente principal PressStart2P - Regular SDF, ya que presenta una estética propia de un juego de píxeles.



Sin embargo para otras escenas y por problemas de legibilidad se usa también en el proyecto el texto por defecto de Unity Text Mesh Pro, Arcade Game (Canva), o Liberations Sans SDF en el Scroll View.

Tipo de diseño

Lalín run presenta una estética centrada en los juegos antiguos de arcade por lo que su diseño de píxeles debe de recordar a ello. Para ello toda la ambientación se nos presenta con este formato de píxeles, tanto los personajes, objetos, pantallas de carga, como escenarios se encuentran con esta estética de píxeles.



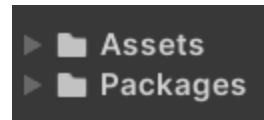
Estos son algunos ejemplos de la temática de píxeles que presenta este juego.



Organización del proyecto

Carpetas

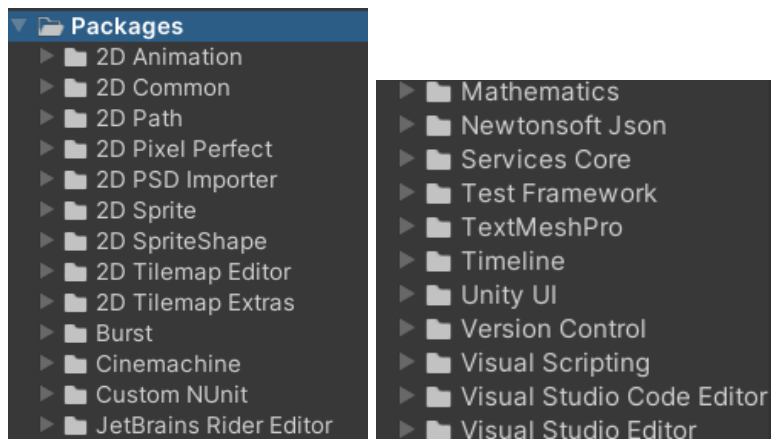
El proyecto cuenta con dos carpetas principales los assets, los cuales serán todas aquellas implementaciones realizadas por el programador y los packages que gestiona los paquetes de software que utiliza el proyecto.



PACKAGES

En ella podemos encontrar:

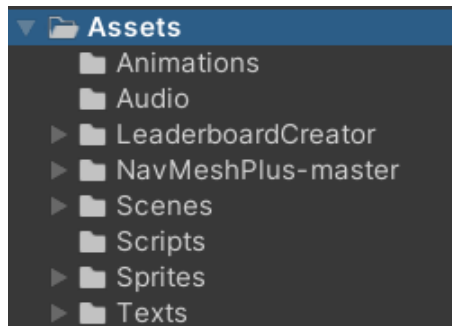
- Manifest.json: es un archivo de configuración que se encuentra en la raíz de unity
- Archivos de metadatos: ayudan a mantener la integridad del proyecto en unity y contiene información de los assets, scripts...Se pueden diferenciar por su .meta
- Carpetas de paquetes: con cada uno de los proyectos que han sido importados





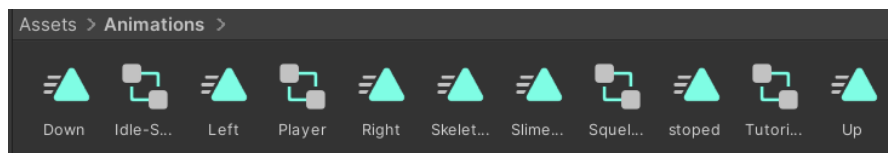
ASSETS

Nos presenta cada una de las carpetas creadas por el usuario, en el caso del Lalin Run poseemos las siguientes:



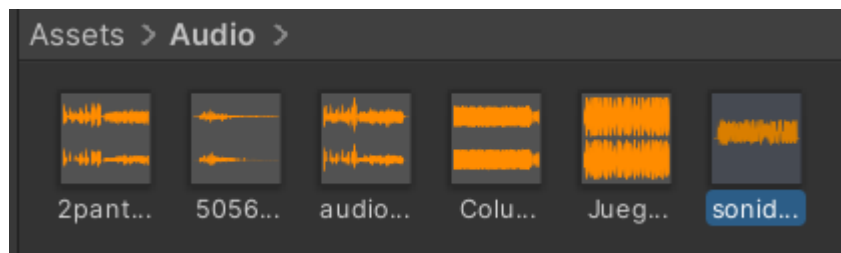
Animations

Presenta un conjunto de movimientos sea como la dirección del jugador corriendo o la animación del npc llamado Tutorial.



Audio

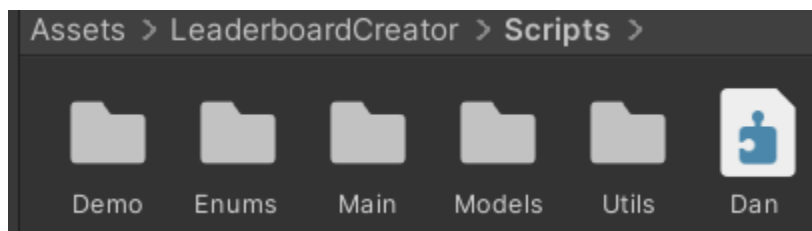
Podremos observar los distintos sonidos que presenta el juego en el menú, al ganar, en el juego...





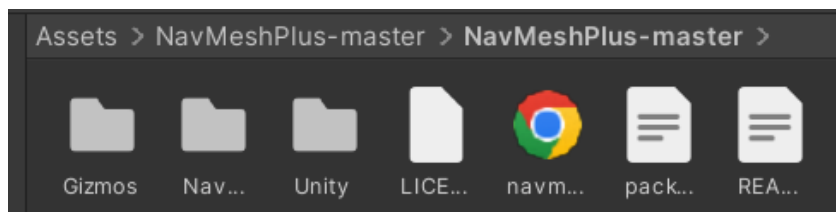
LeaderBoardCreator

Nos muestra un conjunto de archivos importados de Danial Jumagaliyev que servirán para el control y el buen funcionamiento al enlazar la tabla de marcadores del Lalín Run con la página web



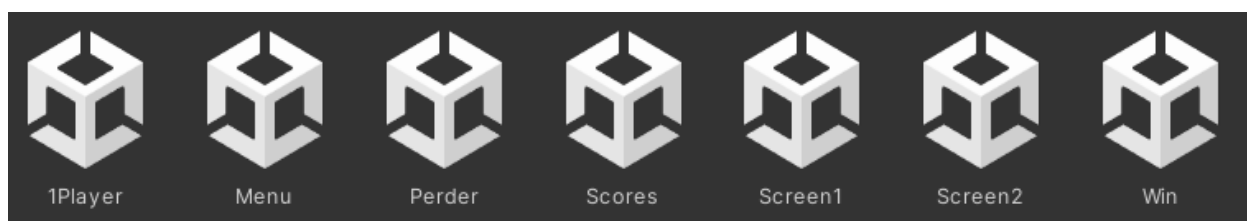
NavMeshPlus - master

Se emplea para ampliar las capacidades del motor de Unity. Por ejemplo en nuestro caso será necesario para el control de los enemigos haciendolos actuar con un comportamiento de IA llamado navMeshAgent.



Scenes

En Scenes encontraremos las distintas pantallas que presenta el juego, como ya hemos visto son no lineales y pasan de una a otra dependiendo de determinados acontecimientos. Las 7 escenas son:





Scripts

En él se encuentran todos los archivos C# que controlan funcionalidades del juego.

-CameraFollow: sigue al personaje con un pequeño delay, está asociado al juego mediante el MainCamera en la escena 1Player.

-Dialogue: acciona el dialogo con el cartel y el tutorial.

-Enemies controla mediante IA que los enemigos sigan al jugador por un área delimitada.

-Escena1, Escena2, Escena3, Menú controlan la transición entre pantallas.

-Escenascores actúa como transición entre pantallas además de controlar que se le introduzcan datos a la tabla de marcadores.

-Food como su nombre indica controla que al accionar la “X” el jugador recoja la comida. Además al aproximarse saldrá una exclamación.

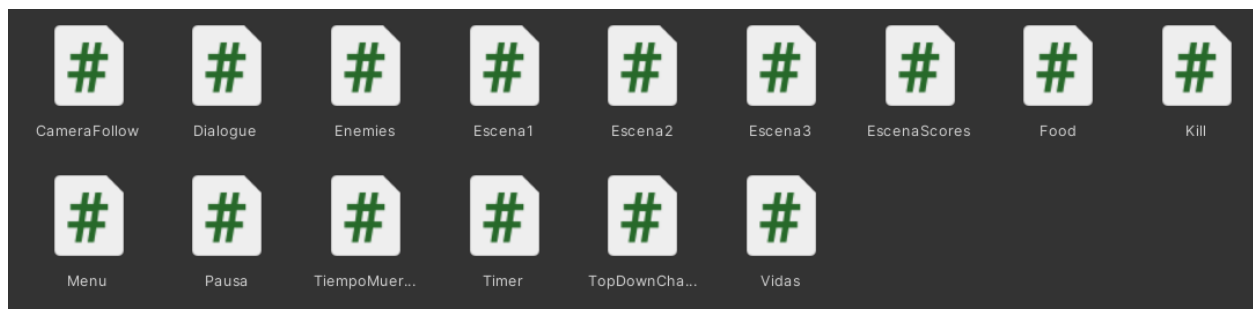
-Vidas y Kill controla que cada vez que nos acerquemos a un enemigo se nos quite una vida además de enseñarnos la pantalla de derrota si acabamos las vidas.

-TopDownCharacter controla el movimiento del usuario en los ejes X e Y.

-Pausa controla el menú de pausa así como pausar la escena o el cronómetro.

-TiempoMuerte recoge el tiempo en el que el jugador pierde y lo muestra en la pantalla de Perder.

-Timer controla todo el tiempo de la partida, actúa conjuntamente a tiempo muerte(pasando el tiempo), a pausa(pausando o reiniciando el contador) y al score(también al igual que a tiempo muerte pasandole el tiempo pero este en forma de puntuación).

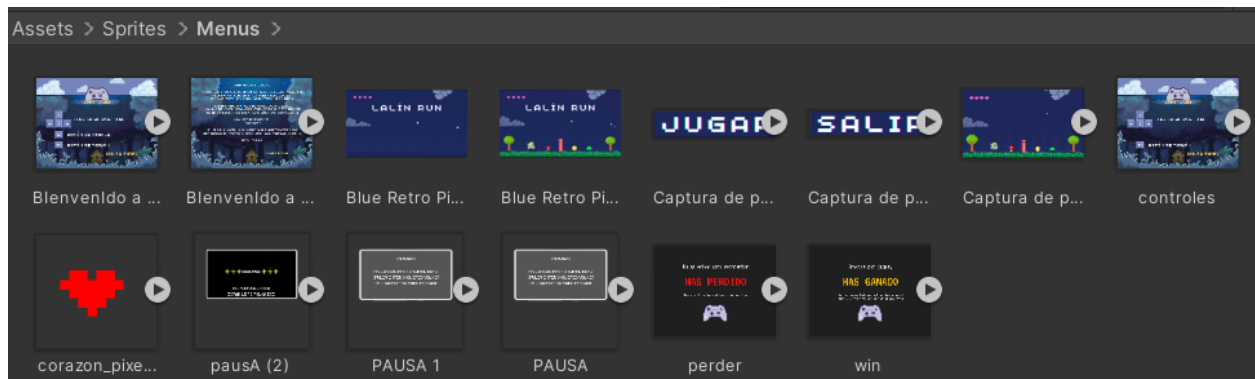




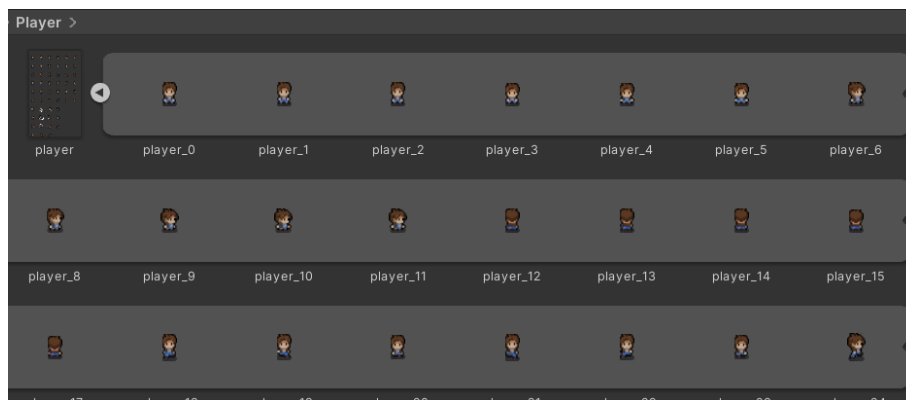
Sprites

Muestra todos los diseños que se ven en el juego, desde el suelo, los personajes, las pantallas de muerte...

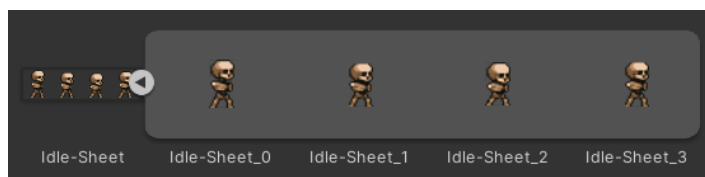
Menús:



Al jugador:

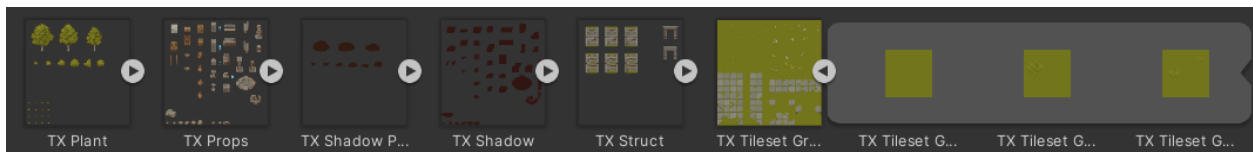


A los enemigos:

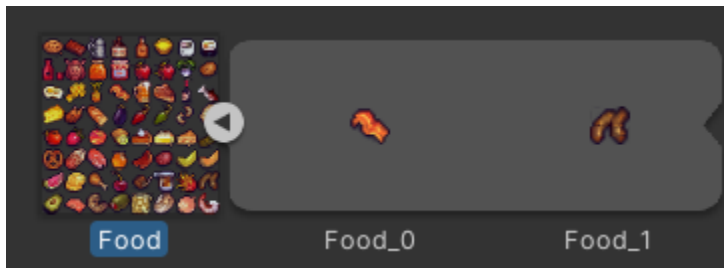




Texturas del entorno:



O la comida:



Text

Muestra documentación variada acerca del proyecto, fuentes, shaders, TMPs, Sprites predefinidos...

