Manual Urby G-Maker

[**Sobre la herramienta:**](#_2kkl28sukf3k) **1**

[**Requisitos:**](#_90hfo4ib2x2d) **1**

[**Introducción:**](#_bhvx9of7rugy) **2**

[**Instalación y usos básicos:**](#_j4sv3b5hmfvh) **2**

[**Repositorio:**](#_w2nxlau1be78) **2**

[Scripts de cámara:](#_rzq2822vzktg) **3**

[Scripts de movimiento de jugador:](#_3zxp4amwhb1d) **5**

[Player Scripts:](#_hfkfz5ixbo1i) **6**

[System Scripts:](#_xq5tail06nse) **7**

# Sobre la herramienta:

El Urby G-Maker es una herramienta/repositorio con una amplia variedad de scripts programados para el motor gráfico Unity con una amplia variedad a escoger por los usuarios que abarca lo más posible en cuanto a scripts necesarios para la creación de un videojuego, la herramienta está pensada para apoyar a diseñadores o desarrolladores independientes que no tengan un amplio conocimiento en el ámbito de la programación permitiéndoles desarrollar sus proyectos haciendo uso de un mínimo conocimiento de programación.

# Requisitos:

Versión del motor Unity 2019.3.1f1 o posterior.

Visual Studio 2019 o posterior o procesador de código compatible con Unity.

# Introducción:

# Instalación y usos básicos:

Una vez descargada la capeta scripts y documento Window.cs exportarlos al proyecto donde se pretende utilizar la herramienta, en el editor de Unity hacer clic en la pestaña llamada “Window” dar clic en la opción llamada Urby “G-Maker”, se creará una nueva ventana con unas herramientas de apoyo como lo son botones para automatizar movimientos comunes durante el desarrollo de un juego como lo son agregar un collider y un rigidbody, el botón “Add collider and rigidbody” te permite agregar con un clic estos dos elementos a cualquier objeto u objetos que se tenga seleccionado en la jerarquía de Unity igualmente el botón “Add animator” te permite agregar a todo los elementos seleccionados un animator component igualmente el botón “Add manager” te permite crear un game manager en la jerarquía.

# Repositorio:

Junto con el script que permite acceder a la ventana previamente mencionada se descargara dos carpetas llamadas “Scripts” y “Scriptable Objects”, estas carpetas contienen todos los scripts disponibles para uso totalmente libre y con libertad de poder ser modificados según lo crea conveniente el usuario.

La primera carpeta llamada “Scripts” contiene dentro subcarpetas que dividen los scripts en diversas categorías:

* Camera Scripts: En esta carpeta se pueden encontrar todos los scripts relacionados a usos de cámaras dentro del juego.
* Object Scripts: En esta carpeta se encuentran todos los scripts relacionados con objetos dentro de los juegos, esta carpeta contiene tres carpetas:
  + Environment objects: todos los scripts relacionados con objetos del ambiente dentro del juego.
  + Item objects: todos los scripts relacionados a ítems interactuables en el juego.
  + Weapon scripts: todos los scripts relacionados con armas en el juegos.
* Player movement scripts: En esta carpeta se encuentran todos los scripts relacionados con el movimiento de los personajes jugadores.
* Player scripts: En esta carpeta se encuentran todos los scripts relacionados a elementos extra para los personajes jugadores como inventarios, habilidades especiales, etc.
* System scripts: En esta carpeta se encuentran todos los scripts relacionados a los sistemas de pausa, guardado de partidas y configuraciones de menús dentro del juego.

Usos de scripts:

# Scripts de cámara:

Ubicación de los scripts: Assets > Scripts > Camera Scripts

FPSCamera: Para hacer uso de este script se requiere tener un personaje jugador en la jerarquía, una vez agregado esté tomar el objeto cámara en la jerarquía de unity y agregarla al objeto jugador para volverlo un hijo de este mismo, de esta forma la cámara se moverá junto con el jugador (Fig 1), posteriormente agregar el script llamado “FPSCamera” a la cámara que previamente colocamos en el jugador, este script contiene una variable llamada Mouse Sensitivity colocada por default en 100 esta variable permite modificar la velocidad de respuesta del movimiento de la cámara con respecto al mouse, también cuenta con un contenedor para un objeto de tipo transform ubicado debajo de la variable de Mouse Sensitivity en este transform deberá de agregarse el objeto dentro de la jerarquía que funcionara como el jugador al que se le agrega esta cámara, esto para que el movimiento de la cámara sea responsivo al jugador como se muestra en la fig.2.

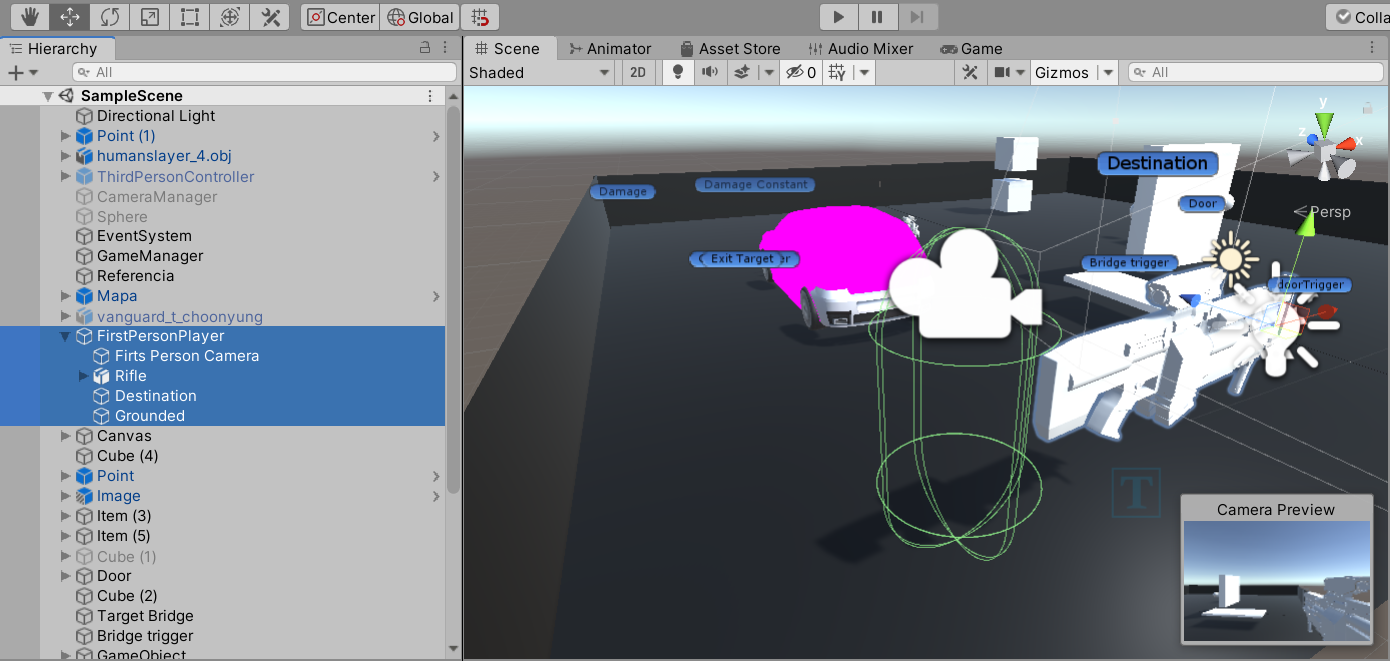


Fig.1

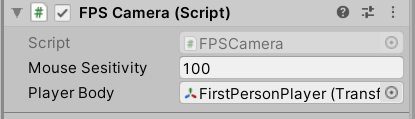


Fig.2

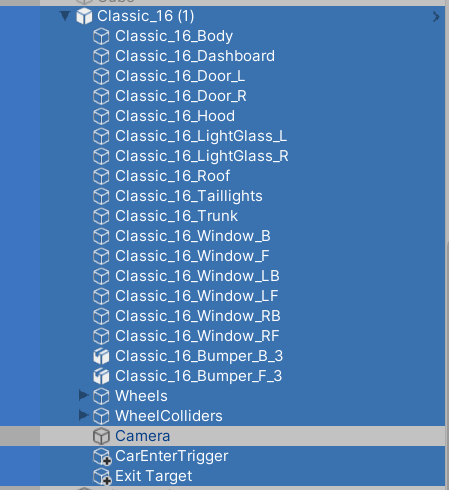
CarCameraController: Para el uso de este script se requiere de un objeto que se pretenda sea controlado por el jugador y la camara colocada como hijo de este objeto en la jerarquía de preferencia un vehiculo como se muestra en la fig.3, una vez agregado el script el objeto se deberan calibrar las variables “Offset” en las cuales son “X”, “Y” y “Z” estos parámetros te permitirán calibra la posición a la que la camara se colocara en el momento de ejecutarse el script y mientras el objeto se mueva, la variable “Follow Speed” te permitirá manipular la velocidad a la que se desea que la camara siga el objeto y “Look Speed” la velocidad a la que la camara se posicionara para observar el objeto al que se le implemento este script fig.4.

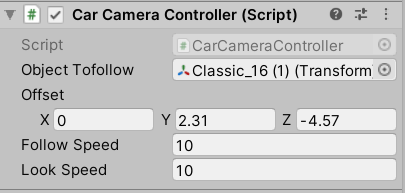
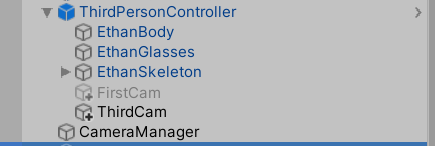
Fig.3 

Fig.4

CameraChange: Este script te permitirá cambiar entre dos diferentes camaras en el juego utilizando la tecla “C” esta pensado para intercambiar entre una camara en tercera persona en el juego y una camara en primera persona, para este script se deberá de crear en la jerarquía de unity un objeto vacío al que se le implementara el script fig.5, una vez echo esto solo se deberán de agregar a los contenedores llamados “Third Cam” y “First Cam” las dos camaras que se pretenden poder intercambiar, la variable “Cam Mode” determina que camara esta activa siendo “0” la camara “Third Cam” activa y “First Cam” desactivada y“1” “Third Cam” desactivada y “First Cam” activada en el juego fig.6.

Fig.5



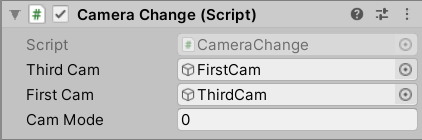


Fig.6

Camara: Este script te permitirá crear una camara en tercera persona que puede rotar al rededor del jugador y seguirlo o mirar al jugador y hacer rotar su posición con respecto al movimiento del mouse para esto el script debe de ser agregado a una camara que sea hija del objeto usado como jugador, el script te solicita hagas referencia al objeto jugador en un contenedor llamado “Player Transform”, la variable “Smooth Factor” te permitirá alterar el tiempo de respuesta o retraso del movimiento de la camara con respecto al movimiento del mouse, las variables llamadas “Look at Player” y “Rotate Arround Player” te permitirá decidir si quieres hacer que la camara pueda rotar en torno al jugador o si se desea que la camara mantenga su vista hacia el jugador y la variable “Rotation Speed” sirve para determinar la velocidad de rotación de la camara fig.7.

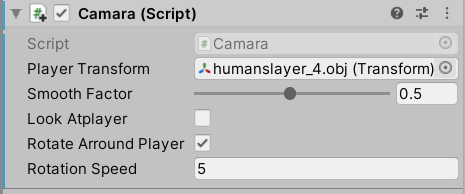


Fig.7

# 

# Scripts de movimiento de jugador:

Ubicación de los scripts: Assets > Scripts > Player Movement Scripts

Roller Movement: Este script te permitirá crear un movimiento de tipo roller pensado para objetos esféricos más especificamente objetos que se pretenda sea controlado por el jugador siendo manipulados por las flechas del teclado o las teclas “W”,”S”, ”A”,”D”, el script permite marcar una casilla llamada “Can jump” si la casilla esta desmarcada el jugador no podra hacer saltar el objeto utilizando la tecla espacio, si es marcada si permitirá saltar al objeto al que se le agregue este script, este objeto tiene una función que impide que el objeto con el script abuse de la tecla espacio para saltar indefinidamente, para esto los terrenos por los que se desplace el objeto deben de tener el tag “Ground” (sin las comillas), la variable “Jump force” te permite calibrar la fuerza con la que la esfera efectuara el salto, la variable “Speed” te permite calibrar la velocidad de aceleración del objeto fig.8.

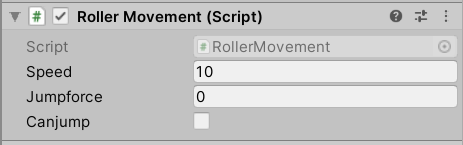


Fig.8

FPSMovement: Para el uso de este script se requiere que el objeto al que se le implemente cuente con un “Character Controller” agregado desde el inspector de Unity y referenciar en el contenedor del script llamado “Controller”, la variable “Speed” te permitirá manipular la velocidad con la que el objeto se mueva, la variable “Gravity” te permitirá manipular la gravedad del jugador esto para afectara a la velocidad de caída del objeto, la variable “Jump Force” te permite manipular la fuerza con la que el objeto efectúe un salto, la variable “Ground Check” te solicitara un objeto de referencia para detectar que el objeto esta tocando el suelo si se usa un modelo o un humano o humanoide se recomienda agregarle a este como hijo un “Empty Object” en este ejemplo llamado “Grounded” fig.9 y posicionarlo a la altura de la planta de los pies del modelo para que sea lo mas realista posible, la variable “Ground Distance” es para determinar la distancia maxima entre el “Empty Object” y el suelo (se aconseja dejarlo en 0.4 como esta predefinido), la variable “Ground Mask” debe esta definida en “Ground” esto con el objetivo de que no se pueda abusar de la tecla espacio para saltar indefinidamente, las teclas de control de este script una vez implementado son “W”, “S”, “A”, “D” y la barra de espacio para saltar fig.10.

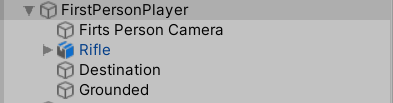


Fig.9

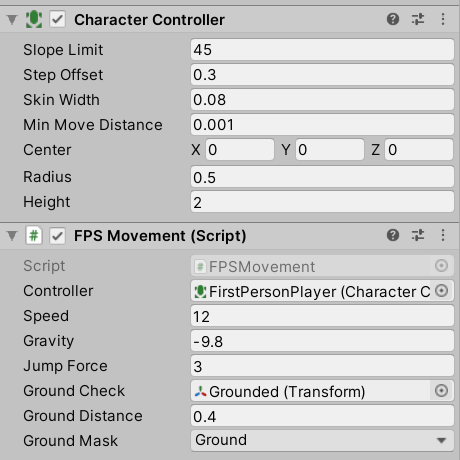


Fig.10

Auto Movement 2D: Este script te esta pensado para juegos en 2D en los que el jugador evada obstáculos, el objeto se movera de manera automática este movimiento al verse interrumpido por una colisión con cualquier objeto que no tenga el tag “Ground”, una vez ocurra esta colisión el objeto cambiara su dirección de movimiento automático al sentido opuesto, la variable “Speed” te permitirá manipular la velocidad con la que el objeto se mueva, la variable “Jump force” te permite calibrar la fuerza con la que el objeto pueda saltar, la casilla “Can Jump” es para decidir si el objeto puede efectuar un salto o no, adicionalmente el script solicita un objeto llamado “Spawn” este con la finalidad de tener un punto de referencia al que el objeto reaparece de ser dañado cuando el usuario oprima la tecla “T” fig.11, los controles de movimiento del objeto son, “D” para aumentar la velocidad de movimiento y la tecla espacio para poder saltar, los terrenos por los que se desplace el objeto deben de tener el tag “Ground” (sin las comillas) con el objetivo de que no se pueda abusar de la tecla espacio para saltar indefinidamente fig.12.

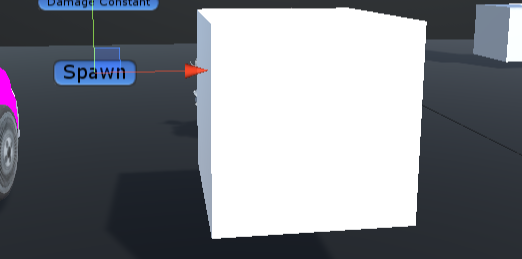


Fig.11

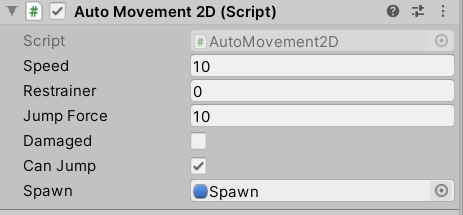


Fig.12

# Player Scripts:

Ubicación de los scripts: Assets > Scripts > Player Scripts

Interaction: Este script agregado el objeto jugador te permite activar y desactivar un mensaje en pantalla cada vez que el jugador entre en un collider con un tag diferente a “Item”, “platform” y “Fall Caster”, el script solicita un objeto de tipo “Text”, el cual debe de estar en un canvas dentro de la jerarquía de Unity, este objeto text debe de estar desactivado de forma predeterminada, con la intención de evitar errores.

Player Healt System: Este script te solicitara una Healt Bar agregada en un canvas, esta barra a su vez deberá de contar con un script llamado “Healt Bar” ubicada en la carpeta “Scripts” > “System Scripts” este script a su vez requiere de un slider que se puede agregar a la Healt Bar desde el inspector y un objeto “fill” (la configuración de este objeto queda a disposición del usuario para decidir como se llenara la barra de vida) el script “Player Healt System” te solicita un “Max Healt” este mismo te permite manipular la maxima salud del jugador, la variable “Current Healt” se actualiza automáticamente con la variable “Max Healt”.

Player PlatformInteraction: Este script debe de ser implementado en el objeto que se pretenda ser el jugador, la intención de este script es que el jugador al subir a alguna plataforma que se mueva dentro del juego el jugador no resbale de ella si no que se mueva junto a esta, para esto solo se requiere que la plataforma tenga el tag “platform”, para que el script funcione sin ningún problema se aconseja que el collider de la plataforma sea un poco más grande que la plataforma al menos por el lado al que el jugador se colocara fig.13.

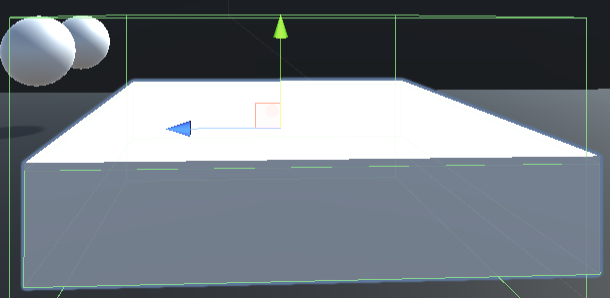


Fig.13

Use Object: Este script permite al jugador recojer cualquier objeto al que se le apunte por medio de un raycast para que este objeto sea recogido este debe de tener el tag “pickable” para recojer el objeto se debe de usar la tecla “E” y para soltarlo la tecla “R”, para recojer un objeto el jugador debe tener como hijo un “Empty Object” que funcione como objetivo al que el objeto recogido por el jugador se movera (se aconseja sea colocado como una mano) para facilitar el uso de este script se aconseja usar colocar un crosshair al centro de la pantalla para saber a donde apunta el raycast.

# System Scripts:

Ubicación de los scripts: Assets > Scripts > System Scripts

Fall Caster: Este script es para determinar puntos en los que pueda caer el jugador fuera del mapa como en un juego de plataformas, para usarlo el script debe de estar colocado en un “Empty Object” con un collider en el que al momento de entrar el jugador este hara respawn a un punto determinado por el diseñador del juego el script solicita un “Target Spawn” este sera el punto al que el jugador hara respawn al caer en este collider y un player que sera el objeto jugador que pueda caer en el collider, para que este funcione el jugador deberá de tener el tag “Player” fig.14.

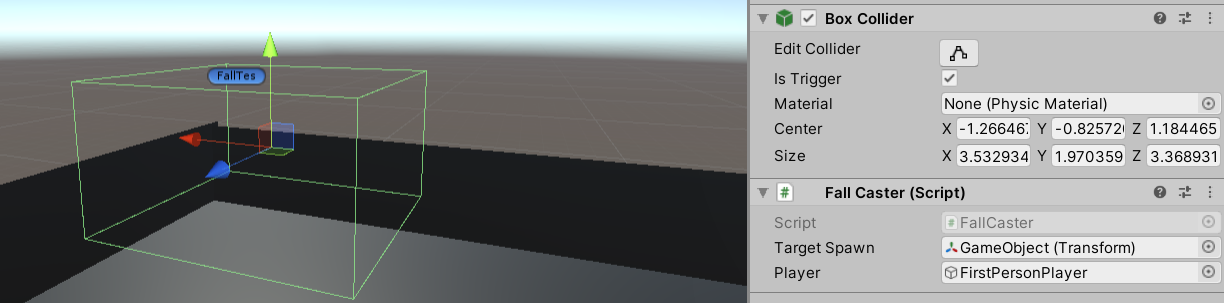


Fig.14

# 