

Implementación de un tutorial interactivo de programación con Unreal Engine en forma de videojuego puzle en primera persona.

Rafael Moreno Serrano

14 de septiembre de 2021

Índice general

I	Introducción	2
1.	Introducción	3
1.1.	Ficheros	3
2.	Código C++	4
2.1.	Clase <i>AbrirPuerta</i>	4

\defcounter {refsection}{0}\relax

\contentsline {subsection}{\numberline {2.1.2}Funci\'on \emph {TickCom

Parte I

Introducción

Capítulo 1

Introducción

En este documento se mostrará todo el código diseñado y realizado para el proyecto.

El código principal está realizado en el lenguaje C++, y tiene las características y funciones propias del motor gráfico Unreal Engine, ya que en todo momento se codifica en base a éste. Así mismo, se mostrará

los Blueprints creados, al ser un lenguaje de script como tal, y se tratará de la misma manera.

Así, este documento se divide en las funciones creadas con C++ y con Blueprints, siendo a su vez separadas por clases y/o documentos.

1.1. Ficheros

Capítulo 2

Código C++

A continuación mostraré cada una de las clases codificadas en lenguaje C++:

2.1. Clase *AbrirPuerta*

Esta clase es creada para poder abrir una clase por medio de un *trigger* de superposición, y que ésta se cierre al cabo de un tiempo.

2.1.1. Función *BeginPlay*

```
void UAbrirPuerta::BeginPlay()Owner = GetOwner();
```

Esta función se activa al comienzo, y guarda en variables el actor que tendrá la capacidad de abrir la puerta (en este caso el jugador), y la rotación inicial de la puerta.

```
// Called when the game starts
void UAbrirPuerta::BeginPlay()
{
    Super::BeginPlay();
    Owner = GetOwner();

    ActorThatOpens = GetWorld()->GetFirstPlayerController()->GetPawn();
    actorRotation = Owner->GetActorRotation();
}
```

2.1.2. Función *TickComponent*

Esta función comprueba en cada tick si el actor está activando el trigger. En caso positivo la puerta se abre y se guarda el momento en el que se ha abierto. Una vez pasado un tiempo, la puerta se cerrará.

```
void UAbrirPuerta::TickComponent(float DeltaTime, ELevelTick TickType, FActorInfo TickActorInfo)
{
    Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
    if (PressurePlate->IsOverlappingActor(ActorThatOpens))
    {
        OpenDoor();
        LastDoorOpenTime = GetWorld()->GetTimeSeconds();
    }

    if (GetWorld()->GetTimeSeconds() - LastDoorOpenTime > DoorCloseDelay)
    {
        CloseDoor();
    }
}
```