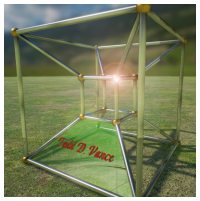

Unreal Code for Building Escape

Notes

from the Udemy course

Table of Contents

1 GitIgnore base for Unreal Engine 4

```
**/*.VC.db
**/*.opensdf
**/*.opendb
**/*.sdf
**/*.sln
**/*.suo
**/*.xcodeproj
**/*.xcworkspace
**/Build
**/Binaries
**/DerivedDataCache
**/Intermediate
**/Saved
**/StarterContent/
```

The above ".gitignore" file contents are aggressive, possibly too aggressive for some people's project styles, but it also works even if the unreal project is put in a subdirectory of the top-level repository directory, as it is done in the Udemy course.

2 Common Editor Keys

- While playing, F8 to eject from pawn
- In editor, F to focus on selected object
- Alt-LMB to rotate around focused object
- Alt-RMB to zoom focused object
- Alt + WASDQE to move around like in an FPS
- W, E, R to change tool to translate/rotate/scale
- Hold L and doubleclick to quickly place a light

3 Logging

UE_LOG(LogTemp, Warning, TEXT("your text here, formats allowed too %d %s"), IntegerValue, *FStringValue);

The contents of the TEXT macro must be a string literal.

4 Useful Component Methods

- GetWorld()→… for various properties of the world.
  - GetWorld()→GetTimeSeconds() returns game time
  - GetWorld()→GetFirstPlayerController() returns the first player controller it finds.
  - GetWorld()->GetFirstPlayerController()->GetPlayerViewPoint(OUT location, OUT rotation); OUT is defined to be nothing. Location and rotation variables will now hold the player's viewpoint.
    - To get the default pawn in code, get it from the player controller.
  - Also, LineTraceSingleByObjectType (see Ray Casting section)
- DrawDebugLine(GetWorld(), StartVector, EndVector, FColor(255,0,0), false, -1, 0, ThicknessFloat); draws a debug line
- GetOwner() returns the actor that owns this component
  - GetOwner()→FindComponentByClass<UPhysicsHandleComponent>()
  - GetOwner()→GetName() returns an Fstring.

5 Ray Casting

```
//Set-up Query Parameters
FCollisionQueryParams CollisionQueryParams(FName(TEXT("")), false, GetOwner());

// ray-cast to reach distance
FHitResult HitResult;
GetWorld()->LineTraceSingleByObjectType(OUT HitResult, location, LineTraceEnd,
FCollisionObjectQueryParams(ECollisionChannel::ECC_PhysicsBody),
CollisionQueryParams);
```

6 “Decorator” macros

- UPROPERTY for marking declarations of fields of a class (in the .h file, may be private fields) for exposure in the Details panel of the component
  - UPROPERTY(VisibleAnywhere)
  - UPROPERTY(EditAnywhere)
  - later in lecture 89: UPROPERTY(BlueprintAssignable)
  - Note, not all types are equally supported. An advanced topic is making a specific type supported.
  - See https://docs.unrealengine.com/latest/INT/Programming/UnrealArchitecture/Reference/Properties/Specifiers/BlueprintAssignable

7 Create a new C++ component for an actor

1. Content Browser→Add New button→C++ File→Actor Component
2. Select an actor instanced in the scene and click Add Component in the details window, and your component you made will be in the Custom section.
3. To add a component to the default pawn, go to its blueprint class (you may have to make it first and set it as the new default pawn) and add the component to the blueprint details panel.

8 Make blueprints from code, etc., objects

- To create a Default Pawn blueprint, play game, eject, select player pawn in game, and go to Blueprints menu→Convert Selected Actor to Blueprint Class.
- To make a blueprint for most instantiated actors, select it and go to Blueprints menu→Convert Selected Actor to Blueprint Class.
  - Component properties that link to objects will be reset to null and have to be re-wired. If you don’t have guard code, running in this state may cause the game to crash from dereferencing a null pointer.

- To create a new Game Mode blueprint, needed to set a new default pawn, say to the blueprint default pawn:
    - Blueprints→Create Empty Blueprint Class → open all class and find <name of game>GameModeBase class and select it. That last name has changed from one UE4 version to the next, so it might be a little different. Then you can change the default game mode and change its default pawn in Project Settings→Maps and Modes.

9 Physics

- Check Simulate Physics on actors to make them moveable and set up a default physics system. Then set a reasonable mass if needed. You may need to add a collision to the mesh to make physics work right.
- Add PhysicsHandle component to the default pawn.
- The collision component of a physics object should simulate physics and generate hit and overlap events, in most cases.

10 PhysicsHandle component

- PhysicsHandle->GrabComponentAtLocation(ComponentToGrab, NAME_None, GrabbedObject→GetActorLocation());
    - This changed from earlier versions, so is slightly different from the lectures.
    - Ensure ComponentToGrab is not null first.
    - Aactor* GrabbedObject = ComponentToGrab→GetOwner()
    - `UprimitiveComponent *ComponentToGrab = CurrentHitResult. GetComponent();`
- `PhysicsHandle->ReleaseComponent();`

11 Player Input

- `InputComponent = GetOwner()->FindComponentByClass<UInputComponent>();`
    - make sure it’s not null before using!
- InputComponent->BindAction("Grab", EInputEvent::IE_Pressed, this, &Ugrabber::Grab);
- InputComponent->BindAction("Grab", EInputEvent::IE_Released, this, &Ugrabber::Release);

- The Grab action is set in ProjectSettings→Engine:Input

12 Iterate over every actor in the world

```
for (TActorIterator<AActor> ActorItr(GetWorld()); ActorItr; ++ActorItr)
{
    AActor* Actor = *ActorItr;
    //do stuff
}
```

13 Open a component in VS2015

Select the object, select the appropriate component in the Details panel, then right-click that component and select "open UsomeComponent.h".

14 Animating a Door Opening

Lecture 89, in section 3. Door closing is analogous (note you can play the timeline in reverse).

- First, door has to be a blueprint. (do this for all openable doors—see section above on making blueprints from objects)
- Add Timeline action in blueprint Event Graph to add animation.
- Create an event called OnOpenRequest to trigger the timeline
    - Make a blueprint assignable field in the UOpenDoor class called OnOpenRequest of type FOnOpenRequest (type is declared in the next item). It should be a public, not private, property.
    - `//Above the class, need to DECLARE_DYNAMIC_MULTICAST_DELEGATE(FOnOpenRequest); // so that the FopenDoorRequest "delegate" type exists.`
    - In OpenDoor.cpp, in the OpenDoor method of class UopenDoor, need to "broadcast" the OnOpenDoor event with OnOpenRequest.Broadcst(); Comment out the old code in the OpenDoor method, as this blueprint activity is going to replace it completely.
    - In the door blueprint, event graph, with the OpenDoor component clicked (to get access to what we declared in OpenDoor), right-click and select Add "OnOpenRequest" to add the event.

    - Test by wiring the new event node into a SetActorRotation node, adjust the rotation, compile, and test in game.
- Edit the timeline curve to set the door rotation.
    - Double-click the timeline to open it and add a float track and name it OpenAngle.
    - Right click and add key(s) to define the curve from 0 degrees to 90 degrees (or whatever) over the time desired. (Use F to “focus”, change scale to see whole curve).
    - Right click on a key and select Auto interpolation to make the curve smooth.
    - Make sure the length field of the animation is correct (or it will “hang” at the end till the time runs out).
- Wire it up in the event graph.
    - Drag the OpenAngle output of the timeline, select Make Rotator, and for opening a door, change the connection to the Yaw instead of the Roll.
    - Wire the Return Value of the Make Rotator to the New Rotation of Set Actor Rotation.
    - Wire the triangle output of the OnOpenRequest event to the Play triangle input of the timeline, and the Update triangle output of the timeline to the (Exec) triangle input of the Set Actor Rotation.
    - Compile blueprint, save all, and test
- Later, the UopenDoor class is refactored (Lecture 91).

15 Miscellaneous

- Rotation.Vector() returns the forward unit vector of a rotator.