Unreal Engine and UnrealTournament

The_Cowboy

June 22, 2019

1 Introduction

These are the notes concerning the Unreal Engine and UnrealTournament that I took while building UnrealTournament 4 on Linux. I will be documenting the "unspoken" code and my modifications to bring 3 giants (the third is of course Linux!) together in harmony and "Greater Good" of the community. I will assume that

- you have a decent knowledge about coding (no PHP, Python and JavaScript don't count) in Java or C++.
- you are quite comfortable with Linux terminal and bash scripting.
- anything else I missed.

2 UnrealBuildTool

The UnrealBuildTool or UBT is an .NET application which has two-fold functionality

- generate the project files for IDEs which include Qt Creator and KDevelop (both are multiplatform supported) and take care of all the relevant dependencies (including the "circular dependencies") in the game project.
- compile the C++ source files into binary files (as per the platform needs). Add more information about the platform dependent compilation

The code for the UBT is present in the `Engine/Source/Programs/UnrealBuildTool` directory. The cross platform IDE MonoDevelop can be used to load the UBT project and modify the tool "as per the need". Then you can either build the project from the IDE or go about the Epic's way (I prefer the latter).

In order to do that make sure that xbuild and Mono are appropriately setup on the box. Then the command

```
1  xbuild Engine/Source/Programs/UnrealBuildTool/UnrealBuildTool.
     csproj /property:Configuration="Development" /verbosity:quiet /
     nologo
```

will yield the executale file (.exe) in the `Engine/Binaries/DotNET` directory. There you have your "manipulated" UBT!

Don't get upset by the .exe extension (Linux users tend to get agitated by the name), the Mono[1] framework will generate the appropriate environment for UBT to work. The real-time building can be performed by the command

```
mono Engine/Binaries/DotNET/UnrealBuildTool.exe UE4Game Linux
    Debug
```

2.1 Generating Project Files

I, for one, find it not only helpful but also natural to work with decent IDE for writing the code. It not only facilitates the "smart" autocomplete features, but also provides a scope for "searching" and "documenting" the beloved code. Therefore UBT's toolmode GenerateProjectFiles comes in handy. The command

```
mono Engine/Binaries/DotNET/UnrealBuildTool.exe -Project=``
    PATH_TO_PROJECT_DIRECTORY/PROJECTNAME.uproject'' -makefile
```

yields the makefile in the Root directory of Unreal Engine. The switch -projectfiles generates all sorts of projectfiles that UBT supports. For Qt Creator the switch is -qmakefile

2.2 Building with UBT

The makefile generated by UBT doesnot seem to compile the project (find out why?). For project building the command is

```
mono UnrealBuildTool.exe -project="/home/the_cowboy/unrealworks/
    Unreal Projects/UnrealTournament/UnrealTournament.uproject"
    Development Linux -TargetType=Editor -Progress -
    NoHotReloadFromIDE
```

3 Coding with Qt Creator

Qt Creator is a decent IDE for coding Unreal Engine projects (or any other C++ project) in Linux. The configuration recommended by Epic is mentioned in their docs.

[1] The words "Cross Platform" and "Open Source" should restore the peace!

In this document I will assume the Unreal Engine Root directory as the main directory of the project[2]. Thus it will facilitate the "shipping" and other formal procedures associated with the professional building "schemes". The commands of Section 2 are in agreement with this structure.

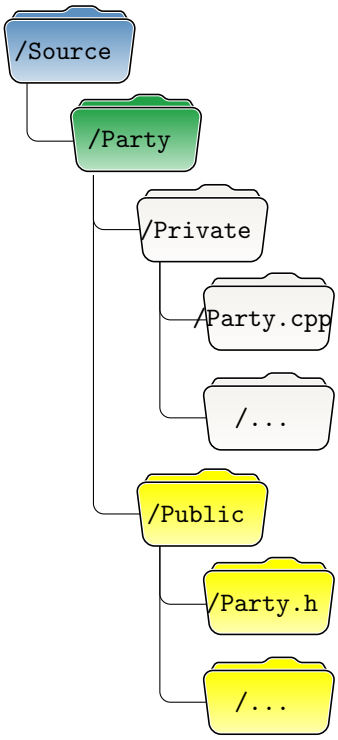
Also make sure to follow the "no space" convention while naming the folders else UBT will tend to mess up the INCLUDEPATH paths in . . . includes.pri.

4 Steps for Porting UT4 to Latest UE

In this section I will log the changes introduced in the classes and header files of UnrealTournament 4 to generate the compatibility with the latest Unreal Engine reseases.

4.1 Party and all that

Party is component of the plugin *OnlineFramework* present in `Engine/Plugins/Online/OnlineFramework/Source/` directory. The directory structure as per the old Engine is shown



In Unreal Engine 4.22 (and hopefully above) the structure of `Party` becomes more complex with highly customized "modularity".

[2]One Unreal Engine project at a time!

- Added the definition for Enum EMemberChangedReason in the file Party.h

```
enum class EMemberChangedReason
{
    Disconnected,
    Rejoined,
    Promoted
};

```

- The compilation error is

```
Error: Couldn't find parent type for 'UTParty' named 'UParty'
    in current module or any other module parsed so far.
```

In order to fix the compatibility issues, I am performing some tests

– Test1: Copied Party.h(.cpp) and PartyGameState.h(.cpp)[3] to `Public(Private)/Online` directories. The hope is that the parent class UParty (of type PARTY_API) defined in Party.h might resolve the definition issues.
Success Seems to work. But not quite. There is a series of confilicts of type

```
/home/the_cowboy/unrealworks/UnrealProjects/
    UnrealTournament/Source/UnrealTournament/Public/Online
    /PartyGameState.h(46) : LogEnum: Error: Enum name
    collision: 'EPartyType::Public' is in both '/Script/
    UnrealTournament.EPartyType' and '/Script/Party.
    EPartyType'
```

– Test2: Removed the following code snippet from PartyGameState.h

```
UENUM(BlueprintType)
enum class EPartyType : uint8
{
  /** This party is public (not really supported right now
    ) */
  Public,
  /** This party is joinable by friends */
  FriendsOnly,
  /** This party requires an invite from an existing party
     member */
  Private
      };

```

[3] Of course from the old Engine.

as already defined in PartyTypes.h (`Engine/Plugins/Online`). Success Seems to work! Note it is not a BlueprintType any more. Will see how it affects the online gameplay.
Consequently, this must be followed by the inclusion of

```
#include "PartyTypes.h"

```

in the file PartyGameState.h and removing the definitions of EApprovalAction and ToString.

- The compilation error is

```
Error: Couldn't find parent type for 'UTPartyMemberState'
    named 'UPartyMemberState' in current module or any other
    module parsed so far.
```

In order to fix the compatibility issues, I am performing some tests

  - Test1: Copied PartyMemberState.h(.cpp) to `Public(Private)/Online` directories. The hope is that the parent class (of type UNREALTOURNAMENT_API) defined in PartyMemberState.h might resolve the definition issues.
    Success Seems to work.

- The compilation error is

```
Error: Couldn't find parent type for 'UTBlurWidget' named '
    UBackgroundBlurWidget' in current module or any other
    module parsed so far.
```

In order to fix the compatibility issues, I am performing some tests

  - Test1: Copied BackgroundBlurWidget.h(.cpp) to `Public(Private)/UMG` directories. The hope is that the parent class UBackgroundBlurWidget defined in BackgroundBlurWidget.h might resolve the definition issues.

- The compilation warning is

```
/home/the_cowboy/unrealworks/UnrealProjects/UnrealTournament/
    Source/UnrealTournament/Public/UTATypes.h(339) :
    LogCompile: Error: Cannot expose property to blueprints in
     a struct that is not a BlueprintType. TextureUVs.U
```

The fix seems to define the structure FTextureUVs as a BlueprintType with the code

```
USTRUCT(BlueprintType)
```

and define FConfigMapInfo, FPackageRedirectReference, FScoreboardContextMenuItem, FUTGameRuleset, FCustomKeyBinding as BlueprintType. Success Seems to work.

- Compilation warning of type

```
some error related to IsHovered override in UTBaseButton.h
```

Just wrote override keyword in both UTBaseButton.h(.cpp) files.

- More BlueprintType conversion in file McpStubs.h of the structure FMcpItemIdAndQuantity.
- In file UTTeamInfo.h changed the UPROPERTY of EnemyList from BlueprintType to EditAnywhere.

There is a dilemma on how to override a UFUNCTION. The error of type

```
Override of UFUNCTION in parent class (someclass) cannot have a
    UFUNCTION() declaration above it; it will use the same
    parameters as the original declaration.
```

spit by the UHT makes one to consider writing UFUNCTION keyword in the .cpp file (and removing the keyword from .h to rectify the error). Also Epic's advise in the file HeaderParser.cpp is

```
Native function overrides should be done in CPP text, not in a
    UFUNCTION() declaration (you can't change flags, and it'd
    otherwise be a burden to keep them identical)
```

but then it no longer remains the UFUNCTION (trust me, I performed the necessary tests by modifying the UHT).

Some un understood concepts

- Consider the example of delegates in Party.cpp

```
    PartyJoinRequestReceivedDelegateHandle = PartyInt->
    AddOnPartyJoinRequestReceivedDelegate_Handle(
    FOnPartyJoinRequestReceivedDelegate::CreateUObject(this, &
    ThisClass::PartyJoinRequestReceivedInternal));
```

The UBT is not recognizing the CreateUObject() method for the delegate FOnPartyJoinRequestReceivedDelegate inspite of the proper definition in OnlinePartyInterface.h. Simillar problems are there for FOnPartyMemberChangedDelegate, PostLoadMapWithWorld

4.2 Blueprints for UT

- In the file BlueprintContextManager.cpp present in `BlueprintContext/Private`, line 103 seems to have improper function call and it results in the following error

```
error: no matching member function for call to '
    AddReferencedObjects' Collector.AddReferencedObjects<
    TSubclassOf<UBlueprintContextBase>, UBlueprintContextBase
    *>( Iter.Value() );
```

As of now I have no clue on how to make the appropriate call, so I am commenting it out.

4.3 UTCharacterContent

It seems a lot has changed since last UT code upgrade expecially for the class AUTCharacterContent so much that it renders the definition of base class AActor incomplete.

- First comes the constructor for AUTCHaracterContent which (for old version was defined by)

```
    AUTCharacterContent(const FObjectInitializer& OI)
    : Super(OI)
{
    RootComponent = OI.CreateDefaultSubobject<USceneComponent
    >(this, FName(TEXT("DummyRoot"))); // needed so Mesh has
    RelativeLocation/RelativeRotation in the editor
    Mesh = OI.CreateDefaultSubobject<USkeletalMeshComponent>(
    this, FName(TEXT("Mesh")));
    Mesh->SetupAttachment(RootComponent);
    Mesh->AlwaysLoadOnClient = true;
    Mesh->AlwaysLoadOnServer = true;
    Mesh->bCastDynamicShadow = true;
    Mesh->bAffectDynamicIndirectLighting = true;
    Mesh->PrimaryComponentTick.TickGroup = TG_PrePhysics;
    Mesh->SetCollisionProfileName(FName(TEXT("CharacterMesh"))
    );
    Mesh->bGenerateOverlapEvents = false;
    Mesh->SetCanEverAffectNavigation(false);
    Mesh->MeshComponentUpdateFlag = EMeshComponentUpdateFlag::
    OnlyTickPoseWhenRendered;
    Mesh->SetCollisionEnabled(ECollisionEnabled::NoCollision);
    Mesh->bEnablePhysicsOnDedicatedServer = true; // needed
    for feign death; death ragdoll shouldn't be invoked on
    server
```

```
    Mesh->bReceivesDecals = false;
    DMSkinType = EDMSkin_Red;

    RelativeScale1p = FVector(1.0f, 1.0f, 1.0f);
    RelativeRotation1p = FRotator(0.0f, -90.0f, 0.0f);

    DisplayName = NSLOCTEXT("UT", "UntitledCharacter", "
    Untitled Character");

    static ConstructorHelpers::FObjectFinder<UClass> GibRef[]
    = { TEXT("/Game/RestrictedAssets/Blueprints/GibHumanHead.
    GibHumanHead_C"),
      TEXT("/Game/RestrictedAssets/Blueprints/GibHumanLegL.
    GibHumanLegL_C"), TEXT("/Game/RestrictedAssets/Blueprints/
    GibHumanLegR.GibHumanLegR_C"),
      TEXT("/Game/RestrictedAssets/Blueprints/GibHumanRibs.
    GibHumanRibs_C"), TEXT("/Game/RestrictedAssets/Blueprints/
    GibHumanTorso.GibHumanTorso_C"),
      TEXT("/Game/RestrictedAssets/Blueprints/GibHumanArmL.
    GibHumanArmL_C"), TEXT("/Game/RestrictedAssets/Blueprints/
    GibHumanArmR.GibHumanArmR_C") };

    new(Gibs) FGibSlotInfo{ FName(TEXT("head")), GibRef[0].
    Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("thigh_l")), GibRef[1].
    Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("thigh_r")), GibRef[2].
    Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("Spine_01")), GibRef
    [3].Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("Spine_02")), GibRef
    [4].Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("lowerarm_l")), GibRef
    [5].Object };
    new(Gibs) FGibSlotInfo{ FName(TEXT("lowerarm_r")), GibRef
    [6].Object };
}

```

Added the proper incude line

```
#include "GameFramework/Actor.h"

```

and the following modifications

```
    Mesh->SetGenerateOverlapEvents(false);
    Mesh->VisibilityBasedAnimTickOption =
EVisibilityBasedAnimTickOption::OnlyTickPoseWhenRendered;

```

in the constructor.

4.4 UTRecastNavMesh.h

- The error

```
        error: no member named 'AddDirtyArea' in '
UNavigationSystemBase'
                        GetWorld()->GetNavigationSystem()->
AddDirtyArea(FBox(FVector(-WORLD_MAX), FVector(WORLD_MAX))
, ENavigationDirtyFlag::All);
```

is rectified by the following code in

```
                        if (bFirstTime && bNeedsRebuild)
{
        //GetWorld()->GetNavigationSystem()->AddDirtyArea(
FBox(FVector(-WORLD_MAX), FVector(WORLD_MAX)),
ENavigationDirtyFlag::All);
        if((UNavigationSystemV1*) GetWorld()->
GetNavigationSystem())
        {
            ((UNavigationSystemV1*) GetWorld()->
GetNavigationSystem())->AddDirtyArea(FBox(FVector(-
WORLD_MAX), FVector(WORLD_MAX)), ENavigationDirtyFlag::All
);
        }
    }
```

- Another depricated function

```
    //return Cast<AUTRecastNavMesh>(World->
GetNavigationSystem()->GetMainNavData(FNavigationSystem::
ECreateIfEmpty::DontCreate));
    return Cast<AUTRecastNavMesh>(UNavigationSystemV1::
GetCurrent(World)->GetDefaultNavDataInstance());
```

4.5 UTCharacter.h

- The override of the function ClearJumpInput() seems wrong (probably due to new version of Unreal Engine). I am declaring the method with DeltaTime and the function call is done *only* in UTCharacter.cpp with DeltaTime=0.
- The property MovementMode in CharacterMovementComponent.h is deprecated. Instead there are now two properties StartPackedMovementMode and EndPackedMovementMode containing the same information at the begining and end of the move. Thus the redefinition of the function

```
        virtual void UTServerMove(float TimeStamp,
    FVector_NetQuantize InAccel, FVector_NetQuantize ClientLoc
    , uint8 CompressedMoveFlags, float ViewYaw, float
    ViewPitch, UPrimitiveComponent* ClientMovementBase, FName
    ClientBaseBoneName, uint8 ClientMovementMode);
2
```

is in order. We define the new paramaters

```
1       virtual void UTServerMove(float TimeStamp,
    FVector_NetQuantize InAccel, FVector_NetQuantize ClientLoc
    , uint8 CompressedMoveFlags, float ViewYaw, float
    ViewPitch, UPrimitiveComponent* ClientMovementBase, FName
    ClientBaseBoneName, uint8 StartPackedMovementMode, uint8
    EndPackedMovementMode);
```

- The affected classes are in file UTCharMovementReplication.cpp
  - In the function PostUpdate, the deprecated property MovementMode is assigned

```
1 MovementMode = UTCharMovement->PackNetworkMovementMode();
```

    which needs to be modified as well. TODO for later!!!

4.6 PartyContext.cpp

There is a typecast error in PartyContest.cpp with the error

```
1     /home/the_cowboy/unrealworks/UnrealProjects/UnrealTournament/
    Source/BlueprintContext/Private/PartyContext.cpp:147:35: error:
     no viable conversion from 'FUniqueNetIdRepl' to 'TSharedPtr<
    const FUniqueNetId>'
```

To resolve the error the type of LocalUserID in method HandlePartyMember-Joined is changed as following.

```
1       //TSharedPt< const FUniqueNetId> LocalUserId
         FUniqueNetIdRepl LocalUserId
3
```

similarly changed the datatype of LocalUserID in several other places in the file.

4.7 UTRecastNavMesh.h

- The error is of type

```
        /home/the_cowboy/unrealworks/UnrealProjects/
    UnrealTournament/Source/UnrealTournament/Public/
    UTRecastNavMesh.h:276:38: note: in instantiation of
    function template specialization 'TWeakObjectPtr<
    UUTPathNode, FWeakObjectPtr>::TWeakObjectPtr<const
    UUTPathNode, void>' requested here
                if (Goals.Contains(FRouteCacheItem(Node,
    EntryLoc, INVALID_NAVNODEREF)))
```

is due to the update in the Engine code

```
                /home/the_cowboy/unrealworks/UnrealEngine/
    Engine/Source/Runtime/Core/Public/UObject/
    WeakObjectPtrTemplates.h:77:3: warning: 'condition' is
    deprecated: Implicit conversions from const pointers to
    non-const TWeakObjectPtrs has been deprecated as it is not
     const-correct. Please update your code to the new API
    before upgrading to the next release, otherwise your
    project will no longer compile. [-Wdeprecated-declarations
    ]

```

The idea to remove the const keyword for UUTPathNode in the function definition of method Eval

```
                virtual float Eval(APawn* Asker, const
    FNavAgentProperties& AgentProps, AController* RequestOwner
    , const UUTPathNode* Node, const FVector& EntryLoc, int32
    TotalDistance) = 0;
```

- The non-const to const typecast error spits the following error

```
                /home/the_cowboy/unrealworks/UnrealProjects/
    UnrealTournament/Source/UnrealTournament/Private/UTBot.cpp
    :3964:89: note: in instantiation of function template
    specialization 'TWeakObjectPtr<UUTPathNode, FWeakObjectPtr
    >::TWeakObjectPtr<const UUTPathNode, void>' requested here
                        NavData->FindPolyPath(NavData->
    GetPolyCenter(StartPoly), AgentProps, FRouteCacheItem(Link
    .Start.Get(), NavData->GetPolyCenter(Link.StartEdgePoly),
    Link.StartEdgePoly), Polys, false);

```

I am trying with the definition of the method GetTransientCost by removing const keyword. So now the method definition is

```
                    virtual uint32 GetTransientCost(
  FUTPathLink& Link, APawn* Asker, const FNavAgentProperties
  & AgentProps, AController* RequestOwner, NavNodeRef
  StartPoly, int32 TotalDistance)
{
  return 0;
}
```

4.8 UTEditorEngine.cpp

Some DEPRECATED functions here were cleaned

- Error of type

```
          /home/the_cowboy/unrealworks/UnrealProjects/
  UnrealTournament/Source/UnrealTournamentEditor/Private/
  UTEditorEngine.cpp:9:29: warning: '
  StringAssetReferenceLoaded' is deprecated:
  StringAssetReferenceLoaded is deprecated, call
  FSoftObjectPath::PostLoadPath instead Please update your
  code to the new API before upgrading to the next release,
  otherwise your project will no longer compile. [-
  Wdeprecated-declarations]
      if (FCoreUObjectDelegates::StringAssetReferenceLoaded.
  IsBound())
```

4.9 SUTProgressSlider.cpp

- Error of type

```
          /home/the_cowboy/unrealworks/UnrealProjects/
  UnrealTournament/Source/UnrealTournament/Private/Slate/
  Widgets/SUTProgressSlider.cpp:98:5: error: no matching
  function for call to 'MakeBox'
  FSlateDrawElement::MakeBox(
```

Commented out RotatedClippingRect as per the depriciation. Might introduce the clipping misfunction in the Editor. Beware of that!

- Also the function call is found in SUTSlider.cpp, BackgroundBlurWidget.cpp,

4.10 UTEyewear.cpp

- The method DetachRootComponentFrromParent() is deprecated. So I am using the DetachFromActor() routine with the FDetachmentTransformRules::KeepWorldTransform. Could be KeepRelativeTransform but I am not sure. But it certainly should be bMaintainWorldPosition equivalent.
- Also found the call in UTCarriedObject.cpp, UTCharacter.cpp (if the function PlayDying())

4.11 UTNavBlockingVolume.h

- No idea about the role of method GetNavigationData but it surely uses function call presumably present in the old engine code. So I am commenting them out and will come back to it later.
- simillar obscurity in UTLift.cpp

4.12 UTCheatManager.cpp

- Commented the entire TestAMDAllocation() method to avoid ENQUEUE_UNIQUE_RENDER_COMMA depreciation.

4.13 UTBasePlayerController.cpp

- In the method ServerSay_Implementation() (and some other calls) the call to Trim() is replaced by TrimStart() probably to trim the whitespaces.
- Another instance in SUTTextChatPanel.h
- Another instance in UTLobbyGameState.cpp

4.14 UTAnalyticsBlueprintLibrary.cpp

- AttrName and AttrValue are declared const in the AnalyticsEventAttribute.h structure. Created a pull request on GitHub to rectify the issue. Right now commenting the lines.

4.15 BackgroundBlurWidget

- The warning of type

```
warning: 'BuildDesignTimeWidget' is deprecated: Don't
call this function in RebuildWidget any more. Override
RebuildDesignWidget, and build the wrapper there; widgets
that derive from Panel already do this. If you need to
recreate the dashed outline you can use
CreateDesignerOutline inside RebuildDesignWidget. Please
update your code to the new API before upgrading to the
next release, otherwise your project will no longer
compile. [-Wdeprecated-declarations]
```

```
        return BuildDesignTimeWidget(MyGuardOverlay.
    ToSharedRef());
```

The simple solution is to replace with the code

```
        #if WITH_EDITOR
        return CreateDesignerOutline(MyWidget.ToSharedRef());
        #else
        return MyWidget.ToSharedRef();
        #endif

```

similar modification in UTLoadGuard.cpp, UTListView.cpp

4.16 SUTPlayerSettingsDialog.cpp

- commented out //ViewFamily.bUseSeparateRenderTarget = true;. No such memberfunction in the new Engine. Furthermore //View-¿ViewRect = View-¿UnscaledViewRect;
- simillar comment out for SUTWeaponConfigDialog.cpp

4.17 SUTGameSetupDialog.cpp

- The method RequestMinimize() seems stub for Linux. Will have to implement from the Engine. Also present in SUTMenuBase.cpp

4.18 UTMapTriangleCountTests.cpp

- Commented out the method GetNumTrianglesInScene() components. Could be for stats purpose.

4.19 UTNavBlockingVolume.h

- Commenting out GetNavigationData method. Don't yet understand Nav-Octree and relevant concepts.

4.20 UMG/SObjectTableRow.h

Turns out that new Engine has predefined class SObjectTableRow so it clashes the class defined in the UT code. Right now I am renaming the UT code class to SObjectTableRowUT.

4.21 Copying FActorComponentTickFunction

- The error of type

```
1       error: object of type 'FActorComponentTickFunction'
    cannot be assigned because its copy assignment operator is
     implicitly deleted
                                CustomDepthMesh->
    PrimaryComponentTick = CustomDepthMesh->GetClass()->
    GetDefaultObject<USkeletalMeshComponent>()->
    PrimaryComponentTick;
3
```

is all over the UT code. Apparently it worked for old engine, but new Engine code explicitly prohibits such behavior. Need to find a good solution as per the thread copy deilema. The code snippet is present in UnrealTournament.cpp, UTWeaponAttachment.cpp, as following

```
                                // TODO: scary that these get
    copied, need an engine solution and/or safe way to
    duplicate objects during gameplay
2       CustomDepthMesh->PrimaryComponentTick =
    CustomDepthMesh->GetClass()->GetDefaultObject<
    USkeletalMeshComponent>()->PrimaryComponentTick;
        CustomDepthMesh->PostPhysicsComponentTick =
    CustomDepthMesh->GetClass()->GetDefaultObject<
    USkeletalMeshComponent>()->PostPhysicsComponentTick;
4
```

4.22 SUTWeaponConfigDialog.cpp

- Error of type

```
1                               /home/the_cowboy/unrealworks/
    UnrealProjects/UnrealTournament/Source/UnrealTournament/
    Private/Slate/Dialogs/SUTWeaponConfigDialog.cpp:684:13:
    error: no member named 'bUseSeparateRenderTarget' in '
    FSceneViewFamilyContext'
        ViewFamily.bUseSeparateRenderTarget = true;
3
```

Right now I don't exactly know the role of bUseSeparateRenderTarget so I am commenting it out.

4.23 UTPlayerState.cpp

- Commented out

```
1     //FJsonSerializer::Serialize(Notification.Payload->
      AsObject().ToSharedRef(), Writer);
2
```

4.24 UTResetLineUpDefaultsCommandlet.cpp

- Commented out PostEditChange() method. Also in UTResetPostProcessVolumes.

4.25 UTJumpPad.cpp

- Again the error of type

```
1     warning: 'condition' is deprecated: Implicit conversions
      from const pointers to non-const TWeakObjectPtrs has been
      deprecated as it is not const-correct. Please update your
      code to the new API before upgrading to the next release,
      otherwise your project will no longer compile. [-
      Wdeprecated-declarations]
```

The reason is clear, the error is produced due to

```
1     note: in instantiation of function template
      specialization 'TWeakObjectPtr<UUTPathNode, FWeakObjectPtr
      >::TWeakObjectPtr<const UUTPathNode, void>' requested here

        B->SetMoveTargetDirect(FRouteCacheItem(Path.End.Get(),
      NavData->GetPolySurfaceCenter(Path.EndPoly), Path.EndPoly)
      );
3
```

- The error can be easily rectified by removing the const keyword

```
1     for (FUTPathLink& Path : MyNode->Paths)
2
```

4.26 UTGameMode.cpp

- some dumb error in SetNative method so commenting it out in the function InitGame

4.27 UTWeaponAttachment.cpp

- Tick error line 138 139 also in UnrealTournament.cpp line 172 and 173