# Links

## API

* Actor
  + <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Engine/GameFramework/AActor/>
* Pawn
  + <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Engine/GameFramework/APawn/>
* Character
  + <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Engine/GameFramework/ACharacter/index.html>
* FVector
  + <https://docs.unrealengine.com/4.27/en-US/API/Runtime/Core/Math/FVector/>

## Information

* Root motion
  + <https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/RootMotion/>
* Blueprint
  + <https://www.youtube.com/results?search_query=zak+parrish+unreal+tutorial+blueprints>
* Animation overview
  + <https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/Overview/>
* AnimInstance
  + Anim instance you are referring to is a C++ object of class UAnimInstance. Animation Blueprint (AnimBP) is in fact a class derived from UAnimInstance (or from any class which inherits from it). In default, you create your animation blueprint just from standard UAnimInstance and extend it solely in blueprint -- in C++ semantics: UAnimInstance is a base class of your blueprint.
  + Example: In C++, I define my own class UMyAnimInstance which inherits from Epic's UAnimInstance because I want to extend or modify its functionality (in some way) in C++. Later in the editor, I can create an animation blueprint which derives from my UMyAnimInstance (instead of default UAnimInstance) which allows me to create state machines, animation states and blending trees which may use my extended functionality.
* Video on making ablend space + having two state machines
  + <https://www.youtube.com/watch?v=4ounmO7KcbQ>
* Reading anim curve data in C++
  + <https://forums.unrealengine.com/t/reading-anim-curve-float-values-at-all-time/128360>
  + <https://answers.unrealengine.com/questions/711277/missing-animation-curves-in-cooked-builds.html>
* Blendspace tutorial
  + https://www.youtube.com/watch?v=3l7XBO9-Gvk