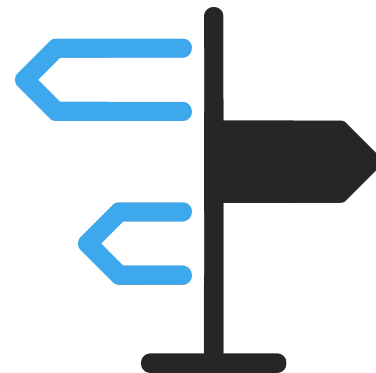


Inhalt

- Installieren
- Projekt erstellen
- UI Bedienen
- Objekte importieren
- Blueprints
- C++



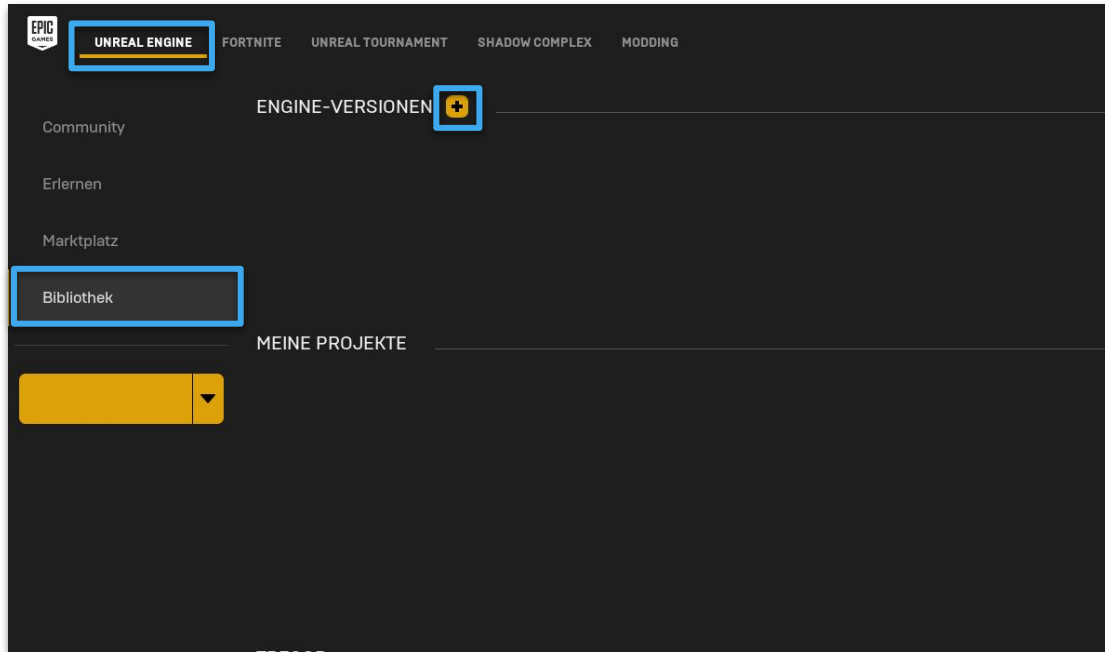
Installieren

Launcher

Engine

Visual Studio

- Registrieren
- Epic Launcher herunterladen (www.unrealengine.com) und installieren

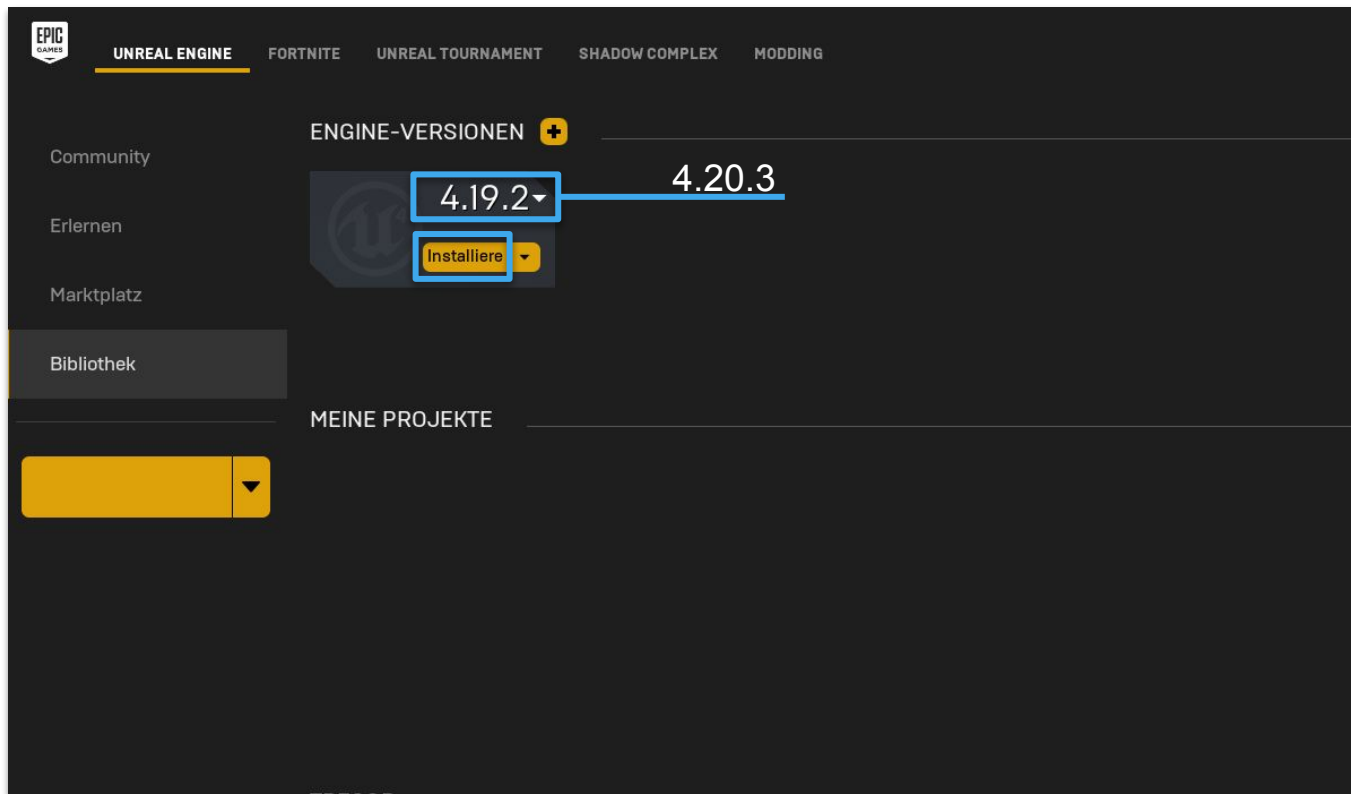


Installieren

Launcher

Engine

Visual Studio

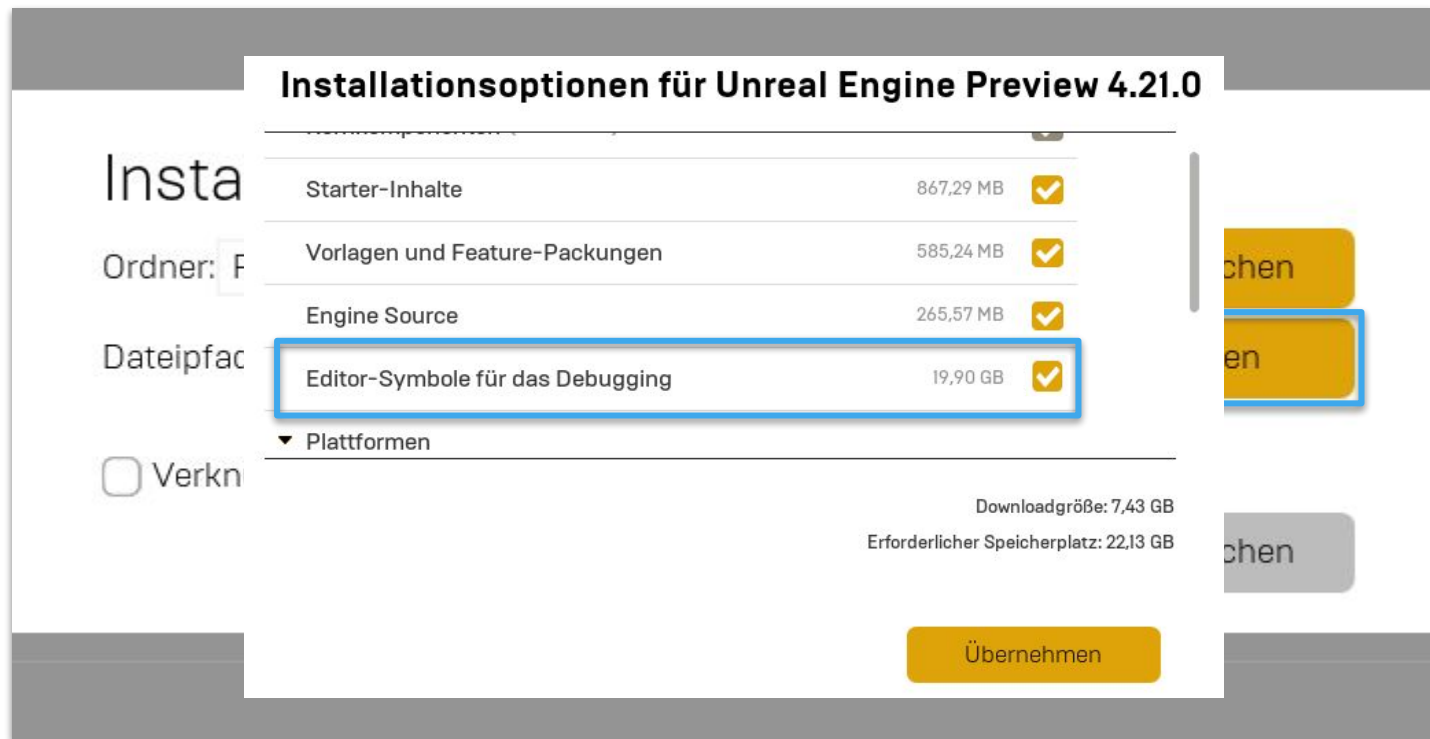


Installieren

Launcher

Engine

Visual Studio



Installieren

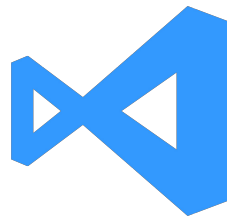
Launcher

Engine

Visual Studio

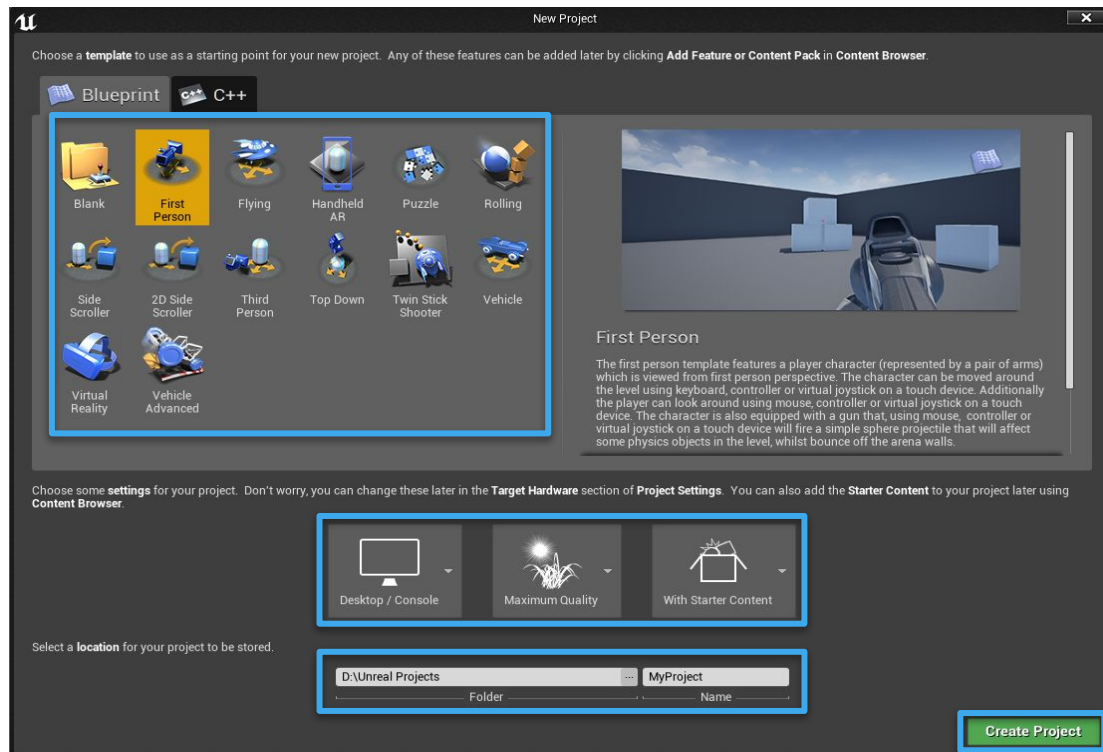
- Empfohlen: Visual Studio (VS) 2017 Enterprise (Dreamspark)
 - Ansonsten VS 2017 Community mit Update 3
 - VS 2015 funktioniert ab UE 4.20 nur mit spezieller Konfiguration

- Empfohlene Pakete bei der Installation:
 - Update 3 (erweiterte Tools)
 - Visual C++
 - allgemeine Desktop Entwicklung
 - Spieleentwicklung



Projekt erstellen

Templates



UI Bedienen

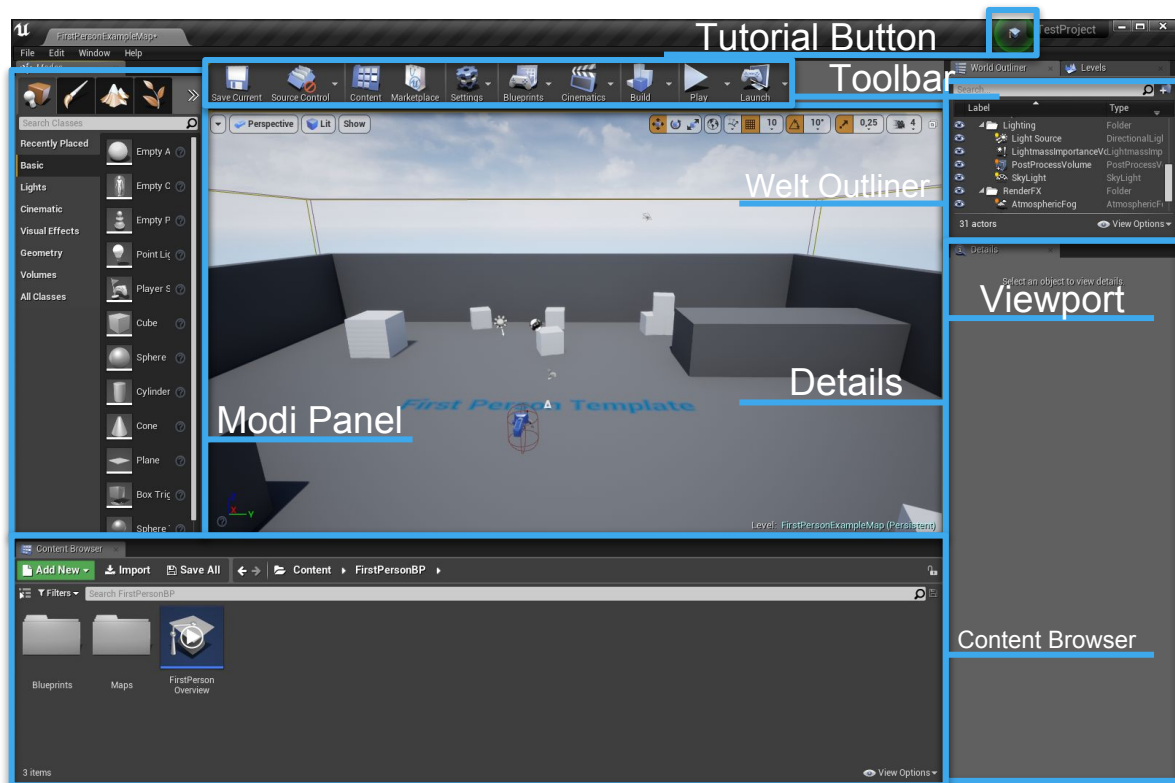
Allgemein

Modi

Explorer

Welt/Level

Details



UI Bedienen

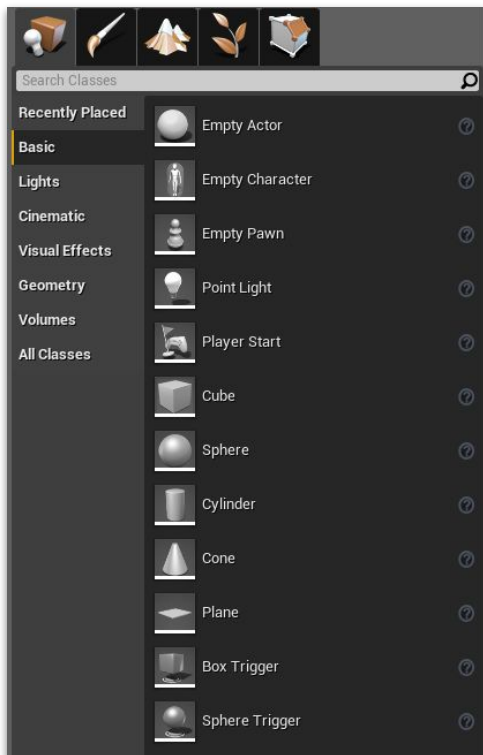
Allgemein

Modi

Explorer

Welt/Level

Details



- Platzieren

- Einzelne Objekte (Blueprint, C++)
- Drag & Drop

- Malen

- Farben, Gewichtungen, Texturen

- Landschaft

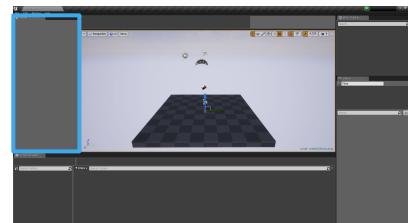
- Landschaften erstellen
- Management, feines Bearbeiten, Malen

- Pflanzen

- Objekttyp zu Liste Hinzufügen
- Fläche entsprechend der Parameter bemalen

- Geometrie

- Erstellen, Drehen etc.



UI Bedienen

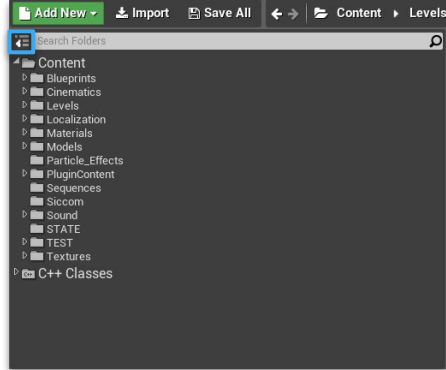
Allgemein

Modi

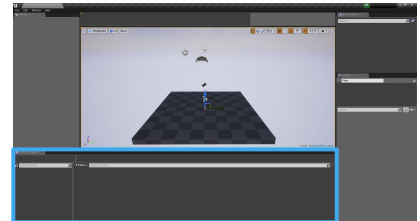
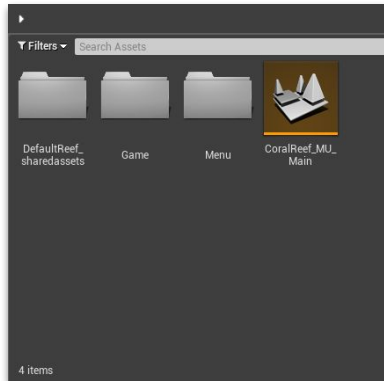
Explorer

Welt/Level

Details



- Übersicht aller verfügbaren Objekte
 - Z.b. Blueprint, C++, UI, Level, Ressourcen, etc.
- C++ Ordnerstruktur wie in Visual Studio
- Rechte Übersicht zeigt den Content Ordner (kein C++)
- Navigation wie im normalen Windows Explorer
- Hinzufügen von neuen Objekten (Inklusive C++ Klassen)



UI Bedienen

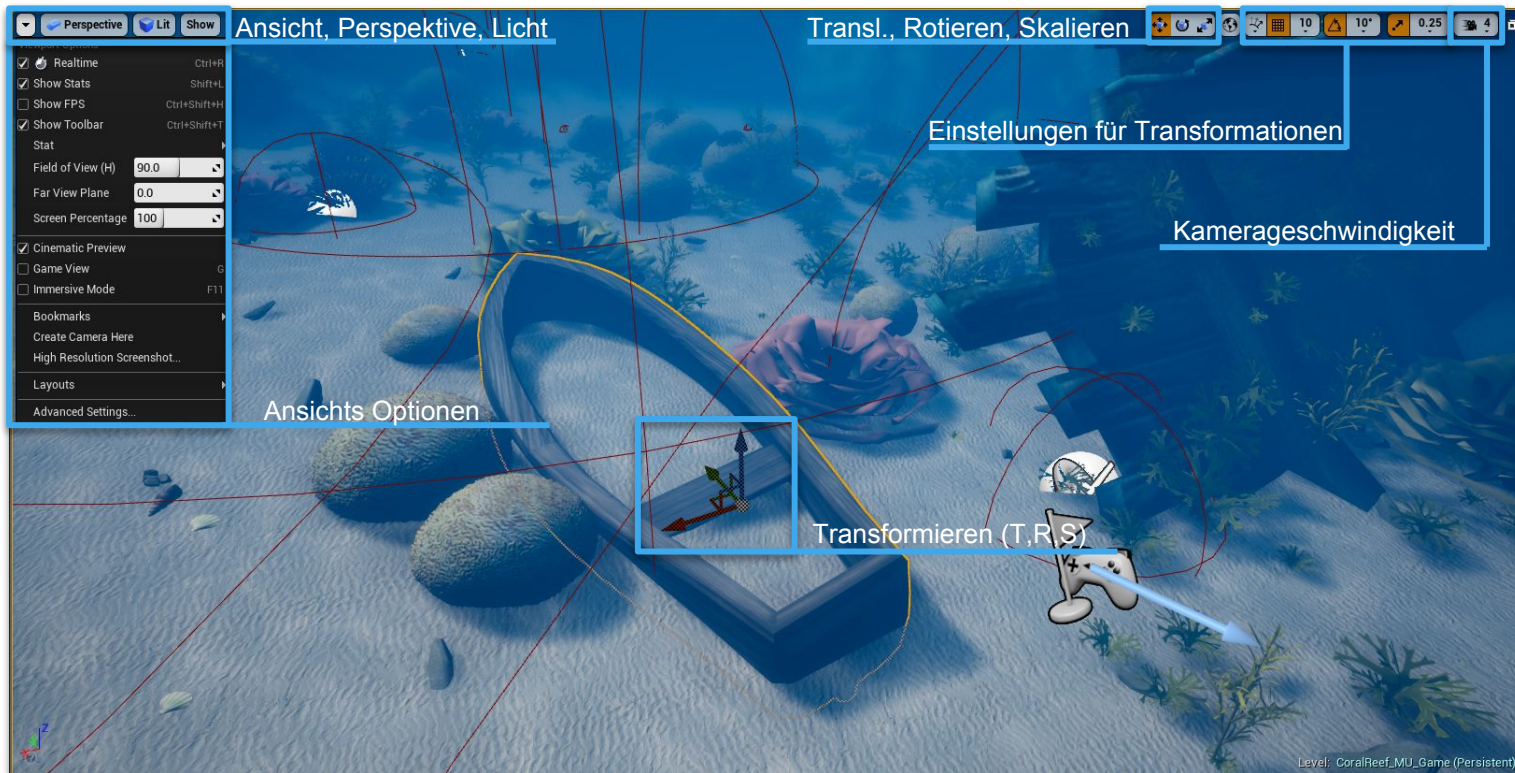
Allgemein

Modi

Explorer

Welt/Level

Details



UI Bedienen

Allgemein

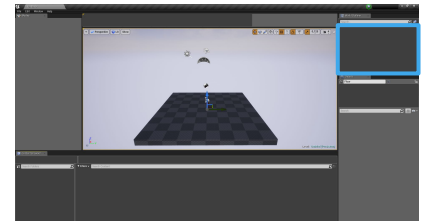
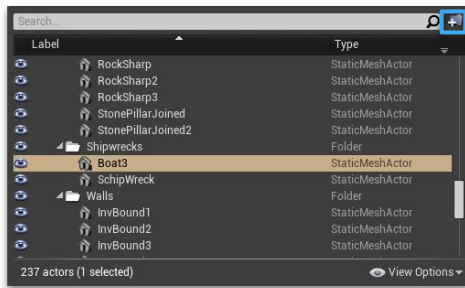
Modi

Explorer

Welt/Level

Details

- Übersicht über alle Objekte im aktuell geladenen Level
- Strukturierung mittels Ordern
- Objekte auswählbar (Detailansicht)



UI Bedienen

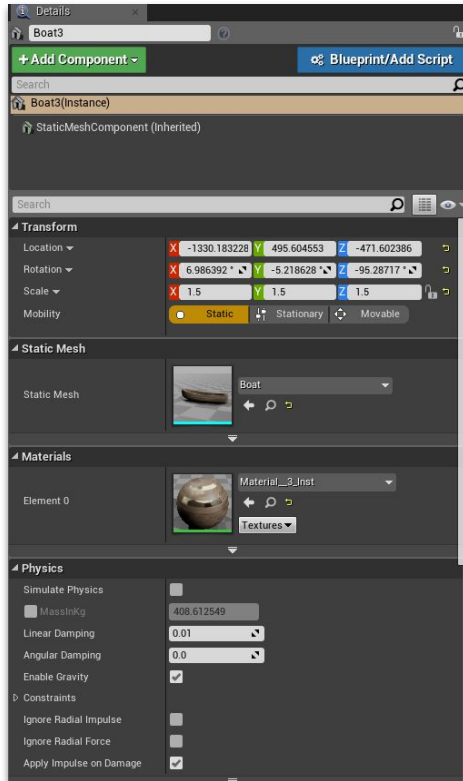
Allgemein

Modi

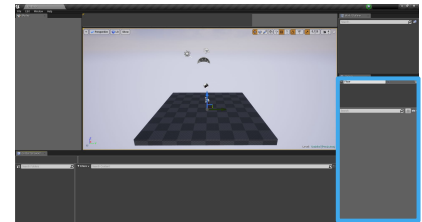
Explorer

Welt/Level

Details



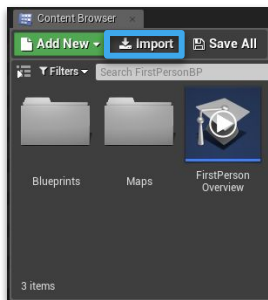
- Bearbeiten aller verfügbaren Einstellungen
 - Transformation
 - Mesh
 - Material
 - Physik, Kollision
 - Etc.
- Interne Struktur des Objektes
- Zugriff auf die Details der Unterobjekte



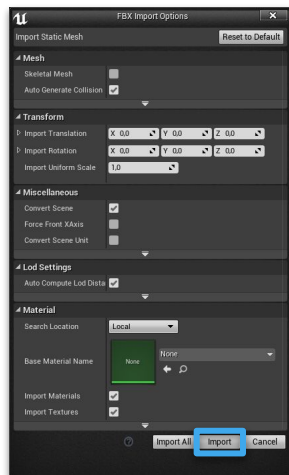
Objekte Importieren

Import

Material Editor



- Import Button oder Drag and Drop
- Unterstützt viele Formate: fbx, png, mp3 ...
- Importeinstellungen variieren bei verschiedenen Formaten
- Modelle können in der Welt platziert werden

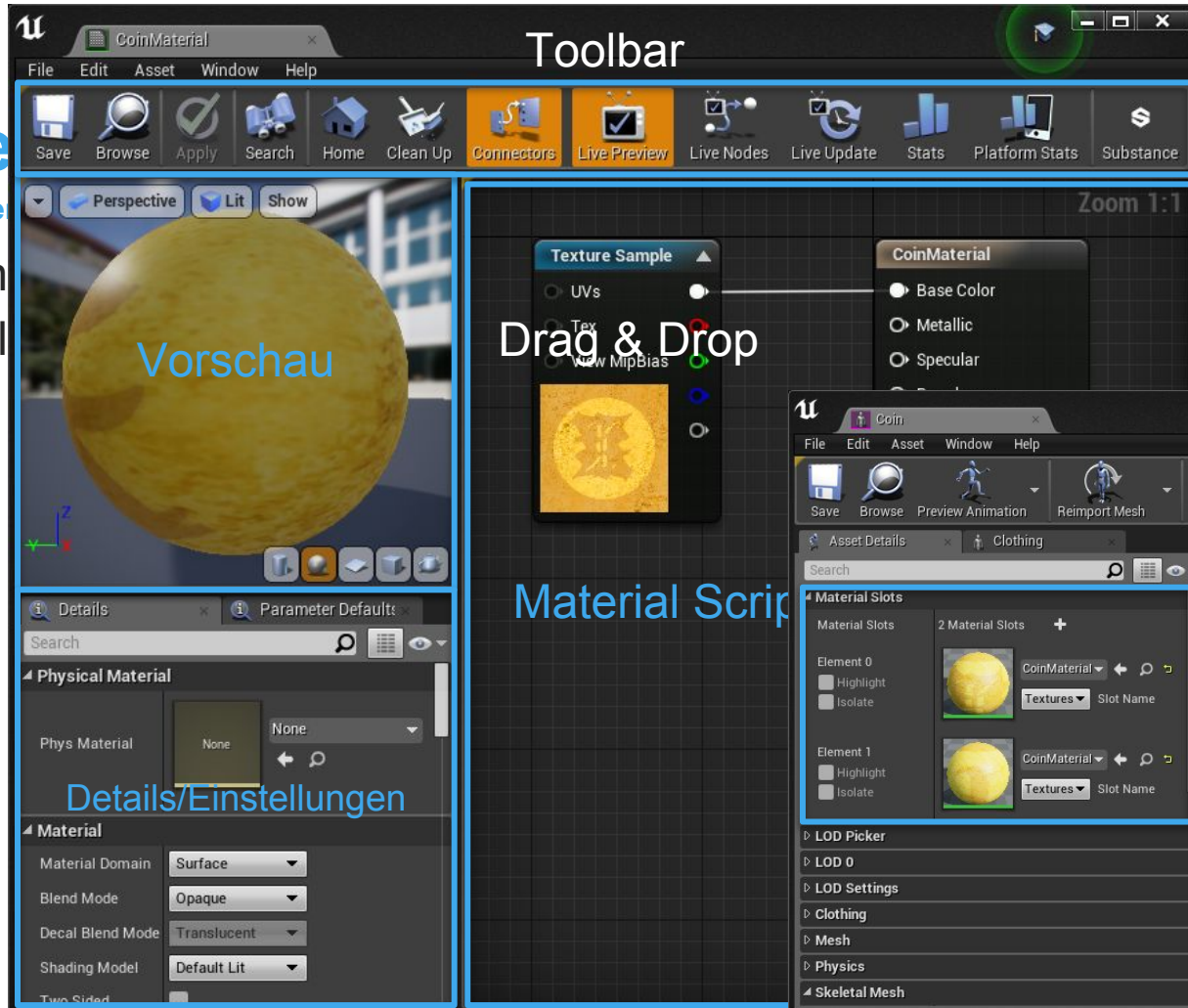
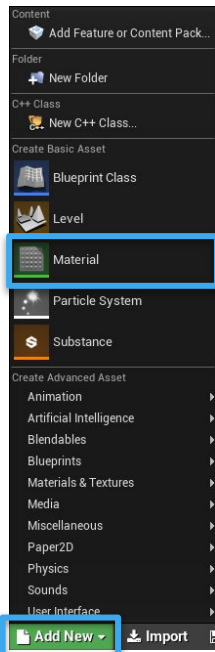


Objekte

Import

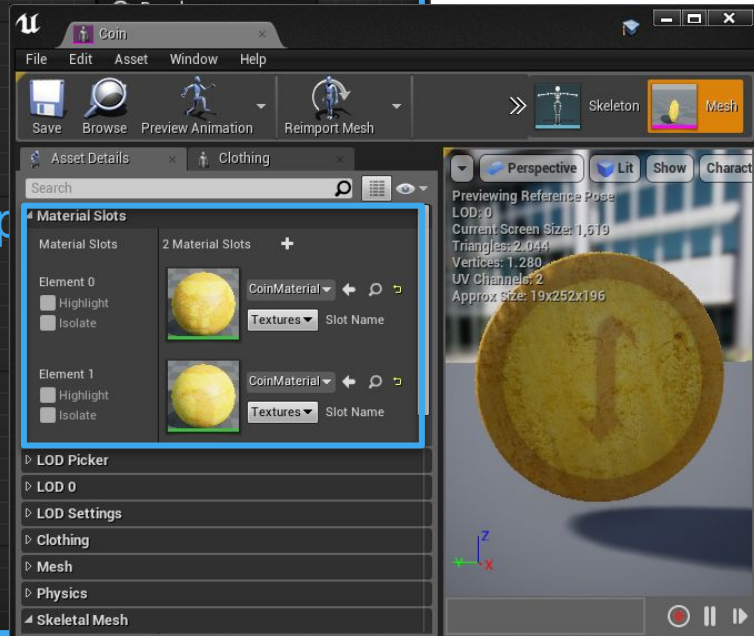
Material

- Mit dem
- Material



Drag & Drop

Material Script



Blueprints

UI

Scripting

Animation BP

- Besteht aus Toolbar, Komponenten-Baum und Objekteigenschaften
- Beinhaltet Viewport Tab und Event Graph Tab zum Scripting

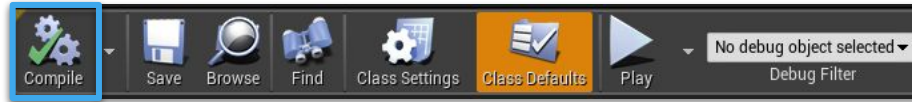


Blueprints

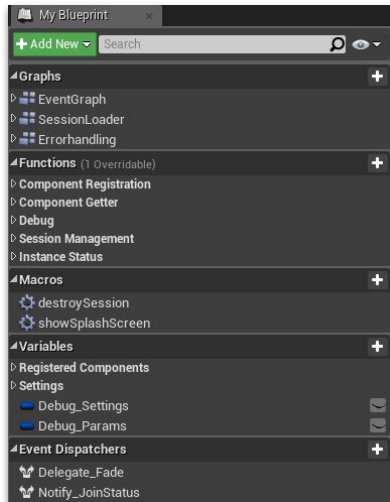
UI

Scripting

Animation BP



- Nach jeder Änderung Compile ausführen
- Findet Fehler in Blueprints



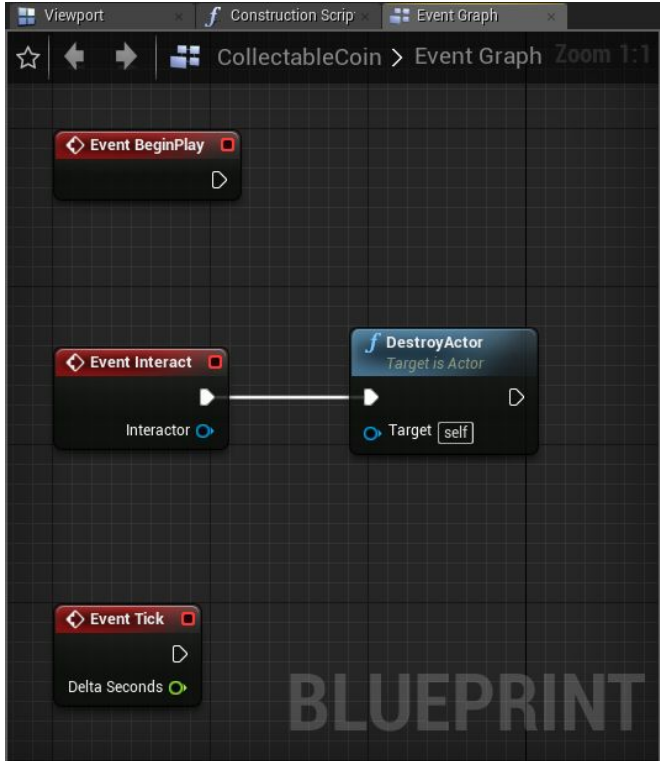
- Graphs: Erstellte Graphen für Skripte
- Functions: Können von anderen Aktoren aufgerufen werden
- Macros: Können nur von diesem Blueprint aufgerufen werden
- Variables: Erstellbare veränderbare Werte, können von anderen Aktoren aufgerufen und verändert werden
- Event Dispatchers: Events können von anderen Aktoren registriert werden

Blueprints

UI

Scripting

Animation BP



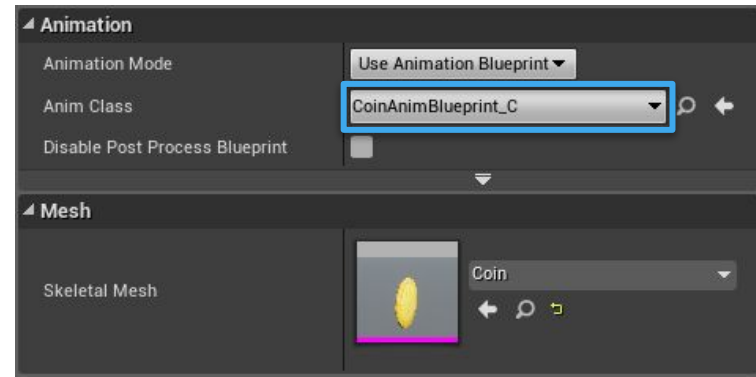
- Events werden bei bestimmten Ereignissen gefeuert
- Der weißen Linie wird gefolgt
- Nach Events folgende Nodes werden zuerst ausgeführt
- Nodes können Parameter brauchen
- Nodes können mehrere Ergebnisse haben
- Nodes können kopiert, gelöscht und auch kommentiert werden

Blueprints

UI Scripting Animation BP

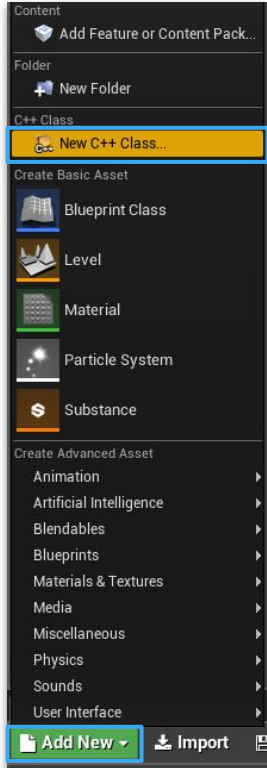
Animation Blueprints:

- Haben einen zusätzlichen Anim Graph
- Werden genutzt um Animationen zu Skripten
- Können auf Variablen von Aktoren zugreifen

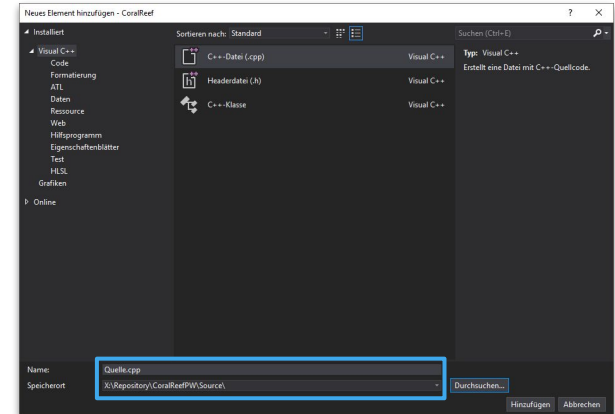




Grundlagen Blueprints

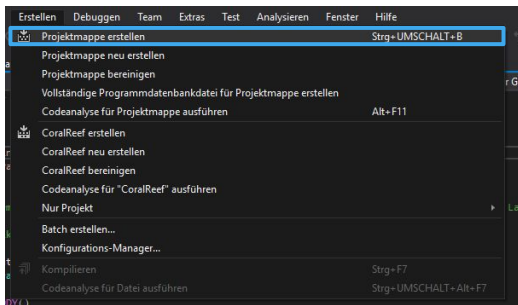
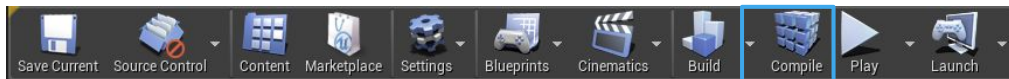


- Erstellen über UE (links) oder VS (rechts)
- Es muss immer eine *.h (Header) und eine *.cpp (Code) geben
- Die *.h Datei muss immer eine [klasse].generated.h includen





Grundlagen Blueprints

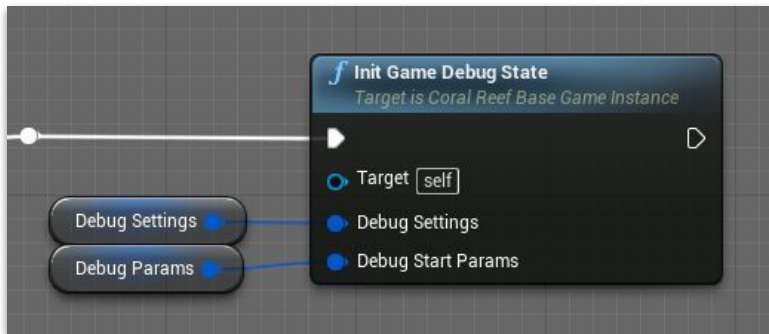


- Kompilieren kann über die Engine (oben) oder VR (unten) geschehen
- Beim Kompilieren während der Laufzeit von UE wird immer eine “Hot-Reload” Datei erzeugt
 - Bugs lassen sich evtl durch neu erstellen des Projektes über VS beheben

C++

Grundlagen Blueprints

- Verbinden von C++ Klassen und Blueprint Klassen ist möglich
 - C++ → BP ist einfacher als BP → C++
- Manche Datentypen existieren in BP nicht
- Notwendige C++ Notationen einfach zu finden



```
UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "BlueprintHelper")
bool VrSwimming = false;

UFUNCTION(BlueprintCallable)
void InitGameDebugState(FDebugSettings DebugSettings, FStartParameters DebugStartParams);
```



Vielen Dank für die Aufmerksamkeit