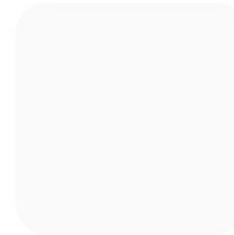
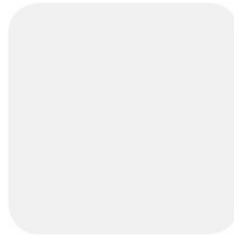


Fachprojekt DET SS 2019 - Einführung in Unity -



Einführung in Unity

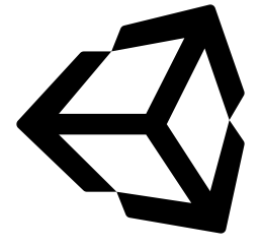


Überblick

1. Was ist und was kann Unity
2. Editor
3. Definitionen
4. Scripting
5. Interaktionen/Kommunikation
6. Ressourcen

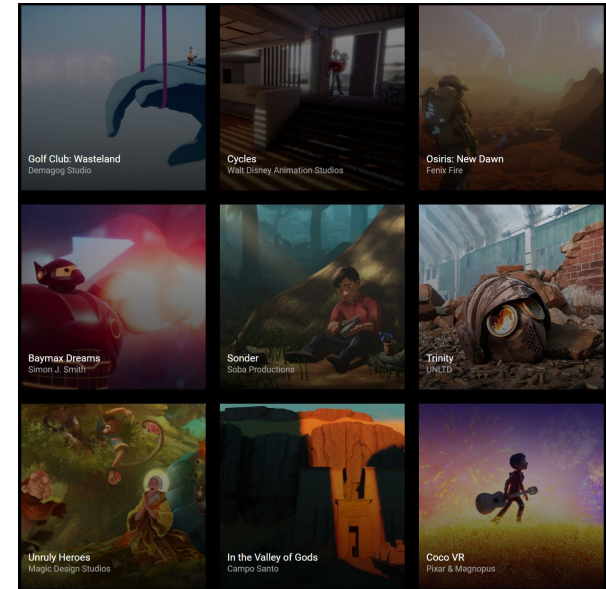
Was ist Unity

- Game Engine für 2D und 3D Anwendungen
- Programmiersprache C#
- Viele unterstützte Plattformen
- Großer Asset Store
- Kostenfrei bis zu einem bestimmten Umsatz



Was kann Unity

- Sehr viele Spielumsetzungen
- Teils schlechter Ruf durch Spiele bestehend aus Assets aus dem Store
- 2D, 3D, VR, AR, ... Anwendungen
- ...

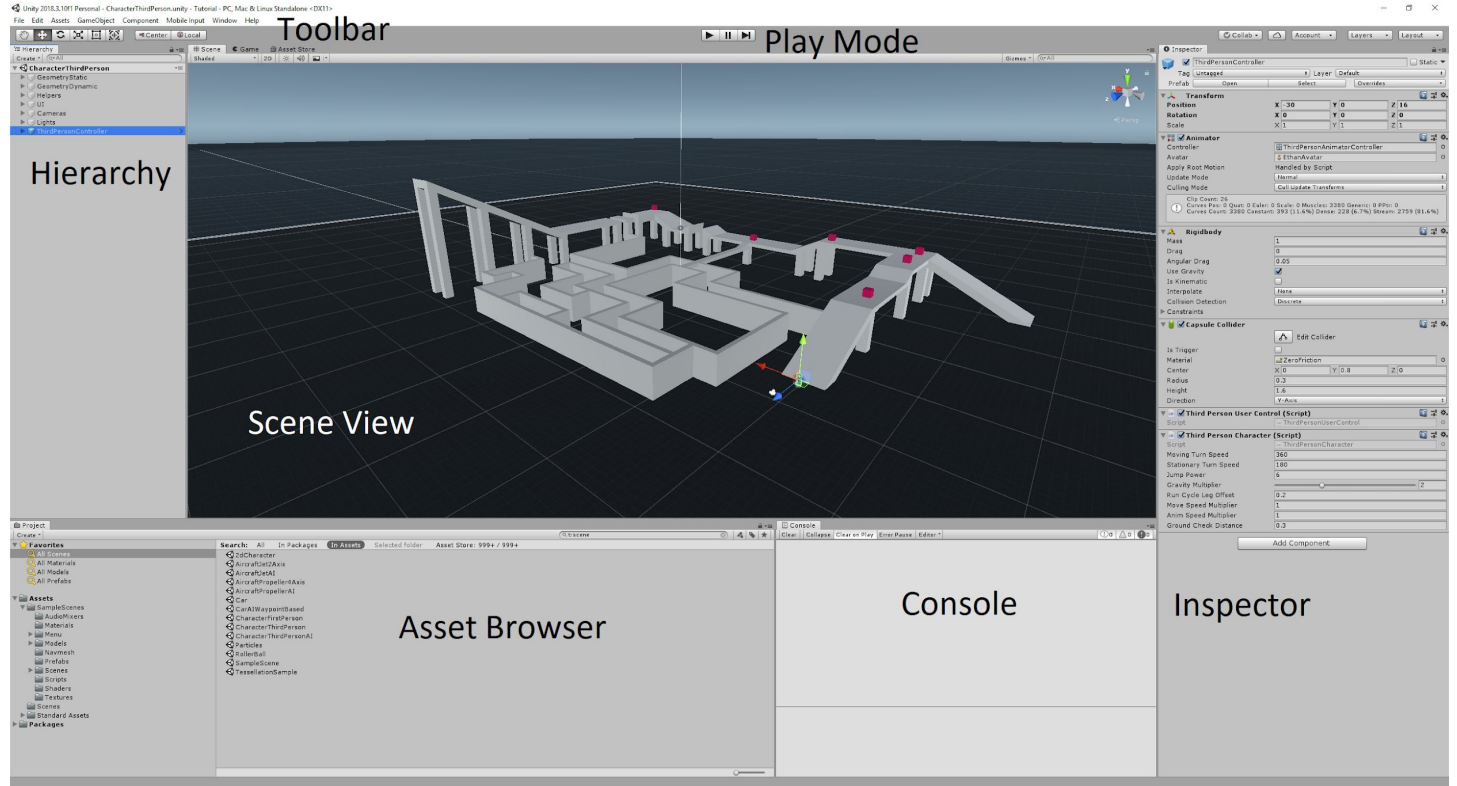


Was kann Unity

Rapid Prototyping

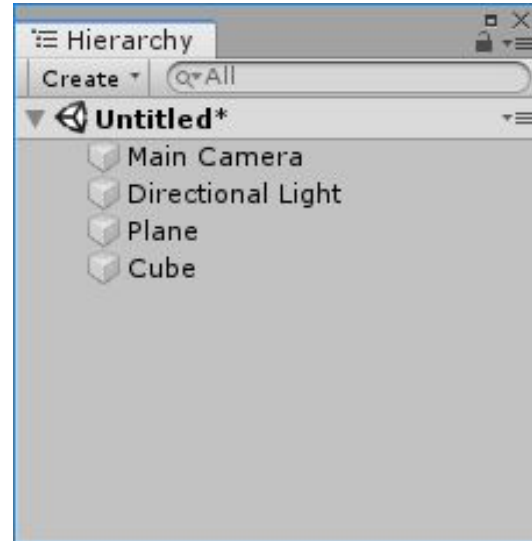
- <https://itch.io/jam/brackeys-2>
- <https://globalgamejam.org/2019/games>

Editor



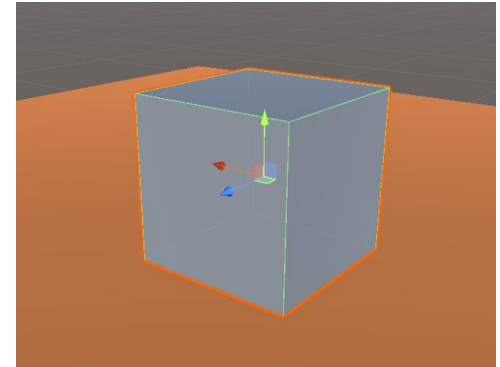
Definitionen: Szene

- Eine Szene setzt sich aus GameObjects zusammen



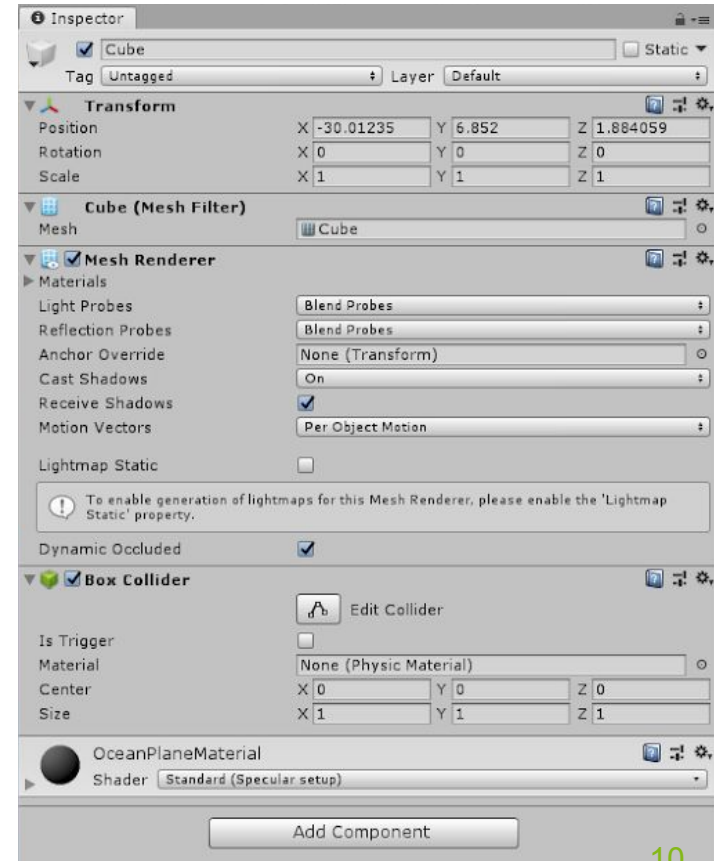
Definitionen: GameObject

- GameObjects bestehen aus Komponenten und erhalten dadurch ein Verhalten
- Ein GameObject hat einen Namen und einen Tag



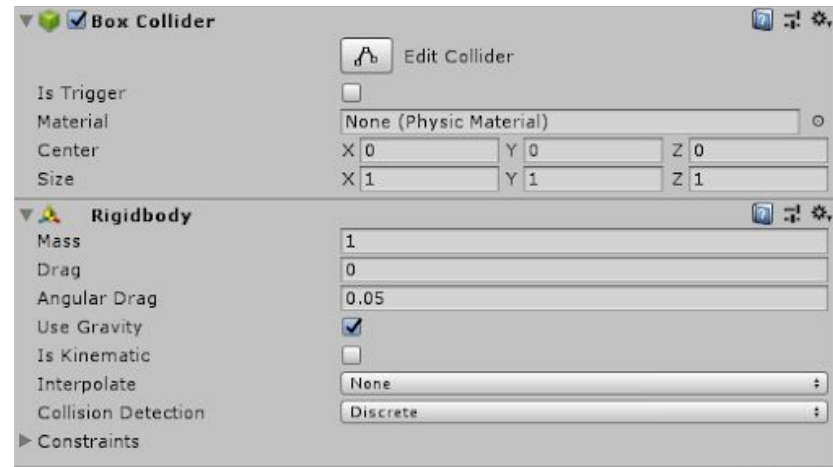
Definitionen: Komponenten

- Jedes GameObject hat eine Transform Komponente
- 3D Modelle haben einen Mesh Filter, einen Mesh Renderer und ein Material



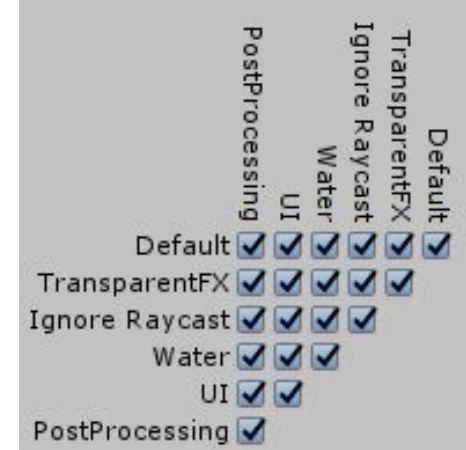
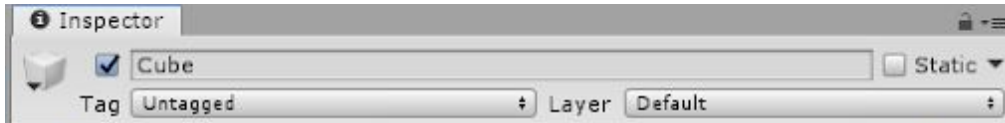
Definitionen: Physikkomponenten

- Rigidbodies und Collider gehören zur Physics Engine
- 3D und 2D Varianten



Definitionen: Layer

- Kollisionserkennung in Abhängigkeit von Layers



Definitionen: Weitere Komponenten

Komponenten für...

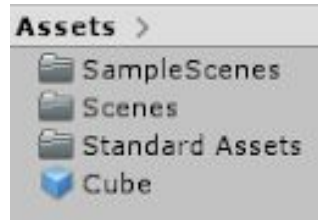
- Networking
- Rendering
- 2D
- Audio
- AI
- UI
- ...

Weitere Beispiele:

- Kamera
- Audio Listener
- Animator
- Line Renderer
- Particle System
- Sprite Renderer
- ...

Definitionen: Prefab

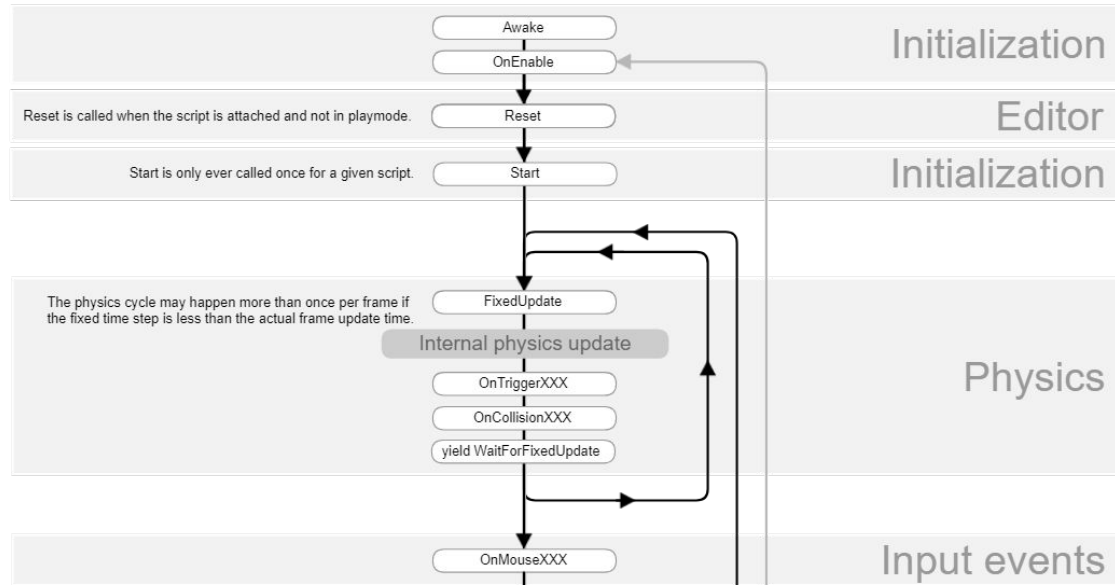
- Ein Prefab ist eine Blaupause für ein GameObject
- GameObjects werden als Prefab im Asset Browser abgespeichert
- Prefabs können zur Laufzeit instanziiert werden



Scripting: MonoBehaviour

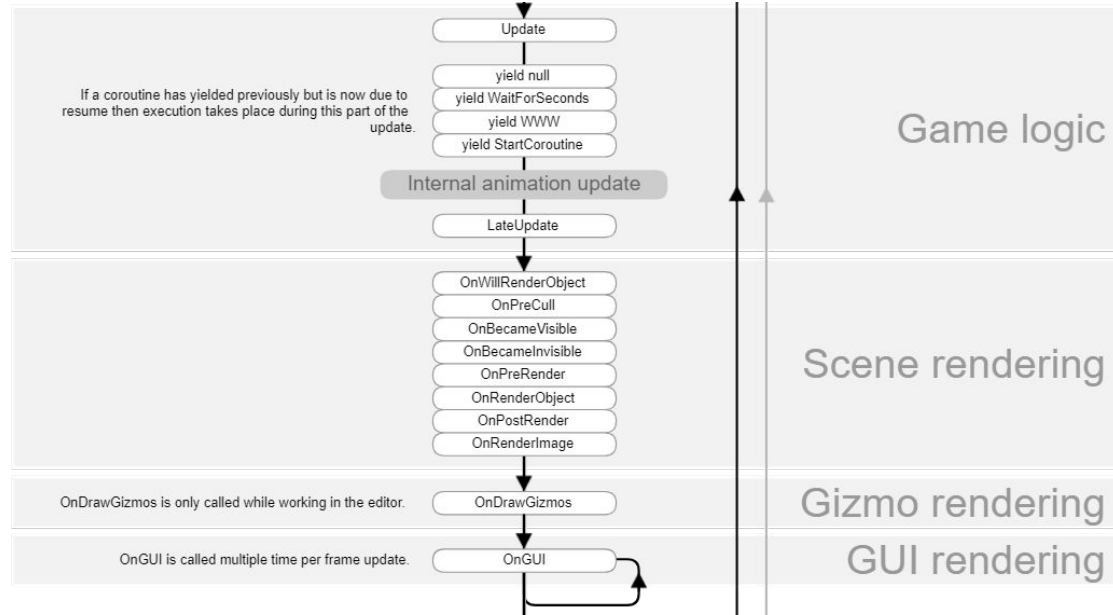
- Komponenten sind abgeleitete Klassen von MonoBehaviour
- Eigene Komponenten (oder Scripte) erben von MonoBehaviour

Scripting: Lifecycle



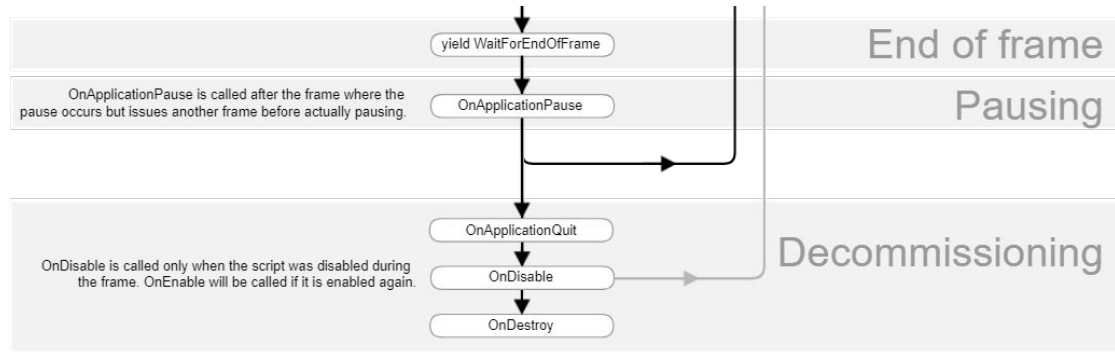
[Unity Manual](#)

Scripting: Lifecycle



[Unity Manual](#)

Scripting: Lifecycle



[Unity Manual](#)

Scripting: Event Functions

- *Awake()*
 - Wird vor Start() ausgeführt
- *Start()*
 - Wird vor dem ersten Frame Update ausgeführt
- *FixedUpdate()*
 - Wird in konsistenten Zeitintervallen aufgerufen
- *Update()*
 - Wird jeden Frame aufgerufen

[Scripting Reference](#)

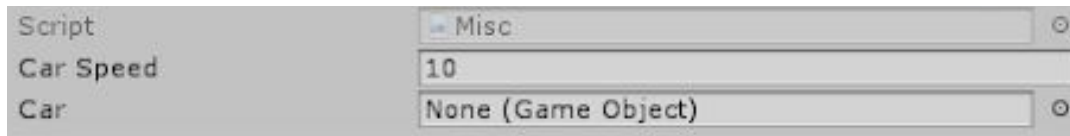
Scripting: Event Functions

- *OnEnable()* / *OnDisable()*
 - Wird bei der Aktivierung und Deaktivierung von einem GameObject aufgerufen
- *OnTrigger()* / *OnCollision()*
 - Wird bei einer Kollision aufgerufen
- *OnDestroy()*
 - Wird aufgerufen, sobald das GameObject zerstört wird

[Scripting Reference](#)

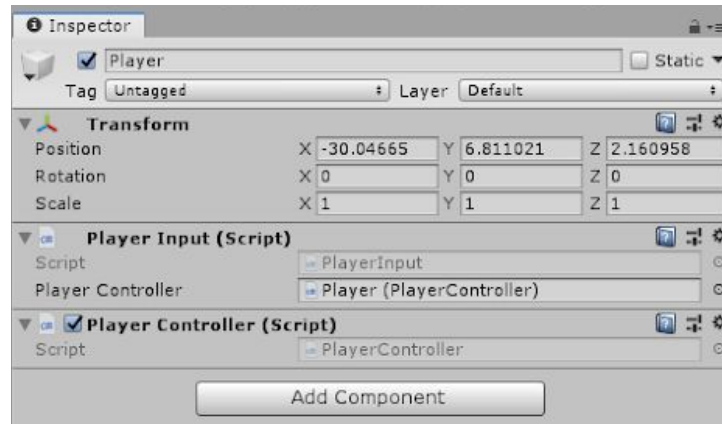
Interaktionen/Kommunikation

- Member Variablen serialisieren:
 - *[SerializedField] private float carSpeed = 1.0f;*
 - oder
 - *public GameObject car;*



Interaktionen/Kommunikation

- Referenzierung von Komponenten per Inspector:



Interaktionen/Kommunikation

- Referenzierung von Komponenten per Script:
 - *PlayerController playerController=
GetComponent<PlayerController >();*

Interaktionen/Kommunikation

- Referenzierung von Komponenten per Script:
 - *GameObject player =
GameObject.FindGameObjectWithTag(“Player”);*
 - *PlayerController playerController =
player.GetComponent<PlayerController>();*

Interaktionen/Kommunikation

- Referenzierung von Komponenten per Script:
 - *GameObject player =
GameObject.FindGameObjectWithTag(“Player”);*
 - *PlayerController playerController =
player.GetComponent<PlayerController>();*

Ressourcen

- [Unity User Manual](#)
- [Unity Scripting Reference](#)
- [Unity Learn](#)
- [Unity Standard Assets](#)
- [Unity3D College](#)
- [Hollistic3d](#)
- [Brackeys](#)
- [Catlike Coding Text Tutorials](#)