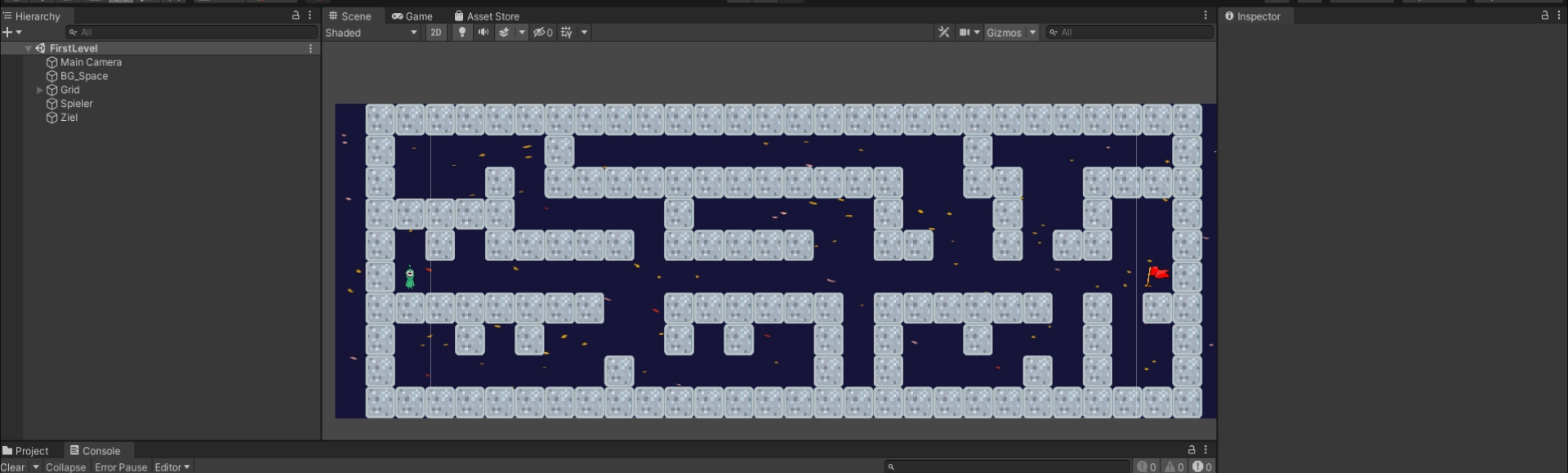


Grundlagen der Medieninformatik 2

T04 - 06.07.2021
Anwendungstutorium Unity



Unity!

Unity - Scenes



- Ein Unity Projekt besteht aus mehreren Scenes (Szenen)
- Jede Scene kann individuell konfiguriert werden und enthält demnach verschiedene Objekte
- GameObjects - Deutsch “Spiel-Objekte” die Objekte die das Spiel verwendet (es gibt auch andere wie z.B. UI-Objekte / Kamera usw.)



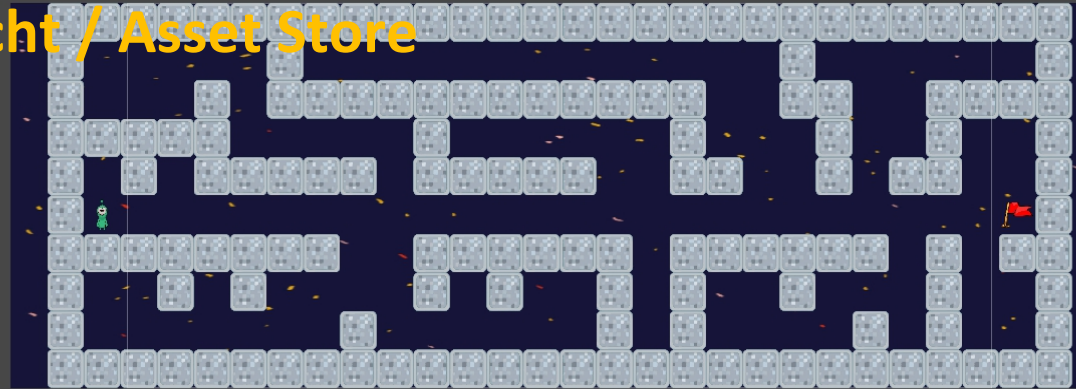
Objekthierarchie

Ansicht der aktiven Szene

Inspector

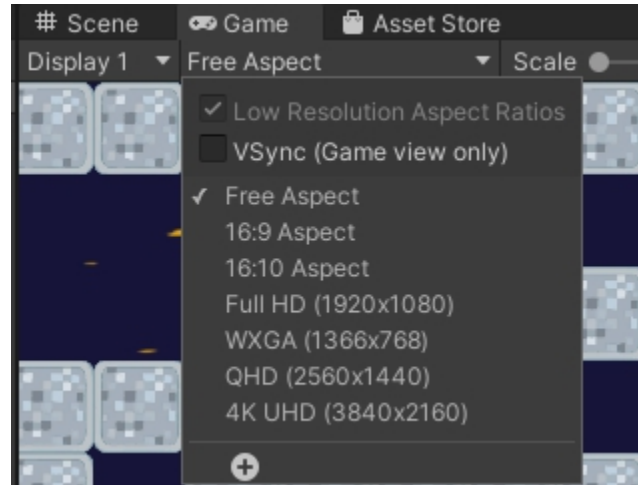
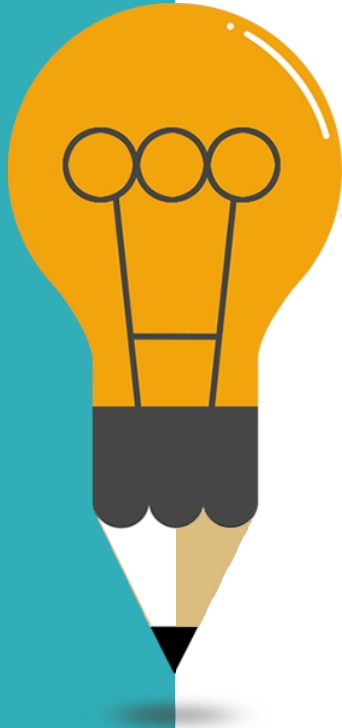
Projekt Dateien
/ Debug Konsole

Szene Übersicht / Spielkamera ansicht / Asset Store



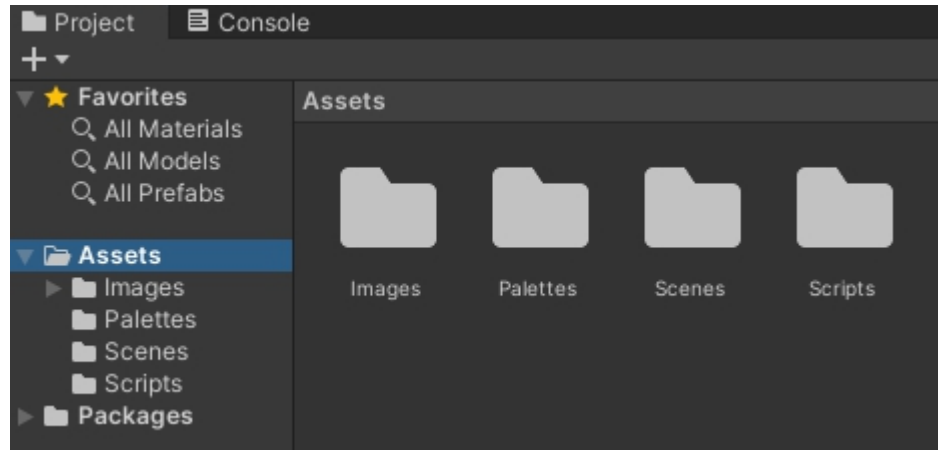
Unity - Game

- Man kann zwischen der Perspektive der aktiven Kamera und der Szenenübersicht wechseln
- Die Kamera hat dabei eine einstellbare Screen-Resolution:



Unity - Projekthierarchie

- Ein Unity-Projekt hat einen Ordner mit dem Namen “Assets”, welche eure Dateien / Fotos / Skripte enthält.
- Scripts enthält dabei die C# Dateien welche ihr verwendet.
(Kann weiter unterteilt werden bei großer Architektur)



Unity - C#



- Die tatsächliche Spiellogik wird in C# Dateien umgesetzt. (**Physik ist eingebaut** und die Komponenten müssen nur verwendet werden - keine manuelle Implementierung notwendig)
- Damit ein C# Skript ausgeführt wird muss es entweder an ein GameObject “rangehängt” werden (**MonoBehaviour** Objekte) oder durch ein anderes Skript aufgerufen werden (wie bei Standardprogrammierung)
- Objekte die von **MonoBehaviour** erben haben mehrere eingebaute Funktionen welche bei Spiel-Start / Jedes Frame / anderen Szenarien aufgerufen werden. (z.B. Start() und Update())

Unity - MonoBehaviour



- Objekte die von **MonoBehaviour** erben haben mehrere eingebaute Funktionen welche bei Spiel-Start / Jedes Frame / anderen Szenarien aufgerufen werden.
- Z.B.:
 - `void Start()` - aufgerufen beim Erstellen des Parent-Objekts
 - `void Update()` - aufgerufen jedes Frame

- **Gesamte Liste:**

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>

Unity - Komponenten anderer Objekte



- Ein `GameObject` kann auf eine Komponente eines anderen Objekts mit der Methode `GetComponent<>()` bekommen
- Eine Komponente kann z.B. ein C# Skript sein, Shader Settings, Texturen, Sound etc.
- Beispiel:
 - Angenommen Objekt A hat eine Komponente D
 - Objekt B kann auf diese mittels **`A.GetComponent<D>()`** zugreifen
- Ist A nicht direkt ein `GameObject`, kann seine `GameObject` Komponente mittels `.gameObject` bekommen werden:
 - **`A.gameObject.GetComponent<D>()`**

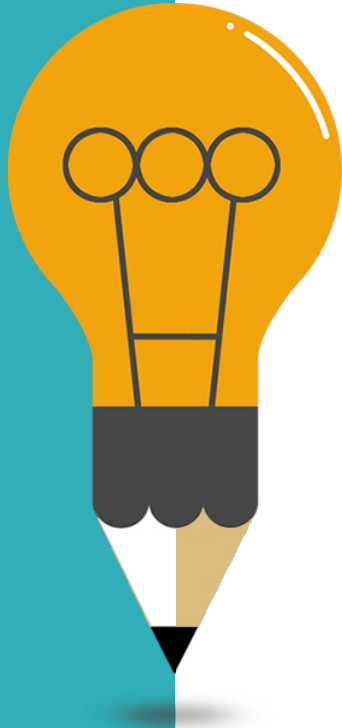
C# Notation



- In C# werden Funktionsnamen groß geschrieben:
 - **void Start()**
- Ebenso werden Variablennamen groß geschrieben:
 - **bool HelloWorld = false;**
- Bei Getter/Setter wird die private Version der Variable mit einem **_** und klein geschrieben:
 - **private bool _hi;**
 - **public bool Hi {**
 - **get { return _hi; }**
 - **set { _hi = value; } // "value" ist der Wert der mitgegeben wird,**
 - **// eingebautes Keyword (wie "this")**
 - **}**

Anwendungstutorium!

- **Auf StudIP unter Zentrale Materialien**



Medieninformatik 2 – Anwendungstutorium



Erstellt von Alex Ackermann und Bastian Dänekas am 31.05.2019

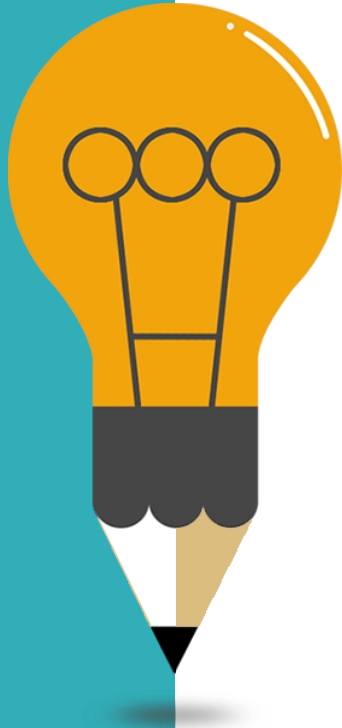
Angepasst von Alex Ackermann am 25.03.2020

Inhalt

1. Game Engine Unity
2. Projekt Anlegen
3. Unity Benutzeroberfläche
4. Unity Assetstore
5. Einsatz der Grafiken – Hintergrund, Labyrinth, Spieler und Ziel
6. Physik, Skripte und Kollision
7. Neue Szene und Szenenwechsel

Blatt E3 - Compositing

- Abgabe bis zum **12.7, 20:00** auf StudIP!



Universität
Bremen

Übungen Sommersemester 2021



Übung E3: 3D Compositing

Einzelaufgabe, 11 Punkte, Abgabe 12.07.21, 20:00 in Stud.IP

Montiere das animierte Insekt aus ÜZ E2 in die vorgegebene Realweltszene.

- » Verwende die Realweltszene `uebungE1bis3-realweltclip.mp4`.
- » Tracke die Kamerabewegung.
- » Passe Pose, Skalierung und Animation des Insektes so an, dass es in die Szene passt.
- » Ein Teil der realen Szene soll das Insekt verdecken.
- » Stelle die reale Lichtsituation sinnvoll realistisch nach.

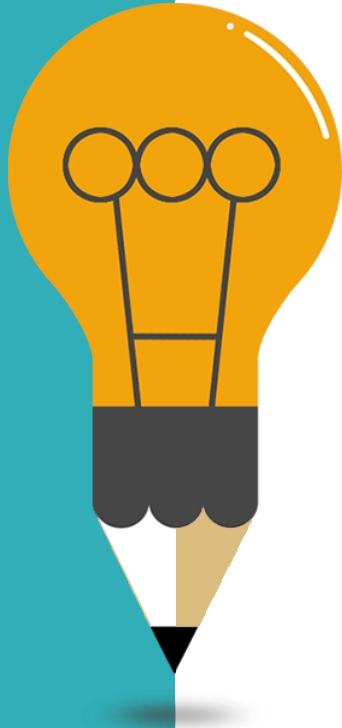


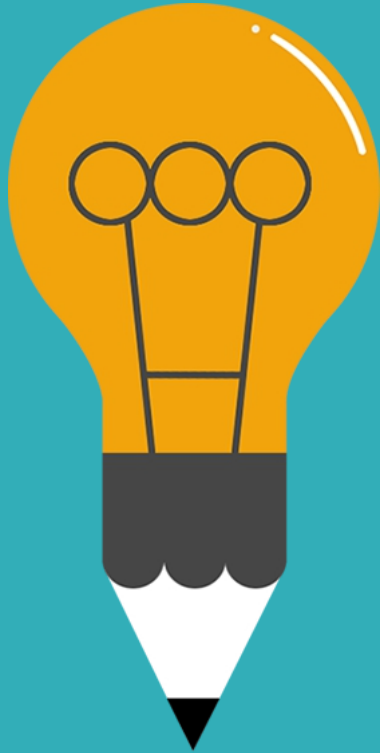
Universität Bremen: Grundlagen der Medieninformatik 2

1/3

Übungsblätter

- Abgabe Vorlage beachten!
- Erlaubte Dateien für Doku: PDF (KEIN DOC/DOCX!)
- Namen, Tutorium, Bearbeitungszeit angeben!
- Benennungsschema Beachten:
mi2_uebung#_nachname1_nachname2_nachname3
.PDF/.ZIP
- Wenn von Hand geschrieben, sauber schreiben, gute Belichtung und vernünftiges Foto, Druckschrift!





Das wars erstmal!

Bis nächste Woche!