# Backup Ressourcen

Offizielle Tutorials:

<https://docs.godotengine.org/de/4.x/about/introduction.html>

Godot Skript Referenz: <https://docs.godotengine.org/de/4.x/tutorials/scripting/gdscript/gdscript_basics.html>

# Übungsziel

Teil 1: Ein Charakter läuft per Tastatursteuerung über einen Hintergrund, per *move\_and\_slide ()*

Teil 2: Ein animierter Charakter (Laufanimation) kann sich nur in bestimmten Bereichen bewegen (auf Wegen und nicht über Häuser hinweg).

# Projekt starten

🡺 Projektname generiert automatisch ein Unterverzeichnis

Ein Bild, das Text, Software, Multimedia-Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

# Character Grafiken bereitstellen

* Teil 1: Im ‚asset‘ Ordner, siehe ‚single\_character.gif‘
* Teil 2: In Aseprite oder Libresprite können Frames aus Bilddateien ausgeschnitten und angeordnet werden (sogenannte Spritesheets), jedes Sprite kann dann per Position aufgerufen werden   
  (links oben = 0, 0 und rechts unten 3, 3) ... siehe ‚ character-sheet.png‘

Ein Bild, das Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Cartoon, Clipart, Animierter Cartoon enthält.

KI-generierte Inhalte können fehlerhaft sein.

# Szenen und Nodes erstellen

* Low code = select & confirm
* Wird eine Node grafisch erstellt, entsteht automatisch Programm Code im Hintergrund   
  (z.B. der Character Node)

Ein Bild, das Text, Schrift, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Folgende Einstellungen werden häufig in der Projektumgebung vorgenommen:

1. Fensterdimensionen ‚Projekt / Projekteinstellungen / Anzeige / Fenster / Größe‘  
   x: 1200; y:800
2. Rendering für Pixel-Grafik einstellen, unter ‚Projekt / Projekteinstellungen / Rendern / Texturen‘  
   Default\_Texture\_Filter auf ‚nearest‘
3. Hauptszene einstellen : ‚Projekt / Projekteinstellungen / Anwendung / Ausführen / Hauptszene‘

In Abschnitt 6 wird noch die projektweite ‚Gravity‘ besprochen, die gilt nur für Side-Scroller und muss daher auf Null gesetzt werden.

**Szenen**:

* ‚Character‘ erstellen (CharacterBody2D)
* ‚Games‘ erstellen (Node2D)
* Im Teil 2: kommt dann noch ‚World‘ (Node2D) hinzu

**Character Node:**

* Sprite enthält das‚single\_character.gif‘

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Games Node** kombiniert**:**

* Hintergrund (TextureRect) mit dem entsprechenden PNG aus dem ‚assets‘ Ordner und   
  Character Node

Für die Zukunft, eine mögliche Character – Node Struktur

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

# Script hinzufügen (Verwendung des Templates)

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

Automatisch generierte Beschreibung

**Warum fällt das Sprite nach unten?**🡺 check CharacterBody2D Hilfe in Godot 4 (Apple: option + space um die Hilfe aufzurufen)

* ‘gravity’ wird von PhysicsBody2D geerbt

Zwei Wege um ‚Gravity‘ zu ändern ...

Quelle <https://www.youtube.com/watch?v=trImNx6vWwY&ab_channel=NADLABS>

* global gravity change under Projekteinstellungen / Physik / 2D (für alle Objekte)
* avoid gravity in func \_process() with
  + velocity.y = 0
  + move\_and\_slide()

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

# Movement Template anpassen

Tipp: Speed = 100

* Alles was nach Plattformer aussieht auskommentieren #
* Die bestehende rechts – links Bewegung (direction\_x) erweitern auf hoch-runter erweitern (direction\_y)
* Scripte sind Erweiterungen bestehender Funktionalitäten, im vorliegenden Beispiel gibt es schon ein function ‚*move\_and\_slide ()‘,* die wir über die Koordinaten *direction.x* und *direction.y* ansteueren können

Optional: Beispiel für einen ‚randomly moving‘ NPC

Quelle: <https://youtu.be/LMSbPkNgnWA?t=904> (ersten 13 min)