

Unity2D CheatSheet

Basics

Scene View	Betrachtung & Bearbeitung der Szene / des Spiels
Game View	Betrachtung & Bearbeitung der Szene / des Spiels
Hierarchy	Alle Objekte in der aktuellen Szene hierarchisch angeordnet
Inspector	Zeigt Eigenschaften / Komponenten des aktuellen Skriptes an
Project	Beinhaltet alle Dateien deines Projekts
Console	Zeigt Fehler, Warnungen und Informationen an

GameObject	Grundlegendes Element. Alles in der Hierarchy ist ein GameObject. Erstellung: Rechtsklick in Hierarchy > Create Empty
Komponenten	Definieren Verhalten von GameObjects Werden im Inspector angezeigt, hinzugefügt oder entfernt Erstellung: Rechtsklick im Project-Tab > Create > C# Script
Sprites	2D-Bilder, welche Objekte, Charaktere, Hintergründe, etc. darstellt

Basics - Code

Awake()	Aufruf bei Skriptinitialisierung genau einmal noch vor Start()
Start()	Aufruf genau einmal bei Script Initialisierung und wenn das GameObject aktiviert ist (zu Beginn des Spiels oder wenn das GameObject aktiviert wird)
Update()	Aufruf in jedem Frame
gameObject	Kleingeschrieben ist es das GameObject an welchem das Skript dranhängt
transform	Kleingeschrieben ist es die Transform-Komponente des GameObjects an welchem das Skript dranhängt
Debug.log("Text")	Ausgabe von Text in der Unity Konsole

Attribute

[SerializeField] private int myNumber;	Serialisiert private Variablen, sodass sie im Inspector angezeigt werden, ohne dass sie public sein müssen
[HideInInspector] public int myNumber;	Verhindert, dass eine public Variable im Inspector angezeigt wird.
[Header("Settings")] public int myNumber;	Erstellt eine Überschrift im Inspector um Variablen visuell voneinander abzutrennen.
[Tooltip("This is a tooltip")] public int myNumber;	Beschreibung einer Variable, wenn man mit der Maus im Inspector über sie fährt

Transform - Position, Rotation, Skalierung

Position	Global (Get; Set;): Vector2 globalPosition2 = transform.position; transform.position = new Vector2(-3, 0.5f); Lokal (Get; Set;): Vector2 localPosition2 = transform.localPosition; transform.localPosition = new Vector2(2, 5);
Rotation	Global (Get; Set;): Vector2 globalEuler2 = transform.eulerAngles; transform.eulerAngles = new Vector2(-270, 90); Lokal (Get; Set;): Vector2 localEuler2 = transform.localEulerAngles; transform.localEulerAngles = new Vector2(180, 1.5f);
Skalierung	Global (Get;): Vector2 globalScale2 = transform.lossyScale; Lokal (Get; Set;): Vector2 localScale2 = transform.localScale; transform.localScale = new Vector2(2, 3.5f);

Klassen & Komponenten

GameObject	Objekt in Spielwelt und Container für Komponenten
Transform	repräsentiert Position, Rotation & Skalierung eines GameObjects
Vector3	Vektor im 3D Raum kann für Positionen, Richtungen & Skalierungen verwendet werden
Vector2	Vektor im 2D Raum kann für 2D Positionen, Richtungen & Skalierungen verwendet werden
Mathf	Mathematische Funktionen (z.B. sqrt()) und Konstanten
Color	Repräsentiert RGBA Farbe
Time	zeitbezogene Variablen, wie vergangene Zeit seit dem letzten Frame
Random	Generierung von Zufallswerten

Camera	stellt die Szene dem Spieler dar
SpriteRenderer	Rendert ein Sprite
Rigidbody2D	Fügt einem GameObject 2D Physik Eigenschaften hinzu
BoxCollider2D	Kollisionskomponente für 2D Kollisionen
Animator	kontrolliert Animationen eines GameObjects
Canvas	rendert UI Elemente auf einer definierten Fläche
CanvasScaler	Skaliert ein Canvas entsprechend der Displaygröße
AudioSource	spielt Audiodateien in 2D oder 3D (positionsabhängig) ab