



Unity Game Vol. 4
Mentoring SoSe 2018

Diane Hanke, Nina Matthias & Toni Wirth
Freie Universität Berlin

18. Mai 2018

Programm

Bildschirmausgabe

Freiarbeit

Programm

Bildschirmausgabe

Freiarbeit

1. Blockkurs
2. Blockkurs
3. Blockkurs
4. **Blockkurs**

Wir sind fast fertig!
Der letzte Feinschliff wartet!

- ▶ Bildschirmausgabe
- ▶ Exkurs: Blender
- ▶ Freiarbeit

Programm

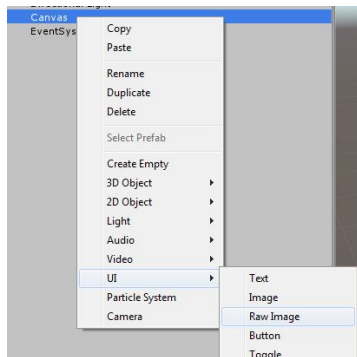
Bildschirmausgabe

Freiarbeit



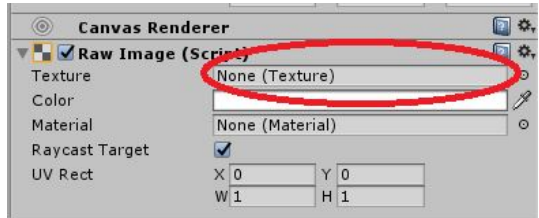
GameObject > UI > Canvas

- ▶ Wähle Bilder, die angezeigt werden sollen
- ▶ Füge die Bilder den Assets hinzu
- ▶ Erzeuge neues Canvas-Objekt
- ▶ Füge dem Canvas in der Hierarchy per Rechtsklick ein Kind-Objekt vom Typ Raw Image hinzu



Lose- & Win-Bildschirm

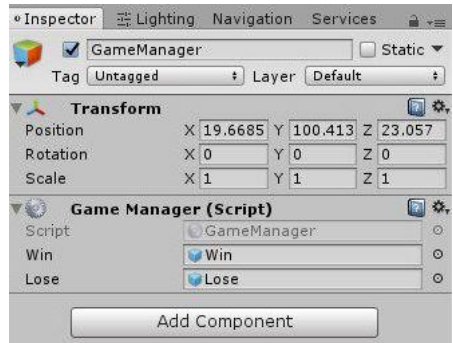
- ▶ Bild in das Texture-Fenster ziehen
- ▶ Bildgröße anpassen
- ▶ Prefab erzeugen: in den Asset-Ordner verschieben



Erzeuge auf diese Weise einen Lose- und einen Win-Canvas. Lösche abschließend die beiden Canvas aus der Hierarchy.

GameObject > Create Empty

- ▶ Empty GameObject erzeugen
- ▶ Umbenennen zu: Gamemanager
- ▶ Skript erstellen: GameManager
- ▶ Lose- und Win-Prefab in den Inspector des GameManager ziehen



Game Manager - Skript

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 //muss importiert werden; beinhaltet alle Komponenten, die zur Verwaltung
5 //der Scene benötigt werden
6 using UnityEngine.SceneManagement;
7 public class GameManager : MonoBehaviour {

9     bool hasEnded = false;
10    public GameObject win;
11    public GameObject lose;

13    public void LoseGame()
14    {
15        //Boolean hasEnded wird abgefragt, damit Spiel nur einmal beendet wird
16        if (!hasEnded)
17        {
18            //erzeugen lose-Prefab, in dessen Position, mit identischen Winkeln
19            Instantiate(lose, lose.transform.position, Quaternion.identity);
20            //Invoke startet die Funktion restart um 3 Sekunden verzögert
21            Invoke("Restart", 3f);
22        }
23    }
}
```

```
1 public void WinGame()
2 {
3     //gleiches Prinzip wie bei lose, nur wird ein win-Prefab erstellt
4     if (!hasEnded)
5     {
6         Instantiate(win, win.transform.position, Quaternion.identity);
7         Invoke("Restart", 3f);
8     }
9 }
11 void Restart()
12 {
13     //sagen dem SceneManager, dass er die aktuelle scene reloaden soll
14     SceneManager.LoadScene(SceneManager.GetActiveScene().name);
15 }
16 }
```

Anpassen von PlayerHealth Skript

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;

6 public class PlayerHealth : MonoBehaviour {

8     //aktuelles Leben
9     int currentHealth = 100;

11    public void TakeDmg(int amount)
12    {
13        //ziehen amount vom Leben ab
14        currentHealth -= amount;
15        //Fragen, ob der Spieler tot ist
16        if (currentHealth <= 0)
17        {
18            currentHealth = 0;
19            //Dianes Idee
20            Debug.Log("HaHa Kackn008...Du bist tot.");
21            //Zerstörung des Spielers
22            Destroy(this.gameObject);
23        }
24    }
}
```

Anpassen von PlayerHealth Skript

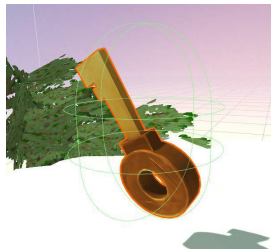
```

2 //wird zusätzlich zu Update frameweise aufgerufen
3 private void OnCollisionEnter(Collision collision)
4 {
5     //wenn das Objekt mit dem Tag Enemy kollidiert, dann rufe TakeDmg mit
6     100 Dmg auf
7     if (collision.gameObject.tag == "Enemy")
8     {
9         TakeDmg(100);
10
11         ////////////////////////////////////DAS IST NEU////////////////////////////////////
12         FindObjectOfType<GameManager>().LoseGame();
13         ////////////////////////////////////
14     }
15 }
16
17 }

```

Add Component > Physics > Capsule Collider

- ▶ Asset in der Welt platzieren
- ▶ Capsule Collider hinzufügen
- ▶ Capsule Collider an das Asset anpassen
- ▶ Skript hinzufügen



Achtung: Spieler benötigt den Tag "Player"!

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;

5 public class schluessel : MonoBehaviour {

8     private void OnCollisionEnter(Collision collision)
9     {
10         //Kam eine Collision mit dem Tag Bullet zustande?
11         if (collision.gameObject.tag == "Player")
12         {
13             //suchen unser GameObject und rufen die Funktion winGame() auf
14             FindObjectOfType<GameManager>().WinGame();
15         }
17     }
18 }
```



Nach der Pause folgt könnt Ihr euer Spiel in Freiarbeit fertigstellen.

Inhalt

Programm

Bildschirmausgabe

Freiarbeit



Ihr könnt nun selbstständig arbeiten.
Bei Fragen, fragt!