Fakultät Informatik/Mathematik

Lehrstuhl für Computergrafik

Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

COMPUTERGRAFIK II

BELEGARBEIT

Dokumentation von

Daniel Kegel (), Eric Schmidtgen (46562) und Linda Garten (44371)

Betreuender Hochschullehrer

Prof. Dr. Marco Block-Berlitz

Eingereicht am: 14. Januar 2020

INHALTSVERZEICHNIS

1 EINLEITUNG UND AUFGABENSTELLUNG

In Gruppen von bis zu drei Personen dürfen kleine Projektideen von der Konzeption bis zur Umsetzung durchgeführt werden. Die Inhalte müssen allerdings mit den Dozenten vorab abgesprochen werden. Grundvoraussetzung für die Projektideen sind Innovation und Interaktion. Es dürfen auch Game-Engines verwendet werden. Bei der Bewertung wird der Eigenanteil entsprechend berücksichtigt. Teil 2 fließt mit mindestens 50kompensieren. In jedem Fall ist die Rücksprache mit den Dozenten erforderlich.
Das Thema für die Projekte lautet in diesem Semester: SCHATTEN

Die Abgabe der Arbeit erfolgt im letzten Praktikum des Wintersemesters 2019/20.

2 VORSTELLUNG DER LÖSUNG

2.1 IDEE

Unsere Idee für die Umsetzung des Aufgabe ist es, dass wir in UNITY eine Taschenlampe durch eine Welt steuern können. Wir sollen dabei die Taschenlampe spielen und durch ein Haus "laufen"können. Unity bietet den Vorteil, dass einfache Objekte wie Lichtquellen oder grafische Primitive (Ebenen, Würfel, Kugeln) einfach im Editor erzeugt werden können. Somit sparen wir uns die Berechnung des Lichtes beziehungsweise des Schattens und wir erzielen schnell Erfolge.

2.2 UNITY ASSET STORE

Damit wir nicht viel Zeit beim modellieren verschwenden, werden wir auf bereits vorgefertigte ASSETS zurückgreifen. Für die Taschenlampe verwenden wir das freie Asset von RRFREELANCE[1].

2.3 ERSTER ÜBERBLICK

Nach ein paar Stunden im probieren mit Unity (Rigidbody, Gravity, Animator, etc.) laden wir unser Asset (siehe Abb. 1). Wir probieren erst einmal die schon fertigen Komponenten des Assetes und stellen schnell fest, dass wir mit der Physik nicht zufrieden sind.

2.4 PHYSIK DER TASCHENLAMPE/KAMERA

Um die Physik der Taschenlampe anzupassen müssen wir ein Script anlegen und etwas C# programmieren.
Am Anfang legen wir ein paar Variablen an und initialisieren diese direkt mit Werten. Wir benötigen diese später für unsere Physik.

```
1  public class FlashLight : MonoBehaviour {
2    public float horizontalSpeed = 2f;
3    public float verticalSpeed = 2f;
4    float mainSpeed = 100.0f; //regular speed
5    float shiftAdd = 250.0f; //multiplied by how long shift is held.
         Basically running
6    float maxShift = 1000.0f; //Maximum speed when holdin gshift
7    float camSens = 0.25f; //How sensitive it with mouse
8    private Vector3 lastMouse = new Vector3(255, 255, 255); //kind of
         in the middle of the screen, rather than at the top (play)
9    private float totalRun = 1.0f;
10   ... }
```

[1] RRFreelance, "Flashlight", im Unity Asset Store, https://assetstore.unity.com/packages/3d/props/electronics/flashlight-18972, abgerufen am: 14.01.2020

In der UPDATE() - Funktion lesen wir die X und Y-Werte der Mausposition aus und geben diese an die Transformation des Elementes weiter.

```
void Update() {
    float h = horizontalSpeed * Input.GetAxis("Mouse X");
  float v = verticalSpeed * Input.GetAxis("Mouse Y");
  this.transform.Rotate(v, 0, h);
  ...
}
```

Damit wir uns nun noch in der Welt mit unserer Taschenlampe bewegen können, implementieren wir uns noch ein paar Tastenkombinationen. Dafür arbeiten wir mit Vector3D.

```
private Vector3 GetBaseInput() {
  Vector3 p_Velocity = new Vector3();
  if (Input.GetKey(KeyCode.W)) {
    p_Velocity += new Vector3(0,0.03f,0);
  }
  if (Input.GetKey(KeyCode.S)) {
    p_Velocity += new Vector3(0, -0.03f,0);
  }
  if (Input.GetKey(KeyCode.A)) {
    p_Velocity += new Vector3(0.03f, 0, 0);
  }
  if (Input.GetKey(KeyCode.D)) {
    p_Velocity += new Vector3(-0.03f, 0, 0);
  }
  if (Input.GetKey(KeyCode.Y)) {
    p_Velocity += new Vector3(0, 0, -0.03f);
  }
  if (Input.GetKey(KeyCode.X)) {
    p_Velocity += new Vector3(0, 0, 0.03f);
  }
  return p_Velocity;
}
```

Die Kamera, durch die wir später schauen werden, schieben wir in den Ordner des Flashlights. Damit ist die Kamera auf die Taschenlampe fixiert und geht jede Bewegung mit.

2.5 WELT - DESERTSETHOUSE

Damit wir mit unserer Taschenlampe nicht einfach auf einer einfachen und langweiligen Oberfläche laufen, werden wir ein altes Wüstengebäude nutzen. Auch dies haben wir im Unity AssetStore gefunden[2]. Damit es für unser Projekt passt mussten wir noch einige Anpassungen in Maya vornehmen. So haben wir die Türen geöffnet, damit wir sauber mit der Taschenlampe durchlaufen können (siehe Abb. 2). Zudem wurden noch einige Wände vervollständigt (siehe Abb. 3).

[2]Crazy Cool, "Desert Set House", im Unity Asset Store, https://assetstore.unity.com/packages/3d/environments/historic/desert-set-house-138105, abgerufen am: 14.01.2020

3 ABBILDUNGSVERZEICHNIS

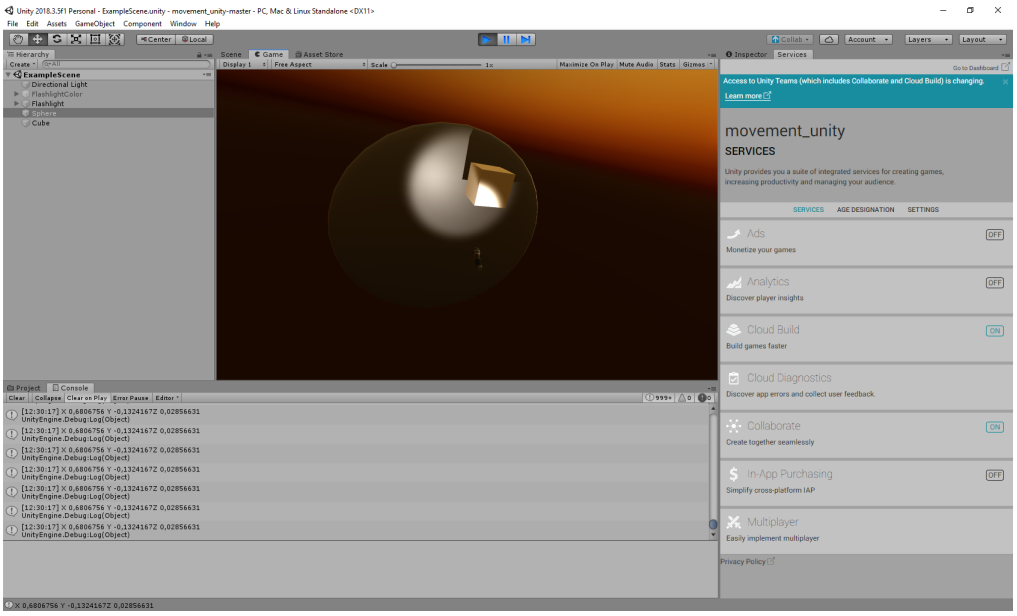

Abbildung 1: Unity UI - Erstes Laden des Assets

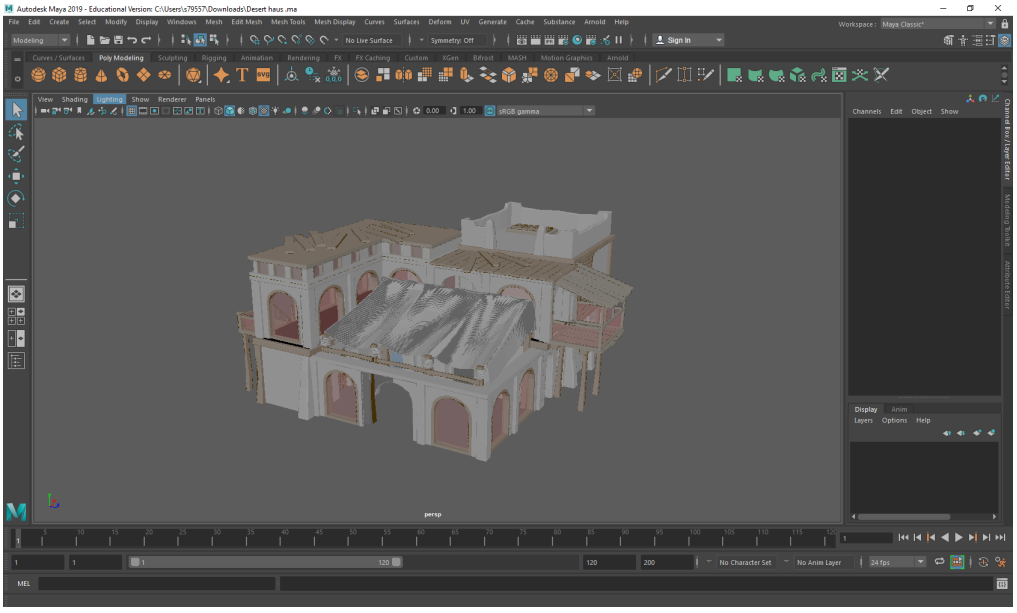

Abbildung 2: Autodesk Maya - Sicht von vorn

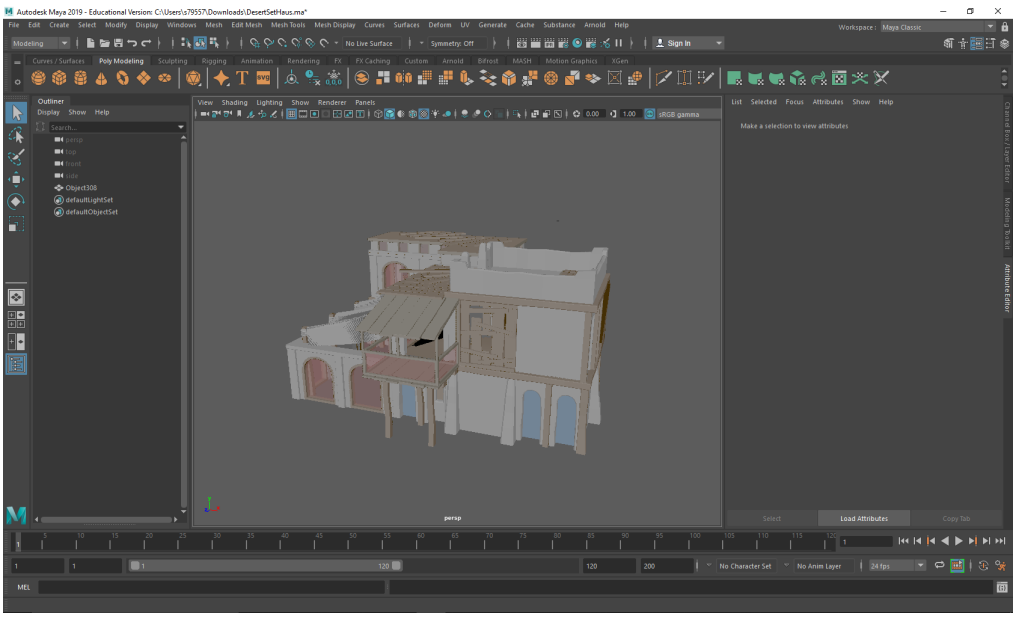

Abbildung 3: Autodesk Maya - Sicht von der Seite