Semesterprojekt WS 2017 / 2018

Katharina Ziegler, Janka Siewert, Patrick Taege

Dokumentation

# Das Projekt

Programmierung neuer UI Elemente in Blender zur Erstellung und Anpassung vorgegebener Lichtsetzungen. Mit wenigen Klicks kann ein fertiges Setup mit Lampen geladen werden, die automatisch an vorgegebenen Positionen generiert werden. Beispielhaft wurden Setups für eine Portrait- eine Packshot- und eine Grid- Lichtsituation programmiert. Die Einstellungsmöglichkeiten beziehen sich auf Helligkeit und Farbe des gesamten Setups.

# Arbeitsmittel

Programmiert wurde teilweise ausschließlich innerhalb der Blender Arbeitsumgebung, teilweise auch mit PyCharm und anschließender Übertragung des Codes in Blender.

# Klassen und Methoden

ViewLightningPanel: Definiert, unter welchen Tabs das neue Menü erreichbar ist

LampAdjustPanel / SetupSelectionPanel: Enthält Namen, ID und mehr für eines der Panel im neuen Menü. Enthält die draw() Methode, welche dem Panel Buttons und entsprechende Funktionen zuweist.

ColourOperator: Operator für den Button zur Farbwahl. Lädt bei Klick drei Slider mit Min und Max Werten, jeweils für Rot, Grün und Blau, die dann an die SetColour() Methode weitergegeben werden.

BrightnessOperator: Operator für den Button zur Einstellung der Helligkeit der Lampen. Lädt bei Klick einen Slider mit Min und Max Wert, der dann bei einer Iteration durch alle Objekte vom Typ LAMP an die SetLampStrength() übergeben wird.

SelectPortraitSetup:

SelectGritSetup:

SelectPackshotSetup:

SwitchOffAllLampsOperator: Operator für den Switch Button. Iteriert durch alle Objekte in der Szene vom Typ LAMP, speichert die aktuelle Einstellung der Helligkeit mittels der GetLampStrength() und stellt diese durch Aufruf der SetLampStrength() auf 0. Bei erneuter Betätigung des Buttons wird die Helligkeit der Lampen auf den zwischengespeicherten Wert zurückgesetzt.

SetColour: Iteriert durch die Objekte in der Szene vom Typ LAMP und setzt die übergebenen Werte für Rot, Grün und Blau.

SetLampStrength: Setzt die Helligkeit für die übergebene Lampe auf den übergebenen Wert.

GetLampStrength: Gibt den Wert der Helligkeit für die übergebene Lampe zurück.

SelectAllLampsOperator: Iteriert durch alle Objekte in der Szene und deaktiviert alle, die nicht vom Typ LAMP sind, bzw. aktiviert alle, die vom Typ LAMP sind.

# Bekannte Bugs

* Es kann auftreten, dass die Buttons zur Anpassung der Lichtsituation auf die falschen Objekte in der Szene verweisen, wenn Lampen zur Laufzeit gelöscht oder umbenannt werden.
* Obwohl der Code es nicht erlauben sollte, wird beim Betätigen des „Switch“ Buttons immer die letzte Lampe in der Szene ignoriert, was dazu führt dass alle Lampen bis auf die letzte aus gehen während die letzte eingeschaltet wird und umgekehrt.

# Schwierigkeiten

Während der Arbeit am Projekt änderten sich durch Updates seitens Blender Teile der Zugriffe im Code. Diese Änderungen waren zwar teilweise sehr nützlich, wurden allerdings nicht sichtbar dokumentiert und konnten erst nach einiger Recherche im Internet entdeckt werden.

Einige Änderungen sind:

Der Zugriff auf die Nodes von Lampen war:

1. +**def** SetLampStrength(context, lampStrength):
2. helligkeit = lampStrength
3. sce = bpy.context.scene
4. **for** object **in** sce.objects:
5. **if** object.type == "LAMP":
6. bpy.data.lamps[object.name].node\_tree.nodes["Emission"].inputs[1].default\_value = helligkeit

Und wurde angepasst in:

1. **def** SetLampStrength(object, lampStrength):
2. helligkeit = lampStrength
4. object.data.energy = helligkeit
5. **return** {'FINISHED'}

Die Values für Rot, Grün und Blau waren die Standard Werte von RGB zwischen 0 und 255

1. redValue = bpy.props.IntProperty(name="Red", description="Red Proportion", max=255, min=0)
2. greenValue = bpy.props.IntProperty(name="Green", description="Green Proportion", max=255, min=0)
3. blueValue = bpy.props.IntProperty(name="Blue", description="Blue Proportion", max=255, min=0)

Mit einem Update wurden diese abgeändert in Werte bis 1.000, was dazu führte dass die Slider in der UI bei jedem Wert über 1 die komplette Farbe übernahmen. Folgende Anpassungen im Script schafften Abhilfe:

1. redValue = bpy.props.FloatProperty(name="Red", description="Red Proportion", max=1.000, min=0)
2. greenValue = bpy.props.FloatProperty(name="Green", description="Green Proportion", max=1.000, min=0)
3. blueValue = bpy.props.FloatProperty(name="Blue", description="Blue Proportion", max=1.000, min=0)

Ähnlich mit den Values für die Helligkeit von Lampen, die vor dem Update ints zwischen 0 – 100 waren, und nach dem Update floats, die zwischen 0.001 und 10.000 liegen.

Es war außerdem zu beobachten, dass der Code bei den Projektteilnehmern zum Teil unterschiedlich funktionierte – so kam es vor, dass bei einem Teilnehmer alle Buttons ihre Funktion erfüllten, während sie bei einem Anderen zu Fehlermeldungen führten. Dies kann jedoch an dem Unter „Bekannte Bugs“ erwähnten Fehler bei der Umbenennung bzw. Entfernung von Lampen liegen.