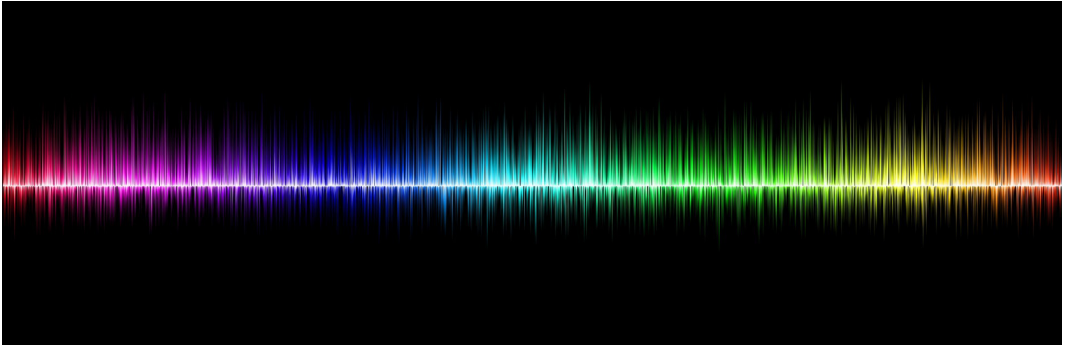


Gem Illuminator

Game Programming

HPI

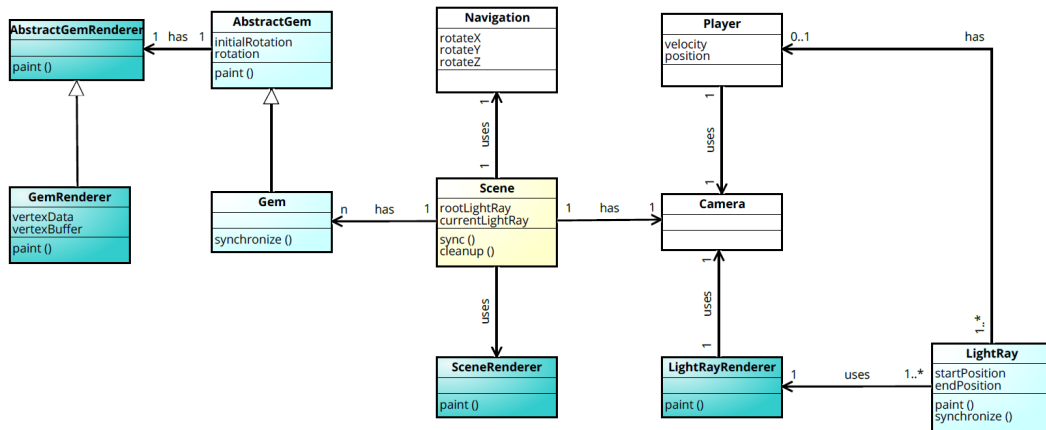


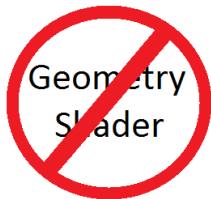
Pascal Lange, Sebastian Koall, Jennifer Stamm

Daniel Limberger Hasso-Plattner-Institut WiSe 2014/2015

Demo



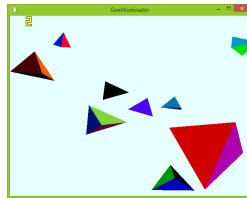




Normalenberechnung

```
onCrystalCountChanged: {  
    var gemComponent = Qt.createComponent("gem.qml");  
    var gems = GemGenerator.generateGems(60, 0.5, -10, 10)  
  
    for (var i = 0; i < gems.length; i++) {  
        console.log("New gem" + i)  
  
        scene.appendGeometry(gemComponent.createObject(  
            scene,  
            {"id": "gem" + i.toString(),  
             "position.x": gems[i][0],  
             "position.y": gems[i][1],  
             "position.z": gems[i][2],  
            }  
        ))  
    }  
}
```

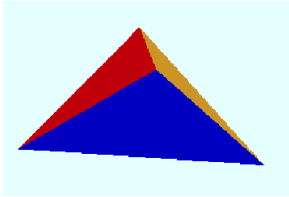
QML-Propertylists.append???



Niedrige Framerate



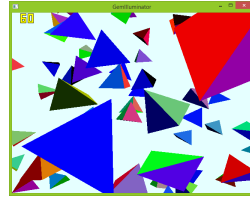
Akkufresser



Normalenberechnung

```
onCrystalCountChanged: {  
    var gemComponent = Qt.createComponent("gem.qml");  
    var gems = GemGenerator.generateGems(500, 1, -10, 10)  
  
    var gemsToJSON = []  
    gemsToJSON.push()  
  
    for (var i = 0; i < gems.length; i++) {  
        gemsToJSON.push(gemComponent.createObject(scene,  
            {"id": "gem" + i.toString(),  
             "position.x": gems[i][0],  
             "position.y": gems[i][1],  
             "position.z": gems[i][2],  
            }  
        ))  
    }  
    scene.geometries = gemsToJSON
```

QML-Propertylists.append???



Niedrige Framerate




Akkufresser

```
1  Connections {
2      target: Qt.application
3
4      onStateChanged: {
5          switch (Qt.application.state) {
6              case Qt.ApplicationActive:
7                  if(Qt.platform.os == "android") {
8                      root.showFullScreen()
9                      var types = QmlSensors.sensorTypes();
10
11                      if (types.indexOf("QRotationSensor") !== -1) {
12                          sensorInputs.rotationSensorId.active = true
13                      } else if (types.indexOf("QTiltSensor") !== -1) {
14                          sensorInputs.tiltSensorId.active = true
15                      }
16                  }
17                  else {
18                      mouseInput.enabled = true
19                  }
20
21                  if(scene !== null)
22                      scene.active = true
23                  console.log("Active")
24                  break
25              }
26          }
27      }
28  }
```

```
25     case Qt.ApplicationInactive:
26         if(scene !== null)
27             scene.active = false
28
29         if(Qt.platform.os === "android")
30             root.hide()
31
32         console.log("Inactive")
33         break
34     case Qt.ApplicationSuspended:
35         if(scene !== null)
36             scene.active = false
37         console.log("Suspended")
38         break
39     case Qt.ApplicationHidden:
40         if(scene !== null)
41             scene.active = false
42         console.log("Hidden")
43         break
44 }
45 }
46 }
```

- Ende November
Prototyp mit Landschafts-Generierung abstrakter „Kristall“-Objekte im Raum.
Idealerweise können wir bereits das zu steuernde Objekt ausfindig machen.
- Zwischenstandspräsentation
Vollständige Implementierung der Steuerung und Berechnung des übernächsten Objekts,
auf das die Strahlen treffen würden.
- Ende der Weihnachtsferien
Umstellung von Listen auf Baumstrukturen. Kollisionserkennung.
- Early Access Convention
Auswahl des Kristalls, auf den das Lichtbündel trifft. (Erster Lichteffect: Spekulare Reflektion.)

- Was passiert, wenn ein Lichtstrahl auf keinen Kristall trifft?
- Woran erkennt man den Kristall, den man steuern kann?
- Wie soll der Hintergrund aussehen?
- Wie verhalten sich mehrere Lichtstrahlen?
- Sound, wer, wie, wo, was?
- Und wer ist eigentlich diese Usability?

 gameprogramming.hpi3d.de/201314/mammut/.
Source for graphics.

 <http://www.signavio.com/de/>.
Creation tool used fo architecture diagrams.

 www.krautscape.net/index.php?page=screenshots.
Source for graphics.

 www.pixabay.com.
Source for graphics.