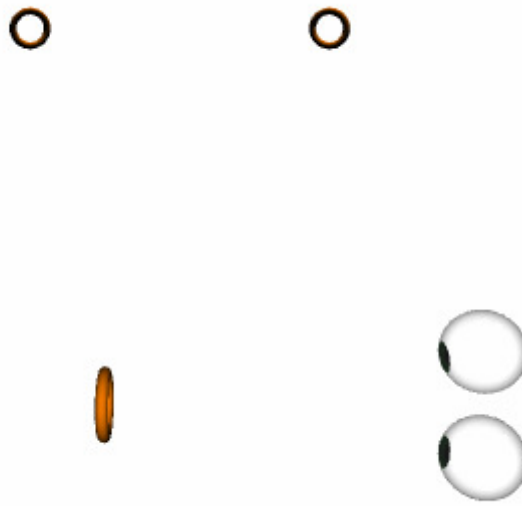


HumanVision

Hintergrund

Die Anwendung visualisiert den Prozess des menschlichen Sehens. Dabei wird deutlich: je näher ein Objekt an die Augen herangeführt wird, desto unterschiedlicher wird es von beiden Augen wahrgenommen. Das menschliche Gehirn muss also mehr Leistung erbringen, um beide Bilder kognitiv zu einer 3-dimensionalen Wahrnehmung zu überlagern. Um den Prozess des menschlichen Sehens, insbesondere der 3-dimensionalen Wahrnehmung, kinetisch nachbilden zu können, ist also eine große Rechenleistung erforderlich.



In der Anwendung werden die menschlichen Augen und das Testobjekt gezeigt. Die beiden Abbildungen im oberen Teil der Visualisierung zeigen dabei die Projektionen des Testobjektes, welche von dem rechten und dem linken Auge wahrgenommen werden.

Tastenbelegung

Die Pfeil-nach-oben- und Pfeil-nach-unten-Tasten tauschen die Demonstrationsobjekte aus.

Die Tasten H, P und R lassen das Objekt rotieren. R steht dabei für, H für Heading (Rotation um die X-Achse), P für Pitch (Rotation um die Y-Achse) und Roll (Rotation um die Z-Achse)

Mit den Tasten + und – können die Rotationen bewegt werden.

Mit der Taste G kann man die geometrischen Grundlagen ein- und ausblenden.

Um das Objekt näher an die Augen heran zu bewegen kann es mit der Maus angeklickt und abgelegt werden.

Screenshots

Anwendung

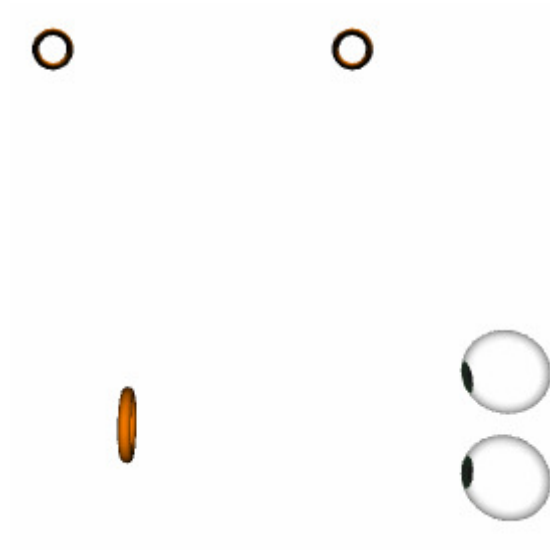


Abbildung 1: Die Visualisierung beim Start

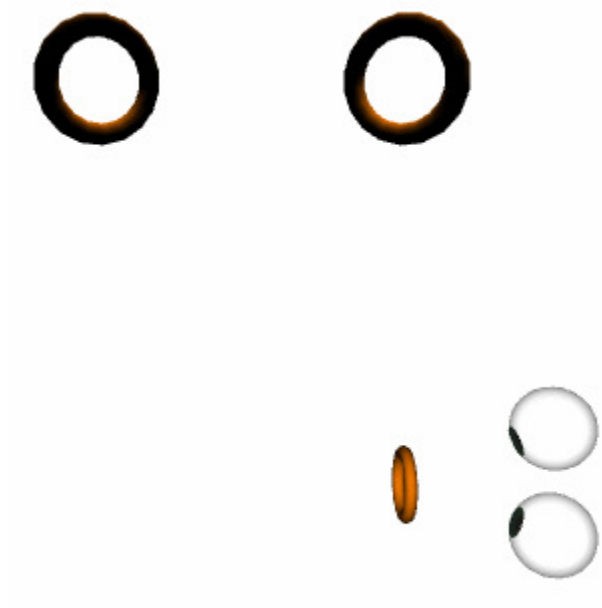


Abbildung 2: Mit der Maus kann das Objekt (auf der X-Achse) bewegt werden

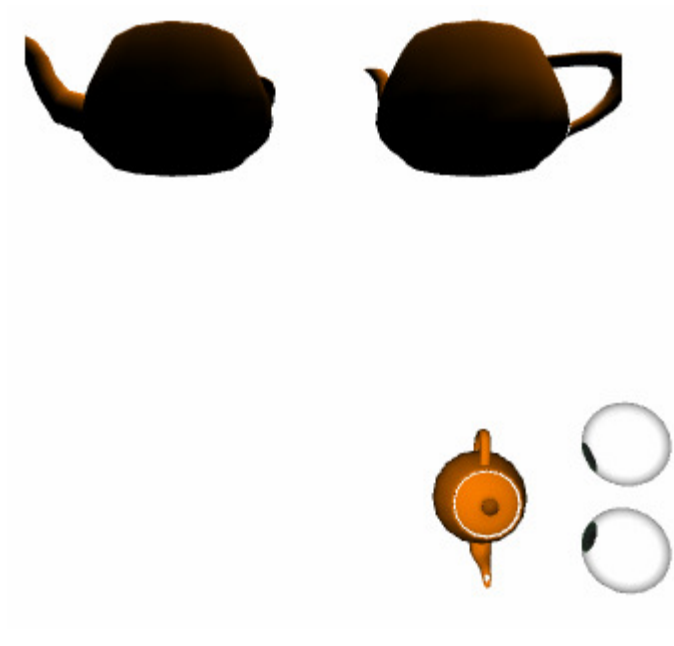


Abbildung 3: Mit den Tasten Pfeil-nach-oben und Pfeil-nach-unten kann zwischen verschiedenen Testobjekten gewechselt werden



Abbildung 4: Mit den Tasten H, P und R kann das Objekt in Rotation versetzt werden. Die Tasten + und - steuern dabei die Geschwindigkeit

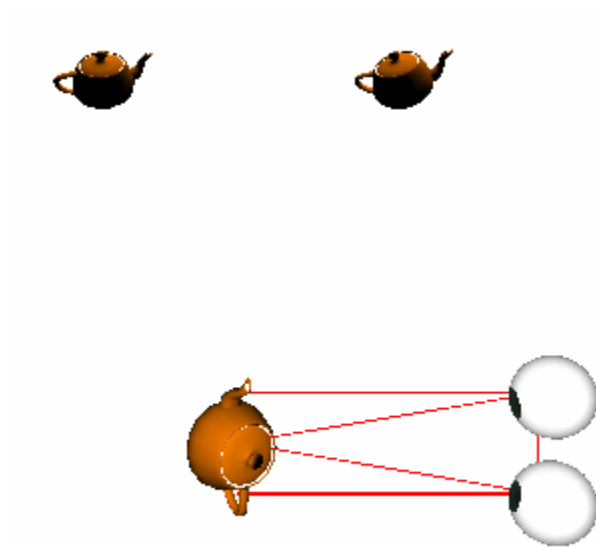


Abbildung 5: Die Taste G zeigt einfach geometrische Grundlagen an

Visualisierung

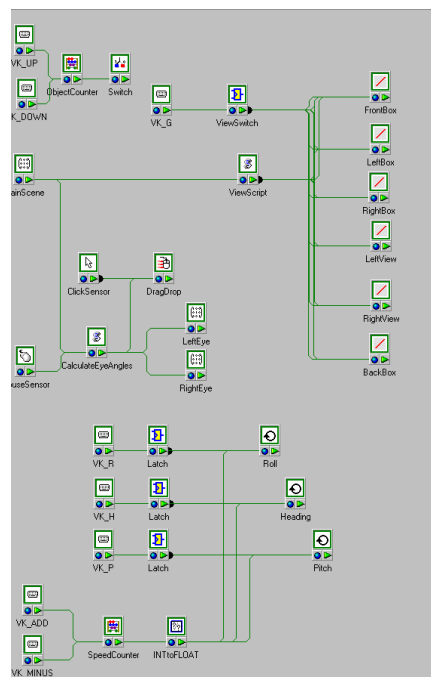


Abbildung 6: der Routes-Graph der Visualisierung

```

// Edit Scriptnode here.
// Add subroutines for each in-event or exposed field
// that you define in the script node's property page.

function initialize()
{
  // Called when simulation starts
  // Add initialization code
}

function On_ObjectPosition()
{
  //var Objects = eon.FindNode("Simulation!Scene!Objects");
  var arr = ObjectPosition.Value.toArray();

  var distance = 3-arr[0];
  var angleRight = Math.atan(1.4/distance)*180/Math.PI;
  var angleLeft = angleRight*-1;

  OrientationLeftEye.Value = eon.MakeSFVec3f(0, 0, angleLeft);
  OrientationRightEye.Value = eon.MakeSFVec3f(0, 0, angleRight);
}

function On_MouseIn() {
  var arr = MouseIn.Value.toArray();
  if (arr[0] > -1 && arr[0] < 0.5) {
    MouseOut.Value = eon.MakeSFVec2f(arr[0], 0.46);
  }
}

```

Abbildung 7: SCRIPT-Komponenten wurden mit JavaScript programmiert

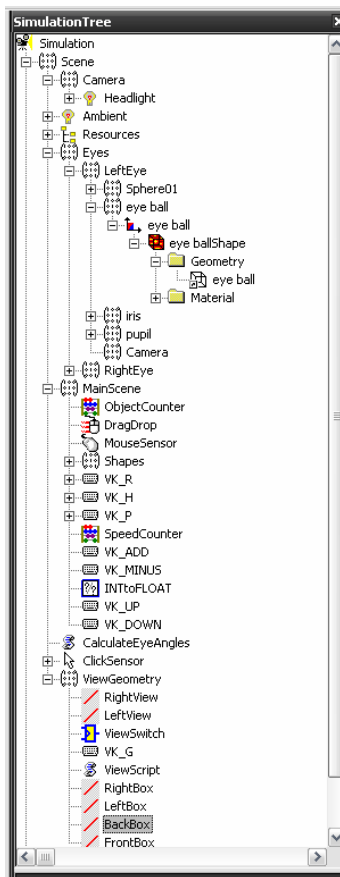


Abbildung 8: Der Simulation-Tree der Visualisierung