Planung des Vorgehens (Zeitplan, Arbeitsteilung, Milestones)

*\_* Konzeption des Spieles

*\_* Beschreibung besonderer Funktionen oder Libraries

*\_* Geeignete Testfälle für das Spiel, welche extern gespeichert werden können (pdf, doc, txt)

[CG] Aufgabenblatt S

|  |  |
| --- | --- |
| **Konzeption** | Ähnlich zu Beispielprogramm, ein Läufer in einem Tubus der Löcher im Tubus ausweichen muss. Wenn er herunterfällt verliert er ein Leben.  Wir setzen uns jeweils am Wochenende zusammen und planen die Aufgaben für die nächste Woche (Rhythmus Samstag bis Samstag). |

|  |  |  |
| --- | --- | --- |
| **Milestone I – erste Woche** | | |
| **Klasse** | **Funktion** | **Bearbeiter** |
| **Tubus** | Erstellen eines Tubus mit x Ecken und y Segmenten pro Seite, muss eine Funktion enthalten um den Tubus in Richtung Kamera zu bewegen (moveZ) | Sebastian Kriege |
| **Exampleprimitives** | Hauptklasse zum Anzeigen von Objekten, Erstellen einer Instanz des Tubus, Aufruf der moveZ Funktion im Loop |  |

|  |  |  |
| --- | --- | --- |
| **Milestone II – zweite Woche** | | |
| **Klasse** | **Funktion** | **Bearbeiter** |
| **Renderable** | Erstellen einer Basisklasse für alle Objekte, die gerendert werden müssen. Diese Klasse übernimmt viele Funktionen von Exampleprimitives. Somit kann jedes Objekt einen anderen Shader verwenden und eine eigene Textur. |  |
| **Insect (extends Renderable)** | Unser Läufer, ein Insekt mit 6 Beinen und einer Laufanimation. |  |
| **Level** | Eine Klasse um Level zu erstellen. Da der Tubus aus Segmenten besteht ist es sinnvoll als Datenstruktur eine Matrix zu verwenden, wobei der erste Index das Segment angibt und der zweite Index die Position im Segment. |  |
| **GUI (extends Renderable)** | Eine Klasse zum Erstellen von 2D-Overlays. |  |
| **Tubus (extends Renderable)** | Abändern vom Tubus, so dass er auch über Renderable verwaltet wird und die Level angezeigt werden |  |

|  |  |  |
| --- | --- | --- |
| **Milestone III – dritte Woche** | | |
| **Klasse** | **Funktion** | **Bearbeiter** |
| **Renderable** | Erweitern der Klasse Renderable um eine Funktion, um die Translation und Rotationsmatrizen zu verwenden. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Wichtigste Funktionen** | | | |
| **Klasse** | | | Insect |
| **Funktion** | **Parameter** | **Beschreibung** | |
| **createGeometry** | - | Erstellt VertexArray, textureArray und IndexArray für die Darstellung des Objektes | |
| **createLegAnimation** | int number, float scaleX, float[] root | Number ist der Index des Beines, insgesamt gibt es 6 Beine.  ScaleX ist der Abstand in X-Richtung zwischen dem Körper und dem Punkt, wo der Fuß den Boden berührt.  Root ist ein Punkt im 3-dimensionalen Raum, von der aus das Bein erstellt werden soll.  Diese Funktion erstellt eine Animationsliste für ein Bein. Das Bein selbst ist Flach und besteht nur aus 4 Punkten. | |
| **animateLeg** | int number, int step | Number ist wieder Index des Beines, Step bestimmt welcher Animationsschritt verwendet werden soll.  Die Funktion schreibt den nächsten Animationsschritt in den vertexArray. IndexArray und textureArray bleiben unangetastet. | |
| **animate** | - | Regelt die Animation aller Beine. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Klasse** | | | Tubus |
| **Funktion** | **Parameter** | **Beschreibung** | |
| **createGeometry** | - | Erstellt VertexArray, textureArray und IndexArray für die Darstellung des Objektes | |
| **createVertexArray** | - | Erstellt ein Segment (Ring) vom Tubus an Position 1f, schreibt ihn in den vertexArray, verschiebt das Segment um segmentSizeZ und schreibt es wieder in den vertexArray. Dieser Vorgang wird renderDepth-mal wiederholt. | |
| **createIndexArray** | - | Um zu überprüfen, an welchen Stellen sich ein Loch befindet ist hier eine Abfrage von curLevel (2d-Array von Booleans). Falls die nächste Position true ergibt wird diese Position übersprungen und das Viereck wird nicht gezeichnet. | |
| **moveZ** | offsetZ | Diese Funktion sorgt für die Bewegung des Tubus in z-Richtung. Falls der Tubus zu weit nach vorne bewegt werden soll wird er um segmentSizeZ nach hinten verschoben und die Variable position, welche die Position im Level darstellt wird erhöht. | |
| **turn** | boolean direction | Anstatt den kompletten Tubus zu drehen bleiben die Vertexe unangetastet. Lediglich die Variable rotation wird angepasst. Diese sorgt dafür, dass bei der Erzeugung vom IndexArray das Segment vom 2d-LevelArray um die Variable rotation verschoben wird. Kurz gesagt es werden nur die Löcher gedreht, nicht der Tubus. | |
| **isHole** | float posX, float posZ | Überprüft, ob sich an der Position posX, posZ ein Loch befindet und gibt dann true zurück. | |
| **progress** | - | Regelt den Übergang zum nächsten Level. | |