

2D Game Engine mit C++ und SFML

Studiengang: BSc in Informatik | Vertiefung: Computer Perception and Virtual Reality
Betreuer: Prof. Urs Künzler
Experte: Dr. Harald Studer (Optimo Medical AG)

Vollumfängliche Game Engine für die Erstellung von 2D Computerspielen

Zielsetzung

Das Ziel dieser Bachelorthesis ist es, eine Game Engine für 2D Computerspiele in der Programmiersprache C++ zu erstellen. Die Engine soll es ermöglichen 2D Spiele der unterschiedlichsten Kategorien und Genres für die Betriebssysteme Windows, Linux und Mac OS zu entwickeln. Das Produkt der Arbeit ist letztendlich ein Software Development Kit (SDK) in Form einer C++ Softwarebibliothek, das vorgefertigte und erweiterbare Lösungen für die vielseitigen Problemstellungen der 2D Spieleentwicklung zur Verfügung stellt.

Game Engine

Die Game Engine stellt unter anderem eine vollumfänglich simulierte physikalische Umgebung bereit, in der physikalische Körper unter Einwirkung von frei definierbaren Kräften mit einander interagieren können. Für diesen Teil der Simulation wird die performante und gut etablierte 2D Physik Engine „Chipmunk Physics“ eingesetzt, welche unter anderem im Physik basierten PC-Spiel „Reassembly“ oder der Cocos 2D Game Engine verwendet wird. Die Game Engine verwaltet die Spielobjekte in einer Scene Graph Datenstruktur, diese macht es möglich, dass sich die Objekte relativ zu einander bewegen können. Wenn zum Beispiel ein Spielcharakter einen Hut trägt, bewegt sich dieser Hut immer mit dem Spielcharakter. Mit dem Scene Graph lassen sich komplexe Gebilde aus mehreren individuell beweglichen Objekten definieren, diese Gestaltungsfreiheit erlaubt es passionierten Spieleentwickelnden ein lebendiges, immersives Spielerlebnis zu erschaffen. Für die Levels/Spielwelten im Spiel, verwendet die Game Engine eine Tilemap Datenstruktur, diese speicheroptimierte Datenstruktur erlaubt es wiederkehrende Elemente (bzw. Tiles) mehrmalig zu verwenden, sodass sie nur einmal im Speicher abgelegt werden müssen. In der Game Engine wird diese Tilemap Datenstruktur noch mit physikalischen Attributen erweitert, dadurch wird es ermöglicht in der Spielwelt auch statische

Begrenzungen (wie Mauern oder Böden) zu definieren. Für die Darstellung der visuellen Elemente auf dem Bildschirm, verwendet die Game Engine das OpenGL basierte C++ Framework SFML (Simple and Fast Multimedia Layer). Die wiederholte Darstellung der visuellen Elemente (auch: Rendering Loop) wird in einem dedizierten Thread ausgeführt, um einen konstant flüssigen Spielablauf zu gewährleisten.

Prototypen

Im Rahmen dieser Bachelorthesis wurden zusätzlich zur Game Engine selbst, zwei unterschiedliche Prototypen von unterschiedlichen Spiel Genres erstellt. Der erste Prototyp ist ein 2D Bottom-Up Arcade Shooter Spiel, in dem gegnerische nicht-Spieler-Charakter versuchen den Spieler mit Projektilen zu zerstören. Der Spieler hat selbst auch die Möglichkeit die Gegner mit eigenen Projektilen zu zerstören. Der Prototyp implementiert eine einzigartige Kamera- und Spielersteuerung und ist der bisher am weitesten entwickelte Prototyp mit einem Endgegner und Spielabschlussbedingungen. Der zweite Prototyp ist der Prototyp eines typisches Rollenspiels, in dem der Spieler in der Lage ist, sich durch ein Dorf zu bewegen (mit einer animierten Laufbewegung) und mit anderen nicht-Spieler-Charakteren einen Dialog zu führen. Ein besonderes Merkmal des zweiten Prototyps ist, dass die Spielwelt 2.5-dimensional ist: Der Spieler kann sich je nach eingenommener Position vor einem bestimmten Objekt oder dahinter befinden.



Alain Alistair Markoczy
a.markoczy@gmail.com



Game Engine Prototyp 2