

Projektskizze

1. Model

Als Projekt wird ein einfaches Rogue-Lite implementiert. Es ist geplant eine Welt zu erstellen, welche aus einem Gitter an Räumen besteht.

Die Welt besitzt eine Hashmap der Räume, wobei die x und y Verschiebung vom Startraum als Key genutzt werden. Damit können Nachbarschaft Beziehungen leicht gespeichert werden und der Abstand vom Startraum, welcher für andere Funktionen benötigt wird, leicht errechnet werden.

In jedem Raum findet man Gegner, welche über einzelne Threads gesteuert werden. Sobald alle Gegner in einem Raum besiegt sind, kann man die Benachbarten Räume betreten. Um so weiter der Raum entfernt von dem Startraum, desto schwerer werden die Gegner, dies könnte bedeuten sie halten mehr schaden aus, richten mehr schaden aus, richten häufiger schaden aus oder eine Kombination der zuvor genannten. Alle Lebende Elemente eines Raumes, werden eine Kreistrunde Hit-Box haben. Alle lebenden Elemente werden, sobald ihre HP kleiner-gleich 0 ist, vom Spielfeld entfernt und ihr Thread gestoppt. Sobald die Lebenspunkte des Spielers 0 erreichen, wird die Welt gelöscht, und bei Bedarf eine neue generiert. Im Grundgerüst wird nur 1 Gegnertyp implementiert, welcher mittels Pathfinding zu dem Spieler läuft, und sobald er in Reichweite zu attackieren ist, einen Angriff in Richtung des Spielers ausführt.

Der Spieler kann in die 4 Himmelsrichtungen, oder einer Mischung dieser laufen. Attackieren kann er in eine Richtung unabhängig von der lauf-richtung.

Angriffe von Gegnern und Spielern werden Kreiselemente als Schadens Areas generieren, welche allen in ihrer Hit-Box stehenden Mitglieder ihres Gegnerteams in einem bestimmten Intervall Schaden hinzufügen. Sobald die Lebenszeit des Elements abgelaufen ist, wird es wie jedes andere Element zerstört.

2. Mögliche Erweiterungen

Falls das gewünschte Zeilenlimit nicht erreicht ist, gibt es viele weitere Funktionen hinzufügen.

Beispielsweise zufällig generierte Hindernisse im Level um für Abwechslung in den Räumen zu Sorgen, Multiplayer über Socket-Kommunikation, Power ups, mehr Gegner Typen, wie Bogenschützen oder Magier, oder verschiedene Waffen, wie Bögen oder Zauberstäbe, für den Spieler implementieren.

3. View

Das View soll nur den aktuellen Raum des Spielers und dessen Elemente anzeigen. Die Graphische Darstellung des Spielers und der Gegner soll einfach die Kreistrunde Hit-Box des Elementes in Farben, welche leserlich kennzeichnen, ob es feind oder Freund ist, sein. Die Richtung

Falls die Hindernisse implementiert werden, sollen sie als schwächer saturierte Element auf den Bildschirm gemalt werden, um eine hohe Leserlichkeit des Spielfeldes zu garantieren.

Der Hintergrund wird als Bild eingeladen, wobei es sich umso weiter man weg vom Startraum kommt, immer weiter und dunkler rot färbt, um die erhöhte Gefahr zu symbolisieren.

4. Controller

Der Controller wird genutzt, um Austausch zwischen Model und View zu ermöglichen. Die Nutzer-Eingaben werden aber mittels des Views an den Controller geschickt.