**Dungeon Crawler** 10.06.2018

## Jan Kunert

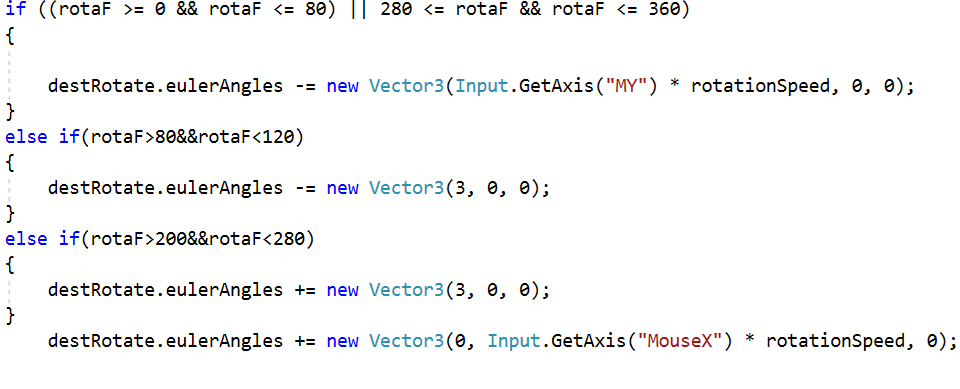
Allgemeine Zusammenfassung:

Meine Version des First-Person-Adventures „Dungeon Crawler“ zeichnet sich hauptsächlich durch eine Steuerungsweise mit der Maus anstatt mit Q und E aus. Somit wird ein präziseres Zielen ermöglicht, außerdem lassen sich neue Ecken des Dungeons entdecken. Während dieser Steuerungsmethode wird der Mauszeiger ausgeblendet und unsichtbar in der Mitte des Fensters fixiert. Um mit der Steuerung mit der Maus weiterhin auf Dinge zeigen/klicken zu können und sie somit aufheben zu können, wurde ein Fadenkreuz eingefügt, welches sich in der Mitte des Bildschirms befindet und das Zielen ermöglicht, um beispielsweise Steine aufzuheben. Jenes geschieht mit einem Linksklick auf diese, welche angesammelt versteckt in einer Ecke des Dungeons zu finden sind. Mit diesen frisch aufgesammelten Wurfgeschossen – welche sich in beliebig hoher Anzahl im Inventar stapeln lassen – kann der Spieler nun seine Gegner bekämpfen: Der Spieler ist nämlich nicht allein im Dungeon, eine Fledermaus hat es in manchen Eckes des Gewölbes auf ihn abgesehen. Diese Fledermaus fliegt zwischen zwei Wegpunkten im Dungeon hin und her, und wird auf den Spieler aufmerksam, wenn dieser sich in das Sichtfeld der Fledermaus verläuft. Nun verfolgt die Fledermaus den Spieler und greift ihn an, wenn sie ihn berührt. Dies sollte dem Spieler selbstverständlich Schaden zufügen, das funktioniert in meinem Spiel jedoch nicht ganz. Der Spieler kann mit seinen aufgesammelten Steinen die Fledermaus jedoch bekämpfen, indem er mit Rechsklicken Steine in seine Blickrichtung werfen kann. Fünf Steine hält die Fledermaus aus, bevor der Spieler diese besiegt hat.

Während der Spieler sich im Dungeon umherbewegt, kann er nicht nur mit normaler Geschwindigkeit laufen, sondern – ähnlich wie in kompetitiven Shootern oder First-Person-Spielen – mit Gedrückt-halten von Shift sprinten und mit Gedrückt-halten von Strg schleichen. Beim Sprinten wurden die Schritt-Geräusche des Spielers modifiziert, sodass sie sich schneller abspielen als beim Laufen mit normaler Geschwindigkeit, um die schnelle Fortbewegung des Helden akustisch darzustellen.

Das Spiel pausieren kann der Spieler mit dem Drücken von ESC. Das nun angezeigte Pause-Menü hebt sich mit einem sanft einblendenem Filter vom Spiel ab. Die nun durch benutzerdefinierte Schriftarten dargestellte Benutzeroberfläche – welche mit dem jetzt wiedererscheinenden Cursor bedient werden kann – verfügt rechts über einen Button, mit dem das Spiel verlassen werden kann. Beim erneutem Drücken von ESC läuft die Zeit weiter und das Spiel wird fortgesetzt – das Pause-Menü verschwindet und der Cursor ebenfalls.

Spezifisches Thema: die Steuerung des Sichtfelds mit der Maus:

****Als *destRotate*, also der Quaternion, in dessen Richtung sich die Ansicht nach jedem Frame dreht, wird die Y-Achse der Mausbewegung vom x-Wert der Blickrichtung (*eulerAngles*) abgezogen, um die Zielblickrichtung im nächsten Frame zu erstellen.

Da sich nun das Problem ergab, dass der Player bei zu starkem Blick nach oben bzw. unten sozusagen „hinten über beugte“ und kopfüber nach hinten schaute, musste ich Grenzen für die Neigung nach oben und unten erstellen. *rotaF* ist hier immer der aktuelle x-Wert der aktuellen Blickrichtung:

Er wird in jedem Frame aktualisiert. Da der Wert 90 einem Blick gerade nach unten entspricht und ich genau diesen Blick ganz nach unten vermeiden will, wird vom Ziel-Wert der Blickrichtung immer 3 abgezogen, wenn die aktuelle Blickrichtung über 80 ist. Somit wird der Blick bei der Blickbewegung nach ganz unten immer etwas hochgezogen, wodurch das „Überschlagen“ des Blickes nach unten verhindert wird.

Analog dazu wird das „Überschlagen“ obenherum damit verhindert, dass bei einer Blickrichtung von weniger als 280 etwas zur Ziel-Blickrichtung addiert wird, sodass der Wert nicht unter 280 kommt.

(Zu beachten: 0 bzw. 360 entspricht einer Blickrichtung gerade nach vorne und 270 dem Blick gerade nach unten.)

In der letzten Zeile des o.g. Codeausschnitts wird die horizontale Änderung der Blickrichtung umgesetzt. Hierbei gab es nichts Besonderes zu beachten, dem y-Wert der Zielrotation, welcher der horizontalen Bewegung entspricht wird die horizontale Maus-Achse – welche bei der Maus wiederum x-Achse heißt – zuaddiert.