Projektarbeit

Entwicklung eines 2D-Spiels mit SFML

Gabriel Gavrilas, G3C
Jan Kunzmann, G3C
Patrick Eigensatz, G3C

Vorwort

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Inhaltsverzeichnis

1 Motivation

1.1 Warum ein 2D-Spiel?

Als wir uns für ein Projektthema entscheiden mussten, haben wir uns die Entscheidung schwer gemacht. Aus 5 verschiedenen Bereichen, für die wir uns alle sehr interessierten, entschieden wir uns für die eines Computerspieles.

1.2 Warum C++ und SFML?

Um die Sache für uns attraktiv zu machen, wählten wir bewusst eine Programmiersprache, die noch nicht alle von uns beherrschten. Natürlich hätten wir genau so gut SDL oder direkt das darunterliegende OpenGL verwenden können. OpenGL schied aus, da der Aufwand bereits ein einfaches 2D-Spiel zu realisieren, schlichtweg nicht möglich gewesen wäre. SDL war unser Favorit, bis wir SFML entdeckten. Ganz im Gegensatz zu SDL schien SFML für C++ ausgerichtet zu sein. So wurden Klassen anstatt Strukturen verwendet, was den (für den Anfang komplizierten) Umgang mit Zeigern reduzierte. Ausserdem besitzen die Klassen eigene Konstruktoren, bzw. Destruktoren, was das Initialisieren oder das freigeben von Speicher überflüssig macht. Davon erhofften wir uns weniger Speicherzugriffsfehler und ein schnelleres programmieren. SFML überzeugte uns schlussendlich, als wir die in der offiziellen Dokumentation gezeigten Beispielprogramme angeschaut haben.