

# Lernprotokoll

(Format: Datum) (Alle Commits: Eintrag Lernprotokoll)

## 15.10.14

Aktueller Stand: Jan und Gabriel haben sich mit C++ vertraut gemacht und verfügen nun über erste Grundlagen. Patrick hat sich über SFML informiert und hat die beiden anderen Teammitglieder beim Lernen unterstützt. Wir sind heute zum Schluss gekommen, dass, nach dem Erlernen von C++, nun fürs Erste die Gameidee im Zentrum unseres Projektes steht. Wir haben beschlossen, dass wir bis am kommenden Freitag alle konkreten Ideen zu den bereits aufgeschriebenen Ideen mitbringen. Zudem haben wir uns den Mittwoch, den 22. Oktober als Datum festgelegt, an dem wir uns für die Spiel-Ideen entscheiden wollen. Aus unserer Sicht sind wir auf gutem Wege und haben ein gutes Gefühl.

## 22.10.14

Wir einigten uns auf eine Spielidee: Wir programmieren ein Einbrecherspiel. Wir nennen es „Robber“. Dazu haben wir bereits erste Überlegungen zu Design, Realisierung und Spiel-Dauer gemacht. Wir haben uns auf ein 2D-Spiel aus der Vogelperspektive geeinigt, bei dem man in ein Haus einbrechen muss. Es soll 3 Spielmodi haben: 1. Einen bestimmten Gegenstand klauen 2. In einer bestimmten Zeit, so viel wie möglich stehlen 3. Einen möglichst hohen Wert der gestohlenen Waren erreichen

## 29.10.14

Die Spielidee haben wir bereits verfeinert und haben uns ein erstes Ziel gesetzt: Bis nächsten Montag soll ein Bild in den Hintergrund des Spieles geladen werden. Und es soll eine Figur geben, die man über den Bildschirm bewegen kann. Ausserdem sollen auch Mauern und eine Kollisionsüberprüfung eingebaut werden. Wir haben heute die main.cpp Datei erstellt und somit offiziell mit dem Programmieren des Spiels angefangen. Dazu haben wir ein Demo-Hintergrundbild, Demo-Level und Demo-Spielertextur erstellt, um das Spiel testen zu können.

## 04.11.14

Wir haben den Projektvertrag nun vollständig zur Abgabe bereit. Das Demo-Level funktioniert bereits. Wir haben noch Probleme mit dem Definieren der Mauern. Dadurch besteht unser Spiel momentan noch aus einer Figur, die sich auf einem gezeichneten Hintergrund bewegt, ohne Hindernisse durch Mauern. Auf Jans Laptop funktioniert das Laden des Hintergrunds und der Spielerfigur nicht auf Anhieb. Wir freuen uns, dass wir unser erstes Ziel grösstenteils erreicht haben. Als nächstes wollen wir weitere Elemente in die Demo-Version einfügen.

## 08.11.14

Heute sind uns einige Errungenschaften gelungen: Zum einen haben wir die `sf::View` Klasse, die

SFML bereitstellt, erfolgreich eingesetzt. D. h. das Spiel müsste jetzt unabhängig von der Auflösung oder der Bildschirmgröße immer gleich aussehen. Schuld, dass es bis jetzt noch nicht funktioniert hat, war ein Missverständnis der SFML-Dokumentation. Dies entstand aus unserer Sicht durch eine Inkonsistenz der SFML-Funktionen. Ausserdem wurden diverse Probleme mit der Kollisionsdetektion gelöst: Es gibt nun keine "Schlupflöcher" mehr in den Mauern und man kann problemlos um die Mauern herumlaufen. Es existieren also auch keine unsichtbaren Mauern mehr. Auf Jans Laptop funktionieren die Skalierungen noch nicht wie gewünscht.

#### **09.11.14**

Gabriel hat durch langes Ausprobieren und Tüfteln unsere Bildschirmauflösungsprobleme fast überall verbannt. Auch hat er es geschafft, die Benachrichtigungsklasse im Spiel richtig einzusetzen, bzw. zum Laufen zu bringen. Jan meldet noch vereinzelt Probleme mit seinem Laptop, das Spiel scheint aber bei allen anderen Versuchs-PCs (mehr als 7) ohne Probleme zu funktionieren. Wir vermuten seine Probleme in der hohen Bildschirm-Auflösung des PCs. Wir arbeiten an einer Lösung und kümmern uns darum, wichtige Teile des Spiels (GUI, Updater, Schätze, Spielfluss) zu implementieren.

#### **12.11.14**

Wir haben uns noch einmal getroffen um das Game auf Jans PC zum Laufen zu bringen. Dabei entschieden wir auch, dass wir das Bild nach dem Spieler ausrichten, sodass der Spieler immer in der Mitte der Ansicht steht. Wir dachten, dass das Problem mit Jans Laptop dadurch vielleicht einfacher zu beheben ist. Durch das Einführen einer Zoom-Funktion können wir nun manuell die Einstellung der Größe und der Ansicht des Spielfeldes einstellen. Damit ist das erste Problem der Ansicht schon einmal behoben. Jetzt bleiben nur noch die Mauern. Beim bearbeiten der Mauern haben wir bemerkt, dass wir 2 Koordinatensysteme haben und Jans PC damit wohl Mühe hat. Dadurch konnten wir jetzt auch das beheben: Jetzt werden die Mauern auch auf Jans Laptop richtig skaliert. Das Einzige, das auf Jans Laptop noch nicht funktioniert, ist zur Zeit die Konsole, die nicht angezeigt wird. Aber wir entschieden uns, dass dieses Problem im Moment nicht weiter verfolgt wird. Wir sind sehr froh, dass wir es hinbekommen haben, dass das Game auf all unseren 3 PCs läuft. Nun fasst jeder von uns ein kleines neues Ziel. Patrick verfeinert die Ansicht unseres Games und einer zusätzlichen Demo Map. Gabriel will sich mit den Türen befassen. Jan übernimmt den Auftrag der Programmierung von Schätzen. Wir haben auch den Projektvertrag zurückerhalten. Wir werden nun die uns darin empfohlenen Verbesserungen übernehmen. Wir haben eine sehr nützliche und gut aufgebaute Kritik erhalten, so dass wir jetzt auch konkret wissen, was wir besser machen müssen und auf was wir achten müssen.

#### **19.11.14**

Durch gute Zusammenarbeit konnten wir heute den Projektvertrag, verbessert, wieder mitbringen und haben den Vertrag zusammen unterschrieben. Jan muss seine Programm-Zeile noch einmal überarbeiten, da wir uns nach kurzer Absprache entschieden haben, die Schätze auch in die Animations-Klasse zu nehmen. Diese hat Patrick kurzerhand selber geschrieben, sie dabei auch gleich für die Mauern kompatibel gemacht und die Pfeile eingefügt. Patrick hat auch aus lauter

Freude einen kleinen Cheat geschrieben, um durch Mauern gehen zu können. Dafür muss man die Konsole öffnen durch drücken der Shift Taste. Gabriel hat bereits die ersten Zeilen Code geschrieben für die Türen. Nach Absprache haben wir eine leicht verbesserte Variante gewählt, um die Türen besser zum Laufen zu bringen. Wir bleiben am Ball. Die letzten „factor“ Funktionen wurden entfernt, da es diese nicht wirklich braucht.

## 26.11.14

Wir haben bis und mit heute einen weiteren Schritt gemacht: Gabriel hat die Türen gezeichnet und nun ist es soweit, dass wir die Tür öffnen können. Heute ist es durch gemeinsame Denkarbeit sogar schon so weit, dass man die Türe öffnen und schliessen kann, wenn man auf dem Feld der Türe landet. Jetzt muss nur noch sichergestellt werden, dass man nicht mehr durch die Türen hindurch gehen kann. Jan hat die Schätze auch so weit, dass man sie auf Abfrage hin entfernen kann, wenn man auf ihnen gelandet ist. Das Bild wird auch noch skaliert und so besser dem Spiel angepasst. Wir schrieben uns auf, was es alles noch braucht, bis wir von einem kompletten Game sprechen können. Das wären sicher Möbel, Zeitlimite, Sound, das Aussenlevel (mit weniger Gewichtung) und ein Startmenü. Bis nächste Woche sollen sich alle ein paar Gedanken machen zum Problem mit den Mauern, durch die man nicht hindurchlaufen kann.

## 03.12.14

Wir arbeiten immer noch an den Türen. Diese lassen sich bereits öffnen und schliessen, doch sind geschlossene Türen immer noch passierbar. Wir haben auch gemerkt, dass das eine viel schwierigere Aufgabe ist, als zuvor angenommen. Wir suchen heute eine Lösung zu diesem Problem. Wir müssen eine Mauer an der Position der Türe mit den Dimensionen und der Rotation der Türe erstellen. Dazu wollte Gabriel noch, dass diese entsprechende Mauer erst entsteht, wenn die Türe auch wirklich geschlossen ist. Wir haben nun eine Lösung gesucht, und haben uns schlussendlich nach kurzer Absprache geeinigt: Wir greifen auf Pointer auf Funktionen zurück. Da uns allen dieses Gebiet noch nicht so bekannt ist, müssen wir uns noch einarbeiten und haben dies heute noch nicht vollendet. Voraussichtlich werden wir in einigen Tagen auch dieses Problem behoben haben. Unsere nächsten Aufgaben werden ein GUI(Patrick), Möbel(Jan), Aussenbereich des Hauses(Gabriel), Gegner und Laserschranken, sowie Ton und Lärmpegeleinbau sein.

## 06.12.14

Das von Gabriel geschilderte Problem, ist behoben. Wie in der eventgesteuerten Programmierung üblich brauchten wir ein Funktionscallback der Animationsklasse. Sprich eine festlegbare Funktion, die aufgerufen wird, wenn die Animation zu Ende ist. Da erste Versuche mit modernem C++11 (d.h. der im `<functional>` Header definierten `std::function`) fehlschlagen, nehmen wir uns die guten, altbekannten C/C++ function pointer zu Hilfe. Wir definierten mit `void (*animationEnde)(int);` einen pointer auf eine Callbackfunktion. Der Wert dieses pointers kann über die (Animations-)Methode `setOnAnimationEnde` gesetzt werden. Nach kurzen Problemen aufgrund der Typisierung, d.h. weil wir unsere Callbackfunktion innerhalb einer Klasse definierten, funktionierte es am Schluss doch. Dazu mussten wir die Funktion global innerhalb der `level.cpp`-Datei definieren.