

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 konkrete Ideen zu den bereits aufgeschriebenen Ideen mitbringen. Zudem haben wir uns 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 an 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. 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 Hintergrundes 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. Das heisst, 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 seines 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 funktionsfähig zu machen. 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 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-Ziele 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 und die Pfeile eingefügt. Patrick hat auch aus lauter Freude einen kleinen Cheat geschrieben, um durch Mauern

gehen zu können. Dafür muss man durch drücken der Shift Taste die Konsole öffnen. Gabriel hat bereits die ersten Zeilen Code für die Türen geschrieben. 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 zum Problem mit den Mauern machen, 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 Lärmpegel einbau sein.

06.12.14

Das zuvor 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.

12.12.14

Die Türen sind noch deutlich ausgebessert worden. Dazu kommt, dass Jan mit den Möbeln begonnen hat. Wir haben jetzt jedoch befunden, dass es am einfachsten sein wird, wenn wir die Möbel am Anfang gleich hinein zeichnen. Dort können wir sie dann als Mauern definieren und so unpassierbar machen. Das Spiel zeigt nun auch die Version, auf der es im Moment läuft, an. Uns ist ein kleiner Fehler aufgefallen. Bei den Schätzen existiert ein zu grosses FloatRect. Somit kann man den Schatz auch auflösen, wenn man daneben steht. Wir haben inzwischen auch eine Zoomgrenze eingefügt. Zusätzlich haben wir auch noch ein ganz simples Startmenü implementiert. Doch leider mussten wir feststellen, dass es auf Jan's PC nicht läuft. Nun suchen wir nach einer Lösung.

14.14.14

Wir haben mit der Dokumentation begonnen. Inzwischen sind die Details zu den genauen Werkzeugen, mit denen wir unser Game erstellt haben, erklärt. Wir machen uns Gedanken, wie wir es anstellen wollen, dass man die Türen auf alle vier verschiedenen Seiten öffnen kann. Wir haben die Pfeile so weit, dass man durch die Berührung der Pfeile zu einem anderen Level wechseln kann. Da dies funktioniert, haben wir das Level testDecke hinzugefügt. Jetzt ist es möglich vom Garten ins Haus einzubrechen. Damit man klar unterscheiden kann, ob man im Haus drinnen oder draussen ist, haben wir zusätzlich rote Pfeile gezeichnet. Die Idee wäre die roten Pfeile im Haus zu brauchen, um den Weg hinaus zu zeigen.

17.12.14

Nun haben wir mit den Lasern begonnen. Für die Laser fügten wir eine neue Animation hinzu. Die gezeichneten Laser sind noch nicht sehr ausgereift, aber sie erfüllen ihren Zweck. Jetzt fehlt noch die Kollisionsabfrage und den GameOver aufruf. Um das Problem mit dem Hauptmenü möglichst zu umgehen, haben wir uns entschieden, ein eigenes Hauptmenü Level zu erstellen. Wir haben zwar damit begonnen, doch es ist noch nicht ganz zu Ende. Es hat erst einen normalen Pfeil um das Level zu starten. Parallel funktionieren nun auch die Pfeile einwandfrei, so dass man auch genau dort startet, wo man es angegeben hat. Das hat bisher noch nicht immer geklappt.

23.12.14

Jetzt haben wir 2 Tage volles programmieren hinter uns. Wir haben grosse Fortschritte gemacht. Zum einen haben wir das Hauptmenü-Level fertiggestellt. Um das Spiel zu beginnen muss man nun auf die Schaltfläche "Spiel beginnen" fahren. Diese ist in Wirklichkeit ein Pfeil, den wir speziell für diesen Level gemacht haben. Beendet wird das Spiel durch eine FloatRect abfrage. Wenn man also auf Spielbeenden kommt, beendet das Spiel auch. Zum anderen kommt man mit ESC nur noch ins Hauptmenü zurück. Bei den Türen haben wir einen Pufferbereich hinzugefügt, dank dem man nun die Türe von allen Seiten öffnen kann. Wir haben auch die Laser Kollisionsabfrage fertig gemacht. Dazu haben wir ein GameOver Level erstellt. Das Problem hier war, dass man zuerst irgendwo im GameOver Level war. Es kam darauf an, wo man erwischte wurde. Dies konnten wir im Verlauf des Tages lösen. Wenn man also einen Laser berührt, fällt man in das GameOver Level. Wir haben auch die Musik von unserem Freund eingefügt. Wir haben drei verschiedene

Musikstücke von ihm erhalten. Diese werden je nachdem, ob man im Hauptmenü, im GamOver oder in einem normalen Level ist, abgespielt. Dies macht das Spiel deutlich interessanter. Natürlich muss man das Spiel auch gewinnen können. Deshalb haben wir auch ein Punktesystem eingeführt. Pro Schatz, den man sammelt, erhält man 10 Punkte. Das Ziel ist den festgelegten Mindestpunktwert zu erreichen. Wenn man den erreicht, kommt man zurück ins Hauptmenü. Um das Spiel noch zu erschweren, haben wir eine Uhr erstellt. Somit hat man für jedes Level nur eine gewisse Zeit zur Verfügung um zu bestehen. Die Uhr, sowie die Punkte werden oben links angezeigt.

26.12.14

Wenn man auf einen Schatz kam, konnte man bis jetzt mehrere Punkte holen. Doch nun haben wir eine Zeitschranke eingeführt. Erst nach einem gewissen Zeitabstand darf die Animation wieder ausgeführt werden und somit Punkte gesammelt werden. Wir haben die alten Laser durch blinkende, ausgereifte Laser ersetzt. Es ist auch möglich die Laser zu passieren, denn sie pulsieren.

03.01.15

Wir haben es nun auch geschafft, dass es den alten Stand der Level speichert. Wenn man eine Tür geöffnet hat und zurückkommt ist diese immer noch offen. Der gesamte Spielstand in jedem Level wird gespeichert. Leider kommt es noch zu Fehlern, dass es nicht alles wieder neu lädt, wenn man verliert oder neu beginnt. Nun haben wir auch Rennen ins Spiel eingeführt. Dazu haben wir eine neue Variable eingefügt, welche beim drücken der Shift-Taste die Geschwindigkeit verdoppelt. Wir haben auch einen Lärmpegel eingefügt. Dieser steigt an, wenn man eine Tür öffnet oder wenn man in eine Mauer läuft. Das Problem ist, dass es bei der Kollision mit einer Mauer jede Bewegung in der Nähe der Mauer als Kollision angesehen wird und somit laut ist. Deshalb haben wir die Mauerkollisionen aus dem Lärm entfernt. Dafür steigt der Schallpegel, wenn man Shift drückt. Dies haben wir nach kurzem so geändert, dass es immer nur dann steigt, wenn man sich auch wirklich bewegt. Parallel dazu haben wir noch eine dunkle Datei erstellt. Diese bezweckt, dass wir nicht mehr das ganze Level sehen. Es wird nur noch ein kleiner Ausschnitt gezeigt. Man kann es sich wie ein schwarzes Blatt vorstellen, mit einem Loch darin. Diese Datei wird der Figur hinzugefügt, sodass sich das Loch immer mitbewegt. Wir finden, das gibt dem ganzen einen ganz anderen Aussehen und eine grössere Spannung. Der oben erwähnte Schallpegel wird links angezeigt. Für ihn haben wir eine Animation erstellt. Wenn man im roten Bereich angekommen ist, wird eine Zufallszahl bestimmt, bei der das Spiel beendet.

17.01.15

Unsere letzten Änderungen waren nur noch kleiner Dimension. Wir haben einen weiteren Zufallsgenerator erzeugt. Dieser kann, wenn es mehrere Möglichkeiten für ein Objekt gibt, per Zufall auswählen, welches er nun einfügt. Dadurch können wir ohne es manuell zu steuern, zum Beispiel verschiedene Schätze einfügen und dasselbe Level hat nicht immer die selben Schätze. Dazu haben wir noch die Pfeile auf die von uns ursprünglich geplante Version gebracht. So führen die roten Pfeile nun tiefer ins Haus hinein und die grünen wieder hinaus. Dies wird durch ein neues Argument für die Pfeile in der Leveldatei erreicht. Momentan ist das Schreiben der Dokumentation im Zentrum unserer Anstrengungen. Darum haben wir beschlossen, das Spiel so zu belassen.