[FOLIE 1 - MAX]

Herzlich Willkommen zu unserer Abschlusspräsentation über unser Spiel „Binding of Newton“.

[FOLIE 2]

Wir werden zuerst einige Worte zu verschiedenen „Vorgängen“ sagen, wie wir das Spiel entwickelt haben. Zuerst einige grobe Punkte zur Spiele-Kategorie dann zu den Designvorgängen und anschließend zu einigen Spielmechaniken. Am Ende werden wir noch das Spiel kurz vorzeigen und spielen.

[FOLIE 3 - MAX]

Das spiel gehört zur Rouge-like Kategorie. Wer nicht weiß, was das bedeutet, hier eine Definiton von Wikipedia:

**Als Rogue-like bezeichnet man eine Untergruppe der**[**Computer-Rollenspiele**](https://de.wikipedia.org/wiki/Computer-Rollenspiel)**, die durch**[**prozedural generierte**](https://de.wikipedia.org/wiki/Prozedurale_Synthese)**Level, die fehlende Lademöglichkeit eines Speicherstandes bei Tod der Spielfigur (sog. permanent death, permanenter Tod) sowie oftmals durch kachel- oder auch textbasierte Karten und**[**rundenbasierte**](https://de.wikipedia.org/wiki/Rundenbasiertes_Strategiespiel)**Spielweise gekennzeichnet sind.**

Zusätzlich hat unser spiel noch eine Pixelgrafik und man spielt aus der Top-Down Perspektive.

[FOLIE 4 - Stajo]

Kommen wir zum Design, genauer gesagt des Sprites. Hier sieht man einige der Sprites: Die drei spielbaren Charaktere Isaac Newton, Thomas Edison und Albert Einstein. Jeder Charakter hat natürlich 4 Ansichten und für jede Ansicht eine Laufanimation mit 10 Frames. Zusätzlich hat jeder natürlich auch einen Todes Sprite, wenn man stirbt und einen Damage sprite wenn man Schaden bekommt. Auch schießt jeder dieser Charaktere ein eigens Projektil (Edison beispielsweiße einen Elektrizitätsball). Auch sieht man hier 2 Gegner: Die Maus und die Fledermaus. Alle Sprites und Animationen wurden mit dem Programm „Aseprite“ gemacht.

[FOLIE 5 - Lober]

Für jede der 3 Ebenen haben wir auch einzelne Level designed. Dies ging recht gut mit dem Programm „Tiled“ hier konnten wir einfach die einzelnen Sprites wie Steine und Feuer einfügen und per drag&drop hineinziehen. Anschließen mussten wir noch die Collisions setzen. Auch brauchten wir mehrere Layher um später im Programm beispielsweiße zwischen Türen und Steinen zu unterscheiden.

[FOLIE 6 - Tobi]

Natürlich braucht das Spiel auch noch Musik. Diese haben wir in FL Studio selbst gemacht, wie man hier sieht. Wir haben versucht ein bisschen diese „mystische“ Kelleratmosphäre auszudrücken wie sie oft in solchen Indie Games vertreten ist. Wir wollte auch keine sehr unruhige actionreiche Melodie, da dass von uns aus gesehen nicht gepasst hätte. Unserer Meinung nach ist die Musik auch gut gelungen. Mit den Soundeffekten wollten wir nicht übertreiben um wiederum das Spiel nicht zu „unruhig“ zu machen. Es gibt natürlich Schuss- und Schadensgeräusche aber beispielsweiße kein Geräusch für einen Flügelschlag der Fledermaus.

[FOLIE 7 - Goller]

Noch kurz zur Spielmechanik wobei wir danach während dem Vorzeigen noch etwas dazu sagen werden. Um die Collisions welche wir in Tiled gesetzt haben zu managen bzw. generell als Game Engine haben wir LibGDX verwendet. Um Ladezeiten zu verringern haben wir außerdem alle Sprites in einen sogenannten Texture Atlas gepackt damit wir dann im Programm schnell darauf zugreifen können. Das spiel ist auch sehr modular programmiert, wodurch es beispielsweiße kein Problem wäre, weitere Gegner oder Ebenen hinzuzufügen. Außerdem kam oft das Singleton Pattern zum Einsatz.   
Jetzt zeigen wir aber das Spiel einmal vor und wir sagen dann währenddessen noch ein paar Sachen wie wir das Spiel programmiert haben.

[GAME VORZEIGEN]

* Wie Gegner funktionieren (Pathfinding…)
* Enemy Spawnpunkte fest gesetzt