Gewerbeschule Lörrach

Projektarbeit

C. Schmidt

Zeitraum  
24.09.2024 – 08.04.2025



**Dungeon Adventure  
Escape JOSCHUA**

Vorgelegt von:

Nico Brauer & Joschua Stammherr

[BrauerNic@mail.gws-loerrach.de](mailto:BrauerNic@mail.gws-loerrach.de)  
[StammherrJos@mail.gws-loerrach.de](mailto:StammherrJos@mail.gws-loerrach.de)

Abgabedatum: 08.04.2025

**Inhaltsverzeichnis**

1. Vorwort…………………………………………………………………………... 2

2. Konzeptionierung……………………………………………………………….. 3

2.1. Spieldetails…………………………………………………………………. 3

2.2. Aufgabenteilung……………………………………………………………. 3

2.3. Programmierung…………………………………………………………… 3

2.4. Grafikdesign………………………………………………………………... 3

3. Spielgeschehen…………………………………………………………………. 4

3.1. Allgemein…………………………………………………………………… 4

3.2. Levelsystem………………………………………………………………... 4

3.3. Krüge………………………………………………………………………... 4

3.4. Darkroom…………………………………………………………………… 5

4. Komplikationen & Lösungen…………………………………………………… 5

4.1 Programmiertechnisch…………………………………………………….. 5

4.2 Graphisch…………………………………………………………………… 6

4.3 Bestehende Fehler…………………………………………………………. 6

5. Fazit………………………………………………………………………………. 7

6. Quellenverzeichnis……………………………………………………………… 8

7. Anlagen………………………………………………………………………….. 9

**Vorwort**

Die vorliegende Dokumentation entstand im Rahmen der Abschlussprüfung für die Fachhochschulreife am Berufskolleg für Informationstechnik der Gewerbeschule Lörrach.

Die gesamte Dokumentation wurde in der dritten Person verfasst, um es dem Leser zu vereinfachen, die Aufgaben und Perspektiven beider Teammitglieder zu verstehen.

Bevor die Entscheidung zwischen einer Cloud mittels eines Raspberry Pi und einem in Java programmierten Videospiel getroffen wurde, war es notwendig zwischen Einzel- oder Partnerarbeit zu wählen.  
Aufgrund von Nicos und Joschuas bisheriger Programmiererfahrung und der mangelnden Kenntnisse in Datenbank- und Serversystemen, entschieden sie sich für die Programmierung eines Spieles, entwickelt mit der objektorientierten Programmiersprache Java.  
Zu der Zusammenarbeit, anstatt der Einzelarbeit, entschieden sie sich, da sie einerseits sehr gut miteinander auskommen, aber sich auch gegenseitig zurechtweisen und unterordnen können.

Das Projekt ist ein 2D Dungeon Adventure, in dem der Spieler zwei Rätsel lösen muss. Beide Rätsel finden in eigenen Räumen statt, aus denen, durch Lösen des Rätsels, ein Signal im Hauptraum aktiviert wird. Die Grafiken dieses Spieles wurden per Pixel Art Technik selbst gezeichnet. Durch das Lösen aller Rätsel, kann das Spiel beendet werden.



**Konzeptionierung**

**Spieldetails**

Anfangs trafen sich Nico und Joschua ein bis zwei Mal pro Woche, um die Einzelheiten des Spieles auszuarbeiten. Sie schrieben vieles auf von dem einiges dann doch wieder verworfen wurde, bis ein Spielkonzept zustande kam, welches beiden gefiel. Auch außerhalb der Treffen tauschten sie sich mittels WhatsApp aus, wodurch manche Ideen umgehend verworfen oder weiterentwickelt wurden, um sie bei den Treffen zu Ende zu führen. Als die Details fertig besprochen wurden, legten sie die Aufgabenteilung fest.

**Aufgabenteilung**

Wird sich für eine Partnerarbeit entschieden, ist es gerade bei der Entwicklung eines Spieles wichtig, eine sinnvolle Aufgabenteilung festzulegen. Dies vermeidet die gegenseitige Arbeitsbehinderung, durch das Arbeiten an derselben Aufgabe. Da sich schon in der Ideenfindung herausstellte, dass Joschua sehr kreativ und Nico eher programmierbegabt ist, entschieden sie, dass Joschua sich um das Erstellen der Grafiken und Nico um das Programmieren kümmern wird.

**Programmierung**

Bevor der Programmierprozess begann, stand die Idee im Raum, mit  
UML-Klassendiagrammen zu starten, um strukturierter arbeiten zu können. Allerdings äußerte sich Nico dagegen, da er bezüglich der Attribute und Operationen eher spontan arbeite. Einige Klassendiagramme sind dennoch in den Anlagen vorzufinden.  
Um immer mit dem Fortschritt der letzten Sitzung vertraut zu sein, führte Nico ein Ergebnisprotokoll zu jeder Arbeit am Projekt, welches ebenfalls Beschwernisse und offene Fragen umfasst.

**Grafikdesign**

Joschua arbeitet mittels eines iPad und eines Apple Pencil. Er brachte sich selbst die Pixel Art Technik bei, um die Grafiken für das Spiel zu entwickeln. Parallel notierte sich auch Joschua auf welche Komplikationen er stieß.  
Um die Grafiken in das Spiel zu implementieren, sendete Joschua diese nach Fertigstellung per AirDrop (Apples Drahtlose Übertragungstechnik) an Nicos iPhone, welche Nico dann per Mail an sich selbst sendete, um sie Schlussendlich am Rechner in das Spiel zu integrieren.

**Spielgeschehen**

**Allgemein**

Ein Bild, das Schaltung enthält.

KI-generierte Inhalte können fehlerhaft sein.Klassisch für ein Dungeon Adventure, spielt sich das Geschehen in unterirdischen Höhlen ab. Der Spieler startet im Hauptraum, von dem er die Möglichkeit hat, jeden weiteren Raum zu betreten. Es existieren zwei weitere Räume, welche jeweils ein Rätsel enthalten. Ein Raum enthält die Aufgabe, sich im Dunkeln zurechtzufinden. Innerhalb dieses Raumes befinden sich zwei Schalter. Der erste lässt Stacheln, welche aus dem Boden ragen, einfahren und macht damit dem Weg zum zweiten Schalter frei. Dieser Lässt den Raum hell werden und löst das Rätsel. Im anderen Raum stellt der Spieler zwei von vier Krügen auf zwei Druckplatten. Sind beide Druckplatten aktiviert, ist das Rätsel gelöst. Allerdings wird ihm hierbei der Weg von einem NPC (=Non-Playable-Character) erschwert, der sich beständig in eine zufällige Richtung bewegt. Kommt der Spieler mit dem NPC in Berührung, wird der Spieler wieder an seine Startposition, innerhalb des Raumes, gesetzt. Der NPC wird mittels eines Bildes von Joschua dargestellt, woraus sich ebenfalls der Spieltitel ableitet. Jeder Raum kann ohne Voraussetzung betreten werden, was in einer flexiblen Rheinfolgenauswahl resultiert. Jedes abgeschlossene Rätsel lässt ein Signal im Hauptraum aufleuchten. Durch Lösen aller Rätsel wird eine Tür entriegelt, die in die Freiheit führt.

**Levelsystem**

Neben dem Bild, welches die Map zeigt, beinhaltet jedes Level ein zweidimensionales Array, dass unbegehbare Bereiche, aber auch Türen kennzeichnet. Nach jeder Spielerbewegung prüft das Programm die aktuelle Position des Spielers. Ist diese Position gleich der Position einer im Map-Array gekennzeichneten Tür, wird jeder graphische Inhalt gelöscht, ein neues Bild so wie Array hinterlegt, neue Spielerkoordinaten festgelegt und anschließend das neue Level und der Spieler wieder gezeichnet.

**Krüge**

Die Krüge im Druckplattenraum sind kein fester Bestandteil der Map. Sie sind einzelne Objekte, die über die Map gezeichnet werden und, wie beispielsweise die Spielfigur, eigene Koordinaten und IDs besitzen. Befindet sich der Spieler nun auf dem Feld eines Kruges, was ebenfalls mittels des Map-Arrays ermittelt wird, bekommt er die Möglichkeit diesen, durch betätigen der „E“ Taste, aufzuheben. Der Krug wird graphisch entfernt und die Spielfigur ändert ihr Bild, zu einer krugtragenden Figur. Wird die „E“ Taste nun erneut betätigt, wird der Krug an der Position gezeichnet, an der sich der Spieler in diesem Moment befindet und das Bild der Spielfigur wird zurückgesetzt. Auch im Map-Array wird beim Aufheben und Ablegen eines Kruges der Array-Wert an der entsprechenden Position angepasst.

**Darkroom**

Der Darkroom wurde realisiert in dem eine Schwarze Fläche, die doppelt so groß ist wie der Raum und eine mittige Aussparung hat, über den Spieler gezeichnet wird. Die Aussparung liegt hierbei über dem Spieler, sodass nur ein kleiner Radius um den Spieler sichtbar und der Rest der Map verdeckt ist. Aufgrund der Größe der schwarzen Fläche bleibt die Map auch dann vollständig bedeckt, wenn der Spieler sich am Rand des Raumes befindet.

**Komplikationen & Lösungen**

**Programmiertechnisch**

Nico hat zwar gute Java Kenntnisse und stieß daher selten auf Syntaxbezogene Hürden, jedoch mangelt es ihm noch an reeller Projekterfahrung, weshalb es ihm nicht leichtfiel, ein funktionierendes Levelsystem zu entwerfen. Zuerst sollte bei Durchschreiten einer Tür eine Variable hochgezählt und je nach Wert das nächste Level geladen werden. Da das vorliegende Projekt allerdings nicht aus klassischen Leveln, sondern Räumen besteht, die immer wieder betreten und verlassen werden können, war eine andere Lösung von Nöten. Nach zwei Wöchigem überlegen und ausprobieren, fand Nico die in 3.2 geschilderte Lösung.

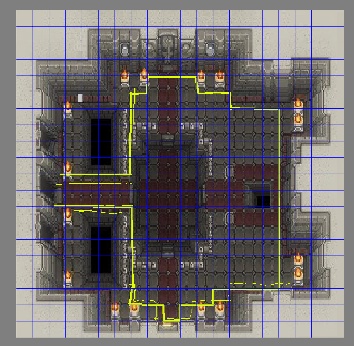
Weiter trat ein Problem mit dem NPC auf. Da eine zufällige Richtungsauswahl ebenfalls in einer nicht begehbaren Richtung resultieren konnte, zeigte der NPC gelegentlich keine Reaktion. Bevor Nico nach einer Lösung suchte, besprach er die Situation mit Joschua und sie überlegten, ob eine Lösung tatsächlich nötig sei. Nico und Joschua entschieden sich diesen Spielfehler zu beheben, also verschachtelte Nico den Code zur Richtungsauswahl in eine Kopfgesteuerte Schleife, die erst dann endet, wenn eine verfügbare Richtung gefunden wurde.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.Nun wurde im praktischen Programmierunterricht das Thema Vererbung vermittelt. Die Vererbung dient der Codereduzierung und Übersichtlichkeit, indem mehrere Klassen auf Attribute und Operationen einer sogenannten Oberklasse zugreifen können. Ein Programm kann somit zeiteffizient fertiggestellt werden. Jedoch befand sich das Projekt bereits in einem fortgeschrittenen Stadium, sodass die Vererbungstechnik, abgesehen von der Übersichtlichkeit, zunächst keinen direkten Nutzen aufgewiesen hätte. Doch aufgrund der verbesserten Übersichtlichkeit und des langfristigen Nutzens, entschied sich Nico, die Vererbung dennoch einzubinden. Hierzu wurde eine neue Oberklasse definiert, und die entsprechenden Unterklassen bestimmt. Etwa 140 Zeilen redundanter Code konnten zu diesem Zeitpunkt entfernt werden, da ihre Funktionalität nun in der Oberklasse bereitgestellt wurde.

Zuletzt verstand Nico nicht auf Anhieb, wie die Dunkelheit des Darkrooms in das Spiel implementiert werden solle. Hier schuf erneut der Austausch mit Joschua und eines weiteren Klassenmitglieds Abhilfe. Das Resultat dieses Austauschs ist in 3.4 aufzufinden.

**Graphisch**

Bei dem Erstellen der ersten Map, passte Joschua die Pixel nicht an das nötige Raster an, welches benötigt wird, um die Positionen von Türen, Krügen oder ähnlichem zu bestimmen. Auch der Spieler bewegt sich ausschließlich innerhalb des Rasters, was das Anpassen der Map noch stärker priorisiert. Dies machte die Zeichnung unbrauchbar, woraufhin Joschua sie überarbeiten musste. Dennoch arbeitete Nico zuerst mit der unfertigen Version des Raumes, um das Map-Array bereits ungefähr zu erstellen. Hierzu zeichnete Joschua gelbe Linien auf das Bild, um klarzustellen, wo sich die Grenzen des fertigen Raumes befinden werden.

Auf ein weiteres Problem stieß Joschua, als er gerade am Zeichnen einer neuen Map war, und plötzlich das Programm abstürzte. Joschuas Fortschritt wurde nicht gespeichert, wodurch er gezwungen war, erneut zu beginnen.

Allgemein ist Joschua sehr neu auf dem zeichnerischen Gebiet, weshalb für ihn alles mit Herausforderung verbunden war.

**Bestehende Fehler**

Auch nach Fertigstellung des Projekts, bleiben einige Spielfehler nicht erspart. Beispielsweise erkennt das Spiel keine Kollision zwischen Spieler und NPC, wenn der Spieler stillsteht und der NPC sich auf die Spielerposition begibt. Da dieser Fehler sehr kurzfristig entdeckt wurde, wird er zum Zeitpunkt der Abgabe dieser Dokumentation, noch im Spiel vorhanden sein. Allerdings strebt Nico an, dieses Problem, bis zur Präsentation des Spiels, gelöst zu haben. Doch das Spiel beinhaltet auch Fehler, die vorerst unlösbar sind. Ohne bestimmte Ursache werden zwischenzeitlich manche Bilder oder Texte nicht dargestellt. Dieser Fehler ist mit hoher Wahrscheinlichkeit auf die, von Herrn Post programmierte, „ZeichenFlaeche14“ zurückzuführen, die in diesem Projekt zum Anzeigen von graphischen Inhalten genutzt wird. Da die dafür verwendete Programmierung Nicos Kenntnisse teilweise übersteigt, bleibt dieser Fehler im Spiel bestehen.

**Fazit**

Trotz einiger Herausforderungen, hatten Joschua und Nico viel Spaß an dem Projekt. Von der Planung bis zur Umsetzung, war es eine spannende und lehrreiche Erfahrung, die ihnen in Zukunft sicher noch einmal nützlich sein wird.  
Nico konnte seine Programmierkenntnisse erheblich verbessern und entwickelte sein logisches denken durch das Projekt weiter. Joschua lernte die Pixel-Art Technik und sehr viel Geduld. Beide durften vor allem in Sachen Teamwork einen großen Schritt nach vorne gehen. Doch auch wenn Nico und Joschua viel lernen konnten und Spaß hatten, planen sie nicht, das Projekt weiterzuentwickeln. So gut dieses Spiel ihnen auch gefallen mag, bleibt die Wahrheit, dass es nicht mit den neusten Trendspielen mithalten kann. Jedoch war dies von Beginn an nicht das Ziel, weshalb beide damit vollkommen einverstanden sind.

**Quellenverzeichnis**

Vereinzelt wurde zwecks Formulierung und Zeichensetzung, allerdings nie für sachliche Informationen, die KI „ChatGPT“ von OpenAI verwendet. Zwischen der Verwendung der KI und der Bekanntgabe, dass der Gebrauch auch für den genannten Zweck gekennzeichnet werden muss, lagen mehrere Wochen, weshalb keine der betroffenen Passagen mehr bekannt ist. Aufgrund dessen, ist auch keiner dieser Abschnitte markiert.

Davon abgesehen wurde ausschließlich auf eine alte Dokumentation des letzten Jahrgangs namens „nPass“ von Leonardo Carvalho da Silva und die von Herrn Schmidt bereitgestellten Matrizen zurückgegriffen. Diese Hilfsmittel dienten lediglich der Struktur. Sie halfen nicht inhaltlich, doch zeigten, was benötigt wird.

Nico verwendete für die Programmierung die IDE (=Integrated Development Environment) „Intellij“ des Anbieters JetBrains, während Joschua mit der Software „Pixquare“ von Son Nguyen arbeitete.

Als informative Quelle für das Projekt an sich, waren hauptsätzlich der theoretische und praktische Programmierunterricht und die gesammelte Erfahrung während des Prozesses gefragt.

**Anlagen**

Ein Bild, das Text, Screenshot, Schrift, Dokument enthält.

KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Bild 4

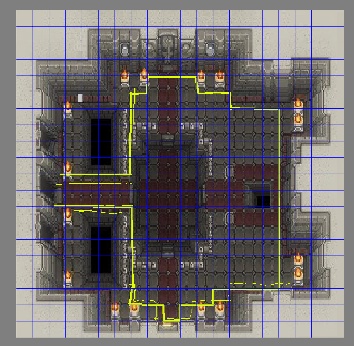


Bild 5

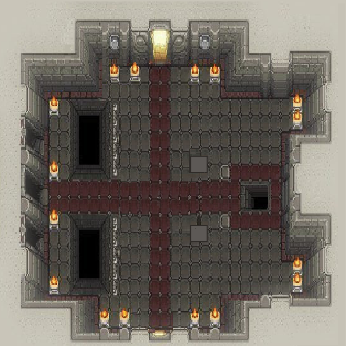


Bild 2



Bild 3



Bild 1