Entwicklungstagebuch Gruppe 7

Michel Romancuk, Severin Memmishofer, Yash Trivedi, Lukas Schmidt

FS 2023

1 Meilenstein I

1.1 23.02.23 - Alle

Heute haben wir uns vor allem mit dem Ideen sammeln beschäftigt.
1. Wir haben uns entschieden kein realtime Spiel zu machen, sondern ein rundenbasiertes Spiel.
2. Wir haben eine Ideen gesammelt: modified Monopoly, Kooperatives Karten Spiel, 2D UNO und Zork (Ein Text basiertes game)
Monopoly schien uns recht aufwendig im Bereich Game logic, da wir kein normales Monopoly machen wollten und beim kooperativen Kartenspiel war das Problem, dass es nicht competitive ist. Schlussendlich haben wir die finale Entscheidung auf Samstag gesetzt, damit jeder Zeit hat sich ein bisschen Gedanken zu machen (Pro, Kontra abwägen).

1.2 25.02.2023 - Alle

Heute haben wir uns nochmals mit der Game Idee und dem weiteren Vorgehen beschäftigt.
Wir haben uns primär darauf fokussiert, die zuvor angedachten Ideen weiter zu vertiefen und einen Favoriten auszuwählen. Nachdem wir alle Aspekte abgewogen haben, haben wir uns für den "Dungeon Crawler" entschieden.
Nachdem unsere Spielidee festgelegt war, haben wir uns alle Bewertungskriterien angeschaut und entschieden, wie wir weiter vorgehen sollen. Wir haben vereinbart, uns am Donnerstag, 02.03. erneut zu treffen und bis dahin haben wir jedem Gruppenmitglied einige Aufträge zugewiesen. Ausserdem soll jeder seinen Github-Account bis dahin vollständig eingerichtet haben und Zugriff auf sein Github-Konto haben, so dass wir schnellstmöglich mit dem Programmieren anfangen können. Zusätzlich werden wir uns noch auf eine Software zur Projektplanung einigen, an der wir alle gleichzeitig mitarbeiten können.

1.3 02.03.2023 - Alle

Wir haben uns heute vor allem mit der Präsentation und dem Projektplan beschäftigt.

Dafür haben wir einige Details für den Meilenstein 1 besprochen und finalisiert. Unter anderem haben wir uns auf den Namen für unser Spiel geeinigt: "Key H(a)unt". Ausserdem haben wir die Präsentation fertig strukturiert und alle Themenbereiche aufgeteilt. Die nächste Sitzung ist für morgen Freitag geplant. Bis dahin werden alle ihren Teil der Präsentation fertig machen und üben, damit wir einen Probedurchlauf machen können. Und wir werden noch das Analyseergebnis fertig stellen.
Ein weiterer grosser Punkt in der heutigen Sitzung war das Erstellen des Projektplans. Dabei haben wir die Zuständigkeitsbereiche und Verantwortlichkeiten untereinander aufgeteilt und auch die Abhängigkeiten der einzelnen Tasks untereinander festgelegt.
Zum Ende sind wir nochmals alle Meilensteine druchgegangen und haben geschaut, ob wir nichts vergessen haben, da es manchmal schwierig ist, den Überblick zu behalten.

1.4 03.03.2023 - Alle

Wir haben uns heute mit der Präsentation und dem Echo Sever beschäftigt.
1. Wir haben die Präsentation durchgemacht und geschaut, dass wir zeitlich drin liegen. Wir haben einige Kleinigkeiten verbessert.
2. Wir haben den Echo Server und den Echo Client wie auf den slides auf einem neuen Branch implementiert: Es gab 2 Schwierigkeiten:
-Echo Client Run configuration set up machen, indem man die korrekte main wählt.
-Bei Echo Client Run configuration noch mehrere Clients erlauben.
3. Luke hat in einem neuen Branch noch den Server gemacht.
Nächste Schritte: NetzwerkProtokoll schon mal anfangen zu implementieren.

1.5 04.03.2023 - Alle

Wir haben heute alle unsere Branches auf Git etwas aufgeräumt und "sauber gemacht". Ausserdem haben wir noch IntelliJ auf Git eingerichtet und unser Netzwerkprotokoll etwas genauer definiert. Die Nächsten Schritte sind die Server-Client Kommunikation und ein funktionierender GlobalChat.

1.6 11.03.2023 - Lukas, Severin; später noch Yash und Michel

Wir haben heute noch einmal von Grund auf die Client- und Serverstruktur überarbeitet und haben eine funktionierende Version, die das Netzwerkprotokoll benutzt.
Ausserdem haben wir uns einige Überlegungen gemacht zum Error-Handling, und zwar möchten wir für Errors ebenfalls einen Netzwerkcommand benutzen.
Zum Schluss haben wir den Code noch etwas aufgeräumt, JavaDoc Kommentare hinzugefügt und auf Git einige, nun unnötige, experimentelle Branches gelöscht.

2 Meilenstein II

2.1 14.03.2023 - Alle

Wir haben eine einfache Playerklasse mit wichtigen Methoden implementiert. Danach haben wir die Hierarchie dementsprechend angepasst und eine "Change Nickname" Methode implementiert.

2.2 15.03.2023 - Yash

Ich habe angefangen Whisper zu implementieren. Es hat nicht ganz so funktioniert wie es sein sollte, falls der Nickname, an dem man es schicken will, nicht existiert, sollte der Server eine Error Message schicken über dem Error-handling vom Netzwerk Protokoll. Dies hat es nicht gemacht und ist einfach abgestürzt. Wir müssen das später nochmal klären.

2.3 16.03.2023 - Alle

Wir haben in dieser Woche vor allem an der Implementation des Ping/Pong gearbeitet. Wir haben dazu einige Überlegungen angestellt, wie wir es am besten implementieren sollten. Dabei haben wir uns dafür entschieden, dass es am meisten Sinn machen würde, wenn der Client den Server anpingt, und jeweils eine Pong-Antwort zurückbekommt. Wenn diese Antwort (zum Beispiel 2 oder 3 mal) ausbleibt, dann sollte sich der Client disconnecten.
Etwas später haben wir gesehen, dass die Anforderung ist, dass sowohl die Server, als auch die Client-Seite einen eigenen Pinger haben muss. Da wir in der Zwischenzeit jedoch den Client-Pinger implementieren konnten, war es eigentlich ein praktisch analoges Vorgehen für die Server-Seite. Wir erstellen dabei jeweils einen neuen Ping-Thread, der nur fürs "Pingen" und die Verarbeitung der Pings zuständig ist. Das ist insofern von Vorteil, da dieser Prozess in einer Dauerschleife ist, und somit keine anderen Prozesse aufhält, wenn er seinen eigenen Thread bekommt.
Nun bekommen sowohl der Server, als auch der Client jeweils eine Pong-Message in regelmässigen Abständen von 15s. Nun müssen wir nur noch schauen, wie wir im Falle eines ausbleibenden Pongs am besten die Verbindung schliessen können.

2.4 17.03.2023 - Lukas

Im Rahmen der Ping-Pong Funktion habe ich etwas Nachforschung betrieben, wie Verbindungen elegant geschlossen werden können und implementiert.

2.5 19.03.2023 - Michel

Ich habe in der "Change Nickname" Funktion ein Bug gefunden und behoben. Es ist jetzt nicht möglich den gleichen Namen wie ein anderer Spieler zu haben.

Die Spieler ändern jetzt automatisch ihren Namen in den System-namen. Ausserdem habe ich noch eine Funktion hinzugefügt, die einzigartige Benutzernamen garantiert.

2.6 21.03.2023 - Lukas

Ich habe in der von Michel implementierten Nickname-Wechsel Funktion einige Änderungen vorgenommen, um Konflikte mit dem Decoding unseres Protokolls zu vermeiden. Ausserdem wurde ein einfaches Interface für den Client hinzugefügt, um die Benutzung eleganter zu machen. Des Weiteren habe ich in der Netzwerk-Struktur einige Klassen umstrukturiert und Superklassen extrahiert um unseren Code sauber und ohne viel Duplikation zu halten.

2.7 22.03.2023 - Alle

Wir haben unseren main Branch updated, indem wir 2 verschiedene Remote Branches gemerged haben. Es gab eine Exception, die wir noch korrekt behandeln mussten. Wir haben eine Liste von noch unerfüllten Tasks gemacht, die noch vor dem 2. Meilenstein erledigt sein müssen. Wir haben weiteres Vorgehen besprochen.

2.8 24.03.2023 - Alle

Wir haben heute unser Protokoll nochmals überarbeitet. Und zwar wollten wir mit einem anderen Trennzeichen arbeiten, als mit "whitespace". Wir haben uns für Tilde "∼" entschieden. Dafür waren einige Anpassungen an unserem Command-Interpreter nötig. Ausserdem haben wir unser Interface noch ergänzt mit einigen weiteren Optionen; unter anderem einer Disconnect/ Logout-Methode, die jetzt keine Fehlermeldung mehr wirft und einer "Ping" Methode, die manuell einen Ping unabhängig der automatischen Pings schickt. Zur Unterscheidung wird dann auch in einem solchen Falle "Pong Manual" ausgegeben, um sie von den normalen Ping/Pongs unterscheiden zu können.

2.9 24.03.2023 - Lukas, Yash

Wir haben die Anpassungen von heute gemergt. Checkstyle durchgeführt, "Code Duplicates" entfernt und die getParameter Methode angepasst (später müssen wir bei dieser Methode aufpassen, falls ∼ im nickname vorhanden ist).

2.10 26.03.2023 - Alle

Wir haben uns heute alle getroffen, um alles zu überprüfen und noch einige Dinge zu verbessern. Zuerst haben wir unser QA überarbeitet, danach unser Protokoll und schlussendlich alles zu PDF konvertiert und auf git hochgeladen.

3 Meilenstein III

3.1 29.03.2023 - Lukas

Ich habe das Grundgerüst für die Lobbies implementiert, zusammen mit dem Lobby-Chat. Noch fehlen auf Client-Seite die Aktionen um diese zu benutzen.

3.2 30.03.2023 - Severin, Yash

Wir haben heute lange versucht, einen Bug in der von Lukas angefangenen Lobby-Struktur zu finden, bei dem mehrere Spieler in dieselbe, bezw. die falsche Lobby gejoint waren. Wir konnten uns das Verhalten nicht erklären, und wir haben viel Zeit mit Debugging verbracht, da wir der Überzeugung waren, dass alle Methoden richtig implementiert waren. Schlussendlich hat Lukas den Fehler gefunden, und zwar war die player-Liste in der Lobby als static deklariert, und deswegen hat nicht jede Lobby eine neue PlayerListe zugewiesen bekommen. Danach haben alle Methoden wie gewünscht funktioniert.

3.3 31.03.2023 - Severin

Ich habe heute einen Entwurf der Klassenstruktur für die GameLogic gemacht. Zuvor habe ich mit Yash und Lukas noch besprochen, welche Struktur sich am Besten dafür eignen könnte. Nach meinem Entwurf habe ich nach Rücksprache mit Lukas noch eine Änderung vorgenommen, und zwar habe ich die Grid-Klasse neu als abstract definiert, damit wir später einfacher grössere Grids/ andere Spielfelder als 5x5 implementieren können. Dies hat auch die hierarchisch höher gelegenen Klassen deutlich vereinfacht, da nicht immer die Grösse als Parameter mitgegeben werden muss. Alles in allem ist die Implementation so deutlich "cleaner".

3.4 31.03.2023 - Yash

Ich habe leave und close Lobby implementiert, es gab einige errors bei close lobby, die wir noch nicht behandelt haben. Ich habe noch einige Bugs im Interface entdeckt, die wir auch behandeln müssen.

3.5 31.03.2023 - Michel

Ich habe angefangen mit der implementation des GUI's. Der Client startet jetzt eine Applikation, sobald er mit dem Server verbunden ist. Dazu wurde eine FXML Datei im SceneBuilder erstellt, die den Chat-Layout representiert. Es gibt wegen kleinen Bugs noch keine Funktionalität. Es gibt schwierigkeiten beim durchreichen von Protocol-Befehlen in den Button-controller.

3.6 01.04.2023 - Severin, Yash

Wir haben heute die von Yash vorher angefangenen leaveLobby und closeLobby Methoden fertig implementiert. Dabei sind wir auf einige Bugs gestossen, wie zum Beispiel, dass wir aus Versehen denselben Protokollbefehl für CreateLobby wie für CloseLobby verwenden. Ausserdem haben wir noch einige Checks und Tests durchgeführt, die vor falschen User-Eingaben schützen, damit der Server bei falscher Benutzung keine Fehler erzeugt. Schlussendlich funktionieren die beiden Methoden nun wie gewünscht. Im späteren Verlauf des Tages hat Severin noch eine 5x5 Map, die Yash vorher entworfen hat und von der Gruppe als gut befunden wurde, implementiert, indem die Türen jeweils für jeden Raum fix "hardcoded" wurden.

3.7 03.04.2023 - Michel, Severin

Wir haben heute einige Bug-Fixes am Gui gemacht, und probiert, das GUI mit unserem Netzwerprotokoll zu verknüpfen. Dabei hatten wir das Problem, dass wir Schwierigkeiten hatten, den ClientSocketThread, mit dem wir die Nachrichten an den Server schicken beim Starten der Chat-App als Parameter mitzugeben. Schlussendlich haben wir eine Zwischenlösung gefunden, bei der wir den Thread als static in der GUI-Klasse definieren, und danach darauf in der start-Methode der Chat-App zugreifen. Vielleicht finden wir später noch eine schönere Lösung, aber vorerst funktioniert es. Nächste Schritte sind jetzt, die Chat-Funktion mit dem Fenster zu implementieren, sodass ein im Fenster geschriebener Text direkt geschickt wird, und ein ankommender Text im Fenster ausgegeben wird.

3.8 06.04.2023 - Michel, Severin, später Lukas

Da wir zu Ostern eine längere Session eingeplant haben, haben wir beschlossen, uns aufzuteilen. Wir haben am GUI gearbeitet, während Yash und Lukas an der Spiellogik weitergemacht haben. Wir haben uns dabei zuerst Gedanken gemacht, wie wir bei der GUI Entwicklung, am Besten fortfahren würden. Da wir das Gefühl haben, dass der schwierigste Teil darin besteht, zuerst die Basic Funktionen zum Laufen zu bringen; da wenn man das System einmal begriffen hat, der Rest vor allem noch Fleissarbeit ist, haben wir uns entschlossen, uns nochmals weiter aufzuteilen. Michel hat sich damit befasst, zu implementieren, wie wir Input im GUI, z.B. eine Chatnachricht in einem Textfeld an den Server schicken können, während Severin sich damit befasst hat, wie wir Input vom Server (der auf dem Clientthread ankommt), ans GUI weiterleiten/ printen können. Michel war erfolgreich, aber es bestehen noch einige Schwierigkeiten, die Nachrichten vom Server am GUI anzeigen zu lassen. Evtl. müssen wir noch einige Anpassungen and er Klassenstruktur vornehmen dafür.

Am Abend haben Lukas und Severin nochmals probiert den Fehler zu finden. Durch einiges Ausprobieren haben wir die Fehlerquelle einschränken können und

schlussendlich auch gefunden. Das Problem war, dass wir die Chatnachricht an den "StageController" schicken wollten, obwohl dieser zu dem Zeitpunkt noch gar nicht instanziert war, da dieser erst implizit aufgerufen und kreiert wird, wenn wir von der WelcomeStage zur LoungeStage wechseln. Da wir aber bereits vorher eine Chatnachricht senden, und zwar die Bestätigung des Nicknamenwechsels, der bei uns am Anfang direkt automatisch gemacht wird (wird ev. später noch geändert, da man den Nickname als Launch-Argument mitgeben können soll...), und wir nicht differenziert haben auf Client-Seite, wurden einfach alle Server-Nachrichten ins System Out geschrieben. Als wir dann versucht haben, diese ins GUI zu schreiben gab es Probleme, weil die entsprechende Klasse mit der ScrollPane noch gar nicht kreiert war zu dem Zeitpunkt.

Das Problem haben wir jetzt so gelöst, dass die Nachrichten vom Server in einer ersten Phase ignoriert werden, was aber nicht weiter schlimm ist, da wir zu dem Zeitpunkt sowieso noch in der WelcomeStage sind, und die Nachrichten deswegen noch nicht relevant sind für uns. So laufen wir auch zukünftig nicht in Probleme deswegen. Ausserdem haben wir noch überlegt, dass wir die Klassenstruktur etwas anpassen können, aber vorerst können wir nun mit der Implementation des GUI wie gewünscht weiterfahren, da wir jetzt kein unerklärliches Verhalten mehr in diesem Bereich haben sollten. Ideal wäre ein Starten und Laden aller Szenen in derselben Klasse, sodass wir nachher nur noch die Root umhängen müssen.

3.9 06.04.2023 - Luke, Yash

Unsere Teilgruppe war zuständig für die Game-Logik Implementation. Wir haben am Morgen angefangen einige Schritte festzulegen wie wir die Implementation machen wollen.

1. Bevor wir beginnen haben wir die Dependences festgelegt, welche Klassen wo in der Hirachie liegen. Zum Beispiel waren wir nicht sicher ob die "Position" Klasse ein Link zum Player haben soll oder umgekehrt. Um circular Dependencies zu vermeiden haben wir ein Link von der Position zum Player. Wir haben auch alle Schritte von der Game-Logik besprochen, wie zum Beispiel was für ein Befehl vom client an dem Server kommt, was der Server damit machen muss etc.

2. Wir haben angefangen zuerst die Server Seite zu implementieren, da dort die grosse Arbeit geschieht. Wir haben nur ein Befehl vom Netzwerk Protokoll, welches vom Client am Server kommt nämlich GAMEACTION, welches als String Argument alle Befehle bezgl. Game beinhaltet. Dieses String Argument wird dann in einer neuen Klasse GameAction sauber decoded und ausgeführt. Diese separate Klasse haben wir gemacht um eine enorm lange Klasse im Server Task oder in einer der Game Klasse zu vermeiden. Wir haben noch einige Überlegungen gemacht, was wir genau

im Game für den Anfang mal haben möchten und haben einige Dinge so implementiert, dass sie später gut erweiterbar wären.

3. Die Client Implementation war nicht gross, im Vergleich zum Server, aber es gab hier einige Fallen. Das erste Problem war unser Reader war im Client (Interface) implementiert, aber wir brauchten ein Reader im Client Task, wir haben um das zu lösen einige Dinge probiert, wie z.B den BufferedReader in der "Client Action" Klasse reinzutun, aber dies gab immer noch Probleme, wie z.B der Input musste 2 Mal eingegeben werden, da es beim ersten Mal in der Client Klasse feststeckte. Das Problem haben wir danach so gelöst, dass wir im Interface danach den Befehl selber geschickt haben.

Nach 2 Stunden debuggen haben wir kleine Fehler behoben und das Programm lief wie gewollt, alle Spieler können sich bewegen und theoretisch schon "Loot" aufsammeln. Es gibt jetzt noch 2 wichtige Implementation zu Game-Logik.

1. Jeder Client wird seinen eigenen Timer haben, somit muss ein Timer Thread auf der Client Seite implementiert werden. Nach Ablaufen des Timers soll der Zug abgebrochen werden, der Spieler passt.

2. Chest und Escape Room (gewinnen) müssen noch implementiert werden.

3. Gefangen werden vom Geist

3.10 08.04.2023 - Lukas

Ich habe den Spielzug des Geists implementiert, der dem Geist Hinweise über den nächsten Player gibt. Dieser nimmt die Stelle des "Looten" ein, sodass auch der Geist einen Zug hier verwenden muss.

3.11 10.04.2023 - Yash

Ich habe wie besprochen den Timer implementiert, aber es gab einige Bugs mit dem interface, die noch beheben werden müssen.

3.12 11.04.2023 - Michel, Severin

In der Zwischenzeit haben wir das Basic Chat GUI fertig implementiert und auch mit einigen weiterführenden Features begonnen, die wir nun jedoch pausieren, da diese für Meilenstein 3 noch nicht relevant sind. Unter anderem hatten wir noch etwas Probleme bei der Ausgabe von Chat-Nachrichten auf dem Display, was sich aber schlussendlich auch gelöst hat. Als wir mal verstanden haben, wie die Arbeit zwischen GUI und Code funktioniert, und wie man Infos vom GUI-Fenster in den Code und umgekehrt hin und herschickt, ist alles deutlich einfacher geworden. Bei der Implementation der Lobbyliste ist uns zum Beispiel aufgefallen, dass noch ein Bug in unserem Code existierte, dass wir zwei gleiche

LobbyNames haben konnten, aber es war trotzdem eine neue Lobby-Instanz, was zu einigen Verwirrungen geführt hatte. Ausserdem waren wir für einige GUI-Features noch auf Features der Game-Logik angewiesen, die zu diesem Punkt noch nicht 100% fertig gestellt waren. Aber insgesamt würden wir sagen, dass das GUI auf einem guten Weg ist.
Auf Severins Laptop gab es noch komische Info-Meldungen, die gedruckt wurden, sobald man das GUI auf Vollbild gewechselt hat, aber es gibt keine Exceptions und es läuft alles normal weiter. Da das Problem aber auf anderen Macs nicht auftritt, scheint es ein Problem der lokalen Installation zu sein, und es ist jetzt auch nicht weiter tragisch. Nun werden wir bei der Game-Logik und einigen kleineren Features weiterhelfen und mit der Dokumentation beginnen; die weitere GUI-Entwicklung kann vorerst warten, bis wir alle Achievements von Meilenstein 3 erfüllt haben.

3.13 12.04.2023 - Yash, Lukas

Wir haben die Chests für die "Hunter" implementiert und die Events, die je nach Spielablauf das Ende triggern und mitteilen. Dabei haben wir noch einige kritische Bugs mit dem Timer und der restlichen Logik gefunden und behoben.

3.14 13.04.2023 - Alle

Wir haben uns alle getroffen und geschaut was schon gemacht worden ist und was noch gemacht werden muss. GUI Team(Michel und Severin) und Game-Logik Team(Luke, Yash) haben ihr stand updated. Bei GUI gab es einige unerklärliche warnings und Game Logik war nicht ganz fertig. Wir haben priorities gemacht um was zuerst erledigt werden muss und die Tasks weiter aufgeteilt.

3.15 13.04.2023 - Yash, Lukas, Sevi

Wir haben zuerst Bugs im GUI behoben, danach hat Severin mit dem GUI weiter gemacht. Luke und Yash haben die Game Logik abgeschlossen und einige Bugs behoben.

3.16 14.04.2023 - Alle

Wir sind zusammengesessen um die gemeinsam Bugs zu finden und das Projekt für die Abgabe von Meilenstein III vorzubereiten. Es wurden alle nötigen Logger hinzugefügt und Javadocs geschrieben. Der Game-logic branch und der GUI-branch wurden in den MAIN-branch gemerged.

3.17 14.04.2023 - Luke, Yash

Bugs behoben, die im Zusammenhang mit Lobby und Game auftraten. Close Lobby funktioniert jetzt richtig und das Spiel wird jetzt beendet falls der Hunter gewonnen hat. Einige Logging statements hinzugefügt in der Game package.

3.18 14.04.2023 - alle

Wir haben uns alle getroffen und haben bugs fixes abgecheckt. Einige Probleme haben wir entdeckt und dann diese gelöst. In der Game Logik gab es einen kleinen Bug und auch in der GUI. Diese haben wir gefixt. Luke und Yash haben in der Zeit als der GUI bug gefixt wurde an der Bedienungsanleitung gearbeitet. Im nachhinein wurden weitere Bugs gefunden, welche Luke und Yash am gleichen Abend behoben haben.

3.19 15.04.2023 - alle

Letzte Schliffe wurden am Code vorgenommen. Wir haben die QA-Messungen durchgeführt, den Projektplan geupdated und alle nötigen Dokumente für die Abgabe geschrieben. Zu letzt, haben wir unsere Präsentation vorbereitet.