

Skip Bo

28. Februar:

Entscheid für das Spiel: Skip-Bo. Lesen der Regeln, erste und sehr abstrakte Überlegungen dazu, was zu implementieren ist.

3. März:

Erstes Treffen, um Zeitplan festzulegen und Aufgaben zu verteilen. Probleme mit git über Intellij. Erarbeiten eines Grundverständnisses für Server-Client Architekturen.

Konkretere Überlegungen zu benötigten Klassen und Reihenfolge der Implementierung. Grundriss von GanttProject erstellt.

3. März - 10. März:

Arbeiten an der Präsentation. Jedes Gruppenmitglied macht seinen Teil der PowerPoint. Erstes individuelles Experimentieren, um selbst einen einfachen Server und Client mit Java zu programmieren.

Erster Versuch, einen simplen Chat zu programmieren.

10. März:

Erstellen eines genauen Zeitplans inklusive Verteilung der Aufgaben. Genaue Überlegungen zu Softwarearchitektur, inklusive Erstellen einer Grafik dazu.

Beenden von Mock-up und diversen anderen Vorbereitungen für die Präsentation am Donnerstag. Grosse Probleme mit Git: Synchronisation klappt bei einem Mitglied nicht.

Schreiben von erstem nützlichen Programmteil, einem Chat, welcher die Nachrichten auf dem Server ausgibt.

11. März:

Chat läuft. Überlegungen dazu, wie er in unser Endprogramm eingebaut werden wird. Feinschliff der Präsentation.

12. März:

Aufnahme der Meilenstein 1 Präsentation.

17. März:

Skype-Konferenz. Bestimmen der Struktur des Netzwerkprotokolls, da es den Grundbaustein für die Kommunikation zwischen Server und Client darstellt.

Überlegungen zu nötigen Befehlen für Chat und Spiel. Klären gewisse Fragen und Unsicherheiten innerhalb der Gruppe. Besprechen des weiteren Vorgehens.

17. - 22. März:

Intensive Code-Arbeit. Server und Client werden geschrieben inklusive jeweiligen Listeners.

Viele Erkenntnisse zu den Anforderungen der jeweiligen Klassen und der Verteilung der Aufgaben zwischen Client und Server: das Login mit dem Starten der Listeners ist komplizierter als gedacht, weil der Listener Thread gewisse Infos braucht, welche auf ersten Blick nur durch einen bereits bestehenden Listener gesammelt werden könnte. Erschwerte Kommunikation wegen Quarantäne.

20. März:

Bei den Spieloperationen sind einige Sachen erstellt worden. Player, Game und Card Klassen sind fast fertig geschrieben. Player Klasse ist besonders wichtig für die Server Implementation, und wir setzen hier Priorität. Kartenverteilung in der Klasse Game ist auch bereit. In einer Methode 'setupGame ()' werden zuerst alle Karten erstellt und dann weiter mit einer Random Variable an verschiedenen Spielern verteilt.

Die Stapelbildung geschieht in der Klasse Pile, die Handkarten der Spieler werden auch wie ein Pile behandelt, aber mit viel wenigen Einträgen. Alle Piles sind in verschiedenen ArrayLists gespeichert, da man die Grösse immer verändern kann!

21. März:

Server ist bereit und mehrere Clients können sich problemlos verbinden. Von der Client Seite ist es möglich, einen Username einzugeben. Server kann sich den Namen auch merken und alle Chat Messages der Clients aufnehmen. Leider ist das Zurückschicken der Nachrichten an alle Clients ein Problem. Die Clients können im Moment Nachrichten schicken, aber sehen die Nachrichten der anderen Clients nicht.

Nach einem langen Debug Session ist es klar, dass der Fehler bei Deklaration vom PrintWriter lag. Wir hatten den zweiten Parameter (autoFlush: true) vergessen; deswegen konnten die Nachrichten und Protokollbefehle nicht an alle Clients verschickt werden. Client-Client Kommunikation läuft zu diesem Zeitpunkt flüssig. Die Gültigkeit der Namen wird auch richtig überprüft vom Server, aber /change name Befehl gibt etwas Falsches aus. Einige Code Abschnitte müssen noch genauer kommentiert/beschrieben werden und das Netzwerkprotokoll muss noch besser dokumentiert werden.

22. März:

Weiterer Testing/Debug Session. /change name Befehl funktioniert ohne weitere Probleme. Der Fehler lag bei einer falschen Deklaration. Bei genauem Nachlesen der Meilensteinpunkte, wurde uns klar, dass der System Username vom Client vorgeschlagen werden muss, also haben wir einige Sachen ändern müssen – bei uns passierte dies auf Seite Server.

Tagebuch besser und anders formatiert. Netzwerkprotokoll wird auch noch fertig geschrieben und schöner formatiert. Die Code Abschnitte, die noch nicht perfekt dokumentiert waren, wurden ergänzt, damit jeder Teil des Codes für jeden verständlich ist. Ob der Client seine eigene Nachrichten ein zweites Mal sehen soll, wurde zu einem Diskussionsthema. Auf dem Terminal sah dies etwas komisch aus, also entschieden wir uns, dass nur die Chatnachrichten anderer Clients auf Terminal sichtbar sein werden, und auf GUI alle Nachrichten im Chat Box zu sehen sind.

Erster Test mit Himachi erfolgreich, obwohl wir zu Beginn nicht ganz verstanden haben, wie Himachi funktioniert. Chat zwischen Gruppenmitglieder funktioniert einwandfrei, aber wir haben manchmal bei Rohan immer wieder eine Fehleranmeldung bei /change name Befehl bekommt. Im Moment nicht ganz klar wieso, da diese beim Rest des Teams nie vorkam!

Letzte Tests und kleine Verbesserungen werden noch im Laufe des Sonntags durchgeführt, aber die Mehrheit der Meilensteinpunkte sind unsererseits erfüllt!

Ende MS2