02.03.2022 – alle Gruppenmitglieder

- Einrichtung Git und IntelliJ

- Spielidee: Jonas: Werwolf-Spiel

6 Spieler, rundenbasiert

Grobe Idee: Es gibt einen Geist und 5 Personen, die in einem Nachtzug in benachbarten Kabinen fahren. Ein Geist kann in der Nacht einen anderen „gesunden“ Spieler (Person) infizieren. Dabei läuft der Geist neben anderen Personen und sie „hören“ den Geist (Benachrichtigung). Wenn die Nacht vorbei ist, müssen sich alle Spieler entscheiden, ob sie sagen, sie hätten etwas gehört oder nicht (mit Chat). Dann gibt es am Tag eine Abstimmung, wer der Geist ist (Stimme der Geister zählt nicht). Der Spieler mit den meisten Stimmen wird aus dem Spiel geworfen. In der nächsten Nacht stimmen die verbliebenen Geister ab, wen sie infizieren wollen. Anschliessend beginnt die Infektionsphase erneut.

Das Spiel ist vorbei, wenn nur Geister oder nur Personen übrig sind.

Schräge/isometrische Grafik, prerendered

08.03.2022 – alle Gruppenmitglieder

- Ziele für heute: Spieldesign/Spielregeln, Analyse, Projektplanung

- Spieldesign:

- Mindestens 6 Spieler (bei weniger als 6 human players wird der Rest mit NPCs aufgefüllt)

- Zwei Arten/Rollen von Spielern: ghosts, humans

- Server kommuniziert Spielern, ob sie humans oder ghosts sind

- Spieler, der neues Spiel kreiert, ist Admin (darf neues Spiel z.B. auch nur mit NPCs anfangen)

- ghosts vs. humans: Geister können Menschen infizieren und zu Geistern machen. Sind nur Geister im Spiel übrig, haben Geister gewonnen. Haben die Menschen den ursprünglichen Geist aus dem Spiel geworfen, haben sie gewonnen.

(- Versucht ein Geist, einen Menschen zu infizieren, ist nicht sicher, ob er es schafft)

- Eine Runde = ein Tag-Nacht-Zyklus

- Geister haben separaten Chat, in dem sie entscheiden können, wen sie in der nächsten Nacht infizieren

- Nacht: Geister aktiv, infizieren Menschen

- Infizierte wissen, wer sie infiziert hat

- Läuft der Geist neben Menschen, ohne sie zu infizieren, werden diese benachrichtigt

- Tag: anonyme Abstimmung, wer aus Spiel herausgekickt werden soll

- time limit: Wenn man in einer bestimmten Zeit nicht abgestimmt hat, kriegt man eine Stimme für sich selbst

- Spieler dürfen vor der Abstimmung sagen, ob sie einen Geist gehört haben oder nicht (Geister können Menschen dabei täuschen)

- Stimmen der Geistern zählen nicht

- Rausgeworfene Spieler werden Zuschauer

- Nach Abstimmung wissen alle, wer ausgewählt worden ist, und ob der Spieler Geist war

Zusammenfassung:

- Spielbeginn: 6 Spielfelder. Erste Runde beginnt. Es ist Nacht, es gibt einen Geist. Geist stimmt für einen Menschen, den er infizieren will (Abstimmung mit time-limit; in möglichen späteren Runden stimmen mehrere Geister über zu infizierenden Menschen). Er infiziert diesen Menschen.

- Infizierter wird benachrichtigt, dass er nun Geist ist; zudem Benachrichtigungen über Bewegung des Geistes

- Es wird Tag: Menschen stimmen ab (mit Chat für Menschen, mit time limit), Geister-Stimmen zählen nicht. Wenn ein Geist die meisten Stimmen erhalten hat, wird er zum Zuschauer. Wenn Menschen für einen Menschen stimmen, werden sie benachrichtigt, dass er kein Geist ist (er wird nicht rausgeworfen) und das Spiel geht mit ihm weiter.

- Spielende: entweder es gibt nur Geister (Geister gewonnen) oder erster Geist durch Menschen-Abstimmung aus dem Spiel geworfen (Menschen gewonnen).

- Spieltitel: “Night train to Budapest”

- Durchspielen auf Papier

- Softwareanforderungen:

- 2D-Grafik

- Event-Handler (kümmert sich um Input/Output)

- Zustandsspeicherung/Zustandsverwaltung (ist Tag / Nacht / Abstimmungszeit / Infektionszeit?)

- Client/Server-Kommunikation

- Separate Chatfunktion

- Grundlagen der Projektplanung

- Netzwerkprotokoll

- Client/-Serverstruktur

- Chatfunktion

- Spiellogik (Speicherung, Methoden für Veränderung)

- GUI

- Klassenstruktur (Datenstrukturen, Client/Server greifen darauf zu)

11.03.2022 – alle Gruppenmitglieder

- Einrichtung, Synchronisation GanttProject

- Projektplanung: Anfang (parallelisierbar): Klassenstruktur (Human, Ghost, …), Spiellogik (Spielzustand, Voting), Client/Server (rundenbasiert), später: GUI (2D-Engine, Animation), Chat (mit speziellem Ghost-Modus)

- Aufgabenaufteilung Meilenstein 1: Vorbereitung Präsentation (Jonas), Übersicht Netzwerk (Sebastian), Softwareanforderungen (Alex), detaillierter Projektplan (Seraina)

- Netzwerktopologie für das Spiel: Stern