



Meeting C 26.10. ab 13:00 Uhr

Projekt - Schiffe versenken



Use Case erstellen ☒



Kommunikation/ Regeln in der Gruppe ☒ ?



Präsentation jeder einzelnen Untergruppe, zur Vorbereitung für Freitag



GIT



Bericht aus der Schnittstellengruppe

Aufgaben klassifizieren: jede Person hat einschl. Kommentare 1000 Zeilen Code zu Verfügung zu stellen am 16.12.22

Schnittstelle zwischen den einzelnen Gruppen —> [Jannik ist die Person die sich mit den anderen bezüglich des gemeinschaftlichen Repository kommuniziert](#)

Projektleitung Organisation —> Katharina, nach Verfügbarkeit

GUI —> [Stefan](#), evtl [Jannik](#)

KI —> Katharina, [Alex](#)

Netzwerk —> [Jann](#)

Logik —> [Lutz](#), [Philip](#), [es gab am 22.10. ein online meeting mit Jannik und Stefan](#)

Bericht aus Jannik Schnittstellengruppe :

Es soll eine abstracte Klasse Spieler geben
Die ist nirgends zugehörig weder GUI noch Logik denn
Sie muss mit allen interagieren. Es muss auf antworten
reagieren der Spieler muss implement wieder

GUI : ist der Client der die anderen Untergruppen darstellt, ist eher ausführend und Darstellend. GUI braucht getter Methoden um den Spielfortlauf darzustellen

Logik : enthält zwei Matrizen, (Gegner/ Spieler) Schiffe Positionieren , überprüfen ob an einer bestimmten Position etwas befindet (boolean getermethoden), informiert die Spielfigur über den Spielverlauf und wird dann an alle anderen weitere. Die gibt die Spielfigur vor. Die Logik ist es egal wer gegen wen spielt

KI : wenn noch nichts getroffen wurde, bei treff in ein neues Array, die KI braucht Arrays in den festgehalten wird in welchen Bereichen man schon war .
KI muss erkennen können ab wann versenkt und getroffen ist

Netzwerk: hat einen Client und einen Server muss einen Spieler extenden. Braucht getterKlassen wie getMove und das Netzwerk muss dann den abstracten Spieler extenden und muss informationen senden können

Eine KI kann ein Spieler sein. Du selbst kannst ein Spieler sein, dein Gegner kann auch ein Spieler sein. Der Spieler ist nicht derjenige der das Spiel spielt, sondern eher die Spielfigur.

Spielfigur soll eine Abstrakte Klasse sein die von allen nutzbar ist

Kommunikation in der Gruppe :

Kommunikation in Telegram Gruppe :

sollte jemand ein Dokument im allgemeinen allen zur Verfügung gestellt haben wird eine kurz Nachricht in die Gruppe geschickt. Diese Info kann zu jeder Zeit mitgeteilt werden

Bis 20 Uhr abends kann erwartet werden zu Chatfragen und Kontexte eine Antwort zu bekommen.

Beginn ist fließend

Es soll vorher angekündigt werden, wenn dringenden Termine im Anschluss anstehen und sich somit früher verabschieden

Ab und an mal ein Pauschen

wichtige Entscheidungen oder große Besprechungen werden an jedem BuchstabenMeeting kurz angesprochen und mit allen besprochen. Um Notizen vorzubereiten kann man die einzelnen Notes.txt files in den Unterordnern nutzen

Allgemein:

um schneller Dinge zu besprechen, wird Discord zum Voice Call benutzt um gemeinsam an den Untergruppenthemen zu arbeiten

Zeitpunkte müssen eingehalten werden

Jede Woche werden kleine Teilaufgaben festgelegt, die bis zur kommenden Woche erledigt sein müssen. Diese Fortschritte werden in den Buschtabenmeeting von jeder Person präsentiert und weiteres Verfahren mit der Gruppe ausgearbeitet

Code ausschließlich in Englisch, beinhaltet auch Kommentare und API, Präsentationen auch in Englisch, Fragen an Philip in Englisch

Logik :

Lukas berichtet, dass es Probleme gab bzgl. der Klassenerstellung bzw. welche von Nöten sind



Demo Game über Konsole zum Laufen zu bringen Bereiche aufgeteilt
Lukas Ship, Player Game
Philip Logik

→ Bis nächste Woche beide Parts zusammenbringen
→ Flussdiagramm

Netzwerk:

Im Netzwerk werden Daten übertragen, wenn Spieler human gegeneinander spielen.

In

https://lab.it.hs-hannover.de/tcm-ss5-u1/progpr22-23-api/-/blob/main/Schnittstellendefinition_1.md
wurde festgehalten was übertragen wird

Das Protokoll überträgt zum Beispiel nur Treffer und erwartet einen Rückgabotyp

Anderes Beispiel: man schießt und über das Netzwerk wird weiter gegeben das geschossen wurde und zurück getroffen oder nicht kommt zurück

Es wird immer im gleichen Semester gespielt muss man vorm Spieler sich einigen

Für den Anfang in der GUI anfrage ich möchte ein Spieler erstellen oder einem beitreten
Chat optional

Hinweis für die Logik: es wurde festgesetzt welche Schiffe in welchem Semester sind
Was alle Gruppen können müssen: Schießen, klicken,

Das Netzwerk bräuchte als nächste Infos von der Logik: Aus der Logik müsste rauslesbar sein in welchem Semester man sich befindet, oder man kann es in der GUI anklicken

Daten werden als Koordinate gesendet das Netzwerk konfiguriert den Standard

GUI:



→ visuelle Struktur

KI:

Kommunikation zur Weitergabe kann von dem Netzwerkprotokoll implementiert werden und angewendet werden

→ künstliche Pause soll implementiert werden, da die KI sonst in ein paar Millisekunden mit ihrem Zug fertig ist.

→Die Möglichkeit ein Feld zweimal zu beschießen, muss entsprechend behandelt werden -> noch keine Lösung

→Über mögliche & etablierte Algorithmen bzw. Taktiken muss sich noch informiert werden.

Konklusion:

Es muss aus allen Untergruppen angefangen werden Klassen zu schreiben um sich das Konzept zu bauen

Hierzu gab es am 22.10.22 eine Bitte über Entwicklung berichten 

Präsentation jeder einzelnen Untergruppe, zur Vorbereitung für Freitag

Jannik :

Prototyp aufbau mit GUI, Logic, Player, Netzwerk, so könnte der Programmaufbau aussehen

Stefan:

Erstellung der einzelnen Szenen mit Übergang zwischen den einzelnen Szenen ohne große Funktion. Die GUI Vorlage ohne Funktionen in die Szenen eingefügt. In die Prototyp eingefügt.

Visualisierung des Spielfelds ohne Funktion

Alex:

Klasse die ein Array entgegen nimmt; bereits getätigte Schüsse & Treffer werden in zwei verschiedenen Listen gespeichert; Methode Shoot() generiert einen zufälligen Punkt innerhalb des Array und gibt ihn zurück, es wird kein Schuss zweimal auf den selben Punkt getätigt

Lukas:

Observer Pattern für Playerlogik grundlegend implementiert. Logik holt sich Änderungen von den Playern und informiert die Player über Änderungen. Player informieren Logik bei bestimmten Events. Ein Zug müsste grundlegend funktionieren.

Philip:

jedes Level aus einer Textdatei zu lesen und das Spielbrett zu erstellen

Katharina

Plan bis nächste Woche:

Jannik:

Generellen Prototyp ausarbeiten und in **Main-branch mergen**, als genereller Grundstein für alle Teilgruppen
Netzwerk weiter ausarbeiten

Lukas:

Schiffe setzen mit Mitteilung an die Logik, dass Schiffe gesetzt sind.
UML Diagramm der Klassen erstellen. Code aufräumen und dokumentieren.

Alex :

Bis nächste Woche: Schiffe platzieren, Priorisierung von Schüssen
konzeptionieren (Bei Schüssen an bekannten Treffern orientieren)

Stefan:

Die Schiffe visualisieren in der Lobby und per drag und drop auf das Spielfeld hinzufügen.

Philip: Verarbeitung der Konfigurationsdatei anstelle von 6 verschiedenen Dateien nur eine und auch die Erstellung von Klassen, die aus dieser Konfigurationsdatei lädt/erzeugt.

Katharina:

Neuer Termin : 02.11.2022 ab 15-16:00 Uhr Meeting

Troubleshooting

MainBranch GUI/Network/Logic

Es ist aufgefallen, dass die Enheritance von Logik ist anders als von Network.
Es wird eine neue Version von Jannik gebastelt.

NetworkPlayer extends Player