Nik Wachsmann, Kevin Bähr

Utneo

Dokumentation Fach CT 13/1

Inhalt

[Ziele: 2](#_Toc123656220)

[Einleitung: 2](#_Toc123656221)

[Technische Umsetzung: 2](#_Toc123656222)

[Game-Engine “Godot”: 2](#_Toc123656223)

[Programmiersprache: 2](#_Toc123656224)

[Netzwerkkommunikation: 2](#_Toc123656225)

[Server/Client: 3](#_Toc123656226)

[Accountsystem: 3](#_Toc123656227)

[Tutorial: 3](#_Toc123656228)

[Projektentwicklung: 3](#_Toc123656229)

[Godot / Multiplayer testen: 3](#_Toc123656230)

[Utneo basic GUI: 4](#_Toc123656231)

[Kommunikation zwischen Client und Host: 4](#_Toc123656232)

[Spielmechanik: 4](#_Toc123656233)

[Timer: 4](#_Toc123656234)

[Datenbank/Accounts: 5](#_Toc123656235)

[Anzeigen: 5](#_Toc123656236)

[Gewinn Optionen: 5](#_Toc123656237)

[Angenehmere Bedienung: 6](#_Toc123656238)

[Dev tools: 6](#_Toc123656239)

[Wiederkehrende Probleme: 7](#_Toc123656240)

[Bedienung: 7](#_Toc123656241)

[Verbinden: 7](#_Toc123656242)

[Lobby Einstellungen: 8](#_Toc123656243)

## Ziele:

#### Muss-Ziele:

* Utneo spielbar
* Unbegrenzte Clients 1 Host
* Spielbar im selben Netzwerk

#### Soll-Ziele:

* Accounts
* Einstellungen zum Spielverlauf
* Punkteberechnung

#### Kann-Ziele:

* Online Spielbar

## Einleitung:

Das Thema dieses CT-Projektes ist die Entwicklung eines UNO-ähnlichen Spiels mithilfe der Game-Engine “Godot”. Das Spiel soll von zwei oder mehr Computern, die über das Internet kommunizieren, gespielt werden können.

## Technische Umsetzung:

### Game-Engine “Godot”:

Wir haben uns dazu entschieden, für die Entwicklung dieses Projektes eine vorgefertigte Game-Engine namens Godot zu verwenden. Es gibt andere Möglichkeiten wie Unity oder Unreal Engine, jedoch wählen wir Godot, da dies die einzige Open-Source Game-Engine ist.

### Programmiersprache:

Für dieses Projekt haben wir die Programmiersprache GDscript verwendet, was eine von Godot entwickelte, python ähnliche Sprache ist. Alternativ könnte man bei Godot auch C# oder C++ verwenden, jedoch sind die meisten Tutorials immer in GDscript, weshalb wir dies wählten.

### Netzwerkkommunikation:

Für die Kommunikation innerhalb des Netzwerkes und in der Theorie auch über das Internet verwenden wir eine von Godot gegebene Funktion. Diese übernimmt das gesamte Networking, wodurch die Verwendung sehr leicht ist. Zum Server erstellen muss man den Port und die maximale Spieleranzahl angeben. Um vom Client auf den Server zuzugreifen, braucht man IP und Port. Innerhalb eines Netzwerkes sowie über Localhost funktioniert es problemlos. Theoretisch können wir auch mehrere PCs über das Internet verbinden, jedoch müssten wir dafür einen Port des Routers öffnen, was Risiken mit sich bringt, oder einen Server mieten, was Geld kostet.

### Server/Client:

Anwendung:

Unser Projekt ist in zwei Bereiche einteilbar. Der Client und der Server. Der Server ist für die Berechnung der Karten, der Rechnungen und des Spielablaufes zuständig. Die Spieler brauchen jeweils einen Client. Die Clients schicken Signale an den Server, woraufhin dieser mit weiteren antwortet. Manche Signale werden nur an den Client geschickt, der eine Anfrage geschickt hat, andere werden an alle Clients geschickt.

#### Anti-Cheat

Da es immer Menschen geben wird, die versuchen, einen unfairen Vorteil zu bekommen, haben wir von Anfang an einen einfachen Anti-Cheat eingebaut. Hätte nur der Client die Handkarten gespeichert, könnte der Server nicht überprüfen, ob der Spieler die Karten tatsächlich hat. Da man auf diese Weise seine eigenen Karten erzeugen kann, um zu gewinnen, weiß der Server zu jedem Zeitpunkt, welcher Spieler welche Karten hat. Ebenso werden neu gezogene Karten vom Server erstellt und dem Client geschickt. Auf diese Weise bringt es nichts, dem Server Karten zu schicken, die man nicht wirklich hat.

### Accountsystem:

Wir haben bei unserem Spiel ein Account System implementiert. Ziel hiervon ist es, die Gespielten Spiele, gewonnenen Spiele und Punkte zu speichern. Ebenso kann man auf diese Weise Namen haben, die nicht mehrfach in einem Spiel auftauchen können. Die Benutzernamen und E-Mail-Adressen müssen einzigartig sein. Falls man einen Namen oder eine E-Mail-Adresse verwendet, die bereits existiert, kann man keinen Account erstellen. Das Passwort wird zusammen mit dem Namen gehasht und dann gespeichert.

### Tutorial:

Da das Spiel nicht selbsterklärend ist, haben wir ein Tutorial gemacht. Dieses ist auf der Internetseite <http://utneo.rf.gd/> zu finden. Beim Starten des Programms bekommt man die Möglichkeit über einen Button zu diesem Tutorial zu gelangen. Das Tutorial selbst ist in PHP geschrieben. Wenn man auf +1/-1 drückt, wird ein anderer Text und ein anderes Bild geladen.

## Projektentwicklung:

### Godot / Multiplayer testen:

Angefangen haben wir mit einem kleinen Programm, mit dem man ein kleines weißes Quadrat über den Bildschirm fahren lassen kann. Daraufhin haben wir dies mit einem Server versucht. Das erste Zeichen, dass unsere Laptops verbunden waren, war, dass bei einem Laptop etwas gedrückt wurde und bei dem anderen ein Fehler erzeugt wurde. Nach einigem Probieren schafften wir es, von zwei getrennten Laptops auf dem gleichen Fenster zwei Quadrate zu steuern. Daraufhin fingen wir mit dem Kartenspiel selbst an.

### Utneo basic GUI:

Die anfängliche GUI beinhaltete nur das Nötigste: Karten und Buttons. Karten sind technisch betrachtet Texturebuttons, also Buttons mit einem Bild als Form. Wenn ein Button gedrückt wird, ruft dieser eine Funktion auf. Die GUI war nicht besonders schön, aber sie erfüllte ihren Zweck.

### Kommunikation zwischen Client und Host:

#### Verbinden:

Die Verbindung wird mit der High Level Multiplayer API, welche von Godot gestellt wird, übernommen. Man übergibt die IP-Adresse und kann sich daraufhin mit dem Server verbinden.

#### Datenübermittlung:

Da es nicht möglich ist, Objekte zu übermitteln, werden einfache Zahlen und Strings verschickt. Funktionen, die über die Internetverbindung aufgerufen werden sollen, müssen mit puppet oder master gekennzeichnet werden.

#### Verbindung trennen:

Um die Verbindung zu trennen, wird das Peer-Objekt, welches sich um die Verbindung kümmert, auf null gesetzt.

### Spielmechanik:

#### Rechenoperationen:

Um auf die aktuelle Karte zu kommen, kann man die Rechenoperationen Plus, Minus, Mal, Geteilt, Hoch und Wurzel benutzen. Bei der Wurzel ist zu beachten, dass es nicht die Quadratwurzel, sondern die n-te Wurzel aus x ist.

#### Andere Optionen:

Weitere Optionen, die man benutzen kann sind der Disconnect-Button, Push Button, Clear Button und der Kartenstapel (Draw Button). Mit dem Disconnect-Button kann man das Spiel verlassen. Der Push-Button, der als Pfeilsymbol dargestellt wird, muss gedrückt werden, wenn man seine Karten mit der Rechenoperation ablegen möchte. Der Clear Button kann gedrückt werden, wenn die ausgewählten Karten und Rechenoperation nicht richtig sind und man seine aktuelle Auswahl ändern möchte. Der Kartenstapel kann gedrückt werden, um eine Karte zu ziehen, wenn keine Karte abgelegt werden kann.

### Timer:

Der Timer hat mehrere Funktionen. Er schränkt die Zeit ein, die ein Spieler zum Nachdenken hat. Auch zeigt er dem aktuellen Spieler an, dass dieser an der Reihe ist. Dies wird dadurch erreicht, dass nur in diesem Fall der grüne Balken zu sehen ist. Die verbleibende Zeit in Sekunden wird allen Spielern gezeigt. Wenn einem Spieler die Zeit ausgeht, bekommt dieser zwei Karten.

### Datenbank/Accounts:

Login:

Um sich anzumelden, gibt man den Accountnamen und das Passwort ein. Dies geht, da der Benutzername einzigartig ist. Das Passwort wird lokal gehasht und dann an den Server übergeben. Dort wird es mit der Datenbank abgeglichen. Stimmt es, bekommt der Client einen Schlüssel, welcher einfach die Uhrzeit mit dem Benutzernamen gehasht ist. Nach dem Szenenwechsel auf dem Client wird dieser erneut an den Server übergeben, um die Anmeldung zu Verifizieren.

Sign up:

Um einen Account zu erstellen, muss man einen Benutzernamen und eine E-Mail-Adresse angeben, wobei beide Einzigartig sein müssen. Das Passwort muss zweimal eingegeben werden, um zu verhindern, dass man es fehlerhaft eingibt. Wenn der Account frei ist, wird man automatisch mit dem Anmeldeprotokoll angemeldet.

Gespeicherte Daten:

Neben den Daten, die beim Erstellen des Accounts benötigt werden, werden auch gespielte und gewonnene Spiele sowie die gesamte Punktzahl gespeichert. Dies kann in der Zukunft dazu verwendet werden, ein gesamtes Leaderboard zu erstellen.

### Anzeigen:

Spielzugverlauf:

Wenn ein Spieler einen Zug macht, wird dieser in einem Textfeld für alle Spieler sichtbar. Auch wenn ein Spieler eine Aktion versucht, die nicht der aktuellen Karte entspricht, wird dies angezeigt. Diese Funktion kann jedoch vom Host deaktiviert werden.

Spielerliste:

Die Spielerliste ist im oberen linken Eck des Fensters. Während der Runde ist der aktuelle Spieler Farbig markiert. Außerdem steht neben jedem Spieler die aktuelle Kartenanzahl. Am Ende der Runde wird die Liste anhand der Gewinnoption sortiert. Ist ein Spieler fertig, so wird dies mit einem (Done) gekennzeichnet.

Punktestand:

Während der Runde kann man nur die eigenen Punkte sehen. Diese stehen am oberen Rand in der Mitte des Fensters.

### Gewinn Optionen:

#### Rundensystem:

Im Rundensystem gewinnt der Spieler, der als erstes keine Karten mehr auf der Hand hat. Die Gewinn Option kann der Host am Anfang einstellen, bevor der Host das Spiel beginnt.

#### Punktesystem:

Im Punktesystem gewinnt der Spieler, der am meisten Punkte hat. Die Punkte werden berechnet, wenn man alle Karten abgelegt hat. Wenn der vorletzte Spieler, der letzte Spieler kann nachziehen, die letzten Karten spielt und somit das Ende des Spiels einleitet, wird abgerechnet und geschaut welcher Spieler am meisten Punkte hat. Die Punkte werden mit dieser Formel, für jeden Spieler einzeln berechnet:

**Absolute Punkte / (1 + 0.05 \* fertige Spieler \* Rundenzahl)**

Die absoluten Punkte sind die Summe der Rechnungen, die man während des Spiels getätigt hat.

### Angenehmere Bedienung:

#### Kartenanzeige:

Um vor allem bei einem kleinen Fenster oder vielen Karten das Auswählen der korrekten Karte zu erleichtern, wird die Karte, über der sich die Maus befindet, undurchsichtig. Wird eine Karte ausgewählt, so bleibt diese ebenfalls undurchsichtig.

Bessere GUI:

Die oben beschriebene Utneo basic GUI wurde überarbeitet, um das Spiel einladender aussehen zu lassen. Das Farbschema ist Blau. Die Rechenoperations-Buttons haben eine bessere Form und die anderen Optionen haben Symbole bekommen.

Shortcuts:

Bisher ist der einzige Implementierte Shortcut Enter, womit man die Karten ablegen kann. Ideen für weitere Shortcuts sind Backspace, um die aktuell ausgewählten Karten zu löschen. Auch gibt es die Idee, über die Zahlen auf der Tastatur die Handkarten auszuwählen, sofern eine mit dem Wert existiert.

### Dev tools:

#### Schneller Login:

Mit vorgefertigten Accounts kann man sich schneller anmelden, um zu spielen, um neue Features zu testen.

Vorgefertigte Accounts sind 0 bis 9 im Benutzernamen und haben auch die gleiche Zahl als Passwort.

#### Local host:

Um beim Testen nicht dauerhaft eine Internetverbindung zu benötigen, haben wir die Möglichkeit eingebaut, über ein einfaches Check Feld anstatt einer IP Localhost zu verwenden. Dadurch kann man schneller kleinere Änderungen testen.

#### Vorgegebene Karten:

Manchmal, um spezielle Events auszulösen, haben wir die Karten nicht Zufällig generieren lassen, sondern die Karten, die wir benötigten, hardgecoded.

### Wiederkehrende Probleme:

#### Division by Zero:

Dieses Problem hat uns einige Male wieder beschäftigt. Bei der Division ist dieses Problem offensichtlich, weshalb hier als Erstes eingegriffen werden konnte. Jedoch gab es mit der Zeit auch bei der Wurzel Probleme. Dies lag daran, dass die Wurzel über x^(1/n) berechnet wird.

#### Timer:

Manchmal wurde der Timer falsch zurückgesetzt, weshalb Spieler ungemäß übersprungen wurden. Das Problem war, dass der Timer nicht zurückgesetzt wurde, nachdem ein Spieler eine Karte zog und somit ein anderer Spieler drankam.

#### Spielerwechsel (Ungeplant):

Längere Zeit hatten wir das Problem, dass Spieler manchmal übersprungen wurden. Hierfür gab es zwei Ursachen. Die eine ist der Timer, welcher oben erklärt wurde. Das andere Problem war schwerer zu finden, da es willkürlich wirkte. Das Problem war, dass der Code zur Erkennung, ob die Aktion korrekt ist, doppelt ausgeführt wurde. Wenn die zweite Karte einer Aktion dem Ergebnis der Aktion entsprach, wurde ein Spieler übersprungen.

## Bedienung:

### Verbinden:

Nach dem Ausführen des Programms hat man sich zu entscheiden, ob man Server oder Client sein möchte. Für den Server muss man nur einen Port eingeben, falls man nicht den Standard Port 7777 benutzen möchte. Als nächstes bekommt man die Möglichkeit, die Einstellungen der Lobby, die das Spiel beeinflussen, zu verändern. Hat man dies getan, kann man den Server hosten. Als Client braucht man die IP-Adresse des Servers, welche bei diesem oben links angezeigt wird. Falls der Server vom Standardport abweicht, muss man diesen nach der IP mit einem Doppelpunkt spezifizieren.

### Lobby Einstellungen:

Der Host kann einstellen, wie viele Spieler es maximal geben darf. Wenn die Spielerzahl unbegrenzt sein soll, kann man die Checkbox deaktivieren. Dies ist zwar in der Theorie nicht endlos, da die maximale Anzahl der Networking API 4095 ist, jedoch wird es praktisch nie so weit kommen. Sobald das Spielerlimit erreicht wird, werden keine neuen Verbindungen akzeptiert. Wird es wieder unterschritten, können neue Spieler sich verbinden. Auch kann der Host die maximale Rundenzeit einstellen. Dies ist die Zeit, die ein Spieler hat, um einen Zug zu machen. Auch hier kann man die Zeit einstellen oder über die Checkbox den Timer deaktivieren. Als nächstes kann man die Anzahl der Handkarten einstellen, die jeder Spieler zu Beginn des Spieles haben kann. Der Host kann entscheiden, ob Spieler nach Beginn des Spieles noch dazu kommen dürfen. Wenn dies erlaubt ist, gibt es 3 Möglichkeiten, wie viele Karten der verspätete Spieler bekommt. Normal ist die normale Menge an Startkarten, die oben festgelegt wurde. Average ist der Durchschnitt der Handkarten aller Spieler, die noch in der Runde sind, fertige Spieler ausgenommen. Unique lässt den Host eine feste, von der normalen Menge abweichende Menge an Karten austeilen. Auch kann der Host entscheiden, ob fehlerhafte Aktionen eines Spielers im Chatverlauf angezeigt werden.

**P.S. Utneo bedeutet U times N exponent O**

Github Repo: <https://github.com/AI-Racing-Company/Utneo>

Quellen:

* <https://docs.godotengine.org/en/stable/index.html>
* <https://stackoverflow.com/>
* <https://docs.godotengine.org/en/stable/tutorials/networking/high_level_multiplayer.html>
* <https://www.youtube.com/watch?v=lpkaMKE081M>
* https://godotengine.org/qa/