

Inhaltsverzeichnis

1	Kurzzusammenfassung.....	2
2	Einleitung.....	3
2.1	Motivation.....	3
2.2	Hypothesen	3
2.2.1	Wettbewerb beeinflusst die Leistung	3
2.2.2	Wettbewerb beeinflusst das empfinden der Nutzungserfahrung	3
2.2.3	Nicht-menschliche und menschliche Gegner besitzen den selben Einfluss	3
2.3	Definitionen	4
2.3.1	Gamification	4
2.3.2	Wettbewerb und Motivation.....	4
3	Ansatz und Vorgehensweise.....	5
3.1	Modi und Testszenarios	5
3.2	Implementation.....	6
3.2.1	Software.....	6
3.2.2	Spielablauf	7
3.2.3	Datensammlung und Spielverlaufslogbuch	8
3.2.4	Quellcode Zusammenfassung.....	9
3.2.5	Fragebogen	10
3.3	Aufbau	11
3.3.1	Beobachtungsgruppen	12
4	Vorstudie	13
4.1	Planung.....	13
4.2	Evaluation.....	13
4.2.1	Teilnehmer.....	13
4.2.2	Ergebnisse und Ablauf	13
5	Hauptstudie.....	15
5.1	Anpassungen.....	15
5.2	Evaluation.....	15
5.2.1	Teilnehmer.....	15
5.2.2	Ergebnisse	15
6	Diskussion	16
7	Ausblick	17

1. Kurzzusammenfassung

2. Einleitung

2.1. Motivation

Gamification ist mittlerweile eine Methode, die häufig dafür verwendet wird um Applikationen neu zu designen mit dem Ziel, dass diese ansprechender und effektiver sind. Auch werden einige Studien zu diesem Thema durchgeführt um die Wirkungen und Einflüsse durch die Anwendung zu ergründen. Dies fokussiert sich oft in einem Anwendungsbereich, zum Beispiel es Schülern beim lernen hilft. Verschiedene Aspekte aus Spielen können angewendet werden um ein Design zu alternieren, die Facette, die in dieser Arbeit betrachtet wird ist der Wettbewerb. Kompetitive Komponenten sind elementare Bestandteile vieler Spiele, aber auch in vielen anderen Kontexten sind sie enthalten. In dieser Thesis wird Wettbewerb als Aspekt von Gamification Kontexten betrachtet und untersucht.

2.2. Hypothesen

2.2.1. Wettbewerb beeinflusst die Leistung

Wenn man Wettbewerb wird das eigen Handel äußerer Einflüssen ausgesetzt. So kann das Ziel eines Siegen einen motivieren oder der direkte Vergleich mit einem Gegenüber vielleicht Stress erzeugen. Daher vermuten wir, dass eine kompetitive Situation ein Individuum in seiner Leistung beeinflusst, dies kann ein förderlicher oder hinderlicher Effekt sein.

2.2.2. Wettbewerb beeinflusst das empfinden der Nutzungserfahrung

Wenn man ein und die selbe Aufgabe unter verschiedenen Szenarien löst, kann dies die Wahrnehmung der Handlung beeinflussen. Dies könnte sich im Empfinden einer Situation auswirken oder sich in der Bereitschaft sich erneut in diese Situation zu begeben äußern. Wir vermuten ein kompetitiver Gedanke könnte eine Tätigkeit lohnenswerter oder abschreckender erscheinen lassen.

2.2.3. Nicht-menschliche und menschliche Gegner besitzen den selben Einfluss

Im Wettstreit vergleicht man seine Leistung mit einer anderen Leistung, daher vermuten wir ist es nicht relevant wer oder die andere Leistung erbringt. Also das es dabei nicht relevant ist ob der Gegner menschlich oder nicht-menschlich ist.

2.3. Definitionen

2.3.1. Gamification

Gamification beschreibt den Vorgang Komponenten, welche für Computerspiele charakteristisch sind, zu einer anderen Anwendung hinzu zu fügen. Dies soll der Verbesserung der modifizierten Anwendung. So soll das Interesse der Nutzer dadurch gesteigert oder ein Lerneffekt erhöht werden.[?]

2.3.2. Wettbewerb und Motivation

Wettbewerb lässt sich als das Vergleichen von Leistungen definieren und dient als extrinsische Motivation. Motivation lässt sich zwei Arten unterscheiden: intrinsische und extrinsische Motivation. Der intrinsische Motivation entspringt einem inneren Antrieb seine eigenen Fähigkeiten zu bestätigen. Im Gegensatz dazu steht die extrinsische Motivation, bei der der Anreiz von einem äußeren Faktor entspringt. Im Fall eines Wettbewerb wäre der Anreiz üblicherweise der Sieg und damit das Besiegen des Gegenübers.[?]

3. Ansatz und Vorgehensweise

Um den Einfluss von Wettbewerb in der Leistung innerhalb eines spielerischen Szenarios beobachten zu können, braucht man eine Umgebung, in welcher dies unter verschiedenen Gesichtspunkten, möglich ist. Für diesen Zweck bietet sich die Implementation eines simplen Computerspiels an, mit dem die kognitiven Fähigkeiten einer Person testen, die Schwankungen in der Leistung aufzeichnen und verschiedene Testszenarien überprüfen kann. Als Spiel wird ein Mathespiel umgesetzt, in dem die Probanden Kopfrechnungen lösen. Zudem sollen die Teilnehmer der Testgruppen einen Fragebogen ausfüllen, der die Beobachtungen der Spielsituationen in einen weiteren Kontext setzt.

3.1. Modi und Testszenarios

Verschiedene Szenarios in denen Wettbewerb in einer Form umgesetzt ist sind denkbar. Im Verlauf der ersten Planung wurden Situationen festgelegt, die interessant für die das Bestätigen oder Widerlegen der aufgestellten Thesen sind. Diese Situationen sollen in Spielmodi abgebildet werden.

Diese Modi sind Variationen eines Standardmodus. Dieser Ursprungsmodus lässt sich mittels der Einstellungen modifizieren. Änderungen durch veränderte Einstellungen sind zum Beispiel die Dauer des Spiels, das Anzeigen der erzielten Punkte oder der verbleibenden Zeit. Das Spiel terminiert entweder manuell nach dem betätigen der 'Escape'-Taste oder nach einer gesetzten Zeit, da diese auch als 'unendlich' eingestellt werden kann, ist das händische Beenden immer möglich. Die Aufgaben können während eines Spieldurchlaufs zufällig generiert werden oder im Vorfeld erstellt, gespeichert und dann für ein Spiel geladen werden. Im folgenden werden alle implementierten Modi beschrieben:

- "Singleplayer"-Modus:

Dieser Modus entspricht dem Basisspiel in dem keine Wettbewerbssituation vorhanden ist. Man spielt ohne Gegner und versucht einen möglichst hohe Punktzahl zu erzielen. Die Anzeige auf dem Bildschirm entsprechen der Abbildung 1a.

- "Halb-Coop"-Modus:

Der Gedanke in dieser Variante ist das zwei oder mehrere Menschen den Einzelspielermodus parallel spielen. Dadurch soll ein indirekter Wettbewerb entstehen, da die Spieler im Grunde für sich alleine spielen, aber ein Vergleich der eigenen Performanz mit einer anderen Leistung möglich ist. Dafür synchronisiert sich der Start des Spiels der Personen. Ansonsten entspricht alles dem Basisspiels. Das Erscheinungsbild entspricht dem "Singleplayer"-Modus

- "Versus"- & "Versus2"-Modus:

Der "Versus"-Modus entspricht einer klassischen Umsetzung von Wettbewerb, zwei Spie-

ler spielen gegen einander. Die Punktzahlen von beiden werden gegenübergestellt, damit ein Vergleich der Leistung stattfinden kann. Die Punkteanzeige auf dem Bildschirm eines Nutzers ist in dieser Variante leicht modifiziert, vergleiche Abbildung 1b, denn sie zeigt auch den Punktestand des Gegenübers an (Bsp. X vs. Y, mit X als die Punktzahl des Spielers und Y als die des Gegners). Die zwei Varianten unterscheiden sich nur im Namen, diese Unterteilung ist für die Logbuchdateien notwendig.

- "Party"-Modus:

Um den Wettbewerb mit mehreren Teilnehmern darzustellen wurde dieser Modus implementiert. Die Teilnehmerzahl ist variabel und wird mit "falschen Spielern" von Spiel aufgefüllt, wenn eine gewünschte Spielerzahl nicht mit Nutzern erreicht wird. Das Userinterface ähnelt mehr dem der Einzelspielermodi als dem der "Versus"-Modus, mit einem unterschied es gibt eine Ranglistenanzeige, die die Punkte der Nutzer und der, falls vorhanden, falschen Spieler in Verhältnis setzt. Die Person ganz oben in der Liste hat die höchste Punktzahl, wenn man nicht an dieser Position ist signalisiert die Liste das, indem sie leicht pulsiert. Abbildung 1c zeigt eine Bildschirmaufnahme dieses Modus.

- "Collab"-Modus:

Ein Modus, der etwas gegen den Gedanken der anderen Modi geht, wurde auch implementiert. In diesem Modus kooperieren die Spieler um einen gemeinsamen Punktstand zu erhöhen. In diesem Modus entspricht der Bildschirm wieder dem des Basisspiels, der Unterschied zwischen einem eigenen Beitrag zum Punktestand und dem eines Mitspielers, ist die Animation ob man einen oder keinen Punkt hinzugefügt, beziehungsweise abgezogen hat. Ähnlich wie der "Halb-Coop"-Modus entspricht die Anzeigeart der des "Singleplayer"-Modus.

3.2. Implementation

Implementiert wird ein Spiel, das die festgelegten Testszenarios beinhaltet, in dem mathematische Probleme mittels Kopfrechnen gelöst werden und ein Fragebogen wird erstellt, der weitere Beobachtungen festhalten soll. Die erste Implementation wird in einer Vorstudie getestet, um eine bestmögliche Version für die Hauptstudie zu entwickeln. Auch wird anhand Daten der Vorstudie ein Durchschnittsnutzer erstellt, der dann in der Hauptstudie als Gegner dienen kann.

3.2.1. Software

Für die Entwicklung wird die Unity Engine 2018.2.2f1 von <https://unity3d.com> verwendet. Diese Entwicklungsumgebung für Spiele bietet einige Grundlagen und Vorlagen, welche sich anbieten, da eine eigene Entwicklung einer Engine nicht zielführend für die Arbeit ist. Um die Fragebögen zu erstellen wurde der Dienst "Google Formulare" (<https://www.google.com/intl/de/forms/about/>) verwendet. Für die Analyse der Daten wird Microsoft Excel(<https://products.office.com/de-de/excel>) verwendet.

3.2.2. Spielablauf

In diesem Abschnitt wird der Ablauf ab Beginn des Programms beschrieben. Die Oberflächen bestehen aus Knöpfen und Eingabefeldern, navigiert und interagiert wird mittels Maus und Tastatur. Der erste Bildschirm ist das Netzwerkmenü (Abbildung 3), in welchen man wählen kann ob man als Host spielen will oder als Client. Für den Fall, dass man als Client spielt kann man in die Netzwerkadresse des Hosts und den dazugehörigen Port ändern.

Nach der Wahl zu hosten oder nicht wird man in das Hauptmenü (Abbildung 2a und 2b) weitergeleitet. Von hier aus kann man eine Runde des Spiels starten, wenn man Host ist, besitzt man die Berechtigung den Spielmodus zu wechseln (Abbildung 2c). Auch kann man in die Einstellungen wechseln, sich vom Netzwerk trennen und damit zurück in das Netzwerkmenü und die Software beenden.

In den Einstellungen kann man den Pfad angeben auf dem die Daten gespeichert werden sollen. Unter diesem Pfad werden die Logbuchdateien einer Spielrunde gespeichert, sowie die generierten Aufgaben. Wenn man Aufgaben erstellen will kann man hier sagen wie viele Funktionen pro Runde generiert werden sollen. Alternativ kann man auch einstellen, dass die Aufgaben während des Spielens zufällig erzeugt werden. Weitere Einstellungen beziehen sich auf die Darstellung des Spiels. So ist es möglich die Anzeigen für die verbleibende Zeit und die Punkte jeweils zu deaktivieren. Die Dauer einer Spielrunde wird auch hier bestimmt man kann, also angeben ob eine Runde nach einer gewünschten Zeit endet oder solange weiter geht bis der Nutzer sie per Tastendruck beendet. Auch die Lautstärke der Musik und der Soundeffekte lassen sich in den Einstellungen regulieren. Informationen zu dem Spieler die für die Logbuchdateien benötigt werden, werden auch hier angegeben, dazu zählt die Identifikationsnummer und das Geschlecht. Auch die Angabe in wie viele Runden bereits gespielt wurden lassen sich hier verändern. Wenn man die Einstellungen werden diese gespeichert, nur die Informationen zu einen Nutzer werden nicht persistent hinterlegt.

Im Hauptmenü kann eine Spielpartie gestartet werden in dem auf den "Start"-Knopf gedrückt wird und alle nötigen Teilnehmer für den Modus bereit sind. Der "Singleplayer"-Modus kann immer sofort starten, die anderen Modi verzögern den Spielstart bis alle mit dem Host verbundenen Spieler mit dem "Start"-Knopf ihre Bereitschaft signalisiert haben. Während der Verzögerung können Nutzer durch erneutes Betätigen des Knopfes die Bereitschaft wieder aufheben.

Im Spiel hat man mehrere Anzeigen und ein Eingabefeld. Die folgende Oberfläche beschreibt einen generellen Fall des Spielmodus und trifft am ehesten auf den "Singleplayer"-Modus zu. Die Anzeige am mittigen, oberen Bildschirmrand ist die verbleibende Zeit, darunter findet man die "Score"-Einblendung. Zentral und im Verhältnis am größten dargestellt findet man die zu lösende Rechnung. Unterhalb der Aufgabe ist das Eingabefeld in dem die Lösung eingegeben wird. Zur Eingabe wird die Tastatur verwendet, wobei nur Zahlen eingeben und das '-'-Zeichen akzeptiert werden. Mit der "Enter/Return"-Taste wird eine Eingabe bestätigt, mit der "Backspace"-Taste können unbestätigte Eingaben wieder rückgängig gemacht werden. Wenn eine Aufgabe richtig beantwortet bekommt man einen

Punkt zum Score hinzufügt, dies wird durch einen Ton und einer Animation zusätzlich visualisiert. Für den Fall einer falschen Lösung wird die Punktzahl um einen Punkt reduziert, dass überspringen einer Aufgabe ist auch möglich wenn eine leere Eingabe bestätigt wird. Überspringen gibt keinen Punkt zum Score, zieht aber auch keinen ab. Auch das Verlieren oder überspringen einer Aufgabe hat jeweils eine eigene Animation und einen Ton.

Wenn man in den Einstellungen gewählt hat, dass man "Supervisor" ist, hat man eine abgewandelte Variante der Modi, die Änderungen sind in allen Modi gleich. Mit diesen Einstellungen muss man nicht mehr selbst beantworten, denn die Aufgaben werden von einem Skript selbstständig richtig gelöst. Nach einer bestimmten Zeit wird die Lösung eingegeben und bestätigt. Bis zur Bestätigung wird im Eingabefeld die verbleibende Zeit bis dahin und etwas unterhalb des Feldes die korrekte Lösung angezeigt. Beide Anzeigen sind so platziert, dass sie eher schwer zu erkennen sind, wenn man nicht von ihnen weiß. Zusätzlich ist es möglich, in dem Zeitraum bis zur autonomen Beantwortung, die richtige Antwort zeichenweise im Eingabefeld sichtbar zu machen, in dem eine beliebige Taste gedrückt wird. Der Zweck dieser Funktion soll es sein, eine Eingabe durch den Betreuer, trotz der selbstständigen Beantworten der Aufgaben vorzutäuschen. Die Zeitspanne zwischen den Antworten ist anfangs anhand gegebener Werte festgelegt, wenn es einen Gegner gibt, ändert sich dynamisch. Denn sie ist das Mittel der letzten drei Abständen in denen der Gegenüber eine Aufgabe gelöst hat.

3.2.3. Datensammlung und Spielverlaufslogbuch

Während eines Spieldurchlaufs wollen wir verschiedene Faktoren beobachten, um Rückschlüsse auf die Einflüsse der Testszenarien machen zu können. Für diesen Zweck sammeln wir Daten des Spielverlaufs und lassen sie uns für eine spätere Analyse speichern. Vom Spiel werden nach einer Runde drei Dateien angelegt, die die gewünschten Ereignisse dokumentieren. Die Logbuchdateien eines Spielers werden in einen Ordner unter einem angegebenen Pfad gespeichert, welcher in der Form "ParticipantX", wobei X der jeweiligen Nutzeridentifikationsnummer entspricht (Bsp. Participant1), benannt wird. Die drei Dateien einer Runde unterscheiden sich im Dateiformat und gering im Informationsgehalt. Die im Spiel zu beobachteten Variablen werden zunächst serialisiert und dann als ".dat"-Dateien gespeichert, diese Dateien dienen ausschließlich als redundante Sicherheitskopie und könnten wieder deserialisiert und geladen werden. Dies ist im Code nicht umgesetzt worden, da keine Notwendigkeit bestand.

Um den Verlauf einer Runde in der Retrospektive nachvollziehen zu können, werden als ".txt"-Dateien stichpunktartige Protokolle der Ereignisse dokumentiert. Im ersten Part der Textdatei findet man ein Resümee der Endwerte, gefolgt von einer Beschreibung des Lösungsvorgangs einer Aufgabe.

Für die Datenauswertung werden hauptsächlich die ".csv"-Dateien verwendet. Solche Dateien können in Programmen, die speziell für die Datenverarbeitung und Analyse dienen, eingelesen und verarbeitet werden. Die Daten einer Aufgabe einer Runde werden tabellarisch hinterlegt, damit sie später in Excel eingelesen werden und ausgewertet können.

3.2.4. Quellcode Zusammenfassung

Die verwendete Programmiersprache ist C#. In der Verwendung mit der Unity Engine wird auch dazugehörige Namespaces verwendet die wichtige Klassen und Funktionen zum erstellen eines Spiels anbieten (<https://docs.unity3d.com/ScriptReference/>). Dazu gehört die Klasse MonoBehaviour, von der einige Klassen erben um in der Spielschleife eingebunden zu sein. Diese Skripten müssen einem GameObject im Editor zu gewiesen werden. Wenn ein Klasse zu greifen UnityNetworking soll, muss sie von NetworkBehavior erben anstatt von MonoBehaviour. Die wichtigsten, selbst erstellten C#-Skripten werden im folgenden zusammengefasst und ihr Nutzen beschrieben.

GameManager.cs

Diese Klasse verwaltet die Einstellungen mit den dazugehörigen Variablen. Diese Klasse ist als Singleton angelegt. Dies bedeutet es existiert immer eine Instanz dieser Klasse und es kann aber auch nie mehr als eine geben. Dieses Objekt kann von allen anderen Skripten referenziert werden, daher beinhaltet diese Klasse Variablen, die den Status des Spiels beschreiben, wie zum Beispiel der aktuell gewählte Spielmodus. Zudem werden in der Datei die Helferklassen ParticipantData, TaskData, GeneratedTasks und OptionsData deklariert, diese werden für die Strukturierung und zum persistenten Speichern der Informationen benötigt. So bietet GameManager je eine Funktion die Variablen der Einstellungen zu sichern und zu laden. Eine andere Funktion die diese Klasse bietet, ist das vorab generieren von Aufgaben für eine Spielrunde. Die von GenerateTask() erstellten Aufgaben können mittels LoadGeneratedTasks() geladen werden und so dann in GameLogic.cs verwendet werden. Ein weiterer Dienst, der hier gestellt wird, ist CreateUserData(ParticipantData participant). Hiermit werden Dateien angelegt die als Logbuch einer Spielrunde dienen.

GameLogic.cs

Diese Klasse beinhaltet die Logik des Mathespiels. Zu Beginn wird die Coroutine StartCountdown() aufgerufen um den Start der Hauptschleife in der Update Funktion mit einem Countdown zu verzögern. In der Game-Loop wird überprüft, ob eine Kondition zum Beenden der Spielrunde erfüllt wird. Hierfür wird überprüft ob es eine Eingabe der "Escape"-Taste gab oder die Spielzeit abgelaufen ist. Beim beenden der Runde wird, mit Hilfe der Instanz der GameManager-Klasse, die Runde in definierten Logbuch-Dateien protokolliert, ein Endbildschirm angezeigt und das Hauptmenü geladen. Zudem wird in dieser Schleife die Hilfe für einen Spieler der als "supervisor" in den Einstellungen gesetzt wurde abgehandelt. Dazu gehört der Countdown der anzeigt wann die aktuelle Aufgabe autonom gelöst wird, das Lösen der aktuellen Aufgabe und das schrittweise Füllen des Eingabefelds mit der Lösung. Wenn allgemein eine Aufgabe gelöst wurde wird die Task()-Funktion aufgerufen. Diese Methode erstellt oder lädt die nächste Aufgabe.

Mit der Solve()-Funktion wird die Antwort im Eingabefeld bestätigt, geprüft und dem ent-

sprechend reagiert. So werden leere Eingaben mit null Punkten, falsche mit minus einem und richtige mit plus einem Punkt zu dem aktuellen Punktenstand vergütet. Auch werden die Informationen über den Verlauf der Beantwortung der Aufgabe als TaskData-Objekt in einer Liste eingefügt.

MenuScript.cs

Hier werden die Funktionen des Menüs zur Verfügung gestellt. Einige Funktionen dienen allein dafür in den Einstellungen die Variablen des GameManager zu ändern oder auf dessen Funktionen zu zugreifen. Diese Funktionen werden dann im Editor UI-Knöpfen oder UI-Reglern zugewiesen, sodass sie in der Anwendung eingestellt werden können.

NetworkMenu.cs

Hiermit wird die Netzwerkverbindung erstellt, verwaltet oder getrennt. Um in der Lage zu sein darauf zugreifen zu können hat er den NetworkManger referenziert, dies ermöglicht es einen Host oder Client zu erstellen und die Verbindungsinformationen anzupassen. verschiedene Funktionen wie PlayAsHost() realisieren dies. Mittels setPort(string port) lässt sich die Portnummer und mit SetNetworksddress(string ip) die Netzwerkadresse des Servers eingeben.

NetworkSync.cs

Eine Helferklasse, als Singleton designt ist, mit der die Abläufe einer Netzwerkverbindung synchronisiert werden sollen. Auch erstellt und verwirft sie NetworkObject-Objekte mittels einer Coroutine für Spielrunden in denen weitere Spieler durch falsche simuliert werden sollen.

NetworkObject.cs

Die Repräsentation eines Clients im Spiel. Auch können hier mit falsche Spieler emuliert werden. Das GameObject, dass dieses Skript zugewiesen hat, muss auch die Komponente NetworkIdentity besitzen. Dies ist notwendig damit das Objekt eine eigene Identität im Netzwerk besitzt, wenn es sogar den Client des eigenen Spiels repräsentiert, besitzt es lokale Autorität. Mit der connectionID identifiziert er sich in in einem Spielsitzung. Die Hauptaufgabe dieses Skripts ist dem Server Änderungen des Clients, wie zum Beispiel eine neue Punktzahl, mitzuteilen. Wenn ein Spieler imitiert werden soll, wird das Verhalten für diesen auch hier erzeugt.

3.2.5. Fragebogen

Neben der reinen Beobachtung des Spielverlaufs sind auch Ausagen und Eindrücke der Spieler nützlich um Daten in Kontexte zu stellen und zu Interpretieren. Hierfür eignet sich die Beantwortung eines Fragebogens durch die Teilnehmer. Der Fragebogen lässt sich

in drei Abschnitte aufteilen, der erste Part ist zu beginn, also vor dem ersten Spielen, zu beantworten, gefolgt von Abschnitten die sich mit dem Spielen eines Modus abwechseln und endet mit einem Teil als Resümee.

Prä-Spiel

Hier wird der Teilnehmer befragt wie er seine Fähigkeit der Kopfrechnungen durchzuführen einschätzt, dies kann man später in Relation mit dem Erfolgserlebnissen setzen. Um diese Fest zu stellen wird der Gefühlzustand auch in diesem Abschnitt abgefragt, um einen Nullwert für die vergleiche zu besitzen. Die Gefühlslage soll vom Spieler insofern angegeben werden ob er sich zu dem Zeitpunkt des Beantwortens gestresst oder erleichtert fühlt. Auch wie die Nutzer zu Wettbewerb in verschiedenen Kontexten sehen wird gefragt.

Spiel

Dieser Abschnitt wiederholt sich vier Mal, da diese Fragen nach jeder Runde, die gespielt wurde, zu dem jeweiligen Modus gestellt werden. Dazu gehört wie der Modus gefallen hat, wie die Zeitdauer war, ob der Modus erneut gespielt werden würde und die Gefühlslage soll angegeben werden.

Post-Spiel

Zum Schluss soll bewertet werden wie das Spiel im Gesamten gefallen hat und die Modi sollen untereinander eingestuft werden, auch soll erneut angegeben werden welche Modi erneut gespielt werden würden.

3.3. Aufbau

Für einen Durchlauf wird ein einzelner Teilnehmer in den Versuchsaufbau gebeten. Es sind zwei Computer aufgebaut auf denen das Spiel läuft und die über das Netzwerk miteinander verbunden sind. Zunächst füllt der Proband den ersten Teil des Fragebogen aus. Anschließend spielt er einen Modus und beantwortet danach den zugehörigen Teil des Fragebogens. Dies wiederholt er bis er alle Modi gespielt hat. Die Reihenfolge der Modi für jeden Teilnehmer durch, sodass erst bei mehr als 24 Teilnehmern die Reihenfolge sich wiederholt. Die verschiedenen Spielszenarien haben unterschiedliche Aufbauten, die alle gemein haben, dass der Proband einen Bildschirm zur Wiedergabe des Spiels und Maus und Tastatur zur Eingabe vor sich hat. In der Einzelspielervariante ist dies bereits die gesamte Aufstellung und der Betreuer dient ausschließlich als Aufsicht. Ausschließlich der "Singleplayer"-Modus nutzt diesen Aufbau. Die nächste Konstellation ist für alle Modi die mehr als einen Spieler benötigen. Besitzt jeder seinen eigenen Grundaufbau, dieser ist von den anderen Spieler nicht einsehbar. Der Betreuer kann hierbei auch ein Mitspieler sein, in diesem Fall kann er die Bildschirme der anderen einsehen. Der "Halb-Coop"-Modus hat

einen leicht abgewandelten Aufbau, den die Bildschirme sollen hier für die Gegenspieler sichtbar sein. Da es organisatorisch leichter abzustimmen ist, werden die alle Modi mit maximal zwei echten Spielern gespielt. Der Opponent wird in jeden Fall die Versuchsleitung sein, somit ist der Gegner eine Konstante in der Studie. Da die Versuchsleitung alles Begutachten muss, darf sie die das Spiel und die Eingabe des Spielers einsehen. Im "Halb-Coop"-Aufbau wird das Spiel der Aufsicht mittels eines zweiten Bildschirms für den Probanden sichtbar. Nach dem vier Modi gespielt wurden muss der Teilnehmer den letzten Abschnitt des Fragebogens ausfüllen.

3.3.1. Beobachtungsgruppen

Für die Untersuchung wurden keine speziellen Kriterien für die Wahl der Teilnehmer im vor hinein gefällt, denn organisatorisch ist im gegebenen Rahmen eine breit differenzierte Teilnehmerauswahl problematisch. Daher werden Gesichtspunkte wie zum Beispiel Altersgruppen, Geschlecht und Bildungsniveau, in der Datenauswertung nicht beachtet. Die Probanden werden versucht über direktes Ansprechen von möglichen Teilnehmern und über Plakate versucht zu an zu werben. Angestrebt werden insgesamt um die 30 Nutzer, diese würden sich in eine kleine Gruppe für eine Vorstudie und eine Gruppe mit circa 24 Teilnehmern für die Hauptstudie aufteilen. Die Zahl 24 ergibt sich durch das Permutieren der Reihenfolge der vier Testszenarien.

4. Vorstudie

Das Ziel der Vorstudie ist mögliche Fehler im Spiel und dem Testverlauf ausfindig zu machen und erste Eindrücke zu sammeln aus denen für die Hauptreihe gelernt werden kann. Außerdem soll anhand der gewonnenen Daten ein Durchschnittsspieler erfasst werden, der dann als programmierte Gegner in der Hauptstudie dienen soll.

4.1. Planung

Die Vorstudie soll im deutlich kleineren Rahmen als die Hauptstudie stattfinden um den Aufwand zu reduzieren. Die Testumgebung soll allerdings im besten Fall, der der Finalen entsprechen. Es sollen vier bis sechs Teilnehmer das Spiel spielen und den Fragebogen ausfüllen. Getestet werden 4 Modi der "Singleplayer", "Halb-Coop", "Versus" und "Party". Jeder Spieler spielt die Modi in einer anderen Reihenfolge. Die Testumgebung ist wie in Abschnitt 3.3 beschrieben. Der "Party"-Modus wird für vier Spieler ausgelegt, dass heißt zwei Spieler sind falsche Spieler, die durch das Spiel erzeugt werden. Dies wird dem Studienteilnehmer auch vermittelt, allerdings kann er im Spiel nicht zwischen den künstlichen und dem menschlichen Gegner unterscheiden.

4.2. Evaluation

4.2.1. Teilnehmer

Es wurden fünf Teilnehmer für die Versuchsreihe akquiriert. Alle Teilnehmer waren in der Altersgruppe 18-29. Alle Probanden studieren, mit einer Ausnahme die eine abgeschlossene Berufsausbildung nachweisen kann. Die Geschlechteraufteilung ist nicht ausgewogen.

4.2.2. Ergebnisse und Ablauf

Wie zu erwarten gab es hin und wieder Probleme mit dem Programm. Sobald ein Fehler im Ablauf aufgetreten ist, wurde die Runde abgebrochen und neu gestartet. Gegebenenfalls wurde dies wiederholt bis eine fehlerfreie Runde gespielt werden konnte. Die Bugs wurden notiert und schnellst möglich behoben.

Bei der Befragung ob sie Wettbewerb als hinderlich oder förderlich empfinden, sagten die Teilnehmer, dass es eher förderlich sein kann, dies am ehesten in sportlichen Kontexten. Diese Aussage lassen die Vermutung aufkommen, dass Wettbewerb einen positiven Einfluss haben könnte, wenn nicht in der Ausübung zumindest in der Empfindung einer Situation.

Die Teilnehmer wurden nach jeder Runde gefragt, ob sie sich gestresst oder erleichtert fühlen. Die Probanden haben sich im Mittel tendenzielle eher erleichtert gefühlt. Als Nullwert für die Vergleiche dienen unter den Runden dient die erste Runde. Mit der Annahme, dass sich die Gefühlslage nicht zwischen den Runden verändert, wurden T-Tests durch-

geführt. Die Ergebnisse widerlegen die Nullhypothese nicht. Auch korrelieren die Werte eher weniger bis auf die dritte Runde, diese steht aber auch nicht in vollständiger lineare Korrelation zur ersten Runde.

Die folgenden Auswertungen richten sich nach dem Mittelwert der Antworten aller Spieler. Der "Singleplayer"-Modus wurde mit einer eher guten mittleren Bewertung eingestuft. Die Nutzer fühlten sich eher erleichtert. Die Spieldauer wurde als eher kurz empfunden. Der "Halb-Coop"-Modus wurde als eher mittelmäßig bewertet. Nach beenden der Runde waren die Probanden eher erleichtert. Wie im ersten betrachteten Modus wurde, die Zeit als eher kurz betrachtet. Die Bewertung des "Versus"-Modus waren höher als die der zuvor gegangenen Modi, er wurde als her gut bis gut gesehen. Auch fühlten sich die Teilnehmer erleichtert nach dem Spielen. Die Zeit wurde kürzer als in dem Einzelspielermodi empfunden. Der "Party"-Modus wurde

5. Hauptstudie

5.1. Anpassungen

Im Spiel gefundene Fehler werden behoben. Zudem werden die Eingang- und Austrittsanimationen für die Aufgaben herausgenommen, da diese den flüssigen Spielablauf gestört haben. Durch diese Änderung können die Spielerdaten nicht für die Erzeugung eines Durchschnittsspieler genutzt werden. Als Ersatz wird eine der künstliche Gegner für die Hauptstudie eine "Sliding Window"-Methode verwenden, um die Zeiten zu generieren in der er agiert. Der artifizielle Spieler antwortet immer in Mittel der letzten drei Zeiten, in denen sein Gegner Aufgaben gelöst hat. Für die Anfangszeiten werden die Durchschnittszeiten der Vorstudienteilnehmer verwendet. Dieser Spieler wird nun immer anstelle des Betreuers spielen, um den Gegner als eine Konstante in den Szenarios zu setzen. Auch werden andere Modi nun in der Hauptstudie verwendet. Der "Party"-Modus wird durch "Versus 2" getauscht, da sich dieser direkter mit "Versus" vergleichen lässt. Außerdem wurde das Laden des Menübildschirms am Ende einer Runde verzögert, damit man seinen finalen Punktzahl besser wahrnehmen kann und gegebenenfalls mit der des Gegners vergleichen kann. Eine erneute Vorstudie lies sich organisatorisch nicht vereinbaren, sodass der angepasste Versuchsaufbau direkt in der Hauptstudie genutzt wird.

Weitere Anpassungen zur Verbesserung des Versuchsablaufs wurden durchgeführt, wie das Umstellen der Ordnerstruktur der Logbuchdateien. Im Fragebogen wird im ersten Abschnitt bereits die Gefühlslage abgefragt, damit ein Wert vor der ersten Spielrunde vorhanden ist. Auch wurde eine Spieldemo als Video angefügt, sodass anhand dieser das Spiel besser erklärt werden kann. Zudem werden die Fragen ins Englische übersetzt und eine leichte Änderung der Darstellungsform wurde vorgenommen, dies hatte aber keine weiteren inhaltlichen Änderungen zu folge.

5.2. Evaluation

5.2.1. Teilnehmer

20 Personen nahmen an der Hauptstudie teil, die gewünschten 24 Teilnehmer konnten nicht in der Laufzeit geworben werden. Die Probanden waren zwischen 18 und 29 Jahren alt. Das Bildungsniveau unter den Teilnehmern war vergleichbar, alle waren Studenten. Auch in der Hauptstudie war die Verteilung der Geschlechter der Teilnehmer stark einseitig.

5.2.2. Ergebnisse

6. Diskussion

7. Ausblick



Abbildung 3 Netzwerkmenü



Abbildung 4 Optionen

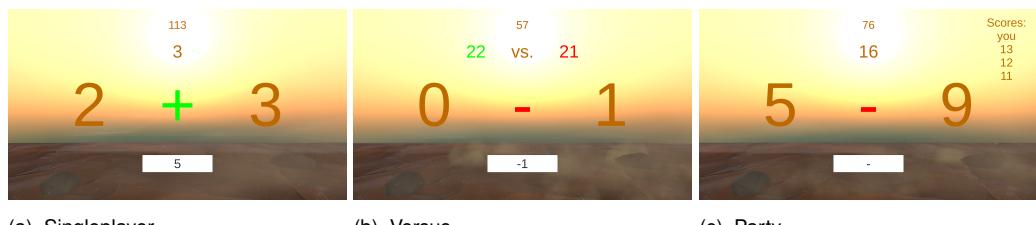


Abbildung 1 Bildschirmaufnahme der Modi



Abbildung 2 Ansichten des Hauptmenüs

Abbildungsverzeichnis

Abbildung 3 Netzwerkmenü	17
Abbildung 4 Optionen	17
Abbildung 1 Bildschrimaufnahme der Modi.....	17
Abbildung 2 Ansichten des Hauptmenüs	18