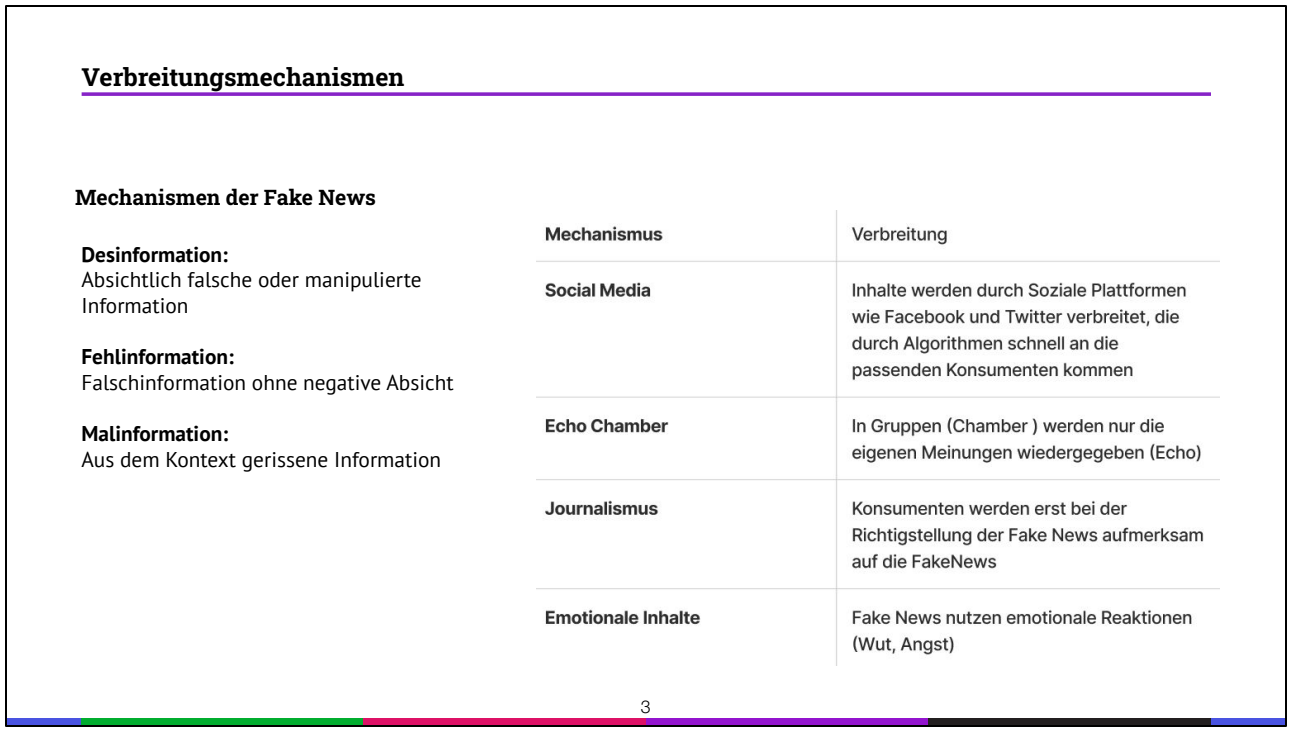
Audit 2

Visual Computing - Entwicklungsprojekt WS24/25
Lukas Diekmann, Amin Merzouk und Morris Schacht
Team Anya

Nachtrag zu Audit 1

Mechanismen, Emotionen, Forschungsfrage

Zu Beginn ein kleiner Nachtrag, zu Kritik- bzw. Verbesserungspunkten aus Audit 1. Es wurden ebenfalls weitere Recherchen innerhalb der Domäne zu den Eltern und Lehrkräften gemacht. Allerdings in neuen anschaulichen Artefakt, welches in diese Präsentation passt, sondern stattdessen in einer weiteren Iteration des Domänenmodells . Die neuen Erkenntnisse werden allerdings bei der weiteren Entwicklung berücksichtigt.



Um Fake News zu verbreiten, werden bestimmte stilistische Mittel verwendet. Ausgehend vom Verbreiter der Fake News handelt es sich dabei um Desinformation, Fehlinformation und Malinformation. Liest der Konsument z. B. einen Artikel, findet er oft Teilwahrheiten, was dazu führt, dass Fake News glaubhaft wirken.

Social Media: Fake News sind häufig in sozialen Medien zu finden. Durch personalisierte Algorithmen erreichen Teilwahrheiten die passenden Nutzer. Das Teilen und Kommentieren solcher Beiträge unterstützt die Verbreitung. Zudem erhält der Nutzer ähnliche Inhalte, was zu einem Confirmation Bias führt (er fühlt sich in seiner Meinung bestätigt).

Echo Chamber: Die "Echokammer" (engl. Echo Chamber, sinnbildlich für Hallraum) beschreibt Gruppen, in denen selektive Informationen ohne wissenschaftliche Grundlage umhergehen. Mitglieder teilen meist ähnliche Meinungen und Ideologien.Das führt dazu dass Informationen die die entsprechende Ideologie unterstützen sich schnell in diesen sogenannten Echo Chamber verbreiten.

Journalismus: Selbst wenn Fake News öffentlich richtiggestellt werden, gelangen diese Informationen oft an ein großes Publikum und fördern, je nach Emotionalität, die Aufmerksamkeit was unfreiwillig zu einer Verbreitung führt

Emotionale Inhalte: Emotionen spielen eine entscheidende Rolle. Je stärker der Konsument emotional involviert wird, desto schwieriger ist es, Fake News

richtigzustellen (z. B.Corona-Pandemie). Diese Art von Verbreitung ist unabhängig von dem Medium und hat auch den größten Impakt das ebenfalls Psychologische Faktoren mitspielen

Quellen:

https://www.boell.de/sites/default/files/2020-08/200825_E-Paper3_DE.pdf /Desinformation Fehlinformation, Malinformation

https://web.stanford.edu/~gentzkow/research/fakenews.pdf / Emotionen

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0159641 / Social Media

https://www.interface-eu.org/storage/archive/files/fake_news_methodenpapier_deutsch.pdf /Journalismus

Emotionen

Emotionen in Fake News

Leser: Je Emotionaler Involviert der Leser ist desto glaubwürdiger wird die Information

Echo Chamber: Gegenseitiges bestätigen von Falschinformationen (Katalysator)

Auswirkung: Fake News werden als wahr wahrgenommen aufgrund des confirmation bias

Bei der Recherche sind und gewisse Emotionale Faktoren bei Fake News aufgefallen. Fake News haben im gegensatz zum relativ neutralen Journalismus stärkere Emotionale rhetoriken die gewisse Emotionen beim Leser auswirken. Unter Anderem werden mehr Adjektive verwendet oder eine gewisse art von sarkastischem Unterton der der Satire ähnelt wobei die Themen mit einer gewissen ernsthaftigkeit durchleuchtet werden z.B. “Papst Franziskus unterstützt Präsidentschaftskandidaten und schockiert die Welt!”. Für den Leser werden bei solchen Artikeln je nach Bindung zum Thema extreme Gefühle ausgelöst wie Wut, Trauer oder Misstrauen. Diese Emotionen führen dazu das irgendwann Artikel Selektiv ausgesucht werden bis ein gewisser grad von confirmation bias entsteht beim Leser entsteht was heißt das nur Artikel die die eigene Meinung bestätigen für richtig empfunden werden. Ein weiterer Katalysator sind Echo Chamber, das sind Gruppen in denen mehrer Personen mit ähnlichen Ideologien und Emotionalen bezug sich gegenseitig bestätigen was dazu führt dass Informationen von außerhalb der Gruppe direkt als falsch wahrgenommen werden.

Quellen:
https://www.tandfonline.com/doi/full/10.1080/21670811.2017.1345645#d1e132
https://www.researchgate.net/profile/Chuan-Guo-5/publication/331543936_Exploiting_Emotions_for_Fake_News_Detection_on_Social_Media/links/5d47deb4299bf1995b6643ff/Exploiting-Emotions-for-Fake-News-Detection-on-Social-

Media.pdf
https://www.mimikama.org/fake-news-und-verschwoerungstheorien-im-internet/

Forschungsfrage

Primäre Forschungsfrage

Wie kann Jugendlichen, durch ein Computerspiel, die Kompetenz der Fake News Erkennung vermittelt werden?

Wir haben uns auf die folgende Forschungsfrage fokussiert: “Wie kann Jugendlichen, durch ein Computerspiel, die Kompetenz der Fake News Erkennung vermittelt werden? ”

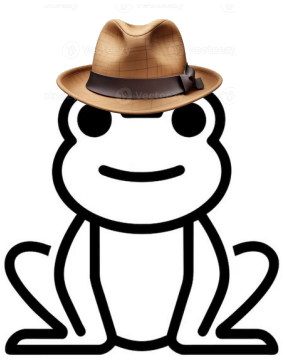
Diese soll im Rahmen des Entwicklungsprojektes beantwortet werden. Die anderen Evaluationspunkte sind auf den Zeitpunkt nach dem Entwicklungsprojekt gerutscht. Mehr dazu in einer späteren Folie.

Gameloop und Pocs

Setting, Gamemechanics, Risiko, Poc, Anforderungen

Setting/Story

“Quibbert Greenwise und das Rätsel der historischen Lüge”:



Worum geht es?
Quibbert Greenwise, ein Detektiv aus der Zukunft reist in die Vergangenheit um Fehler welche in der vergangenheit durch die Verbreitung von Falschinformationen Entstanden sind zu Korrigieren.

Infos zum Setting:
Zeitraum
- Hochmittelalter (ca. 12. Jahrhundert)

Ort:
- Ein abgelegenes Dorf namens Eichenwald

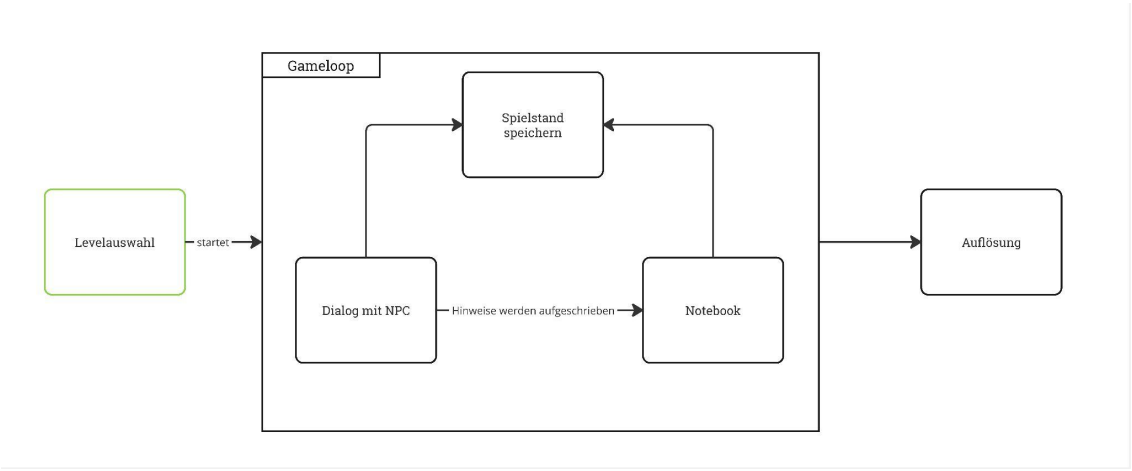
Atmosphäre:
- Das Dorf ist belebt, aber die Gerüchte um den Kometen und die drohende Katastrophe sorgen für eine unruhige, leicht paranoide Stimmung. Manche Dorfbewohner sind abergläubisch, andere skeptisch.

Technologie:
- Typisch mittelalterlich, jedoch mit Quibberts futuristisches Werkzeugen und den Robotieren als Kontrast.

Quibbert Greenwise, ein Detektiv und unser Protagonist, reist in die Vergangenheit um die Verbreitung von Fehlinformationen welche die Zukunft negativ beeinflussen zu verhindert. Diese Fehlinformationen werden von bösartigen Roboter-Tieren aus der Zukunft verbreitet. Im Laufe eines Levels interagiert der Spieler dann mit den verschiedenen Charakteren des dorfes und entlarvt am ende den Täter und verhindert damit einen Fehler in der Zukunft. Charaktere im Dorf sind beispielsweise ein Bauer, ein Zimmermann, der Dorfälteste, eine Jägerin und noch viele mehr. DIese geben dem Spieler jeweils ihre sicht der dinge und helfen ihm gegebenenfalls mit wichtigen Informationen und Hinweisen.

Details zur Geschichte und seinen Charakteren:
https://github.com/Morris-hub/EPWS2425DiekmannMerzoukSchacht/blob/main/Artefakte/WS2425_DiekmannMerzoukSchacht_GeschichteUndCharactere_LVL1.pdf

Gameloop



Der grobe Ablauf eines Levels startet mit der Auswahl eines Levels in einem Menü. In diesem soll der Nutzer die Möglichkeit haben alle bereits gespielten und neue Level einzusehen und anschließend eins Auszuwählen. Wenn der Nutzer sich für ein Level entscheidet und dieses dann auswählt startet dieses im Anschluss und man befindet sich nach einer Ladezeit in der 3D Umgebung und dann sich frei bewegen. In dieser Umgebung sammelt der Nutzer dann, durch Interaktionen mit den verschiedenen Charakteren, Informationen über das geschehen und die Involvierten Charaktere. Der Nutzer soll jederzeit die Möglichkeit haben gesammelte Informationen, in form eines Notizbuchs, einzusehen und für sich zu ordnen. Für alle wichtigen hinweise welche der Nutzer sammelt steigt eine “Ermittlungsanzeige”. Ab 80% erfüllter Ermittlung hat der Nutzer die Option den “Täter” in einer Auflösung zu Entlarven. Liegt der Spieler richtig endet das Level mit einem Erfolg. Falls nicht wird der Spieler zu seinen Ermittlungen zurückgebracht und er hat die Möglichkeit seine Informationen neu zu strukturieren und weiter Hinweise zu sammeln.
Detailliertes Ablaufmodell:
https://github.com/Morris-hub/EPWS2425DiekmannMerzoukSchacht/blob/main/Artefakte/WS2425_DiekmannMerzoukSchacht_Ablaufdiagramm_Level_V1.pdf

Levelauswahl - PoC

Ziel: User soll aus einer Auswahl von Leveln eines durch Klick auswählen und starten können

Exit: User kann ein Level auswählen und dieses durch Anklicken starten

Fail: Level kann nicht ausgewählt werden

Fallback: Start-Button welcher das erste Level startet, folgende Level durch einen Code der dann bis zu einem bestimmten Punkt Level freischaltet/lädt

Bei diesem PoC wurde bewusst kein Risiko formuliert, da dieser bei einem Fail nicht Systemgefährdend ist. Da er aber dennoch einerseits zu einer Übbersichtlicheren Bedienung und andererseits zur Wiederspielbarkeit beiträgt wurde ein PoC verfasst.

Dieser hat das Ziel dem User eine Auswahl an Leveln zu präsentieren welche er dann per Klick auswählen kann. Im Idealfall startet das jeweilige Level, nachdem der User es freigeschaltet und angeklickt hat. Sollte dies nicht der fall sein wäre der Fallback ein Start-Button welcher das erste Level startet und für die folgenden Level ein System welches wenn man das erste Level abschließt einen Code ausgibt welche dann verwendet werden kann um das Zweite Level freizuschalten und zu spielen. Für folgende Level wird dann das selbe Prinzip verwendet.

Levelauswahl - Anforderungen

Funktionale Anforderungen:

- Der User muss aus einer angezeigten Liste von Leveln ein Level auswählen können.
- Die Auswahl des Levels erfolgt durch einen Klick auf das gewünschte Level.
- Die Auswahl muss das Level unmittelbar nach dem Klick gestartet werden.
- Nach Abschluss eines Levels kann der User ein weiteres Level freischalten.
- Wenn ein Level nicht gestartet werden kann, muss eine Meldung angezeigt werden, die den User über Das Problem Informiert.

Nicht-funktionale Anforderungen:

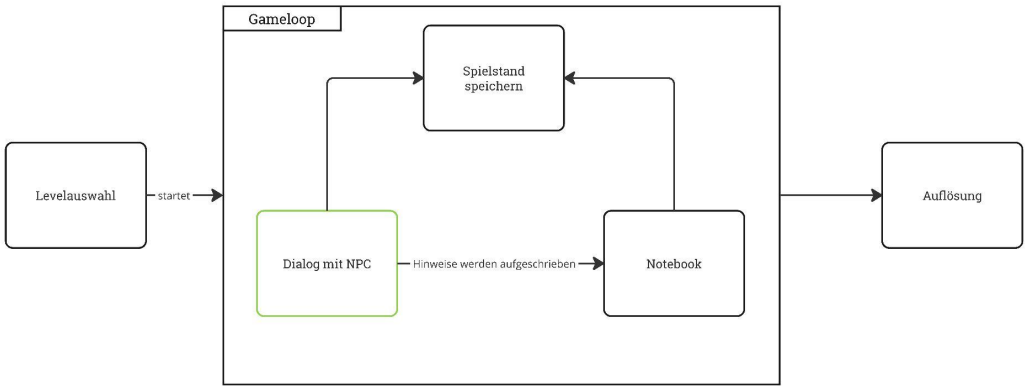
- Die Levelauswahl muss intuitiv gestaltet sein um jedem Nutzer die bedienung zu ermöglichen.

Technische Anforderungen:

- Die Levelauswahl muss als graphische Oberfläche implementiert werden *(z.B. als Liste oder Raster)*.
- Fehler bei der Levelauswahl oder dem Start müssen im Hintergrund protokolliert werden, um die Ursachen nachverfolgen zu können.

Die Anforderungen welche sich aus diesem PoC ergeben sind aufgeteilt in funktional, Nicht-funktional und technisch. Funktionale Anforderungen sind beispielsweise das der User aus der Auswahl Leveln eins Auswählen kann oder dass, das Level durch einen klick ausgewählt und unmittelbar danach gestartet werden soll. Weitere funktionale Anforderungen beziehen sich auf die Freischaltung weiterer Level nach Abschluss des Vorherigen und das falls dies nicht funktionieren sollte eine Meldung ausgegeben wird welche dann zur Identifikation des Problems genutzt werden kann. Nicht-funktionale Anforderungen sind das die Auswahl intuitiv gestaltet sein muss um jedem Nutzer die Bedienung zu ermöglichen. Zusätzlich muss der Fallback Start-Button immer Verfügbar sein um das erste Level zu starten.Die technischen Anforderungen beziehen sich einerseits auf die form der graphischen Oberfläche und auf die Programmierung des Fallbacks. Bei der Oberfläche muss die Auswahl z.B. in Form einer Liste oder einem Raster angezeigt werden. Diese bieten dem User ein übersichtliche Auswahl. Andererseits muss der Fallback-Start-Button so implementiert sein das er funktioniert selbst wenn die ursprüngliche Auswahl dies nicht tut.

Gameloop


Gameloop
Spielstand speichern
Levelauswahl
startet
Dialog mit NPC
Hinweise werden aufgeschrieben
Notebook
Auflösung

Dialog - PoC

Ziel: User kann mit NPC interagieren eine Dialogbox wird angezeigt

Exit: Textbox mit korrektem Dialogtext bei Interaktion mit NPC

Fail: Keine bzw.es werden falsche Dialoge werden angezeigt

Fallback: NPC/Spielgegenstand gibt bereits geschehene Dialoge wieder

Das Ziel des Dialogsystems ist es, dass der Spieler mit NPCs interagieren kann und dabei eine Dialogbox mit korrektem Text angezeigt wird. Somit kann der Spieler die Strukturelle Verbreitung von Fake News im Spielverlauf nachvollziehen Im Normalfall erscheint bei der Interaktion die richtige Textbox mit dem vorgesehenen Dialogtext im schlimmsten fall werden entweder keine oder falsche Dialoge angezeigt. Als Fallback werden über einen NPC bereits geführte oder allgemeine Dialoge wiedergegeben. Was auch den Vorteil hat das der Spieler die Dialoge nochmal für sich durchgehen kann

Dialog - Anforderungen

Anforderung Prototyp entwicklung

- Einfache Integration
- Hohe Flexibilität

Kriterien	Ink	Fungus	Yarn Spinner
Komplexität	Hoch	Niedrig	Mittel
Usability	Skriptbasiert	Drag & Drop	Skriptbasiert
Flexibilität	Sehr Hoch	Mittel	Hoch
Integration	Plugin	Github integration	Github integration
Einsatzbereich	Komplexe Handlungen	Schnelle Prototypen	Verzweigte Dialogbäume

Aufgelistet sind kostenlose Textmanagementsysteme, die für dialogbasierte Spiele verwendet werden. Die Bewertung erfolgt aus der Perspektive eines Unity-Anfängers, um unterschiedliche Wissensstände zu berücksichtigen. Zusätzlich wurden die Verfügbarkeit von Tutorials sowie die Verständlichkeit der Dokumentationen in die Bewertung einbezogen.

Ink ist ein natives Plugin, das über den Unity Asset Store heruntergeladen werden kann. Die hohe Flexibilität ergibt sich aus der skriptbasierten Konfiguration. Als Textformat wird Markdown verwendet, das eine einfache Anpassung bei der Formatierung (z. B. fett oder kursiv schreiben) ermöglicht. Online stehen leicht verständliche Kurzvideos zur Verfügung. Die einzige Hürde ist das Programmieren, das mit einer gewissen Einarbeitungszeit verbunden ist.

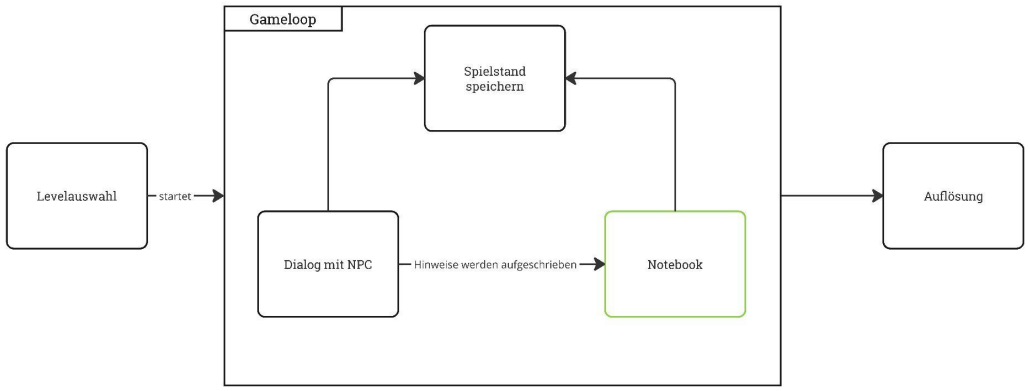
Fungus ist ein Tool, das mit Drag-and-Drop funktioniert. Im Unity Store ist das Plugin nur für ältere Unity-Versionen verfügbar. Alternativ kann eine neuere Version über GitHub installiert werden. Komplexe Verzweigungen sind möglich, allerdings ist Fungus nicht optimal dafür ausgelegt, was zu einer gewissen Unübersichtlichkeit führen kann. Vom Anbieter werden Dokumentationen und Tutorials bereitgestellt, diese sind jedoch oft veraltet oder nicht mehr verfügbar. Das Tool scheint in der Usability einfach zu sein, jedoch besteht das Risiko, bei Problemen auf sich allein gestellt zu sein.

Yarn Spinner ist kostenpflichtig im Unity Store verfügbar, kann aber wie

Fungus ebenfalls über GitHub integriert werden. Die Dokumentationen des Anbieters sind klar verständlich, allerdings sind nur wenige Tutorials von Drittanbietern verfügbar. Das Tool selbst ist eher für komplexe Verzweigungen ausgelegt und eignet sich somit hervorragend, um später komplexere Szenarien hinzuzufügen. Das Problem bei diesem Tool ist jedoch, dass es schwierig sein kann, Hilfestellungen bei auftretenden Problemen, insbesondere beim Skalieren, zu finden, da die Community relativ überschaubar ist.

Quellen:
https://assetstore.unity.com/packages/tools/integration/ink-integration-for-unity-60055
https://fungusgames.com/
https://v1.yarnspinner.dev/docs/unity/installing/

Gameloop


Gameloop
Spielstand speichern
Levelauswahl
startet
Dialog mit NPC
Hinweise werden aufgeschrieben
Notebook
Auflösung

Notebook - PoC

Ziel: Nachdem Infos gesammelt wurden, sollen relevante Informationen im Notizbuch angezeigt werden

Exit: beim öffnen des Notizbuchs werden durch Dialoge erlangte Informationen angezeigt

Fail: Informationen werden fehlerhaft/gar nicht angezeigt

Fallback:
1. Wichtige Informationen werden am Ende einer Konversation eingeblendet
2. Helfer welcher einem Ermöglicht bereits abgeschlossene Interaktionen und ihre wichtigsten Erkenntnisse einzusehen

Für diesen PoC wurde ebenfalls kein Risiko formuliert, da das Feature eines Notebooks dem Nutzer zwar eine Intuitive Möglichkeit bietet gesammelte Informationen zu strukturieren aber bei einem Fail nicht die das System gefährdet und einfach zu umgehen ist.Dennoch wurde ein PoC und damit einhergehen Anforderungen Formuliert. Das Ziel des Notebooks oder Notizbuchs ist es dem User relevante Informationen die gesammelt wurden anzuzeigen. Dabei wäre ein Erfolg wenn der User das Notizbuch öffnet die Infos angezeigt werden. Sollten diese entweder fehlerhaft oder gar nicht angezeigt werden würde man sich auf 1 von 2 oder gar beide Fallbacks stützen. Der erste wäre eine Anzeige der wichtigsten Informationen einer Konversation am ende dieser in form einer Einblendung. Der zweite ist ein Helfer oder Begleiter welcher einem die Möglichkeit bietet bereits abgeschlossenen Interaktionen und ihre wichtigsten Informationen erneut zu besuchen oder einzusehen.

Notebook - Anforderungen

Funktionale Anforderungen:

- Das Notizbuch soll beim Öffnen alle relevante Informationen anzeigen , die der User durch Dialoge gesammelt hat
- Die Informationen müssen nach Themen, Zeit oder Relevanz sortiert sein, um die Übersichtlichkeit zu gewährleisten
- Nach Abschluss eines Dialogs sollen die gesammelten Informationen automatisch im Notizbuch gespeichert werden

Nicht-funktionale Anforderungen:

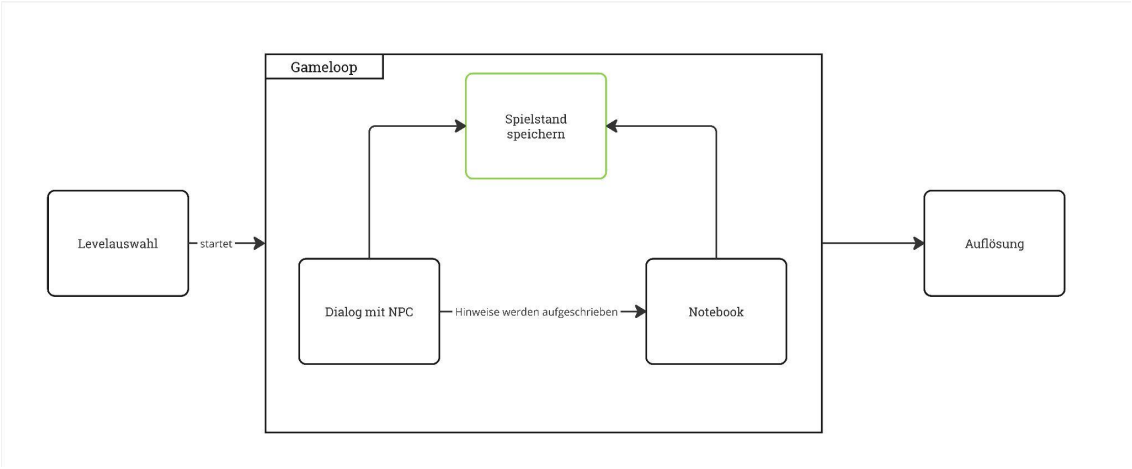
- Das Notizbuch muss mit einer einfachen Aktion schnell zugänglich sein
- Informationen müssen klar und übersichtlich angezeigt werden (z.B. mit Absätzen, Überschriften oder Symbolen zur Hervorhebung)
- Das Notizbuch darf die Gesamtperformance des Systems nicht negativ beeinflusse (z.B. durch lange ladezeiten)

Technische Anforderungen:

- Das Notizbuch muss als UI-Komponente implementiert sein, die sich bei Bedarf öffnen lässt
- Daten müssen persistent gespeichert werden, sodass Informationen auch nach Neustart verfügbar sind

Aus diesem PoC ergeben sich dann diverse Anforderungen welche wie vorher auch in 3 Kategorien unterteilt sind. Die funktionalen Anforderungen sind beispielsweise dass, die Informationen beim öffnen einerseits Angezeigt werden und das dies in einer strukturierten Form geschieht. Zusätzlich soll dieser Prozess automatisch nach Konversationen mit NPCs geschehen. Die Nicht-funktionalen Anforderungen sind dass, das Notizbuch einerseits schnell, und übersichtlich einzusehen sein muss, da es das Hauptwerkzeug des Spielers Darstellt. Zusätzlich dazu darf das Notizbuch die Gesamptperformance des Systems nicht mit langen Ladezeiten negativ beeinflussen. Die Technischen Anforderungen beziehen sich Einerseits auf die Spezielle UI-Komponente welche das Notizbuch darstellt und andererseits auf die reibungslose Speicherung von persistenten Daten um sicherzustellen das der Spieler keinen fortschritt verliert.

Gameloop


Gameloop
Spielstand speichern
Levelauswahl
startet
Dialog mit NPC
Hinweise werden aufgeschrieben
Notebook
Auflösung

Speichern - PoC

Ziel: Der Spielstand wird aktuell gehalten und der Spieler kann das Spiel vom letzten Spielstand fortführen

Exit: User führt das Spiel vom letzten Speicherpunkt vor dem verlassen des Spiels fort

Fail: Fortschritt wird beim verlassen des Spiels nicht gespeichert

Fallback:
1. Festgelegte Speicherpunkte
2. Speicherstand wiederherstellen durch z.B. einen Code

Das Ziel des Speichersystems besteht darin, den Spielstand stets aktuell zu halten, sodass der Spieler das Spiel jederzeit vom letzten Speicherpunkt aus fortführen kann. Der Spieler soll beim Wiedereinstieg die Möglichkeit haben, genau dort weiterzuspielen, wo er zuletzt aufgehört hat.Der geplante Ablauf sieht vor, dass der Nutzer das Spiel genau an dem Speicherpunkt wieder aufnimmt, der zuletzt vor dem Verlassen gespeichert wurde. Sollte dieses Ziel nicht erreicht werden, würde der Spielfortschritt verloren gehen. Um solche Szenarien zu vermeiden, sind fesgelegte Speicherpunkte vorgesehen die der Spieler nutzen kann. Diese stellen sicher, dass der Fortschritt nicht vollständig verloren geht, selbst wenn das automatische Speichern nicht erfolgreich ist. So wird ein grundlegender Fortschritt wiederhertellbar sein über einen Code bei dem das Spiel einen entsprechenden Spielstand erstellt

Speichern - Anforderungen

Funktionale Anforderungen:

- Automatisches Speichern von neuen Informationen nach Abschluss eines Dialogs
- Speicherung in einer persistenten Datenstruktur, damit Daten nach Neustart verfügbar bleiben

Nicht-funktionale Anforderungen:

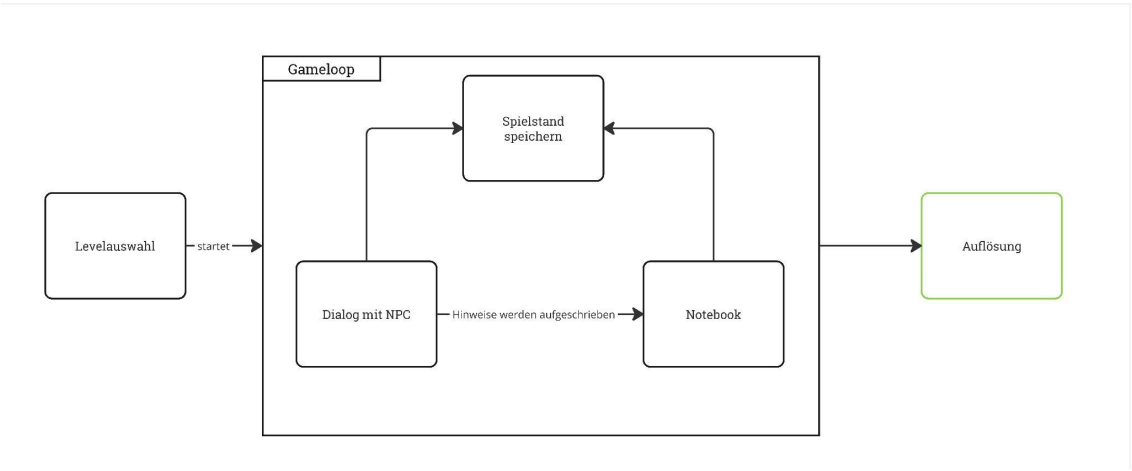
- Schnelle Speicherung ohne Verzögerung.
- Klare Organisation der Daten, damit das die gespeicherten Informationen leicht aufgerufen werden können

Technische Anforderungen:

- Speicherung erfolgt in einem Textbasierten Format, die einfach lesbar und bearbeitbar ist
- Die Speicherlogik ist modular und unabhängig vom UI- oder Dialogsystem

Zu den funktionalen Anforderungen gehört, dass neue Informationen automatisch nach Abschluss eines Dialogs gespeichert werden. Diese Daten werden in einer persistenten Datenstruktur hinterlegt, sodass sie auch nach einem Neustart wieder verfügbar sind. Bei den nicht-funktionalen Anforderungen stellen wir sicher, dass die Speicherung schnell und ohne merkbare Verzögerung erfolgt. Die Daten sind klar organisiert, um einen einfachen und effizienten Zugriff zu ermöglichen. Auf technischer Ebene wird die Speicherung in einem textbasierten Format (z.B JSON oder Markdown) durchgeführt. Diese Formate sind leicht bearbeitbar und benötigen kein großes Hintergrundwissen. Außerdem ist die Speicherlogik modular aufgebaut, was bedeutet, dass sie unabhängig vom UI oder anderen Systemkomponenten funktioniert bzw. das wir festlegen können an welchen Punkten und wie wir Speichern.

Gameloop


Gameloop
Spielstand speichern
Levelauswahl
startet
Dialog mit NPC
Hinweise werden aufgeschrieben
Notebook
Auflösung

Auflösung - PoC

Ziel: Der Spieler bewertet die Information als Wahr oder Falsch basierend auf den gesammelten Informationen

Exit: Die Spielmechanik überprüft gesammelte Hinweise und zeigt bei korrekter Entscheidung die Auflösung

Fail: Die Bewertungskriterien greifen nicht korrekt.

Fallback:
1.Alle Hinweise werden in einer Liste gespeichert und können vom Spieler erneut eingesehen werden
2. Es wird statisch eingetrage ob eine falsche oder richtige Lösung gewählt wurde

Der Spieler sammelt während des Spiels Hinweise, um Informationen als wahr oder falsch zu bewerten. Bei einer korrekten Entscheidung überprüft die Spielmechanik automatisch die gesammelten Hinweise und zeigt sofort Feedback, das dem Spieler erklärt, warum die Entscheidung richtig war. Im Fehlerfall, wenn die Bewertungskriterien nicht korrekt greifen oder die Entscheidung falsch ist, wird eine klare Erklärung angezeigt, die dem Spieler hilft, die falsche Beurteilung zu verstehen. Als Fallback werden alle Hinweise in einer Liste gespeichert, sodass der Spieler diese erneut einsehen kann, um die Informationen zu überprüfen und eine fundierte Entscheidung zu treffen. Sollte dies nicht greifen, kann anhand der Antwortmöglichkeiten entschieden werden, ob die richtige Lösung gewählt wurde, und eine Erklärung, welche Indizien es gab, wird angezeigt.

Auflösung - Anforderungen

Funktionale Anforderungen:

- Alle gesammelten Hinweise müssen automatisch gespeichert werden, nachdem ein Dialog abgeschlossen ist
- Das System bewertet automatisch, ob die gesammelten Hinweise eine Fake News bestätigen oder entkräften
- Alle relevanten Hinweise und die Entscheidung des Spielers müssen für die spätere Analyse zugänglich sein

Nicht-funktionale Anforderungen:

- Die gespeicherten Informationen müssen strukturiert und leicht zugänglich sein, um schnell darauf zugreifen zu können
- Die gespeicherten Daten dürfen nicht verloren gehen und müssen auch beim Neustarten des Spiels korrekt geladen werden

Technische Anforderungen:

- Die gespeicherten Daten müssen in einer strukturierten, textbasierten Form (z. B. JSON oder XML) abgelegt werden, die einfach zu bearbeiten und zu analysieren ist
- Die Speicherung der Hinweise und Entscheidungen erfolgt durch eine separate, modulare Logik.
- Das System muss in der Lage sein, Daten flexibel zu erweitern, ohne die bestehende Struktur zu verändern

Aus dem Proof of Concept zur Auflösung von Fake News ergeben sich ebenfalls mehrere Anforderungen, die in funktionale, nicht-funktionale und technische Kategorien unterteilt sind. Der Spieler soll in der Lage sein, anhand der gesammelten Informationen zu entscheiden, ob es sich um Fake News handelt oder nicht. Die nicht-funktionalen Anforderungen stellen sicher, dass die gespeicherten Hinweise zugänglich sind. Das System muss eine benutzerfreundliche Möglichkeit bieten, alle gesammelten Informationen zu prüfen, um eine fundierte Entscheidung treffen zu können. Die technischen Anforderungen betreffen die strukturierte Speicherung der Daten in einem textbasierten Format sowie die Modularität der Logik, die eine flexible Erweiterung des Systems ermöglicht.

Aussichten

Evaluation, Langzeitmotivation, Zeitplan A3

Evaluation

Zielgruppe Jugendliche
Das Spiel sollte auf die Zielgruppe Jugendliche zugeschnitten sein

Usability
Die Spielmechaniken sind einfach und intuitiv zu verstehen

Fake News Bekämpfung in Schule
Wie kann das Spiel in Schulen eingesetzt werden zum fördern kritischen Denkens bei Jugendlichen

Das Spiel soll hinsichtlich seiner Eignung für Jugendliche evaluiert werden. Dabei ist zu prüfen, ob Design, Sprache und Themen die Zielgruppe ansprechen. Tests mit Jugendlichen sollten zeigen, ob diese sich vom Spiel angesprochen fühlen und ob ihre Interessen ausreichend berücksichtigt werden.

Die Usability wird bewertet, indem untersucht wird, ob die Spielmechaniken leicht verständlich und intuitiv bedienbar sind. Besondere Fokus liegt auf einer einfachen Benutzerführung, die den Usern einen reibungslosen Start ins Spiel ermöglicht. Dazu könne freiwillige erste Tester herangezogen werden die unter Beobachtung das Spielen und ihre Erfahrungen kommentieren.

Das Thema "Fake News bekämpfen in der Schule" ist eine Domäne, die wir in Zukunft beleuchten möchten. Da dies den Rahmen des Projekts überschreiten würde, liegt der Fokus vorerst auf Jugendlichen allgemein. Zur Evaluation in diesem Bereich gehört das Erkennen der Effektivität beim Verstehen und Anwenden von kritischem Denken bei Schülern, da dies sowohl beim Identifizieren von Fake News als auch im akademischen Bildungsweg essenziell ist. Die Evaluation umfasst Tests der Szenarien auf Realitätsnähe, die Reaktionen der Zielgruppe sowie die Unterstützung durch begleitende Materialien für den schulischen Kontext.

Langzeitmotivation

RAMP für intrinsische Motivation

Was ist RAMP?
Relatedness, Autonomy, Mastery, Purpose

Die 6 Spieler-Typen
Achiever, Socialiser, Philanthropist, Free Spirit
+ Player
+ Disruptor

Um die Langzeitmotivation der Spielenden zu sichern, greifen wir in den Werkzeugkasten der Gamification. Die Motivationstheorie nach Marczewski ist hier ein guter Ausgangspunkt, da er nach verschiedenen Spielertypen unterscheidet. Diese haben alle eigene Motivation und Beweggründe für ihre Handlungen. Ziel ist es, ein System zu entwickeln, welches für alle Spielertypen ansprechend ist. Daher wird eine neue Kontrollinstanz neben unserer Persona eingeführt, die 6 Spielertypen.

Nach Vollendung des vertikalen Prototypens wird das System aus der Perspektive aller Typen betrachtet, um mögliche Schwachstellen im Design frühzeitig zu finden.

Quellen:

https://www.gamified.uk/

Marczewski, A. (2015). User Types. In *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design* (1st ed., pp. 65-80).

Zeitplan Audit 3

Hauptaufgabe	W1	W2	W3	W4	W5	W6	wer?
Audit3							Alle
PoC1 - Levelauswahl							Morris
PoC2 - Dialoge							Lukas, Morris
PoC3 - Notebook							Lukas, Amin
PoC4 - Speichern							Morris
PoC5 - Auflösung							Lukas, Morris
Leveldesign							Morris, Amin
Sprites							Amin, Lukas
Story							Amin
Poster?							Alle
Spaß haben							Alle

Für Audit 3 steht die Umsetzung der PoCs im Vordergrund, weshalb sich die Zeitplanung nach genau diesen richtet. Es wurde darauf geachtet, non-PoC Aufagben, zeitlich passend zu platzieren. So muss die Geschichte vor dem Dialogsystem geschrieben sein oder das Leveldesign angefangen worden sein, bevor die Dialoge mit NPCs implementiert werden können.
Der angegebene Zeitplan ist eine grobe Darstellung, da sich die einzelnen PoCs in weitere Unteraufgaben aufteilt, welche über den gesamten Bearbeitungszeitraum der Aufgaben spannen werden. Nach erhalten des Feedbacks für Audit 2 steht eine weitere Iteration des Zeitplans an.