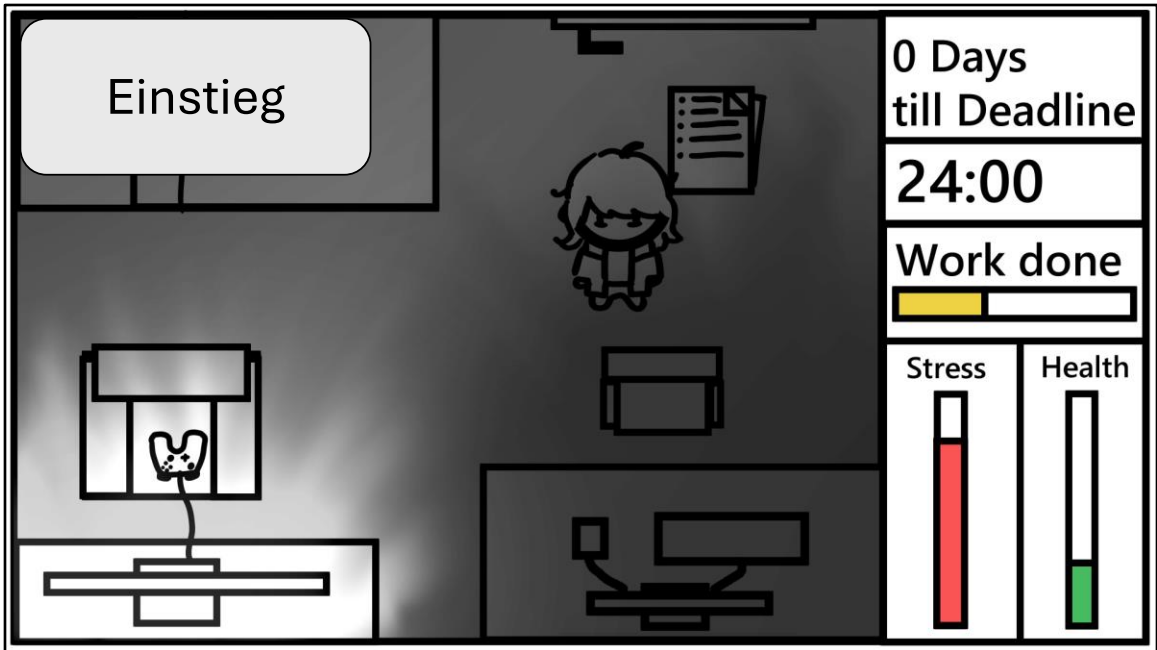


EP Audit 2

Robin Schlegel und Ben Pritzkau

Willkommen zu Audit 2.

Wir geben Ihnen gleich einen Wiedereinstieg in das Projekt. Darauf folgend besprechen wir beispielhaft die Risiken und damit verbunden die PoCs. Im Anschluss gehen wir noch kurz auf den aktuellen Projektplan ein und geben einen kurzen Ausblick auf den nächsten Audit.



Wie Sie sich vielleicht erinnern, ist der Fokus dieses Projekts die Entwicklung eines Spiels, welches Prokrastination simuliert und die Problematik wie mögliche Lösungsansätze nahbar macht. Hier im Bild können sie auch die im letzten Audit präsentierte Vision sehen. Ein Raum, mit verschiedenen Auswahlmöglichkeiten, wobei einige (durch das „Licht“ unten links) mehr Aufmerksamkeit fordern als andere. Doch kommen wir nun zu den Neuerungen des Projekts.

Risiken – Übersicht

- Iteriert auf Basis der Risikoanalyse des Audit 1
- Risiken sollen Projektrisiken abdecken
 - „Was sind Risiken für unseren Projekterfolg?“
- Erlauben eine frühzeitige Einschätzung von:
 - Risiko Wahrscheinlichkeit: „Wie hoch ist die Chance, dass das Risiko eintritt?“
 - Risiko Einfluss: „Was wäre der Einfluss, den dieses Risiko auf den Verlauf oder Erfolg des Projekts hätte?“
- Erlauben eine frühzeitige Planung von Gegenmaßnahmen, zur Minderung oder gänzlichen Verhinderung negativer Konsequenzen

In unserer weiteren Arbeit am Projekt, beschäftigten wir uns weitergehend mit der Herausarbeitung von Risiken. Hierbei galt es die in bis Audit 1 erstellten Risiken zu erweitern und zu iterieren. Dabei erweiterten wir alle Risiken um eine Schätzung der Chance eines Risiko-Eintritts und um den Einfluss, welchen ein solches Risiko auf das Projekt nehmen würde. Des weiteren erweiterten oder iterierten wir die Risiken und ihre Gegenmaßnahmen.

Risiken – Transformation von Text

R508 – Transformation des Textes nicht möglich

Risiko:

- Da im Spiel Interaktions-Texte während des Spielgeschehens auf multiple Arten manipuliert werden sollen, also in ihrer Größe, Form und Farbe verändert werden sollen, kann es dazu kommen, dass die Texttransformationen nicht wie gewünscht oder gar nicht stattfinden, was dazu führt, dass der Spieler nicht zum Zielverhalten, also der bevorzugten Wahl von Prokrastinations-Entscheidungen, geführt werden kann.

Risiko Wahrscheinlichkeit:

- Geringe

Einfluss aufs Projekt:

- Hoch

Gegenmaßnahme:

- Standard-Eigenschaften für die Textelemente festlegen, welche anstelle von Transformationen verwendet werden können.

Schauen wir uns den Aufbau der Risiken einmal an Risiko R508, der Transformation von Texten.

Dieses Risiko beschäftigt sich mit dem Risiko, dass Interaktions-Text, welcher beispielsweise in TextBoxen vorkommt, nicht in seiner Größe, Form oder Farbe verändert werden kann. Dies würde ein Risiko für das Projekt darstellen, da die Darstellung des Texts als Medium dienen soll, welches dem Spieler den Reiz einer Aktivität vermittelt. Ohne diese Funktion wäre eine zielführende Realisierung des Rapid Prototypes gefährdet. Es galt also einzuschätzen, wie hoch die Chance des Risiko-Eintritts ist und wie hoch der Einfluss ist.

Aus unserer Sicht ist es die Eintritts-Chance nicht sehr hoch, da die Implementation von Text-Transformation kein zu hohes Maß an Komplexität hat. Dies ist aber aus unserer Sicht der Gegenteilige Fall, wenn es um den Einfluss eines solchen Risikofalles geht. Dies liegt daran, dass das Spielprinzip im Rapid Prototype ohne solche Funktionen nur schwer vermittelt werden kann, beziehungsweise würde es die Qualität der Vermittlung unseres Erachtens nach stark verringern. Trotz des aus unserer Sicht hohen Einflusses haben wir diesem Risiko dennoch nur eine Gegenmaßnahme zugewiesen, da die Risikowahrscheinlichkeit nur gering ist.

PoCs – Übersicht

- Iteriert auf PoCs aus Audit 1
- Einige PoCs leiten sich aus Risiken ab
- Andere PoCs haben sich durch Mockups und Projektplanung abgeleitet
- Sie erlauben besser zielführendes Arbeiten in der Implementation

Wir haben die PoCs aus Audit 1 weiter Iteriert und uns mehr mit ihnen beschäftigt. Ein Großteil der PoCs haben wir uns aus der Risikoanalyse abgeleitet viele haben wir aber wir aber noch zusätzlich dazu geschrieben welche sich aus der konkreteren Projektplanung und der Erstellung von den Mockups noch als relevant für das Projekt erwiesen haben. Durch die Nutzung von PoCs wird die Entwicklung und Implementation der einzelnen Funktionen Zielgerader.

PoCs, welche aus R508 resultieren

PoC 404: Text Transformation – R508

Es gilt zu beweisen, dass Text in TextBoxen bei Interaktion mit Aufgaben auf besondere Art und Weise, also durch Veränderung der Größe und/oder Farbe transformiert werden können.

Exit Kriterium: Ein Beispiel Text, Text in einer Textauswahl, wie „Lorem Ipsum“ verändert bei Auswahl die Font Size um 2 Punkte, die Farbe von #00CC00 zu #FF0000.

Fail Kriterium: Font Size und Textfarbe der Auswahl verändern sich nicht, auch wenn die Auswahl selektiert ist.

- Fallback 1: Statt dynamischer Farbe und Font Size, welche nur bei Auswahl stattfinden, werden statische Werte bei Aufgabentexten angewendet.
- Fallback 2: Es werden nur Transformation von Font Size ODER Farbe angewendet.

PoC 405: Text Verformung – R508

Es gilt zu beweisen, dass die Text-Geometrie des Textes in TextBoxen bei Interaktion mit Aufgaben verformt/animiert werden kann.

Exit Kriterium: Ein Beispiel, Text in einer Textauswahl, wie „Lorem Ipsum“ verändert bei Auswahl sichtbar die Form und wird verzerrt.

Fail Kriterium: Form der Auswahl verändern sich nicht, auch wenn die Auswahl selektiert ist.

- Fallback 1: Statt dynamischer Verformung, welche nur bei Auswahl stattfinden, werden durchgehende Verformungen bei Aufgabentexten angewendet.
- Fallback 2: Es werden keine Verformungen verwendet.

Aus dem Risiko R508 – Transformation des Textes nicht möglich haben wir das PoC 404 und 405 abgeleitet. Diese PoCs befassen sich mit den Anforderungen, welche für eine Funktion welche sich mit der Transformation von Text befassen würde. In diesem Fall wäre es dass, Text in Textboxen in der Lage sein soll sowohl in ihrer Font Größe und Farbe transformierbar zu sein, wie auch in der Geometrie verformt und animiert zu werden. Für den Fall dass, die Implementation dieser Funktionen fehlschlägt gibt es Fallbacks, welche alternative Ansätze vorschlagen.

PoC: das Aufzählen von dem Zeitparameter

PoC 307: Aufzählen des Zeitparameters

Es gilt zu beweisen, dass der Zeitparameter in regelmäßigen Zeitabständen um 1 inkrementiert, und so das Voranschreiten der Zeit im Spiel simuliert.

Exit Kriterium: Der Parameter wird mit jedem Inkrementieren in einem regelmäßigen Zeitabstand in der Konsole ausgegeben und die Werte haben eine Differenz von 1.

Fail Kriterium: Die Inkrementierung, welche in der Konsole ausgegeben wird, findet nicht regelmäßig statt und die Werte haben eine Differenz größer oder kleiner als 1.

- Fallback: Aktuelle Systemzeit in einer Variable speichern, um Zielzeit (z.B. 1000 Sekunden) addieren, und regelmäßig prüfen, ob die Systemzeit größer als die Zeit in der Variable ist.
- Fallback: Statt Zeit zu zählen, wird jede Interaktion in einem Array oder einer Queue gespeichert, wobei jede Interaktion zugewiesen „Zeit“-Werte hat, welche die Dauer der Interaktion bestimmen.

Ein weiteres PoC welches sich nicht aus der Risikoanalyse ableitet ist das PoC 307 Aufzählen des Zeitparameters. Da ein Großteil der internen Logik mit der Zeit arbeitet ist es besonders Wichtig dass es einen Zeitparameter welcher zuverlässig eine Zeit angibt. Dafür soll der Zeitparameter jeweils immer in regelmäßigen Abständen um 1 inkrementieren können. Falls wir dies nicht möglich ist wäre ein Fallback sich an der Systemzeit zu orientieren, ein weiterer wäre es keine aktive Zeit zu implementieren sondern einen Zeitwert zu haben, welcher jeweils nur bei der jeder Interaktion abgezogen wird.

Übersicht der Änderungen im Projektplan

- Die Risikoanalyse hat 2 Wochen mehr gedauert als erwartet.
- Der Projektplan wurde erweitert um:
 - User-Interface (UI) | In Audit 3 (Rapid P.)
 - Tooltips, als Erweiterung zum UI | Nach Audit 3 (Funkt. P.)
 - Kollision, separiert von Raum und Character | Nach Audit 3 (Funkt. P.)
 - Visuelle Effekte – Text | In Audit 3 (Rapid P.)
 - Visuelle Effekte – (Funkt. Prototyp) | Nach Audit 3 (Funkt. P.)

Hier sehen Sie einmal zusammengefasst die Änderungen bzw. Iteration am Projektplan.

In Audit 2 hat die Risikoanalyse 2 Wochen mehr gedauert als ursprünglich angenommen, wobei dies Primär durch kleinere Ergänzungen und Iteration zu Stande kam. Dies stellte aber keine weiteren Probleme dar, da während Audit 2 hauptsächlich an den Risiken und PoCs gearbeitet wurde.

Zusätzlich zu den bisher bestehenden Meilensteinen im Projektplan wurden noch 5 weitere Punkte ergänzt. Das User-Interface, welches zuvor nicht eigens im Projektplan vorkam, Tooltips, welche als Ergänzung und Lernhilfe dienen sollen, Kollision, welche zuvor in anderen Meilensteinen enthalten war, Visuelle Effekte für Text-Elemente, welche im Rapid Prototype erhöhte Relevanz haben, und Visuelle Effekte wie Shader, welche im Funktionalen Prototypen mehr Relevanz haben werden.

15:03 Days till deadline 4 Days

Work completed 27%

Stress 94 %

Health 3 %

work

game

phone

sleep

Ausblick

Hier können Sie noch mal einen Blick auf ein Mockup werfen, welches das Ziel für Audit 3 aufzeigen soll. Für den rapid Prototype soll hauptsächlich nur mit Textelementen gearbeitet werden, welche ausschließlich dafür da sind um die interne Logik und das generelle Spielprinzip zu demonstrieren. Hier zu sehen ist was passieren soll wenn der Produktivitätsparameter niedrig ist. Elemente, wie das Arbeiten, hier „work“, sind unauffälliger und Elemente wie „game“ und „phone“ sind durch Ihren hohen Kontrast auffälliger und sollen in sehr abstrakter Form eine Motivation gegenüber dieser Optionen darstellen. Dafür muss der wert des Produktivitätsparameters auf bestimmte weise mit der Helligkeit der Optionen work und sleep multipliziert werden.



Fragen

Nun kommen wir zu Ihren Fragen.

Nun kommt der Frageteil.