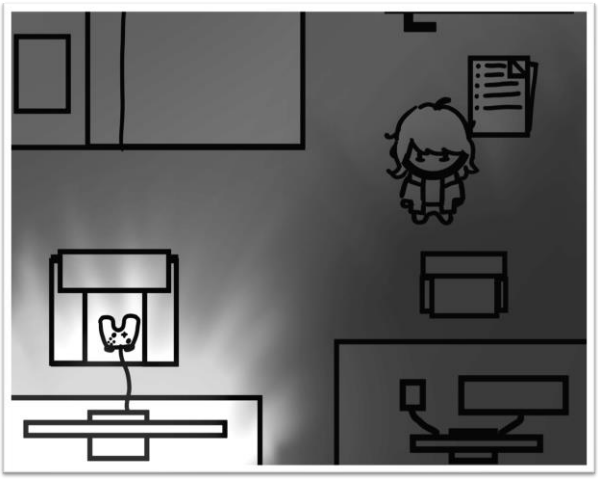
EP Audit 3

Ben Pritzkau und Robin Schlegel

Wiederholung

- Ziel: Erstellung eines Spiels, welches Prokrastination simuliert und damit verbunden Probleme und Lösungsansätze nahe bringt.
- Es sollen also durch das Spiel Rückschlüsse auf Sachverhalte in der Realität möglich sein, welche so zu einem positiven Ergebnis zu führen.
- Was das Projekt nicht ist:
  Ein Prokrastinations-Helfer, welcher den Nutzer über den Alltag coacht.



Hier eine kurze Wiederholung zum Wiedereinstieg in das Projekt.
Der Fokus des Projekts ist die Entwicklung eines Spiels, welches Prokrastination simuliert und die Problematik und damit verbundene Lösungsansätze dem Spieler näher bringt. Es soll also dem Spieler durch das Spielen des Spiels möglich sein, Rückschlüsse auf den Sachverhalt in der Realität zu ziehen, welche es dem Spieler ermöglichen Anpassungen im eigenen Leben durchzuführen.
Da es hier in den Open Spaces Unklarheiten gab, gilt es auch kurz zu erwähnen, dass das Ziel des Projekts es nicht ist, einen Prokrastinations-Helfer zu erstellen, welcher den Nutzer über den Alltag begleitet und den Nutzer coacht.

Vorstellung: Rapid Prototype

- Vorstellung vor Ort
- Oder: Video im Audit 3 Ordner

Hier bitte das Video aus dem Audit Ordner abspielen.

Übersicht durchgeführter PoCs

- Der Rapid Prototype enthält folgende Durchgeführte PoCs
- PoCs 301-310 (Parameter PoCs)
  - 301 – Parameter Manipulation (Parameter wird durch Buttons verändert)
  - 302 – Parameter Zugriff (Zugriff auf Parameter von Funktionen möglich)
    - 306 für Zeitzugriff im UI
  - 303 – Konstante Addition (Parameter werden regelmäßig um ihr Wachstum addiert)
  - 304 – Parameter Grenzen (Parameter können nur zwischen bestimmten Werten liegen)
  - 305 – Berechnung des Produktivitätsparameters
  - 307 – Inkrementierung des Zeitparameters
  - 308 – Deadline-Meilenstein (Die Annäherung an die Deadline wird angezeigt)
  - 309 – Deadline (Bei erreichen der Deadline wird die Endsequenz angezeigt)
  - 310 – Endsequenz (Je nach Arbeitsfortschritt wird die korrekte Endsequenz angezeigt)
- PoCs 401, 402 und 404 (UI PoCs)
  - 401 – Parameter UI (Korrekte Werte, Werte gleichen im UI der Konsole)
  - 402 – UI-Lesbarkeit (Das UI bietet vermittelt Informationen deutlich)
  - 404 – Text Transformation (Text verändert sich z.B. Farbe oder Position)

Hier zu sehen ist eine Übersicht der durchgeführten PoCs. Wie hier zu sehen wurden bisher nur die Parameter spezifischen PoCs und einige UI PoCs implementiert. Die anderen PoCs sind oft auf den Inhalt des 4.ten Audits angedacht.

Diese Ansicht soll aber lediglich kurz zeigen welche PoCs durchgeführt wurden, in der Präsentation vor Ort wird dies hier lediglich Überflogen.

Code Inspektion

```gdscript
35 func calcGes() -> void:
36     var g_mod: float = 0.0 ## Gesundheits_Modifikitor, also der Wert, um den die Gesundheit verän
37     var ticksSinceSleepPortion: float = 0.0 ## Der Anteil den ticksSinceSleep auf g_mod hat
38     var stessPortion: float = 0.0 ## Der Anteil den stress auf g_mod hat
39 
40     ticksSinceSleepPortion = g_modTicksSinceSleep.sample(ticksSinceSleep/(48.0*60.0/MINUTETICK))
41     # Berechnet den Anteil von ticksSinceSleep auf g_mod. Begrenzt auf maximal 48 stunden
42 
43     stessPortion = g_modStress.sample(stress/100.0) # Berechnet den Anteil von stress auf g_mod
44 
45     g_mod = -(ticksSinceSleepPortion + stessPortion - 1) # Rechnung
46     g_mod = clampf(g_mod, -1, 1) # Clampen
47     gesundheit = clamp(gesundheit + g_mod, 0, 100)
```

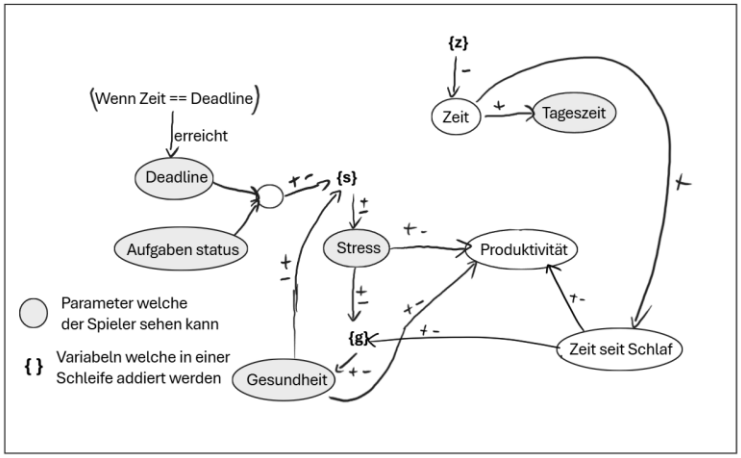
```gdscript
63 ## Produktivitätsparameter berechnen
64 func calcProductivity() -> void: # Überarbeiten??? Berechnet die Produktivität, Formel könnte überarbeit
65     var timeSinceSleepPortion: float = productivityTSS.sample(ticksSinceSleep/(48.0*60.0/MINUTETICK)) * 2
66     productivity = clampf(gesundheit * 0.5 + (-stress + 100) * 0.5 * timeSinceSleepPortion, 0, 100)
```

Hier sind zwei Funktionen zu sehen, einmal die Funktion calcGes() welche berechnet, um welchen Wert die variable Gesundheit, pro Sekunde ansteigt, und die Funktion calcProductivity, welche die productivity variable berechnet, welche später benutzt wird, um zu berechnen, wie effektiv der Spieler seine Aufgaben erfüllen kann. Wenn man ins Interaktionsmodell schaut, sieht man dass, {g} (hier g_mod) sich zum einen aus der Zeit seit Schlaf und dem Stresswert zusammen setzt. In der Funktion kann man das hier an den variables ticksSinceSleepPortion und stressPortion sehen, welche in Zeile 45 zusammen gerechnet g_mod ergeben. Um den Einfluss dieser variables zu berechnen benutzen wir hier Graphen welche sich einfach verändern lassen. Hier kann man zum Beispiel gut sehen dass, für ticksSinceSleep die variable erst ab einen bestimmten Wert einen negativen Einfluss hat (hier, ungefähr 16 stunden ohne Schlaf) und für stress der Einfluss konstant steigt. Über den Einfluss von Stress auf Gesundheit haben wir uns grob auf Studien bezogen, da wir aber keine Studien finden konnten die uns konkrete Werte liefern konnten wir uns nur daran orientieren dass, höherer Stress zu schlechterer Gesundheit führt. Die konkreten Werte haben wir dann im Playtesting auf unser Ziel angepasst. In Zeile 47 wird die variable g_mod dann auf die variable gesundheit addiert.
Die funktion calcStress welche den Stresswert berechnet ist ähnlich aufgebaut

wie calcGes
Im Interaktionsmodell kann man sehen dass die productivity sich aus dem Stresswert, dem Gesundheitswert und der Zeit seit Schlaf zusammen setzt. Im Code geschieht das in Zeile 66 wo die hälfte Gesundheit und die hälfte Stress, dann mit einem weiteren graphen für timeSinceSleep multipliziert wird.

Interaktionsmodell

Interne Interaktion der Parameter



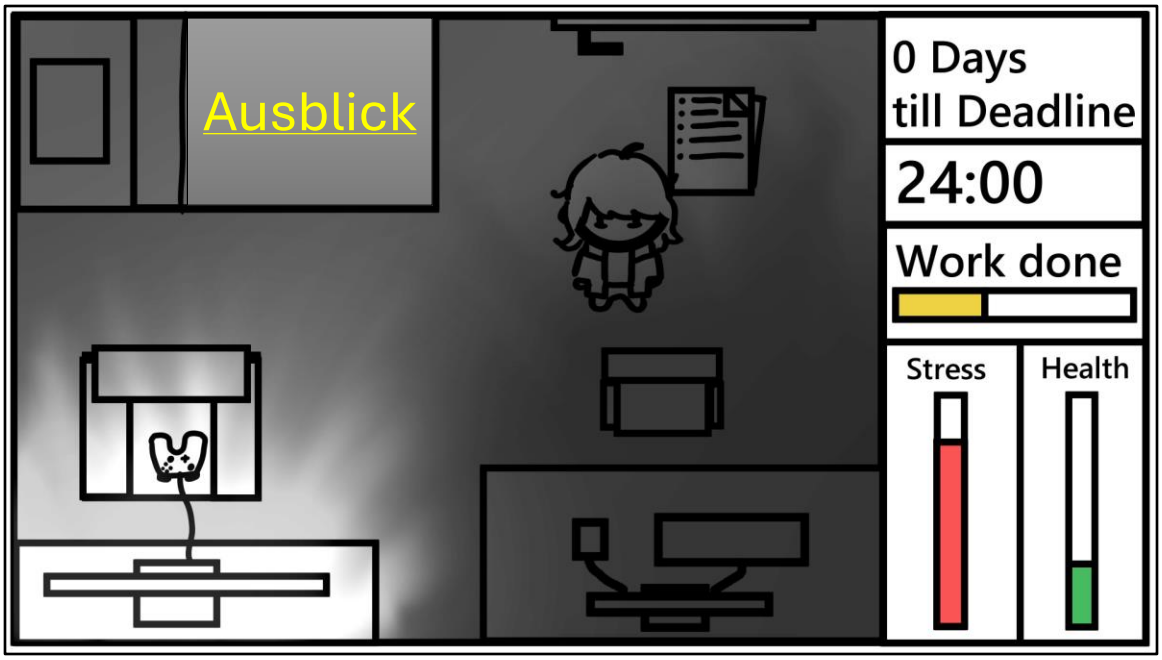
Siehe Code Inspektion.

Projektplan – Änderungs-Übersicht

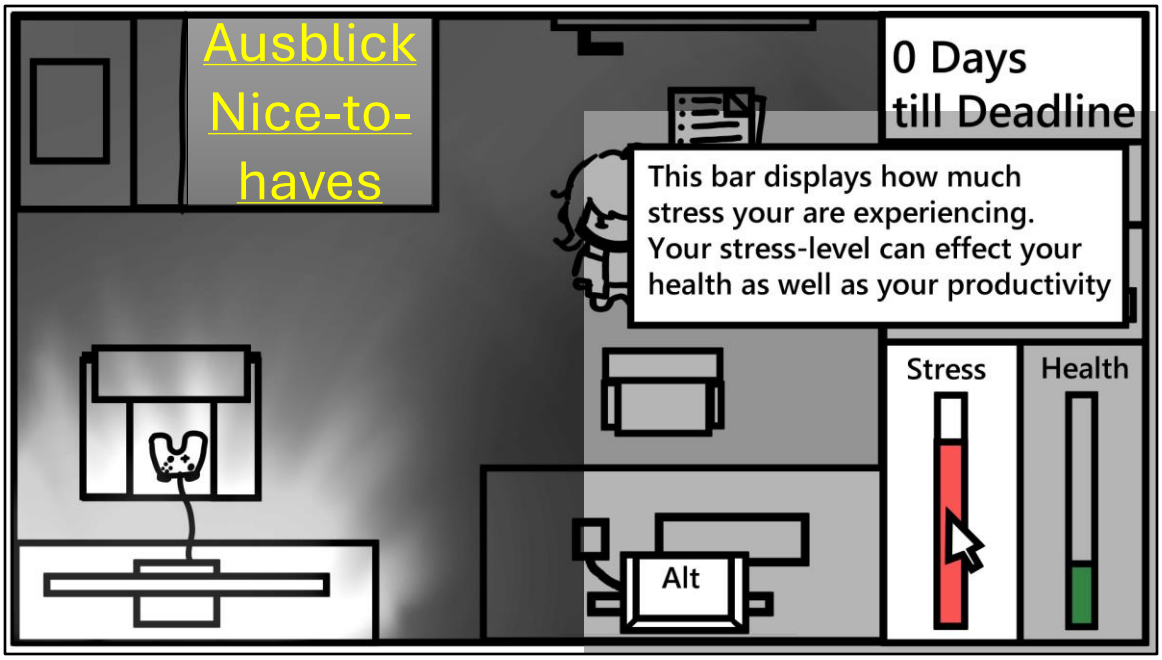
- Implementation der Parameter dauerte länger als erwartet
  - Umstieg von Mathematischen Formeln zu Curve-Sampling
  - Keine negativen Auswirkungen auf andere relevante Aktivitäten
  - Weitere Änderungen werden nun unter „Balancing“ dokumentiert
- Spielenden wurden verspätet aber schnell implementiert
- Visuelle Effekte weiter unterteilt
  - Lichteffekte
  - Charakter-Effekte
- Story-Line als Aktivität hinzugefügt

Hier eine kurze Übersicht der Änderungen im Projektplan. Die Implementierung der Parameter und der für diese relevante Funktionen dauerte länger als zunächst erwartet. Dies kam vor allem durch den Umstieg von Mathematischen Formeln hin zu Curve-Sampling, da letzteres die Parametermanipulation übersichtlicher macht und somit erleichtert. Des weiteren hatte dies keine negativen Auswirkungen auf andere relevante Aktivitäten, da zu genüge Zeit für potenzielle Verzögerungen eingeplant war und andere Aktivitäten wie die Implementation von UI-Elementen keine auch Test-Parameter verwenden konnten. Anpassungen an den Parametern und der damit verbundenen Kalkulation werden in Zukunft unter Balancing festgehalten. Die Spielenden wurden zwar später als erwartet implementiert, aber die Implementation ging hier sehr schnell.

Für das nächste Audit wurden die Visuellen Effekte weiter unterteilt und die potenzielle Implementation einer Story-Line wurde hinzugefügt.



Im nächsten Audit soll es nun der visuelle Aspekt des Projekts im Vordergrund stehen. So sollen die zuvor in Buttons abgebildeten Aktivitäten nun durch Gegenstände im Raum abgebildet werden, mit welchen der Spieler über die Spielfigur interagieren kann. Die Parameter, welche zuvor als einfach Zahlen abgebildet wurden, sollen als Balken/Leisten dargestellt werden. Wobei die Parameterwerte, welche sich zuvor lediglich auf den Text der Buttons widergespiegelt haben, nun für visuelle Veränderungen im Raum sorgen sollen, wie z.B. die im Mockup sichtbare Hervorhebung der Spiele-Konsole in Form eines Lichtes. Auch der Sprite des Spieler-Charakters soll optisch durch die Parameter beeinflusst werden und so den Zustand der Parameter widerspiegeln.



Abseits von den zuvor erwähnten Funktionalitäten, gäbe es Funktionen wie Tool-Tipps (siehe Mockup) oder auch die Inklusion einer Story-Line, welche Vorteilhaft für das Onboarding der Spieler wären, welche aber im Falle von Zeitmangel vernachlässigt werden würden.

Fragen?

Bei Bedarf können wir gerne näher auf den Code eingehen.

Fragen?