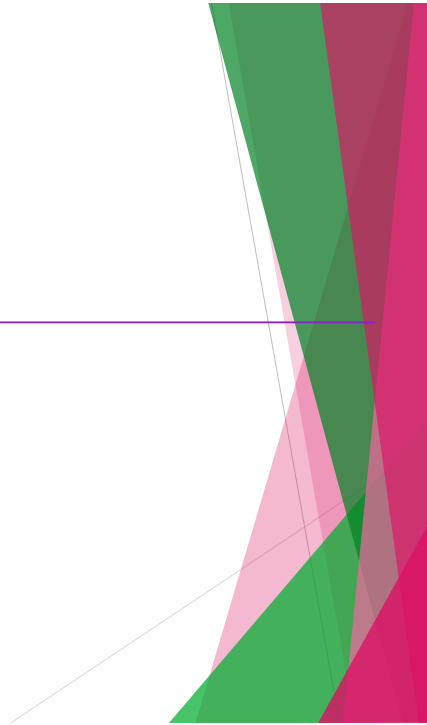


Entwicklungsprojekt WS 21/22

Audit 4



Funktionaler Prototyp – Overview implementierte Konzepte

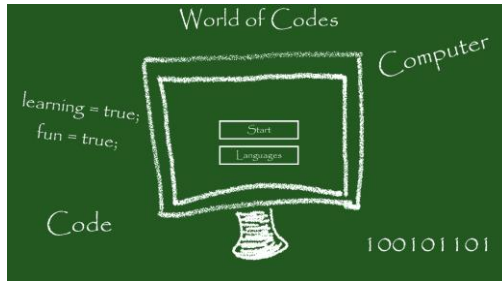
- Speichern der Spielstände -> für Prototyp in Prefabs -> für finales Produkt in eigener Datenbank, um PC zu entlasten
- Erstes Minispiel -> für finales Produkt mehrere Minispiele -> eine "Story" als Strang um Spannung aufzubauen und Nutzer dazu zu bringen die "Story" und somit alle Minispiele beenden zu wollen
- Design -> freundlich, froh, hell, neutraler Charakter mit dem man sich identifizieren kann
- Animationen -> unkompliziert, für mehr Interaktion, ermöglichen dennoch gutes Laufzeitverhalten
- Sounds / Hintergrundmusik -> freundlich, nicht zu aufdringlich, Sounds wieder für bessere Nutzerinteraktionen
- Leistung durch Highscore verfolgen
- Prototyp mittels WebGL gebildet und gepublished: <https://play.unity.com/mg/other/webgl-builds-152903>

Konzepte, die wir implementiert haben sind zum einen das Speichern von Spielständen und Daten wie dem Highscore. Durch den Highscore wollen wir den Ehrgeiz der Kinder anregen, dran zu bleiben und auch besser als ihre Schulkameraden zu werden.

Unser erstes Minispiel gilt als Beispiel eines von vielen Minispielen, welche in unserem Lernprogramm implementiert werden sollen. Anhand dieses Spiels sieht man, wie wir versuchen Wissen mit Spaß zu verbinden. Es soll allerdings nicht nur Spiele mit „Quizcharakter“ geben, sondern auch andere Möglichkeiten, in denen die Kinder beispielsweise etwas tippen oder sortieren sollen. Diese Minispiele sollen in einer großen Story zusammenhängen, jedoch auch gespielt werden können, wenn man die Story nicht beachtet. Die Story sorgt für eine Spannung und soll die Kinder dazu bringen, das Spiel beenden zu wollen, um zu erfahren, wie die Story ausgeht.

Sowohl unser visuelles, als auch unser auditives Design sind freundlich, hell aber auch neutral gestaltet. Die Neutralität war uns wichtig, damit sich kein Kind ausgegrenzt fühlt, weil es sich beispielsweise nicht mit dem Charakter identifizieren kann. Bei den Animationen haben wir uns auf einfache, aber lustige und leicht ausgefallene Charakteristika konzentriert. Sie sollen nicht zu hohem Laufzeitverhalten führen, aber das Spiel optisch anspruchsvoller und reaktiver gestalten.

Design



Main Menu

- Simples Design -> könnte auch von Kind gemalt sein -> Identifizierung
- Erinnerung an Tafel + Kreide und damit an Schule + Lernen
- Schlagwörter damit man weiß, worum es geht

Charakter

- Neutral -> grau, keine Haare, Jungen und Mädchen können sich identifizieren
- Kindlich -> klein, große Augen, kein Mund, süßes Aussehen

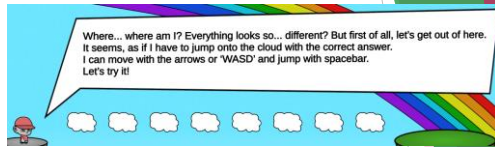


Das Design des Hauptmenüs ist an eine Tafel mit aufgekritzten Sprüchen und einem Monitor angelehnt. Damit soll dem Spieler bewusst werden, dass es im Grunde dennoch um das Lernen von Informatik und dem Umgang mit Computern geht. Der Charakter ist grau und es sind keine Haare zu sehen unter der Kappe. Damit könnte es sowohl ein Junge, als auch ein Mädchen sein. Alle sollen sich mit dem Charakter identifizieren können. Er ist relativ klein durch seine kurzen Beine und seinen verhältnismäßig größeren Kopf. Seine Gestaltung soll möglichst niedlich sein, was durch die großen Augen und den fehlenden Mund unterstützt wird.

Design - Story

Um Spannung & Motivation bis zum Ende aufrecht zu erhalten:

Geschichte eingebaut! →
Kinderfreundlich



This is the story of a lost boy in an unknown world. The only way to get back home is to solve all the riddles he will be given.

...

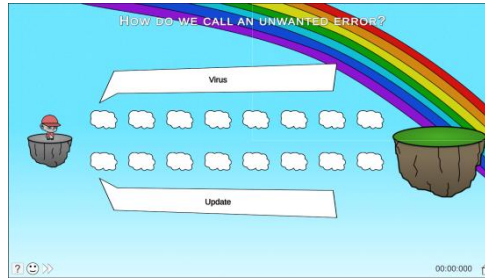
Will he be able to solve them all

Um Atmosphäre nicht zu zerstören:
Sprechblasen mit Instruktionen

Geschichte eingebaut in der die Spielfigur in einer fremden Welt landet und nur durch das Bestehen verschiedener Levels/Spiele wieder zur eigenen Welt zurückkehren kann.

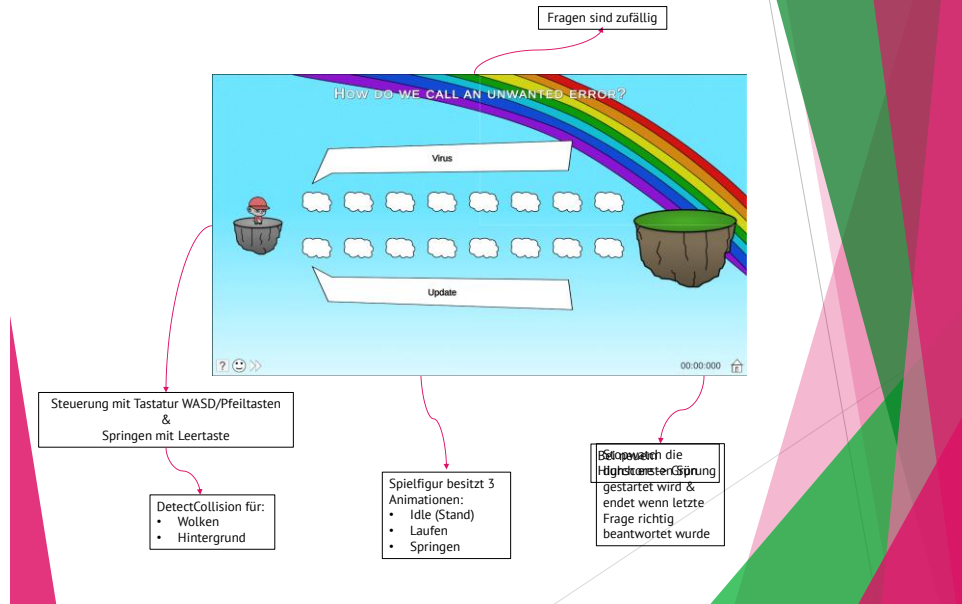
Level Design

- Hintergrund -> Himmel, hell, mit Regenbogen -> in Richtung des Ziels
- Startplattform farblos -> Zielplattform mit satter Grün -> schöner im Ziel
- Spieler springt über Wolken -> Leichtigkeit
- Frage alleine oben -> genug Platz, keine Ablenkung
- Alle Buttons + Timer unten -> alle Interaktionsmöglichkeiten gebündelt
- Antworten in Sprechblasen -> genug Platz auch für längere Antworten + leerer Raum wird gefüllt



Das Level Design unseres ersten Minigames ist so gestaltet, dass die Startplattform relativ trist und kahl ist, während die Zielplattform deutlich farbiger und froher ist. Außerdem endet dort der Regenbogen um diese Extreme zu verstärken. Der Charakter springt über Wolken und strahlt damit eine Leichtigkeit aus, welche das kindliche Design unterstützt. In der oberen Hälfte des Bildschirms steht die Frage für sich alleine. Dort gibt es keine große Ablenkung außer dem Hintergrund. Um die Frage von diesem abzuheben, hat der Text eine Umrandung und einen Schatten erhalten. Alle Buttons, sowie der Timer für den Highscore sind unten am Bildschirmrand platziert. Der Home-Button getrennt von den anderen Buttons auf der rechten Seite, damit man eher versucht noch das Spiel zu beenden, anstatt zum Startbildschirm zurückzukehren.

Level Design – Detaillierte Funktionen



Fragen sind randomized -> somit kann sich der Spieler die richtige Reihenfolge nicht merken, sondern muss die Fragen trotzdem richtig beantworten.

Die Steuerung des Charakters erfolgt über die etablierten WASD-Tasten bzw. die Pfeiltasten und man springt mit der Leertaste. Kollisionen des Charakters mit den Wolken werden erkannt und je nachdem, ob das Feld richtig oder falsch ist, dementsprechend der passende Code ausgeführt. Bei einem richtigen Feld, wird die nächste Frage mit den nächsten Antworten angezeigt. Bei einem falschen Feld wird die Szene zurückgesetzt und der Spieler muss von vorne anfangen. Er hat das Spiel geschafft, wenn er alle 8 Fragen hintereinander richtig beantwortet hat.

Der Charakter besitzt 3 Animationen. Eine während des Stehens, eine während dem Laufen und eine für das Springen.

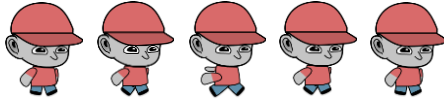
Der Highscore wird mit dem ersten Sprung gestartet und beendet, wenn die letzte Frage richtig beantwortet wurde. Sollte der Highscore geknackt wurden sein, wird der neue Highscore gespeichert und der Timer wird grün.

Animationen



Idle-Animation (Stehen)

- Arme hin und her schwingen
- Leicht auf und wieder ab (4 & 6)
- Keine Bewegung in den Beinen -> soll nicht aussehen, als würde der Charakter gehen oder laufen

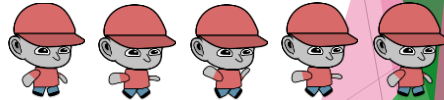


Run-Animation (Laufen)

- Arme nach hinten (spezieller, lustiger Laufstil)
- Frame 3 am höchsten -> in der Luft
- Bewegung in den Beinen

Jump-Animation (Springen)

- Arme nach oben -> Spaß beim Sprung
- Beine weit auseinander -> für weiten Sprung
- Landung wie Start des Sprungs



Um den Stand aufzulockern, wackelt der Charakter mit den Armen.

Während des Laufens bewegt der Charakter die Beine dementsprechend und nimmt die Arme nach hinten. Dadurch entsteht ein spezieller Laufstil (ähnlich wie der von Naruto).

Für den Sprung nimmt der Charakter die Hände nach oben, damit es so aussieht, als hätte er Spaß am Springen.

Sounds

- Hintergrundmusik -> bei Start des Levels, harmonisch, nicht zu laut, damit Kinder sich trotzdem konzentrieren können
- Sound bei klicken eines Buttons -> responsive UI
- Sound beim Springen -> hoch gepitchte Stimme, passend zum "süßen" Aussehen des Charakters
- Sound bei Landen auf richtiger Plattform -> akustische Belohnung und man weiß, dass man richtig ist
- Sounds von der Plattform „Pixabay“ -> kostenlos, auch für kommerzielle Zwecke verwendbar
- Jump-Sound selbst aufgenommen und editiert

Die Sounds haben wir von der Plattform „Pixabay“ heruntergeladen. Für die Hintergrundmusik haben wir eine kindgerechte Musik ausgewählt, die Heiterkeit vermittelt. In der Musik ist auch eine Rassel zu hören, welche erneut das Kindliche verstärkt. Um das Programm reaktiver zu gestalten, wurden Sounds für das Klicken eines Buttons, für das Landen auf einer richtigen Plattform und für das Springen hinzugefügt.

Den Sound für das Springen haben wir als einzigen selbst aufgenommen, die Geschwindigkeit des Clips etwas angehoben und die Stimme hochgepitcht. Die gepitchte Stimme unterstreicht wieder den niedlichen Charakter, der dadurch ebenfalls wieder neutral ist, da weder Jungs, noch Mädchen so eine hochgepitchte Stimme haben.

Reflektion & Fazit

Unsere **Ziele** waren:

- Selbstständig Design erstellen ✓
→ Altersentsprechend
- Angemessener Schwierigkeitsgrad ✓
- POC alle erfüllen können ✓
- Simple technische Anforderungen ✓
→ Simple Animationen und Steuerung

Fazit:

Unity war die richtige Wahl aufgrund von den Funktionalitäten und technischen Anforderungen. Probleme konnten schnell behoben werden & C# sehr verständlich!

Unsere Ziele haben wir größtenteils erfüllen können. Zwar haben wir es nicht geschafft, verschiedene Sprachen zu implementieren. Für einen Prototyp haben wir dies allerdings nicht als allzu tragisch erachtet. Wir konnten durch unseres selbstständiges Design die Altersgruppe von 10 bis 14-Jährigen treffen. Kinder in diesem Alter sollten ebenfalls in der Lage sein, dieses Spiel zu absolvieren, wenn sie beispielsweise durch Hinweise in der Story die Fragen beantworten können.

Unsere POCs konnten wir, wie bereits erwähnt außer dem mit der Sprache, alle erfüllen.

Auch haben wir stets die simplen technischen Anforderungen berücksichtigt, damit jeder PC (im besten Falle selbst ein Raspberry Pi) unsere Web-Anwendung ausführen kann.

Quellen

- Hintergrundmusik: <https://pixabay.com/music/happy-childrens-tunes-background-delicate-presentation-peaceful-happy-friendly-music-20859/>
- Buttonclick: <https://pixabay.com/sound-effects/mouse-click-6687/>
- Richtiges Feld: <https://pixabay.com/sound-effects/short-success-sound-glockenspiel-treasure-video-game-6346/>