

EVERYWHERE
WHAT MAKES

U

A MONSTER



REDLIGHT
ORACLE



STUDIOS

Individualportfolio als Programmierer von Cristian Carrasco



REDLIGHT
ORACLE



Du findest dich als Student in einer
mysteriösen Parallelwelt wieder.
Besuche eine Universität für
Monster, integriere dich und
entkomme ihrem Schöpfer!

Schaffst du es, Mensch zu bleiben?





Unser Team



**Simon
Gremm**

Game-Design
Sound-Design

**Eric
Glasser**

Team-Lead
Game-Design
Sound-Design

**Cristian
Carrasco**

Programmierer

**Georg
Winterfeld**

Programmierer

**Robin
Holz**

Story-Design
Dokumentation
Composer

**Jannik
Reimers**

Story-Lead

**Elisa
Langer**

Art-Lead
Organisation

**Hannah
Rauch**

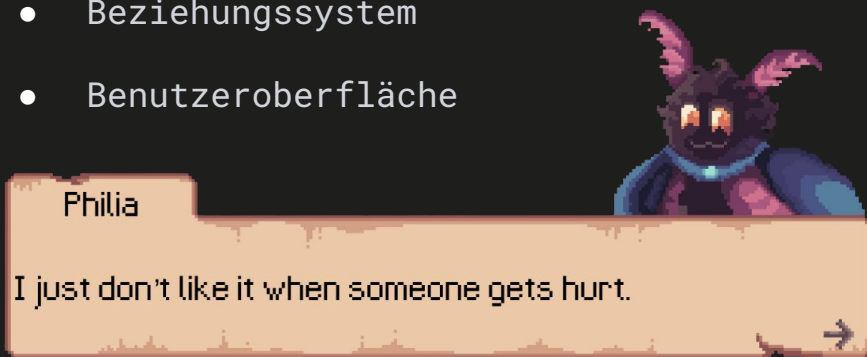
Game-Artist

* Eigene Arbeiten

Meine Aufgaben

Meine Hauptverantwortung lag in der Entwicklung und Implementierung von:

- Dialogsystem
- Beziehungssystem
- Benutzeroberfläche





Dialogsystem



Dialogsystem mit dem Unity-Plugin Ink

- Das Dialogsystem ist eine zentrale Komponente des Spielerlebnisses in unserem Videospiel.
- Es war hauptsächlich mein Vorschlag dem Spieler die Möglichkeit zu geben verschiedene Dialogoptionen zu haben um auch den Beziehungssystem miteinzubauen für den Ich mich ebenfalls stark eingesetzt habe.
- Mithilfe des Unity-Plugins Ink habe ich ein interaktives und dynamisches Dialogsystem entwickelt.
 - Ink ist eine von Inkle Studios entwickelte Programmiersprache.
 - Sie dient zur Erstellung interaktiver Geschichten und Dialoge in Spielen.
 - Mit Ink können Entwickler komplexe Dialogbäume und Entscheidungsstrukturen erstellen.
 - Es ermöglicht Verzweigungen und Variationen in der Handlung basierend auf den Spielerentscheidungen.
- Der Spieler kann verschiedene Dialogoptionen auswählen und somit Einfluss auf den Spielverlauf nehmen.
- Bei jedem Dialog gibt es außer dem Dialogtext der in einem Typewriter-Effekt ausgegeben wird das Portrait und der Name des Sprechers.



Philia

I just don't like it when someone gets hurt.





Dialogsystem



- Dies ist eine Beispiel Ink Datei, die erste Zeile, `INCLUDE globals.ink`, bedeutet, dass die Datei "globals.ink" eingebunden wird, in der globale Variablen definiert sein können.
 - Der Abschnitt `-> main` ist der Einstiegspunkt für den Dialog.
 - Im Abschnitt "main" wird dem Spieler die Frage gestellt, welches Pokémon er wählen möchte. Der Charakter "Philia" spricht und es wird ein Porträt von ihr angezeigt.
 - Der Spieler kann zwischen den Optionen [Charmander], [Bulbasaur] und [Squirtle] wählen. Je nach Entscheidung wird mit `~` auf eine externe C# Methode zugegriffen.
 - Abhängig von der Auswahl wird der Abschnitt "chosen" mit dem ausgewählten Pokémon als Argument aufgerufen.
 - Im Abschnitt "chosen" wird die Variable `pokemon_name` auf den Namen des gewählten Pokémon gesetzt und eine Bestätigungsnachricht wird angezeigt.
 - Wenn der Spieler bereits ein Pokémon ausgewählt hat (also `pokemon_name` einen Wert hat), wird der Abschnitt "already_chose" aufgerufen und eine entsprechende Meldung wird angezeigt.
- Am Ende wird der Dialog beendet (`-> END`)

```
1  INCLUDE globals.ink
2  { pokemon_name == "" : -> main | -> already_chose }
3
4  -> main
5  === main ===
6  Which pokemon do you choose? #speaker:Philia #portrait:philia_Portrait
7  + [Charmander]
8  ~ subtract_hearts("Philia", 1)
9  -> chosen("Charmander")
10 + [Bulbasaur]
11 -> chosen("Bulbasaur")
12 + [Squirtle]
13 -> chosen("Squirtle")
14 //~ add_hearts("Philia", 1)
15
16
17 ~ === chosen(pokemon) ===
18 ~pokemon_name = pokemon
19 You chose {pokemon}! #speaker:Philia #portrait:philia_Portrait
20 -> END
21
22 ~ === already_chose ===
23 You already chose {pokemon_name}! #speaker:Philia #portrait:philia_Portrait
24 -> END
```

End of story



Dialogsystem

- Dementsprechend muss ein DialogManager in der Szene sein der beim anklicken der NPCs den Dialog startet und auf die Ink-Files zugreift.
- Hier wird gezeigt wie ein Dialog aussehen kann:





Beziehungssystem



Beziehungssystem zwischen Spieler und NPC

- In unserem Videospiel war es wichtig, dass der Spieler mit den NPCs interagieren und Beziehungen aufbauen kann.
- Ich habe ein Beziehungssystem implementiert, das es dem Spieler ermöglicht, Freundschaften oder Feindschaften zu entwickeln.
- Je nach den Dialogentscheidungen oder den Quests des Spielers können sich die Beziehungen zu den NPCs und der Spielverlauf verändern.
- Dies ermöglicht große Entscheidungsfreiheit und gesteigerter Wiederspielwert.
- Die Beziehung wird durch Herzen in der UI angezeigt.
- Der Spieler kann von -5 bis zu +5 Beziehungspunkte haben.





Beziehungssystem

Beziehungssystem zwischen Spieler und NPC

- Der Beziehungswert wird in NPCScriptableObjects abgespeichert, dies ermöglicht eine einfache Verwaltung und Änderung der Beziehungswerte ohne direkte Änderungen am NPC-Skript oder an anderen Datenstrukturen.



✖ Benutzeroberfläche

Funktionalität der Benutzeroberfläche

- Eine benutzerfreundliche und funktionale Benutzeroberfläche ist entscheidend für ein reibungsloses Spielerlebnis.
- Ich habe die Benutzeroberfläche entworfen und implementiert, um sicherzustellen, dass sie alle erforderlichen Funktionen bietet.
- Die Benutzeroberfläche ermöglicht dem Spieler eine klare Darstellung von Informationen und eine intuitive Interaktion.



✖ Benutzeroberfläche

Funktionalität der Benutzeroberfläche

- Ich hatte die Idee eines Tab-System für unser Spiel, die Ich dank den Components von Unity und einen eigenen Script ganz gut umsetzen könnte.
- Es gibt eine Seite fürs Inventar, eine für die Beziehungseinsicht, eins um die Fähigkeiten des Monster-Companions einzusehen und eine Kartenseite, die durch einen animierten roten Punkt dem Spieler anzeigt, wo er sich gerade befindet.
- Natürlich sind alle UI-Elemente immer richtig skaliert und pixel-perfect.





Sonstiges

Sonstige Beiträge zum Spiel

- Zu Beginn hieß das Spiel "What makes YOU a Monster", da kam Ich auf die Idee es auf "What Makes U a Monster" umzubenennen um das U im Logo hervorzuheben und um eine Anspielung auf das Wort Universität zu machen um unser Thema wieder aufzugreifen.
- Dies wird besonders durch den College Font auch rübergebracht.
- Diese sind die allerersten Versionen unseres Spiellogos, die von mir entworfen und mit Illustrator und anschließend Aseprite designed wurde und später von den Artists überarbeitet wurde.
- Außerdem gibt es mit der gerundeten Form des Textes um das "U" herum eine Anspielung auf ein Auge der unser Studio-Logo ähnelt, und der Font war extra kindisch gezeichnet um den Gott der verlorenen Kindheit, also unser Erzfeind im Spiel, wieder aufzugreifen.





S T U D I O S

Kontakt:



carrascocristian@gmail.com



@cristiancarrascocreative

<https://www.youtube.com/channel/UCWSqgCg5YM5mjZU4XoLLyJA>



@kuriasuko



<https://criix23.itch.io/>

Schauen sie gerne in mein eigenen Youtube Kanal um meine anderen Projekte zu sehen!