



Projektdokumentation

Michael Ziesemann (2189162)

Anh-Thu Nguyen (2171169)

Betreuer: Patrick Hilgenstock

Git Repository: <https://github.com/miczzz/PainterlyCat>

28. Juli 2017

Image Processing

Media Systems (B.Sc.)

Hochschule für angewandte Wissenschaften Hamburg/

Department Medientechnik

Fakultät Design, Medien und Information

Inhalt

1. Konzept	3
2. Endprodukt	4
3. Was lief gut?	6
4. Was lief nicht so gut?	7
5. Unterschied Konzept und Endprodukt	8
6. Credits	9

1. Konzept

Painterly Cat ist ein 3D Rätsel Spiel, in dem ein Spieler verschiedene Denk- und Geschicklichkeitsaufgaben lösen muss, um von einem Raum in den nächsten zu gelangen. Jeder Raum besitzt ein farbliches Thema, wie beispielsweise schwarz-weiß, regenbogenfarben oder pastellfarben. Die Räume unterscheiden sich außerdem in Größe, Farbe und Beleuchtung.

In jedem Raum ist eine Tür, welche in den nächsten Raum führt und verschlossen ist. Sobald der Spieler meint, eine Aufgabe richtig gelöst zu haben, kann er sich vor die Tür stellen und ein Codewort sagen. Ist die Aufgabe richtig gelöst, öffnet sich die Tür.

Der Spieler erhält keine Anleitung zu den Aufgaben. Er muss seine Umgebung erkunden um seine Aufgabe selbst herausfinden. Die Schwierigkeit erhöht sich im Laufe des Spiels.

Der Spieler steuert in der Third-Person-Perspektive eine Katze, die sich im Raum bewegen und umsehen kann. Die Figur der Katze besteht dabei nur aus einem runden Kopf mit Katzeneigenschaften. Anderen Tiere im selben Stil begegnet man im Laufe des Spiels. Um mit dem Objekten im Raum zu agieren, kann der Spieler Farbbälle in verschiedenen Farben verschießen.

Ziel des Spiels ist es, alle Aufgaben zu lösen. Es geht dabei in erster Linie nicht um Schnelligkeit oder Rekorde, sondern darum, die verschiedenen Räume zu erkunden und in Ruhe über die Rätsel nachzudenken.

Beispielräume:

1. Farben mischen: An einer Wand steht "gelb + rot = ?" und im Raum sind verschiedene Farbböcke verteilt. Der Spieler muss den richtigen Farbton finden um die Aufgabe zu lösen.
2. Monster abschießen: Der Spieler muss Monster einer vorgegeben Farbe abschießen, um die Tür zu öffnen.
3. Farbverlauf: Der Spieler muss Farbböcke richtig anordnen, sodass sich ein Farbverlauf ergibt.

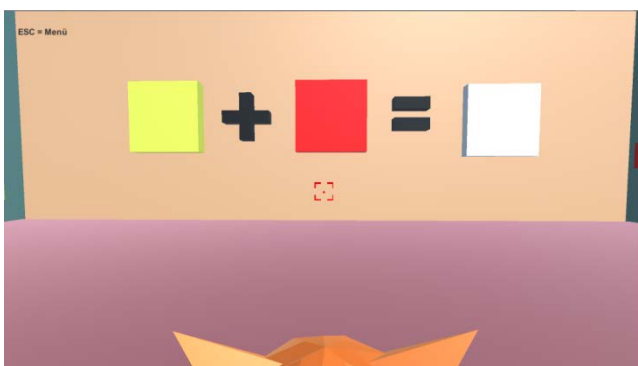
2. Endprodukt

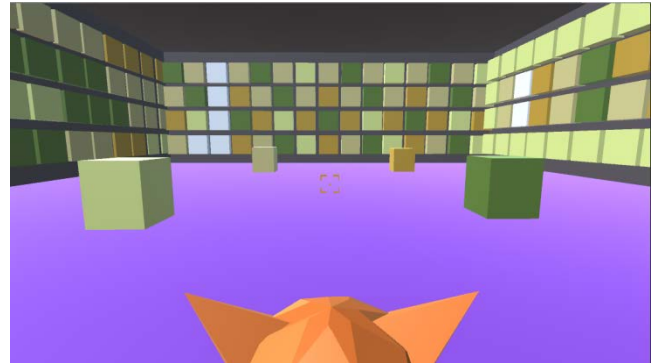
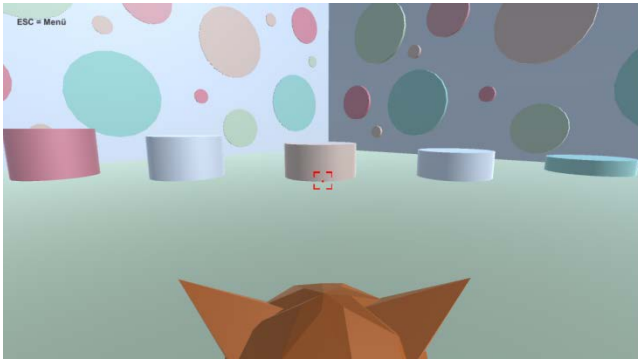
2.1 Menü



Im Hauptmenü hat der Spieler die Möglichkeit das erste Level zu starten, das Spiel im Multiplayer zu spielen, die Anleitung zu lesen oder das Spiel zu beenden.

2.2 Levels





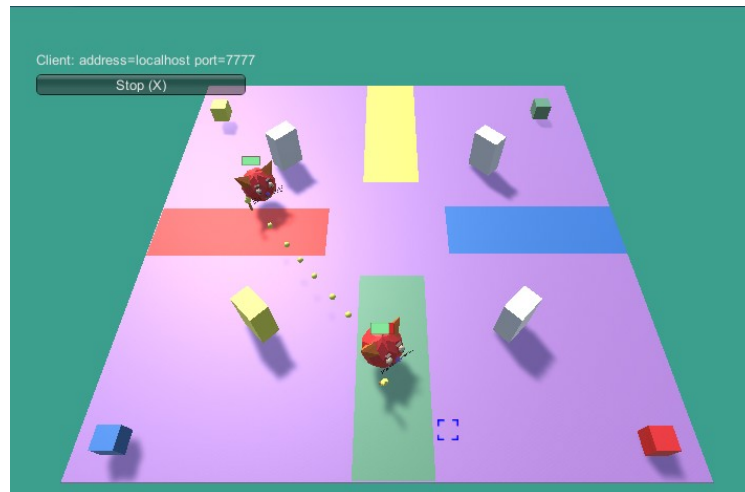
Painterly Cat hat 8 verschiedene Levels, die hintereinander gespielt werden können. Jedes Level hat ein anderes Farbschema und eine andere Aufgabe. Bei einigen Levels müssen Farben gemischt werden, bei anderen Farbverläufe oder Farbmuster vervollständigt werden. Das Rätsel in den Räumen muss der Spieler selbst finden, jedoch gibt es bei einigen Levels Tipps, wenn man mit einem Schweinchen interagiert.

Sobald der Spieler das Rätsel richtig gelöst hat, ertönt ein Geräusch und die Tür im Raum öffnet sich. Der Spieler wird zum nächsten Level teleportiert, wenn er durch die Tür geht. Beim letzten Level gibt es keinen Teleport und das Level kann nicht übersprungen werden. Durch das Menü kann der Spieler dort dann das Spiel beenden.

Das Fadenkreuz in der Mitte vom Bildschirm zeigt die Farbe an, die der Spieler im Moment verschießen kann. Wenn der Spieler einen Farbblock abschießt und dessen Farbe annimmt, verändert sich das Fadenkreuz dementsprechend.

Mit der ESC-Taste kann der Spieler ein Menü öffnen und auswählen, ob er das Level weiterspielen, neustarten, überspringen oder beenden will. Wenn der Spieler das Spiel beendet, wird der Spielstand nicht gespeichert und er gelangt zurück zum Hauptmenü. Durch die Möglichkeit Levels zu überspringen, kann der Spieler das Spiel leicht dort fortsetzen wo er möchte.

2.3 Multiplayer-Modus



Im Multiplayer über Netzwerk spielt man zu zweit gegeneinander. Die Steuerung und Ansicht sind ein wenig anders als im Singleplayer-Modus, aber der Umstieg sollte nicht allzu schwer fallen. Man schießt weiterhin mit Farben, diesmal jedoch mit Farbkugeln (und kein Laser wie im Singleplayer). Mit der Maus zielt man, wohin die Kugeln fliegen. Spielziel ist es dabei den Gegner mit der richtigen Farbkugel abzuschießen – mit der Farbe, die der Körper des Gegners zurzeit hat.

Am Anfang des Spiels werden die Farben der beiden Spieler zufällig verteilt, wobei die Farbkugeln, die man verschießt zunächst die gleiche Farbe haben wie der eigene Körper (also Vorsicht, man kann sich auch selbst abschießen!). Die richtigen Farbkugeln kann man sich an den vier Ecken des Spielfelds abholen, indem man in die farbigen Würfel hineinläuft. Ist der Gegner einmal mit der richtigen Farbe getroffen, wird der Körper neu eingefärbt mit einer Zufallsfarbe (aber definitiv eine andere als zuvor). Wer den Gegner zuerst fünf Mal mit der richtigen Farbe getroffen hat, hat gewonnen.

3. Was lief gut?

- Das Arbeiten mit Git lief gut. Es gab wenige Merge-Konflikte und durch die Kommentare konnte man immer sehen, was am Projekt geändert wurde.

- Unsere Zusammenarbeit funktionierte gut. Wir haben uns gut absprechen können, zueinander passende Vorstellungen gehabt und konnten uns so gut bei der Arbeit ergänzen.
- Generell das Arbeiten mit Unity funktionierte auch gut, da wir schon etwas Vorerfahrung hatten, die Inhalte aus der Vorlesung nutzen konnten und im Internet für die meisten Probleme schnell Hilfe gefunden werden konnte.

4. Was lief nicht so gut?

- Unity-Versionswechsel:
Während der Entwicklung haben wir Unity auf eine neue Version geupdatet. Dadurch kam es kurzzeitig zu einigen Problemen beim Starten.
- Kamera/Steuerung:
Eine schöne Steuerung und Ansicht in der Third-Person-Perspektive für unser Spiel war nur schwierig hinzubekommen, da man sich auch gleichzeitig immer noch überall hin umschauen können muss. Letztlich haben wir daher eine Wahl getroffen, die Single-Person sehr nahe kommt und nur im Multiplayer-Level (mit fester Kamera) eine richtige Third-Person-Perspektive verwendet, wie sie ursprünglich fürs ganze Spiel angedacht war.
- Blender-zu-Unity:
Anfangs hatten wir den Origin in Blender von der Spielerfigur nicht richtig gesetzt, womit es zu Problemen in Unity kam, wobei zunächst unklar war, wieso sich die Spielfigur ausgerechnet so verhält.
Wenn man von Blender Objekte importiert und dabei fahrlässig die Materialnamen gibt (keine Unikate), kann es auch zu unerwünschten Nebenwirkungen in Unity kommen (die bisherigen Materialien werden überschrieben).
- Netzwerk
Netzwerkprogrammierung war ein wenig ein Kampf. Was hauptsächlich bei Painterly Cat übergeben wird, sind Farben. Colors oder Materials sind keine primitiven Variablen, daher kann man sie nicht mal eben so übers Netzwerk schicken, also muss ein Umweg gegangen werden, z.B. über die

RGB-Werte oder die Farben anders abspeichern.

Es müssen immer vier Seiten sich richtig zueinander verhalten (in unserem Fall vor allem die gleichen Farben darstellen): Client zu Client, Client zu Server, Server zu Server und Server zu Client. Alles muss gleich sein, sonst funktioniert das Spiel nicht richtig. Das war schwieriger als erwartet und oftmals als wir dachten „Jetzt funktioniert endlich alles“, haben wir dann erst etwas später gemerkt, dass der Client eben doch nicht die richtige Farbe der Farbkugel vom Server bekommen hatte oder Ähnliches.

Das Testen von den Netzwerkabläufen ist generell eher zeitaufwendig, da man immer wieder neu builden muss usw.

Ein Problem war auch, dass etwas zwar im lokalen Netz richtig funktionierte (Farben werden am Anfang zugewiesen), das aber mit dem bisherigen Code dann im Netzwerk nicht immer richtig umgesetzt wurde, da der Server anscheinend nicht immer auf den Client gewartet hatte und der Client die neuen Farbinformationen dann nicht jedes Mal (rechtzeitig) erhielt. Das haben wir dann code-technisch etwas umständlicher umgesetzt, so dass es dieses Problem nicht mehr geben sollte.

Erschwerend kam noch hinzu, dass selbst in Tutorials zu Networking oft einiges nicht klappte, oder z.B. das Asset von einer kostenfreien (hübscheren) Netzwerk-Lobby wie in einem Video angepriesen mittlerweile so nicht mehr importiert werden konnte, da der Code auf der aktuellen Unity-Version nicht mehr ohne weiteres funktionierte.

5. Unterschied Konzept und Endprodukt

Der größte Unterschied besteht darin, dass wir die geplante Sprachsteuerung nicht eingebaut haben und stattdessen einen Multiplayer-Spielmodus erstellt haben. Im Laufe der Entwicklung haben wir gemerkt, dass die Sprachsteuerung dem Spiel keinen Mehrwert bringt und den Spielfluss eher stört.

Die Levels mit den verschlossenen Türen und den verschiedenen Farben und Rätseln wurden so umgesetzt wie geplant. Ein weiterer Unterschied der im Laufe der Entwicklung geändert wurde ist das Abschießen der Farbböcke. Es war geplant, dass der Spieler die Böcke mit Farbbällen abschießt. Dies wurde zuerst auch so umgesetzt und ist im Multiplayer Spielmodus weiterhin so, aber im normalen Spielmodus verschießt der Spieler einen Laserstrahl.

Der Beispielraum aus dem Konzept, in dem der Spieler Monster abschießen muss haben wir in abgeänderter Form im Multiplayer Modus umgesetzt.

Soundeffekte waren im Konzept nicht mit eingeplant, wurden aber im Endprodukt mit eingebaut.

Die meisten Ideen aus dem Konzept wurden genauso umgesetzt wie sie geplant waren, bis auf einige kleine Ausnahmen, durch die das Spielprinzip jedoch nicht verändert wurde.

Anfangs war angedacht, dass es auch gewisse Passagen gibt, die eher aus Geschicklichkeitsspielen bekannt sind, z.B. zu bestimmten Zeiten etwas abschießen. Durch den Farblaser war das nicht mehr so leicht umzusetzen, daher haben wir so das bisher nicht implementiert.

Es sind generell noch viele Weiterentwicklungen denkbar, mal abgesehen davon, den Umfang des Spiels auszubauen. Weitere Rätselideen sind sicher noch zu finden und andere Modi sind auch denkbar.

Das Multiplayer-Level ließe sich auch erweitern, z.B. mit mehr Spielern (bis vier wäre gut vorstellbar – dann müssten aber wohl andere Figuren her um sich unterscheiden zu können), oder z.B. eine Erweiterung mit Powerups. Auch schön wäre natürlich ein lokaler Multiplayer.

Bugs sind, wie in jedem anderen Spiel auch, auch hier bei uns noch vorzufinden. Zumindest kleinere Verbesserungen hier und da werden wir mit Sicherheit in der kommenden Zeit noch vornehmen.

6. Credits

Alle Grafiken sind von uns.

Der Sound, der verkündet, dass man ein Level geschafft hat, ist mit dem kostenfreien Programm Bosca Coeil erzeugt worden.

<http://online.boscaceoil.net/#!/>

Die anderen Sounds sind mit Bxfr erzeugt worden:

<http://www.bfxr.net/>

Für die 3D-Objekte wurden einfache Objekte in Unity erzeugt oder in Blender

gebaut.

Für das Networking wurde das offizielle Unity Tutorial zu Rate gezogen:
<https://unity3d.com/de/learn/tutorials/topics/multiplayer-networking>

Sowie das im Unterricht empfohlene Video von Bytes Crafter:
<https://www.youtube.com/watch?v=V6wEvT6G92M>

Außerdem sehr hilfreich gewesen:

„Create a Game (Unity 5)“ Youtube-Tutorial-Serie von Sebastian Lague:
https://www.youtube.com/playlist?list=PLFt_AvWsXI0ctd4dgE1F8g3uec4zKNRV0

Und:

<https://github.com/SebLague/Blender-to-Unity-Character-Creation>
Vom selbigen, woher das PlayerControllerNetwork Skript aus unserem
Multiplayer-Level adaptiert wurde.

StackOverflow und Unity Answers waren ebenfalls immer wieder nützlich.