

Dokumentation VR Projekt

Idee/Planung

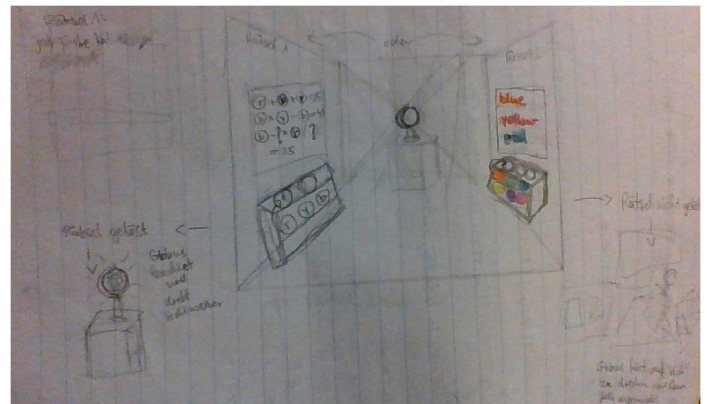
Für die erste Projektbeschreibung, wählten wir zwei Wörter von einer Liste auf einer Website aus. Diese sollte uns bei der Ideenfindung helfen. Ich entschied mich für die Wörter Mystery und Power. Die Voraussetzungen für das VR-Game waren ein Raum, ein rotierender Globus, Interaktion und ein Rätsel zum Lösen.

Technisches SetUp

Änderungen

Das Spiel sollte ursprünglich für eine Oculus Rift entwickelt werden. Da es aber eine Nachabgabe war, hat sich das geändert. Dennoch recherchierte ich das Technische SetUp für die Oculus Rift.

Meine erste Idee war ein rotierender Globus, der in einem Raum steht und von Lasern umgeben ist bzw. geschützt wird. Um diese Laser zu deaktivieren, muss man ein Rätsel lösen. Ein Rätsel war aber am Anfang noch nicht festgelegt. Die Laser werden nicht alle gleichzeitig, sondern teilweise deaktiviert. So kann man sich näher zum Globus hin bewegen. Wenn man das Rätsel löst und somit zum Globus gelangt, sollte dieser aufleuchten (zB Kontinente in einer Farbe, Ozeane in andere, usw.) Wenn man das Rätsel aber x mal falsch löst, sollte der Globus aufhören sich zu drehen und der Raum würde auseinanderfallen.



Nachdem ich mich ein bisschen im Internet umschaute, hatte ich zwei Ideen für das Rätsel.

Das eine war ein Mathe-Rätsel. Man hat eine Tabelle vor sich mit Rechenaufgaben und darunter 3 verschiedenfarbige Objekte. Jedes Objekt entspricht einer Zahl mit der man dann eine Rechnung lösen muss. Die Rechnung wird gelöst, wenn man die Objekte mit der richtigen Farbe auf den Platten platziert.

▲	+	▲	+	▲	=	30
▲	+	●	+	●	=	18
●	-	■	=	2		
▲	+	●	+	■	=	?

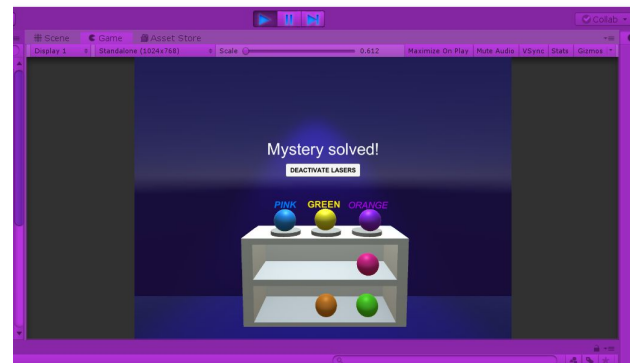
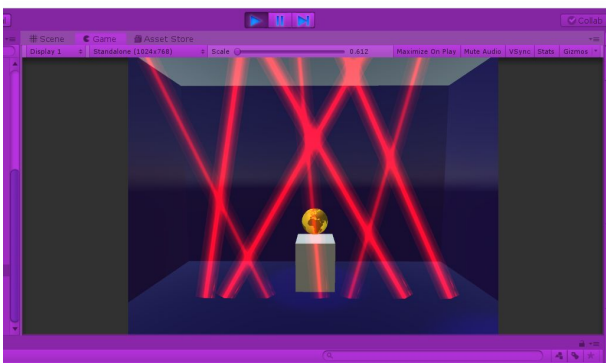
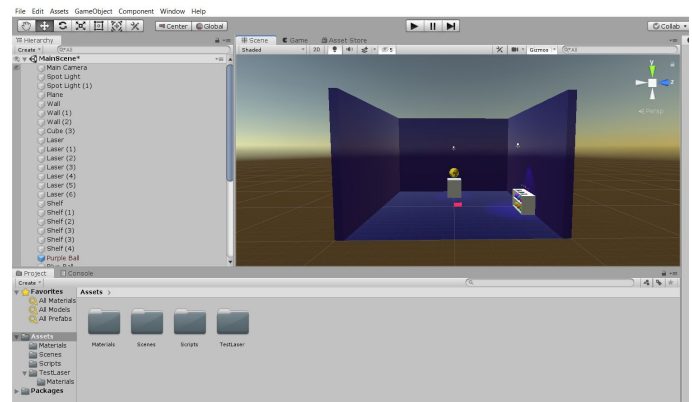
Das andere war ein Farb-Rätsel. Man hat wieder eine Tabelle auf der drei Farben geschrieben sind und darunter 3 verschiedenfarbige Objekte. Das Rätsel wird gelöst, indem man die gleiche Farbe des Objektes wie des Wortes auf den Platten platziert. Der Spieler weiß aber nicht, ob die Farbe des Wortes oder das Farbwort zur Lösung führen.

Want to play a brain game?
Say the color you see not the word that is written
as fast as you can. Ready... go!

BLUE
GREEN
BROWN
RED
GREY
WHITE
BLACK
PINK

Umgesetztes Projekt

Man befindet sich in einem Raum mit einem rotierenden Globus, Laser und ein Rätsel sind vorhanden. Das Rätsel kann gelöst werden, indem man mit den Bällen interagiert. Über den Bällen findet man die drei Wörter (Farben), zu denen man die Bälle richtig zuordnen soll. Wenn das Rätsel gelöst wurde, erscheint ein Text und ein Button. Durch das klicken des Buttons werden die Laser deaktiviert.



3D Models & Level Design

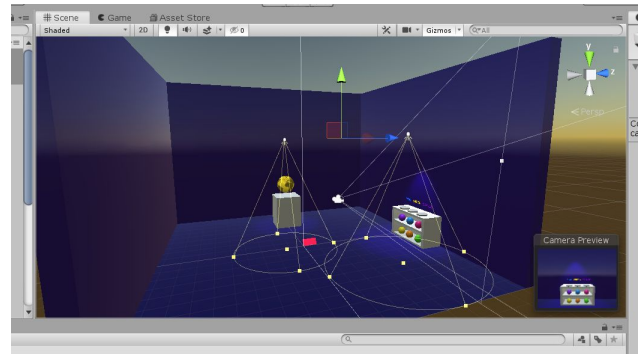
Anfangs baute ich den Raum nur mit WhiteBoxes , d.h. es wurde noch keine Textur, Farbe oder sonstiges hinzugefügt. Alles wurde selbstständig modelliert.

Room

Der Raum besteht aus einer Plane und drei Walls. Für diesen kreierte ich dann ein Material, bei dem ich die Smoothness ganz hochdrehte. Dadurch wirkt der Raum mysteriöser und hat einen transparenten Effekt.

Light

Für die Environment verwendete ich Scene lighting, um eine mysteriöse und spannende Atmosphäre zu schaffen. Außerdem werden die Farben des Raumes und der Objekte schöner betont. Um den Globus und das Rätsel hervorzuheben, wies ich beiden ein Spot Light zu.



Balls

Die Bälle sind Spheres und haben Materials, bei denen für die Farben eine Albedo Map erstellt wurde.

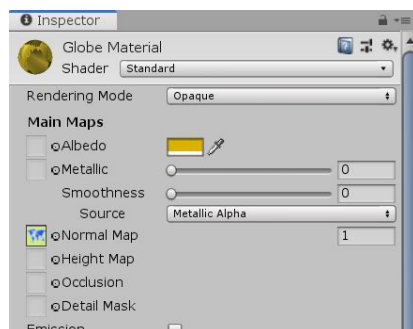
3D-Text

Für die Farben nahm ich 3D-Text her, weil man den Text hier irgendwo platzieren kann und dazu keine Canvas braucht und diese "schweben", was dann toll aussieht.



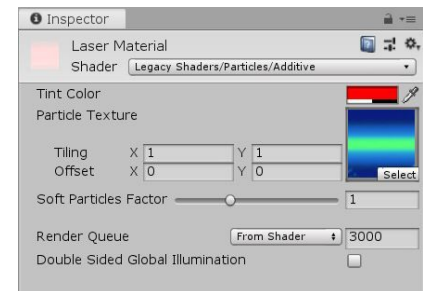
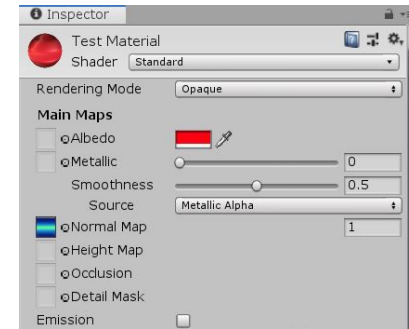
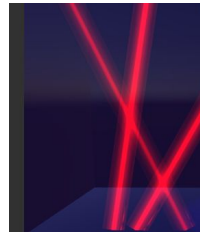
Globe

Der Globus ist eine Sphere und hierfür suchte ich eine passende Textur im Internet, erstellte eine Normal Map und zusätzlich eine Albedo Map, damit der Globus so aussieht, als wäre er aus gold.



Laser

Da ich nicht wusste wie man einen Laser macht, betrieb ich Recherche und fand ein Tutorial. Zuerst erstellte ich ein Empty GameObject, dann fügte ich eine Line Renderer Component hinzu, damit die Linie eben wie ein Laser verläuft. Um den "Laser-Effekt" zu erzielen, kreierte ich zuerst ein Material, bei dem eine Albedo Map für die Farbe und zu der Laser-Textur zusätzlich eine Normal Map erstellt wurde. Das hat aber nicht gut funktioniert. Dann erstellte ich ein neues Material, wählte beim Shader "Legacy Shaders/Particles/Additive" und fügte bei Particle Texture die Normal Map Laser Textur ein. Mit dem Ergebnis war ich dann zufrieden.



Game Mechanic Scripts

Balls

Damit hier eine Interaktion stattfindet, schaute ich mir ein paar Scripts im Internet an und fand schließlich ein passendes.

Man sollte die Bälle mit der Maus bewegen können, aufheben und fallen lassen sowie irgendwo platzieren. Um eine Bewegung durchführen zu können, werden den Bällen die Components Sphere Collider und Rigidbody hinzugefügt. Weiters merkte ich, dass die Bälle auf den Plattformen immer wieder abrutschen, deswegen fügte ich jeder Einzelnen eine Box Collider Component hinzu.

Laser

Der Laser sollte sich in beiden Richtungen ausbreiten. Dafür orientierte ich mich an einem Tutorial, bei dem ich dann am Ende den Laser beam - Effekt erzielen konnte. Danach stellte ich noch im Inspector die Position und Rotation ein.

Globe

Für die Rotation fand ich einen simplen Script im Internet, bei dem die Geschwindigkeit im Inspector entsprechend verändert werden kann, was fürs Ausprobieren sehr praktisch ist. Eine Sphere Collider Component wurde auch hinzugefügt, um die Bewegung zu ermöglichen.

Text

Das Ziel war, ein "Win Text" erscheinen zu lassen, nachdem man die drei Objekte richtig zugeordnet hat. Nach langem suchen, ausprobieren und unzähligen Codes und Tutorials, fand ich einen simplen und passenden Script. Dieser funktionierte dann im Endeffekt auch nicht, der Grund dafür war auch nicht klar. Dann entschied ich, diesen Teil wegzulassen. Nach einer Zeit fing ich dann wieder an zu recherchieren. Ich beschloss die Sache ganz anders anzugehen. Der Text sollte dann erscheinen, wenn man über die Plattform hovers. Das war nicht die beste Lösung aber immerhin hatte ich dann was. Nachdem ich einen Code dafür fand, wies ich es der Plattform zu und fügte noch eine Box Collider Component hinzu. Weiters passte ich den Code an. Das funktionierte dann alles gut, trotzdem war ich nicht ganz zufrieden damit. Also schaute ich mir den ursprünglichen Script nochmal genauer an. Ich änderte beim Ball den Sphere Collider zu Box Collider, deaktivierte "is Trigger" und erstellte einen neuen Tag "Platform". Bei der Plattform fügte ich den Box Collider hinzu, aktivierte "is Trigger" und erstellte einen weiteren Tag "BlueBall". Es klappte aber weiterhin nicht. Dann ersetzte ich im Script ".enabled" durch ".SetActive", das änderte aber nichts. Das Problem wurde im Endeffekt doch gelöst. Ich ersetzte "[SerializeField] private Text winText;" durch public GameObject winText; taggte den Ball, nicht die Plattform.

```
[SerializeField]
private Text winText; // Insert your text object inside unity inspector

void Start()
{
    winText.enabled = false; // You may need to use .SetActive(false);
}

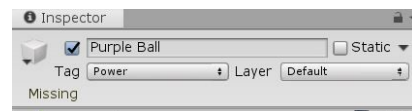
void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.tag == "BlueBall")
    {
        // This is where you make your text object appear on screen
        winText.enabled = true; // May need to use .SetActive(true);
    }
}

void OnCollisionExit(Collision collision)
{
    // Here is where you make the text disappear off screen
    if(collision.gameObject.tag == "BlueBall")
    {
        yourText.enabled = false; // May need to use .SetActive(false);
    }
}

public GameObject winText;
public GameObject buttonText;

void Start() {

    winText.SetActive(false);
    buttonText.SetActive(false);
}
```



Button

Dann wollte ich das Ganze noch erweitern und das Deaktivieren der Laser auch machen, welches über ein Button erfolgen soll. Erst suchte ich einen Code, bei dem ein Objekt bzw. Objekte beim anklicken eines Button verschwinden und der Button selbst auch. Danach fügte ich den Script in Canvas ein. Den Code passte ich entsprechend an. Im Button Inspector bei der OnClick () Liste wurde eine Funktion hinzugefügt, das GameObject hineingezogen und die Methode vom Script ausgewählt.

