



MirrAR

Ein AR-Projekt von Alexander Eggers und Linh Do

Inhalt

- Idee/Konzept
- Aufbau
- Erste technische Umsetzung
- Laserscript
- GUI
- Show-Off Version 1
- Karten als Spiegel
- Show-Off Version 2

Inhalt

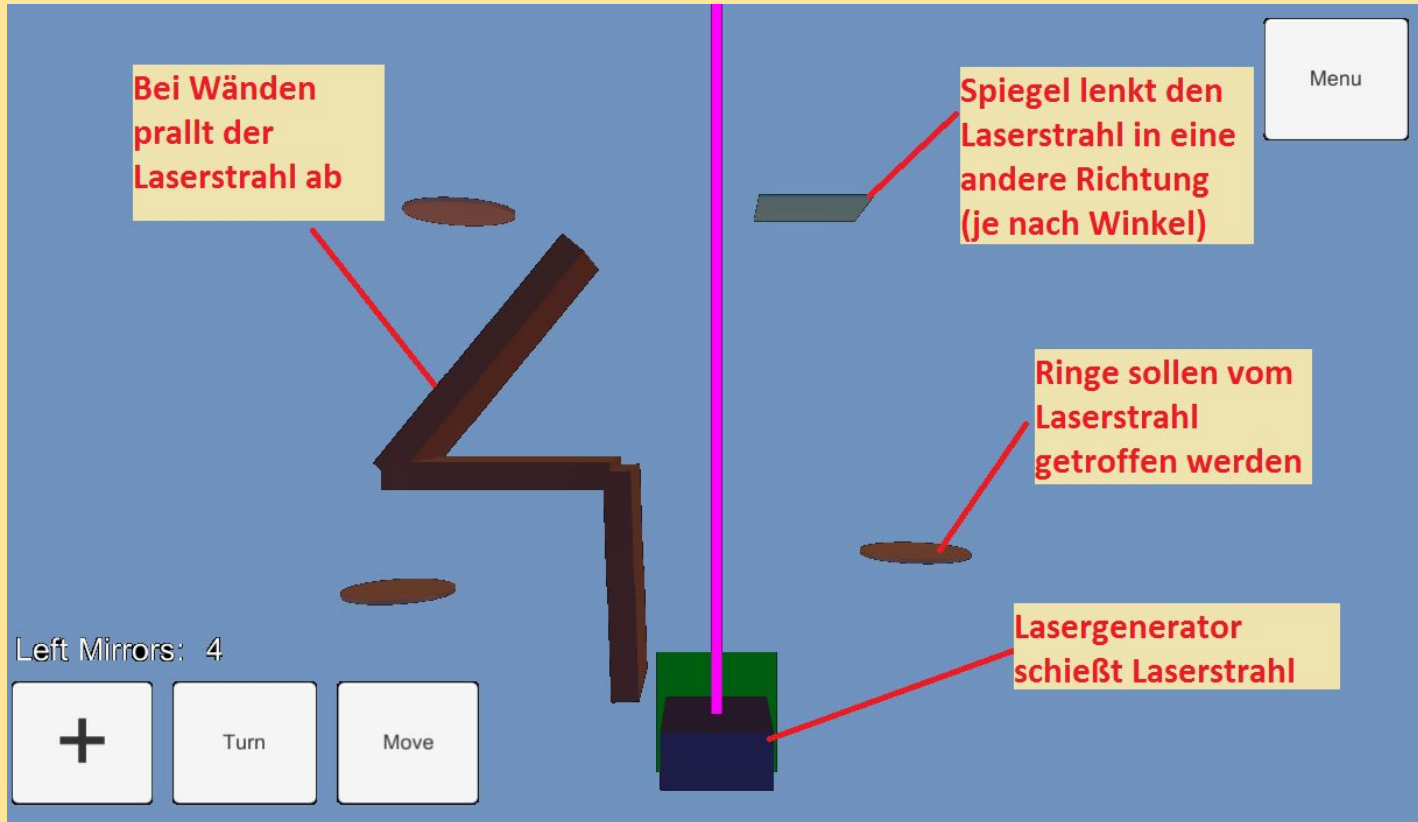
- Platzieren von Objekten
- Image Targets vs. Frame Markers
- GUI Version 1 vs. GUI Version 2
- Probleme in der Entwicklung
- Lösungsvorschläge
- Fazit

Idee/Konzept

- Grundidee: Sinnvolles Konzept finden in dem AR einen Vorteil bietet
- Anfangs Konzepte durchgegangen und auf Basis ihrer Umsetzbarkeit und ihrem AR-Vorteil bewertet → z.B. einen Lego Bauplan
- Grundkonzept basiert auf vorhandenen Spielen, welche die Idee umgesetzt haben
 - Wir wollten mit AR dem Spieler die Möglichkeiten geben, das Rätsel aus mehreren Perspektiven zu lösen



Aufbau

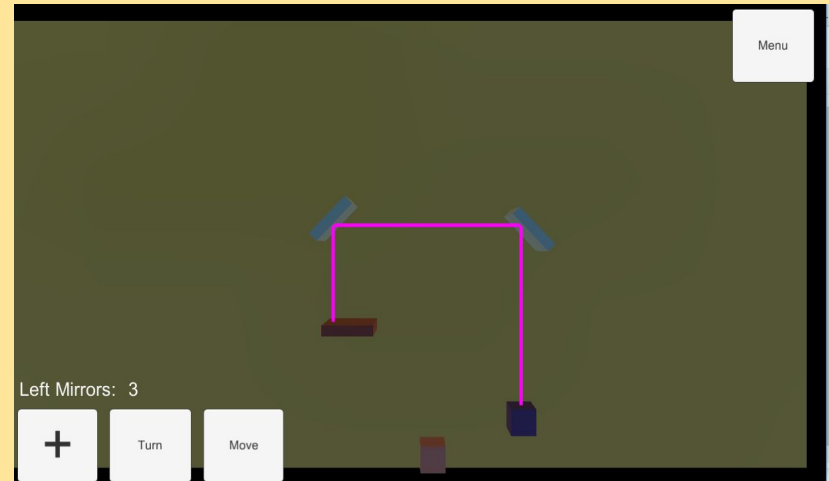
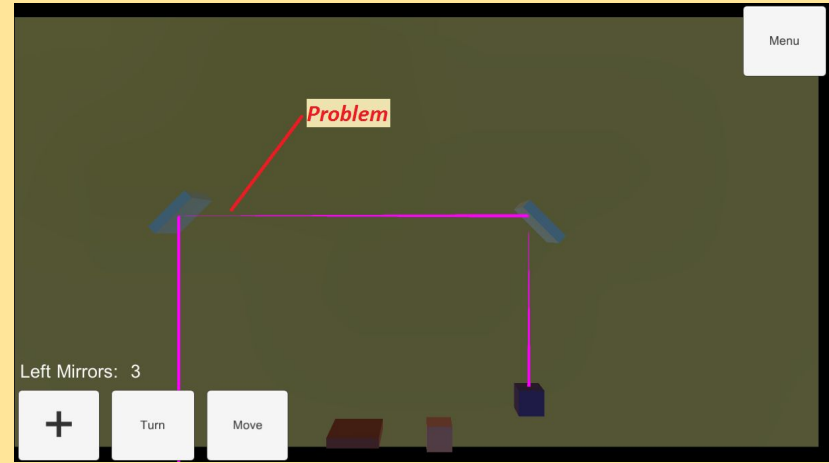


Erste technische Umsetzung

- Realisierung mit Unity und Vuforia
- Es gibt ein Image Target, welches als Referenz dienen sollte für alle Objekte
- Die Objekte (Spiegel, Wände, Generator, Target) sind nur via Gerät verschiebbar
- Die Spiegel auf dem Spielfeld sind auswählbar und können dann verschoben oder/und gedreht werden
- Es ist nur über die GUI möglich weitere Spiegel zu setzen
- Der Spieler kann nur die Spiegel manipulieren, die übrigen Objekte werden vom Spiel gesetzt (via Level 1, Level 2, ...)
- Der Spieler gewinnt wenn alle Targets durch den Laser getroffen wurde

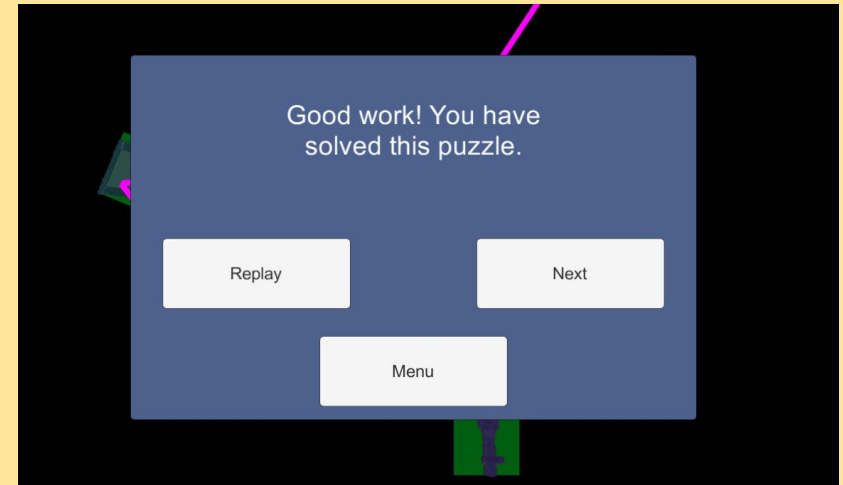
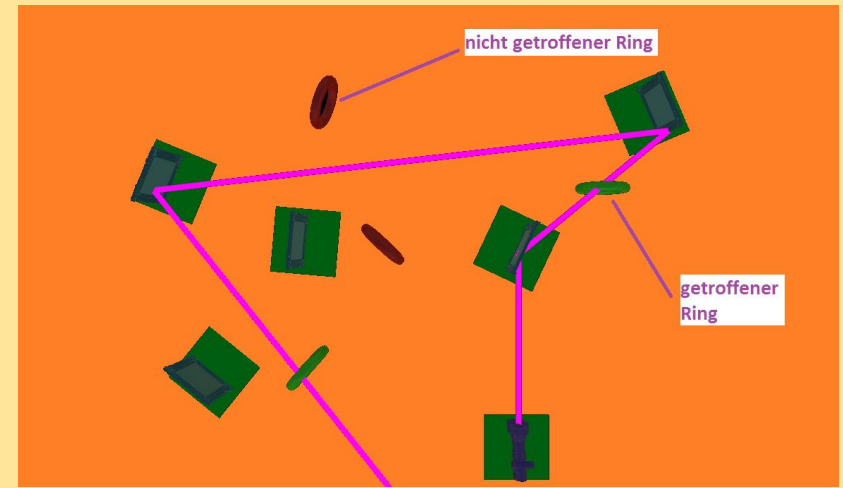
Laserscript

- mithilfe von LineRenderer
- Wand: Laserstrahl prallt ab
- Spiegel: Laserstrahl wird weitergelenkt
- Position der Points vom LineRenderer wird durch Raycast bestimmt
- Ausfallswinkel = Einfallswinkel
- Problem:
 - Linien werden gegen Ende schmaler
- Lösung:
 - nicht nur einen sondern mehrere LineRenderer instanziiieren

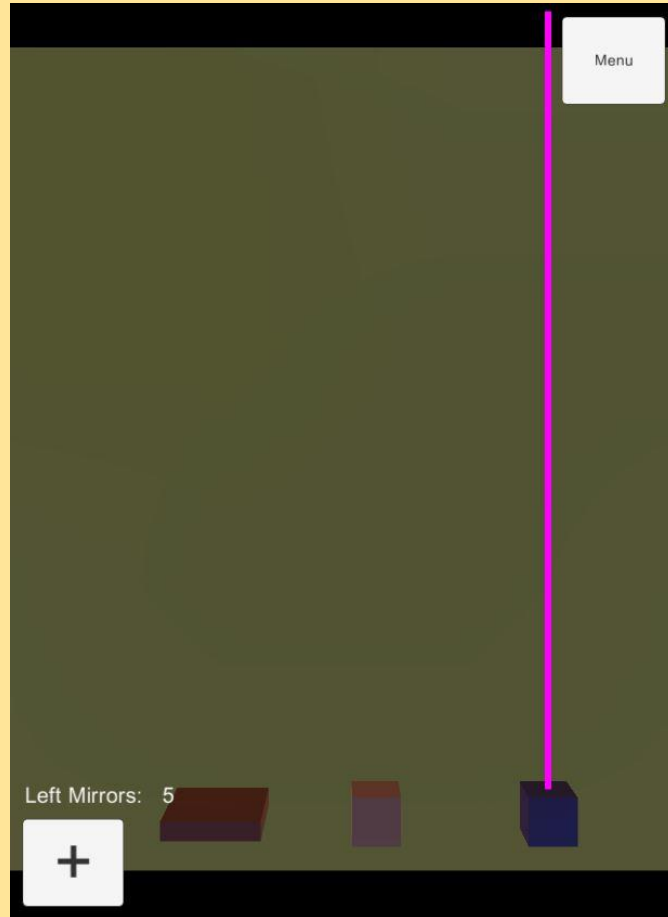


Laserscript

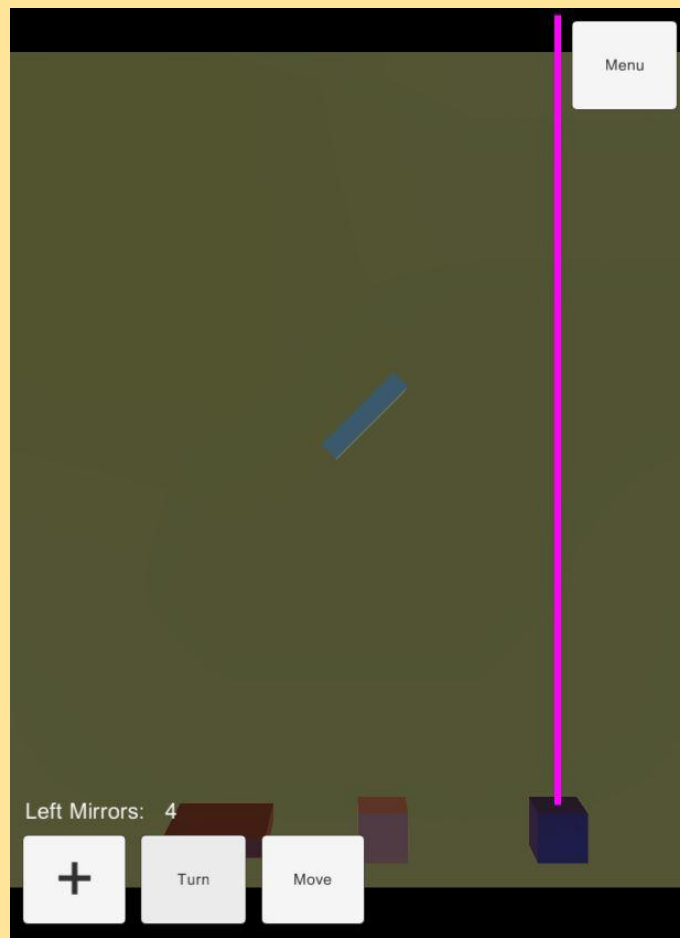
- beim Durchdringen eines Ringes ändert sich die Farbe von rot auf grün
- Laser soll alle Ringe treffen
 - es erscheint ein Gewinn-Fenster



GUI



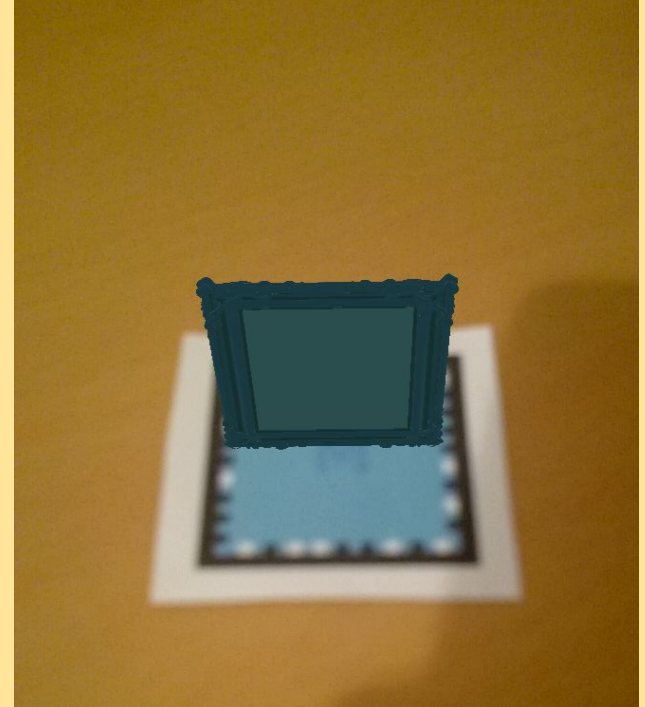
GUI



Show-Off 1

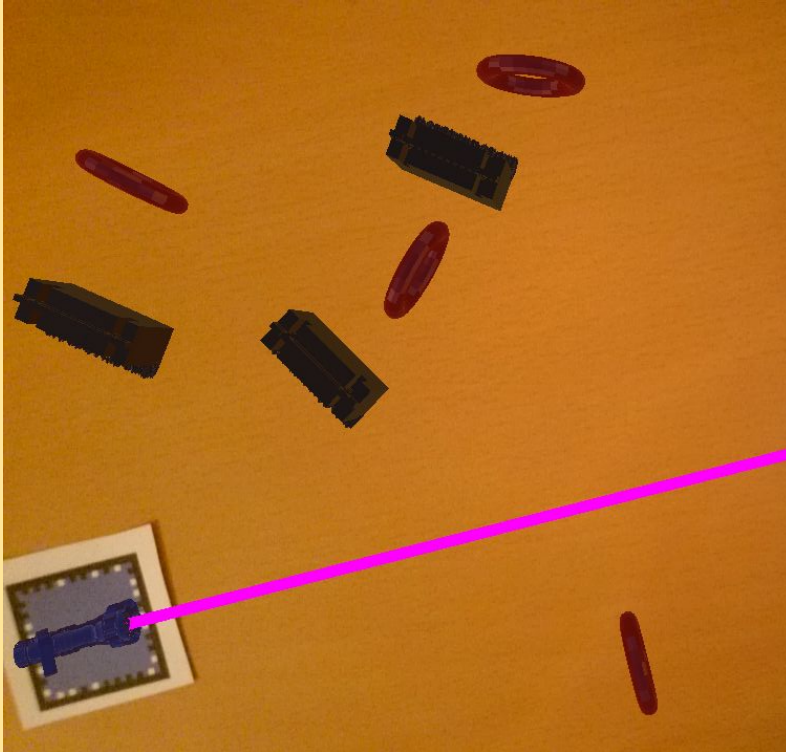
Version 2: Kärtchen als Spiegel

- vermischt mehr die virtuelle mit der realen Welt
- intuitiver die Spiegel über Kärtchen zu platzieren als über den Bildschirm/die GUI
- Rotation des Spiegels einfacher und schneller
- Smartphone wird nur noch als Kamera gebraucht
 - muss nicht in der Hand gehalten werden (z. B. zum Platzieren der Spiegel)
 - beide Hände frei



Show-Off 2

Lasergeneratorkärtchen



- auf dem dazugehörigen Frame Marker ist der Lasergenerator
- um diesen Frame Marker wird die Spielewelt mit den Ringen und den Wänden aufgebaut

Spiegelkärtchen

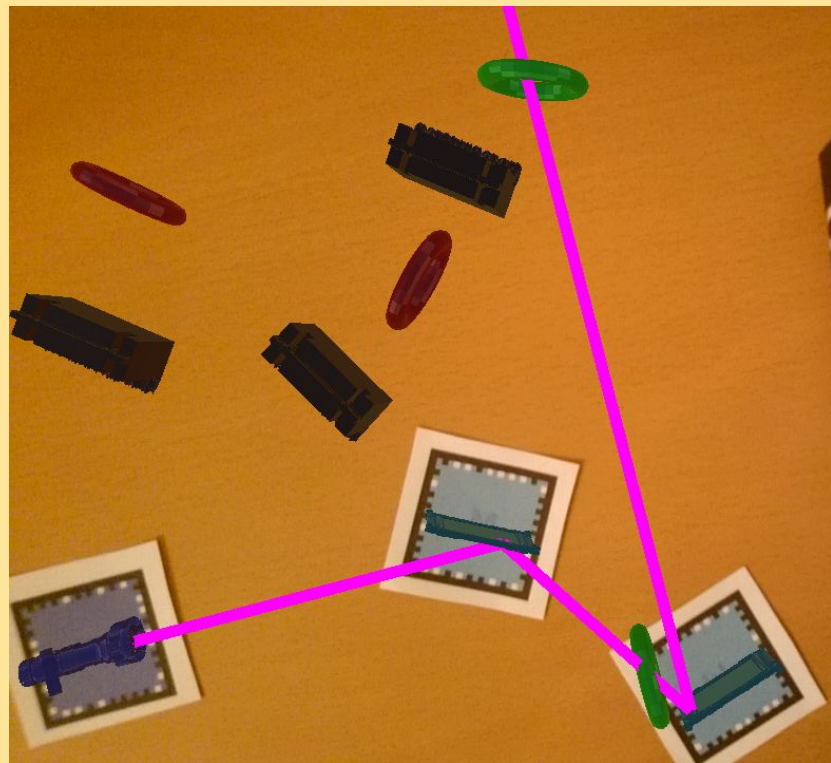
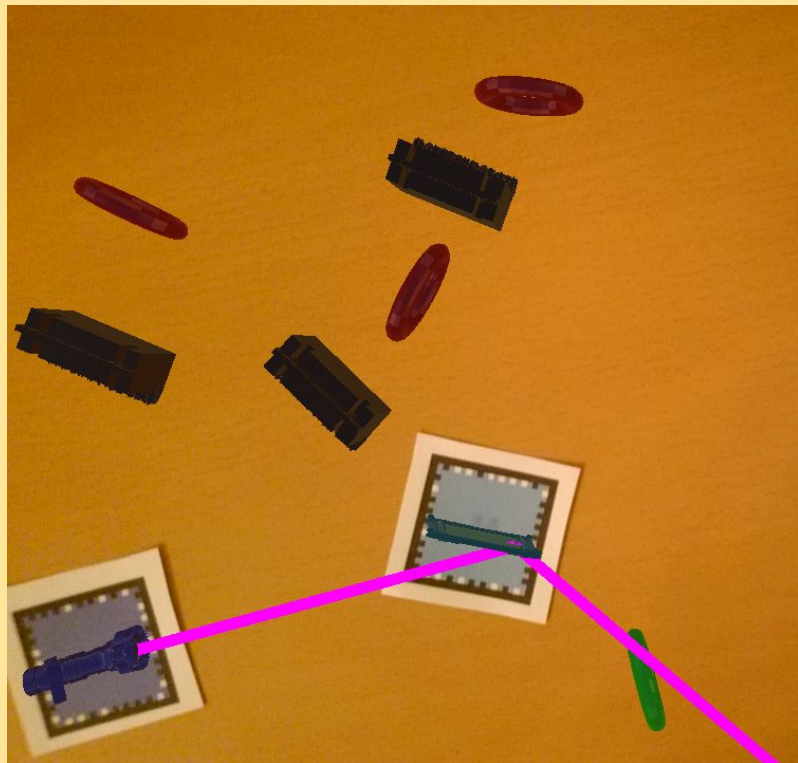


Image Targets

VS

Frame Marker

Image Target

- Bild werden durch seine Struktur erkannt
- Device Database kann bis zu 100 Bilder speichern
- wird auch weiterhin getrackt, wenn Teile des Bildes(Targets) fehlen

Nachteile für das Spiel

- Größe: mindestens 12cm x 12cm
- nur bis zu 5 Bilder können gleichzeitig getrackt werden



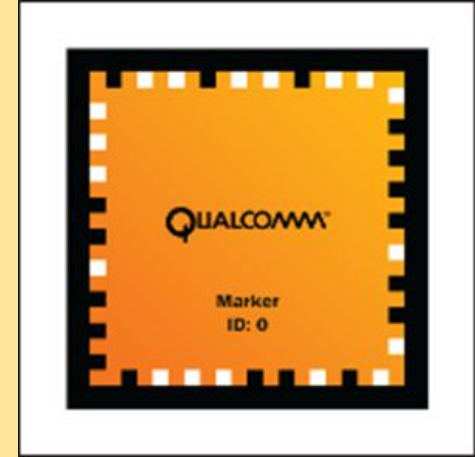
Frame Marker

- Erkennung durch die binären Muster an den Rändern
- Nachteil: Wenn Teile des Frame Markers fehlen, wird das Tracking nicht mehr fortgesetzt

Vorteile für Spiel

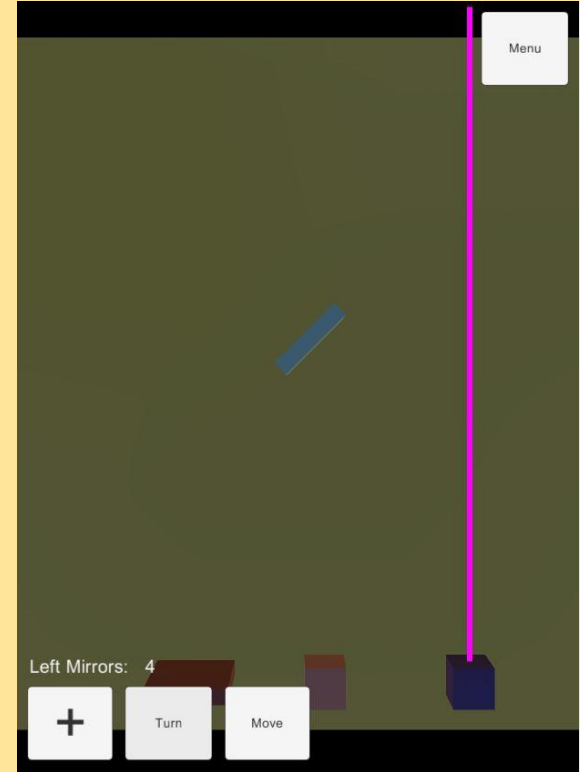
- bis zu 512 verschiedene Frame Marker möglich
- Größe: schon 3cm x 3cm möglich
 - ideal als Spiegel

Entscheidung für Frame Markers



GUI Unterschied Version 1 zu Version 2

- GUI Elemente zum Großteil rausgeworfen
 - Hat keine Notwendigkeit mehr, weil wir die meisten Inhalte via Kärtchen umgesetzt haben (bzgl. GUI)
- Menüs weiterhin vorhanden
- GUI dient nur zur Strukturierung der unterschiedlichen Level und für ein besseres UX Design beim Spielstart

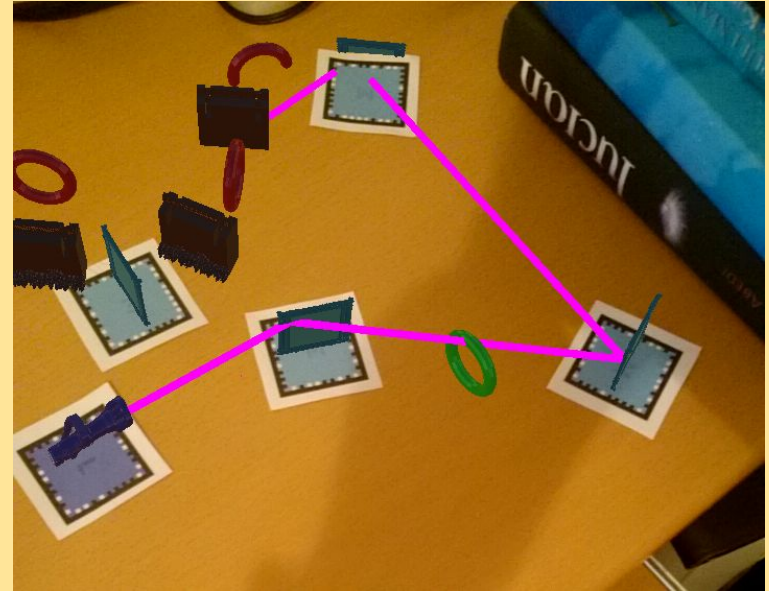


Probleme bei Version 2

- Unterschiedliche Frame Marker, die gleichzeitig getrackt werden
 - verursachen Probleme bei der Interaktion der Objekte (z.B. Spiegel) miteinander, bspw.
- beim Tracken rotiert jedes Objekt je nach Kameraperspektive
 - Verlauf des Laserstrahls ändert sich, wenn die Kamera bewegt wird
 - Kamera muss starr in einer bestimmten Position bleiben
 - Versuch, das Problem zu lösen, indem sich der Laserstrahl nur in der X-Achse und der Z-Achse bewegt (und nicht in der Y-Achse)
 - hat nur teilweise funktioniert: sobald die Kamera zu schräg auf die Frame Marker zeigt, gibt es immer noch Probleme, dass der Laserstrahl über die Spiegel hinweg weitergeht

Probleme bei Version 2

- das Tracking bricht schnell ab (z.B. wenn die Distanz etwas zu groß ist oder die Beleuchtung zu hell)
- zuviele Objekte (z.B. Wände) im Level → Ruckeln
 - Performance des Handy noch nicht gut genug (vor allem wenn es zu viele Objekte mit Collider gibt)
 - Level muss angepasst werden, z.B. das mehrere Wände, die nebeneinander platziert sind zusammen einen Collider bekommen



Lösungsvorschläge/Verbesserungen/Erweiterungen

- Laserscript umschreiben: Reparenting
 - Sobald sich das Kärtchen nicht mehr bewegt, ist nicht mehr der eigene Frame Marker der Parent, sondern der Frame Marker vom Lasergenerator
 - dadurch hat man nicht mehr das Problem mit der Rotation, weil die Position und Rotation der Spiegel auf die Welt (das sogenannte World Center) des Lasergenerator Frame Markers angepasst wird
- die Scripte von Vuforia anpassen
 - auch wenn der Frame Marker nicht mehr getrackt wird, bleiben der Lasergenerator und die Spiegel im Spiel (die letztbekannten Position bleiben einfach erhalten)
- mehr Level bauen
- Sound

Fazit

- Dokumentation zu Vuforia spärlich → teilweise veraltet, fehlerhaft oder es fehlen wichtige Details
- Technik noch nicht ganz ausgereift → Versuche mit Cardboard scheiterten deswegen
- Allgemein hätte man von Anfang an experimenteller an das Projekt rangehen sollten (Image Target vs. Frame Marker)
- Wir haben uns ein anderes Endergebnis vorgestellt, aber insgesamt sind wir zufrieden mit dem, was wir geschafft haben

Quellenverzeichnis

https://vuforiacontent.qualcomm.com/Images/devGuide_ImageTargets.jpg

https://vuforiacontent.qualcomm.com/Images/frame_marker_example_create_targets.png

<http://artishock.com/system/img/augmented-reality-apps/lego-connect.jpg>

<https://www.getdigital.de/web/getdigital/gfx/productsMore/khet-ingame.jpg>