




EMOJI EXARCISE

EMOJI EXARCISE

Ein (AR-) Spiel von Simon Wedel und Dominik Keller,
entwickelt mit  in Ilmenau, Berlin, Aarhus und Trondheim.

©2016, 2017

Team



Dominik Keller

Level-Design

Scripting

Audio

Story

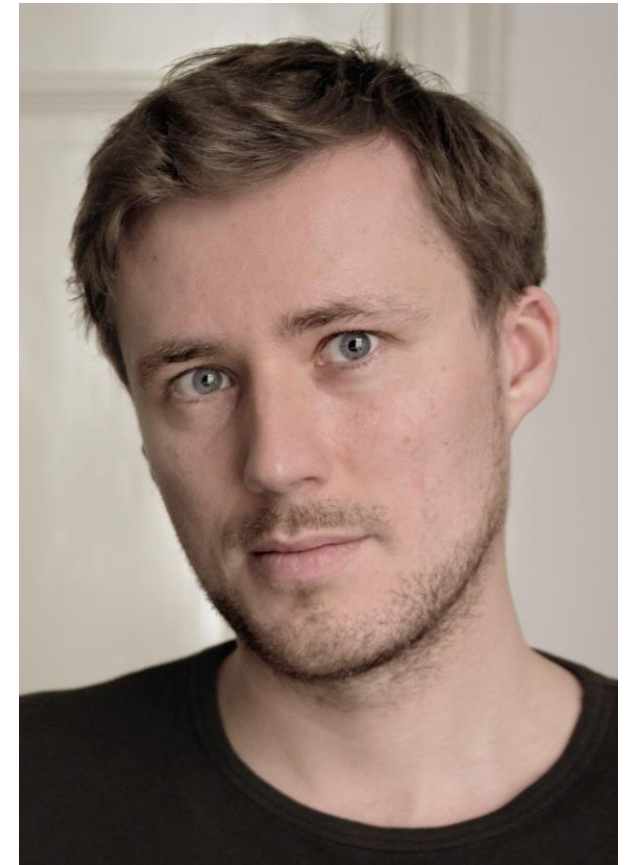
Design

Technik

Scripting

Basteleien

Simon Wedel



- Medientechnologie-Studenten
- keine Erfahrungen im Bereich Game Development
- Erstellung an vier verschiedenen Standorten

Spielidee

- entstanden im Rahmen der Lehrveranstaltung *Game Development* an der TU Ilmenau
- ursprüngliche Idee: virtuelles Spielbrett in Realität verankern und mit Markern steuern
- Problem: Markersteuerung ohne visuelles Feedback schwierig
- Lösung:
 - Projektion eines virtuellen Spielbretts auf reale Wand
 - Steuerung der Spielobjekte über Positionierung von realen Markern
 - Augmented Reality -> Spielsteuerung per Marker-Tracking

Spielkonzept

- Zweidimensionales Spielbrett
- Nutzung von Gravitation und Spielobjekten
- Ziel des Spiels: Kugel erreicht Bestimmungsort
- diverse Hindernisobjekte und Aktionsflächen im Spielbrett
- Aufgabe des Spielers: Leiten der Kugel durch Positionierung (Translation / Rotation) spezieller Spielobjekte

Story

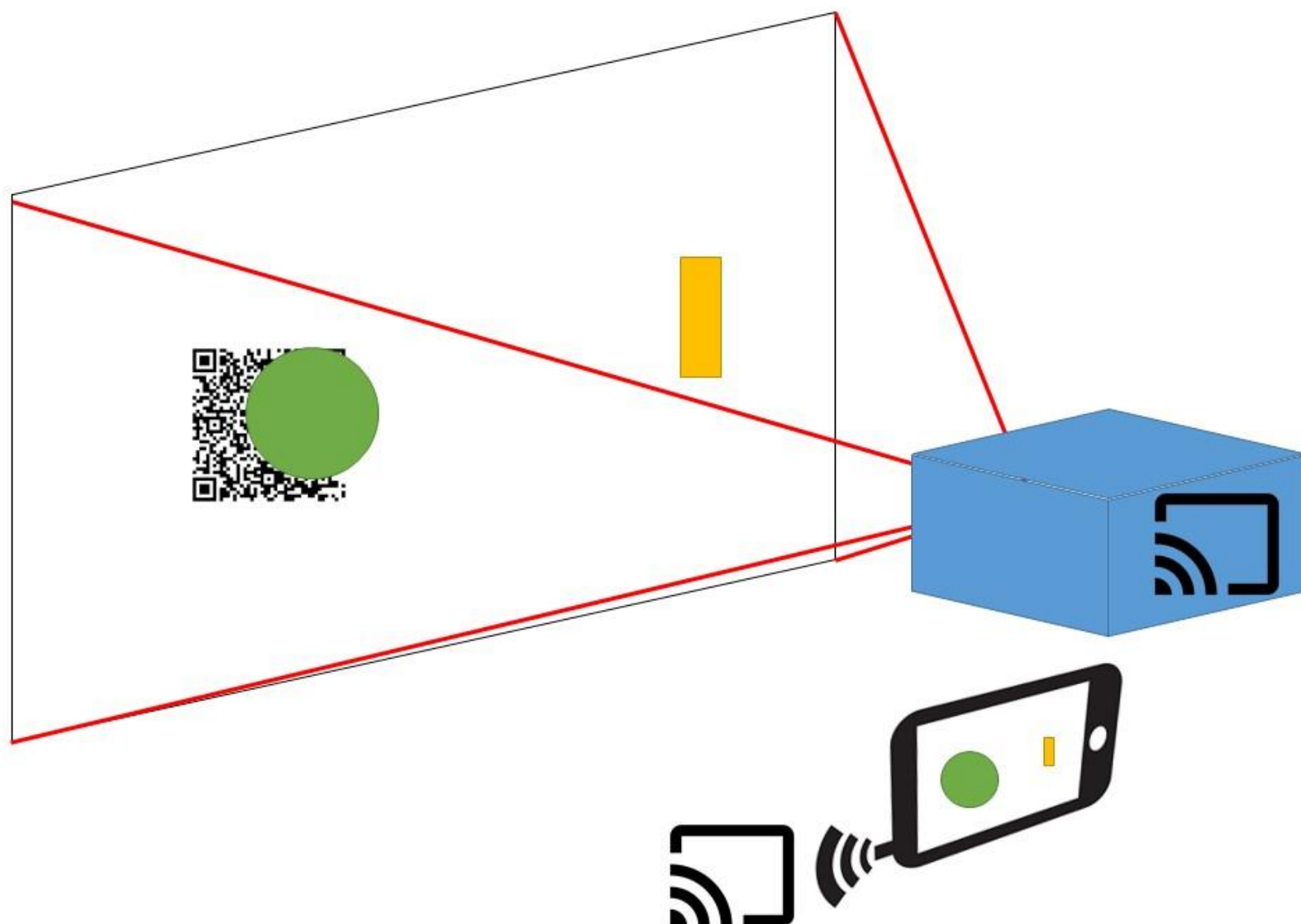
- Von Krieg, Kampf und Terror geplagte Welt
- Wenig sozialer Austausch, Verbot von Gefühlen
- Emojis als Möglichkeit, Emotionen zu übermitteln
- Hans und seine Emoji-Freunde als Hauptakteure
- Wunsch: versendet werden!
- Verwendung von Asciiis (Schriftzeichen) als Hilfsmittel zur Versendung der Emojis



Hans und seine Freunde

Technisches Konzept

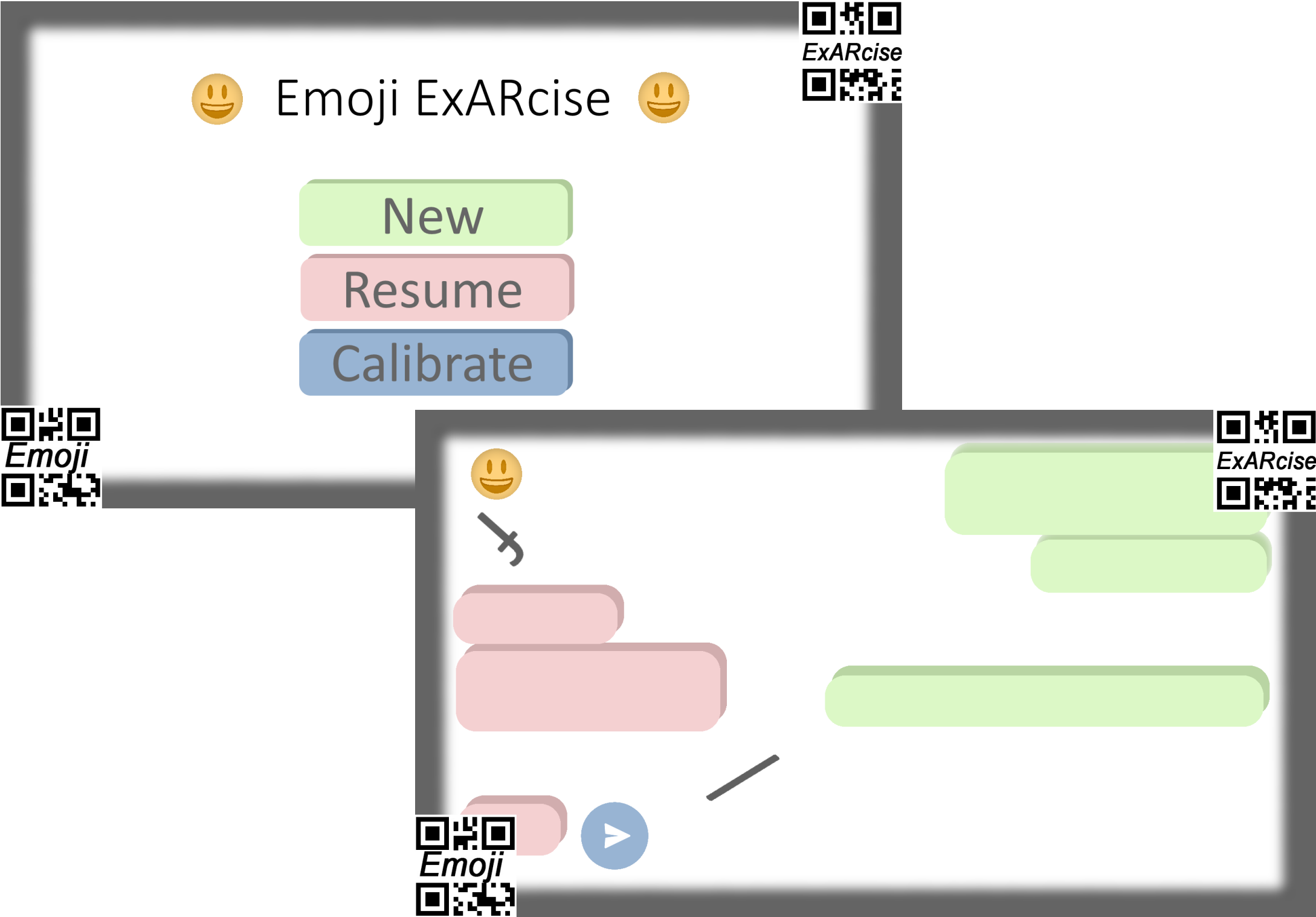
- Android-App, erstellt mit Game Engine Unity und Vuforia als AR-SDK



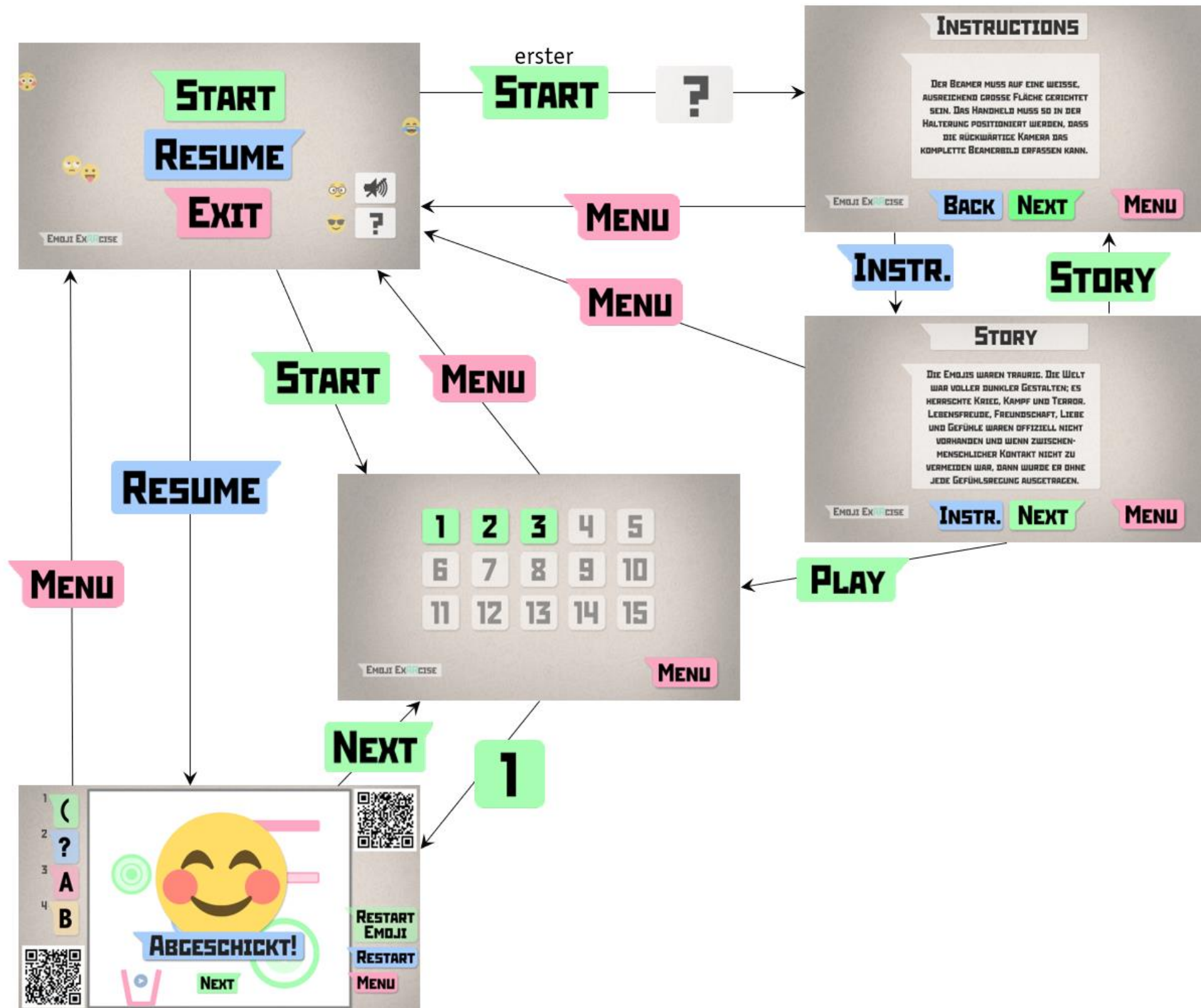
Entwicklungsverlauf

- Oktober/November: Idee, Story, Sondierung
- November/Dezember: Einarbeitung in Unity und Vuforia, technische Prototypen, Mockups
- Januar/Februar: Game-Prototypen, Audio
- Februar/März: Design, Level-Design, Menü, Instruktionen, reale Spielobjekte, Beseitigung von Bugs
- März: Finalisierung, Optimierung

Mockups

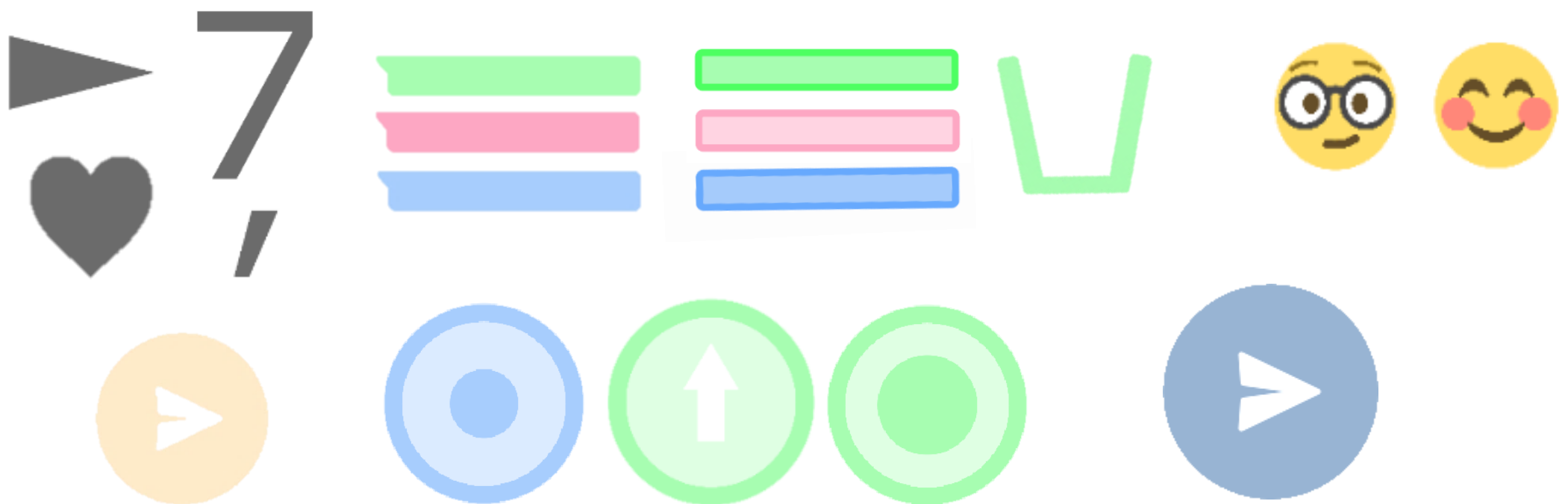


Struktur



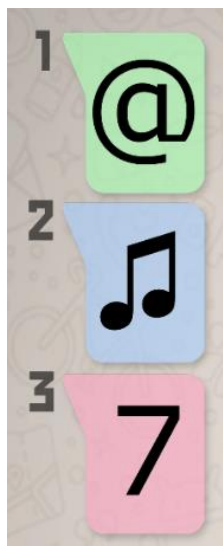
Spiel- und Designelemente

- Design-Anleihen von diversen Messengern
- Optimierung und Anpassung aller Elemente an Projektor-Situation und Markererkennung
 - helle, kontrastarme Flächen
 - einfache Formen
 - Maximalkontraste bei den Markern



Spiel- und Designelemente

- Design-Anleihen von diversen Messengern
- Optimierung und Anpassung aller Elemente an Projektor-Situation und Markererkennung
 - helle, kontrastarme Flächen
 - einfache Formen
 - Maximalkontraste bei den Markern








**START
EMOJI**

RESTART

MENU

Funktion der Spielelemente

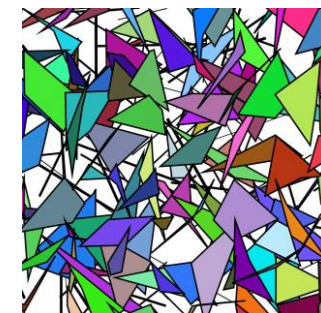
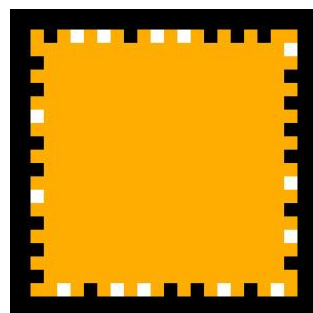
-  ... muss vom Emoji erreicht werden
- Änderungen der Gravitation im Spielfeld
 -  ... negiert die Gravitation
 -  ... verringert die Gravitation
 -  ... erhöht die Gravitation
-  ... schaltet den Send-Button frei

Audio

- Erstellung von Spielmusik und Spielsounds
- Inspiration durch frühe Messenger-Sounds
- Komposition eines Haupt-Themas
- Verwendung in mehreren Abwandlungen sowohl als Musik als auch als Sounds

Technische Herausforderungen: Markererkennung

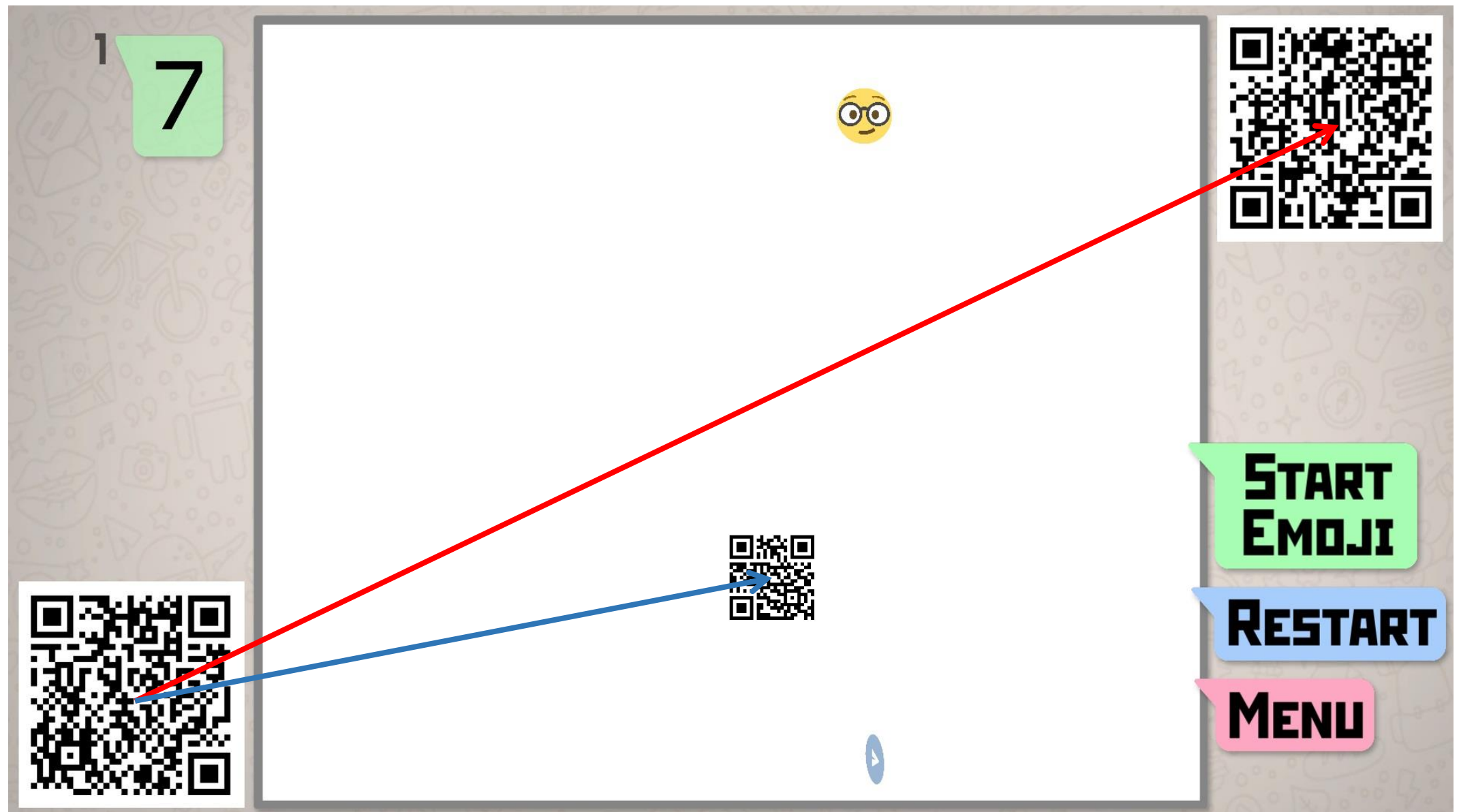
- Erkennung weit entfernter Marker unter gegebenen Bedingungen unzuverlässig (Vuforia)
- Projektion des virtuellen Objekts auf Marker
→ erschwert Markertracking
- Hohe Anforderung an Qualität der Kamera und Projektor
- Größe und Typ der Marker entscheidend für Erkennung



Technische Herausforderungen: **Positionierung der Kamera**

- Ausgangssituation: Virtuelle Spielobjekte mit realen Spielobjekten verankern
- Ziel: Unabhängigkeit von Kameraposition
 - Einzige Bedingung: Kamera muss gesamte Projektionsfläche erfassen
- Lösung: „Kalibrierungsmarker“

Technische Herausforderungen: Positionierung der Kamera



Komplikationen & Bugs

- Einschränkungen im Spieldesign
- Nutzung von AR-Elementen in unüblicher Weise
- Großteil der Entwicklung nicht an gemeinsamen Standort
- Jitter der marker-erkannten Spielobjekte
- Levelabschluss cheatbar
- Darstellung bei Seitenverhältnissen anders als 16:9 nicht optimal

Lernerfolg

- Teamzusammenarbeit über Ländergrenzen hinweg
- Zeitmanagement
- Nutzung von Versionsverwaltungssystem
- Unity, Vuforia, C#-Scripting
- allumfassende Entwicklung eines Spiels

Fazit

- Ziel *Entwicklung eines (AR-) Games* erfüllt
- neuartiges, unterhaltsames Android-Spiel
- Game-Development mehr als nur Programmierung und Design
- Markererkennung als Spielsteuerung nicht optimal
- hoher Zeitaufwand