General Technological Overview

The main components of the setup are as follows:

The Twitch-Chat-Scraper Script: If a Chatter types a command into the chat window this script receives it via the Twitch-API and parses it into Keyboard-Commands.

The main Unity Application: (Has two purposes) Receives those Keyboard-Commands and executes them. Accordingly, it adds votes to the ballot, switches cameras and relays commands to the Gravitrax Connect Client Script via other specific keyboard command.  
Aside from that it arranges and displays all camera feeds - with visualizations for all of the voting processes and interaction options.

The Gravitrax Connect Client Script:  
Receives the Keyboard-Commands output by the Unity Application and transfers those via Bluetooth to the “Gravitrax Power Connect Brick”.

Gravitrax Power Connect Brick:  
This in Turn relays this information to the respective Gravitrax Power Elements (e.g. Switches, Starters or Levers) as a Signal on their “red”, “green” or “blue” channel.

Cameras:  
A mix of mirrorless DSLMs and Webcams capture the Marble track in its entirety with emphasis on important hotspots, such as controllable Switches. Their video feed gets ingested into the PC running the Unity Scene Manager Application in real time through either USB or HDMI links.

OBS: Captures the entire Unity graphical Frame and streams it to Twitch for everyone to see.

OG:

1. Abstract
2. Introduction
   1. State of the Art
   2. Motivation
   3. Goal
3. Finding Ideas and creative processes [taking a step back here]
   1. Extraneous Circumstances and initial request
   2. Creative Process Paradigm: 4D-Phases
      1. Discover: (Get Ideas via looking at Creative Tools/Processes & Models<such as Player Types>)
   3. Game Idea A: Melody Playground
   4. Game Idea B: Logistics Puzzler
      1. Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)
   5. Streaming Idea: Interactive Livestream
      1. (Basic Concept)
      2. Inspiration and History of interactive live Video Feeds
      3. Multicam
      4. Moderation?
         1. A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…
      5. Streamers
         1. Interacting with a built track
         2. Building a track with feedback
4. Implementation (/Transfer?)
5. OR Technical Implementation
   1. Physical Set
   2. Core Software Framework (Unity + 2 PyScripts)
   3. Unity
      1. Video Input
      2. Camera Array (Video-Mixer?)
      3. More interactivity: Sfx
   4. Light-Up Switches
   5. Ausblick: Virtual Particle Systems in 3D Space
   6. Ausblick2: Clickmaps
   7. Ausblick (more vague): Build of an Independent webapp
6. AND Evaluation
   1. Ausblick hier?

Klemens ver.

1. Abstract
   1. Tags and Buzzwords! (unity3d, 3dPrint, CAD-Software)
   2. Ergebnisse (coole latenz…
2. Introduction
   1. Motivation
   2. Goal (ausgearbeitet)
3. Finding Ideas and creative processes *[taking a step back here]*
   1. {Extraneous Circumstances and }initial request
   2. Creative Process Paradigm: 4D-Phases
      1. *Discover: (Get Ideas via looking at Creative Tools/Processes & Models<such as Player Types>)*
   3. Game Idea A: Melody Playground
   4. Game Idea B: Logistics Puzzler
      1. *Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
   5. Streaming Idea: Interactive Livestream
      1. *(Basic Concept)*
      2. Inspiration and History of interactive live Video Feeds
      3. Multicam
      4. Moderation?
         1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
      5. Streamers
         1. Interacting with a built track
         2. Building a track with feedback
4. State of the Art
   1. Grundlagen? (Technische basics, api, twitch how does it communicate)
5. Hardware Implementation
   1. Camera Setup
   2. Lighting and Set
   3. Marble Tracks
   4. *Expansion*: Configuring a custom Daughterboard for a tightly integrated Set
6. OR Technical Implementation
   1. Physical Set
   2. Core Software Framework (Unity + 2 PyScripts)
   3. Unity
      1. Video Input
      2. Camera Array (Video-Mixer?)
      3. More interactivity: Sfx
   4. Light-Up Switches
   5. *Ausblick*: Virtual Particle Systems in 3D Space
   6. *Ausblick2*: Clickmaps
   7. *Ausblick3*: Development of an independent Webapp

Manu ver.

1. *Abstract*
   1. *Tags and Buzzwords! (unity3d, 3dPrint, CAD-Software)*
   2. *Ergebnisse (coole latenz…*
2. Introduction
   1. Motivation *(kurz halten)*
      1. *WICHTIGSTER:Streams immer cooler, interaktive noch cooler! -> immer mehr Firmen (z.B. Product launches)*
      2. *(Digitale testversionen easy) Physische produkte testen und ausprobieren*
      3. *Digitalspielzeuge habens schwer, ich hole die leute ab wo sie eh schon sind (kein produktkauf nötig, soziales zusammenspielen)*
   2. Goals and /Abgrenzung/ (ausgearbeitet)
   3. Structure *(Inhaltsverzeichnis in prosa)*
3. Conception: Finding Ideas and creative processes *[taking a step back here]*
   1. {Extraneous Circumstances and }initial request
   2. Creative Process Paradigm: 4D-Phases
      1. *Discover: (Get Ideas via looking at Creative Tools/Processes & Models<such as Player Types>)*
   3. Game Idea A: Melody Playground
   4. Game Idea B: Logistics Puzzler
      1. *Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
   5. Streaming Idea: Interactive Livestream
      1. *(Basic Concept)*
      2. UI and 3 Wireframes
      3. Inspiration and History of interactive live Video Feeds
      4. Multicam
      5. Moderation?
         1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
      6. Streamers
         1. Interacting with a built track
         2. Building a track with feedback
      7. //Gibt keinen Programmablauf, nonlinear, come in and have fun…oder eben Phasen
      8. *ANforderungsanalyse? (Ziele optimierung visual quality vs latency/performance) (vielleicht später)*
4. State of the Art
   1. Grundlagen? (Technische basics, api, twitch how does it communicate)
   2. Conflicts: Telegames (I have: no fail condition, lower laterncy, social chaos is feature)
   3. Potential (Motivation an konkreten beispielen) (konkrete beispiele vlt hier und mein Wunsch/pipe dream oben in konzeption)
5. Hardware Implementation
   1. Camera Setup
   2. Lighting and Set
   3. Marble Tracks
   4. *Expansion*: Configuring a custom Daughterboard for a tightly integrated Set
6. OR Technical Implementation
   1. Physical Set
   2. Core Software Framework (Unity + 2 PyScripts)
   3. Unity
      1. Video Input
      2. Camera Array (Video-Mixer?)
      3. More interactivity: Sfx
   4. Light-Up Switches
7. Evaluation
8. Fazit & Ausblick
   1. *Ausblick*: Virtual Particle Systems in 3D Space
   2. *Ausblick2*: Clickmaps
   3. *Ausblick3*: Development of an independent Webapp

Manu – redacted ver.

1. *Abstract*
   1. *Tags and Buzzwords! (unity3d, 3dPrint, CAD-Software)*
   2. *Ergebnisse (coole latenz…*
2. Introduction
   1. Motivation *(kurz halten)*
      1. *WICHTIGSTER:Streams immer cooler, interaktive noch cooler! -> immer mehr Firmen (z.B. Product launches)*
      2. *(Digitale testversionen easy) Physische produkte testen und ausprobieren*
      3. *Digitalspielzeuge habens schwer, ich hole die leute ab wo sie eh schon sind (kein produktkauf nötig, soziales zusammenspielen)*
   2. Goals and /Abgrenzung/ (ausgearbeitet)
   3. Structure *(Inhaltsverzeichnis in prosa)*
3. Conception: Finding Ideas and creative processes *[taking a step back here]*
   1. {Extraneous Circumstances and} initial request
   2. Creative Process Paradigm: 4D-Phases
      1. *Discover: (Get Ideas via looking at Creative Tools/Processes & Models<such as Player Types>)*
   3. Game Idea A: Melody Playground
   4. Game Idea B: Logistics Puzzler
      1. *Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
   5. Streaming Idea: Interactive Livestream
      1. *(Basic Concept)*
      2. UI and 3 Wireframes
      3. Inspiration and History of interactive live Video Feeds
      4. Multicam
      5. Moderation?
         1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
      6. Streamers
         1. Interacting with a built track
         2. Building a track with feedback
      7. //Gibt keinen Programmablauf, nonlinear, come in and have fun…oder eben Phasen
      8. *ANforderungsanalyse? (Ziele optimierung visual quality vs latency/performance) (vielleicht später)*
4. State of the Art
   1. Grundlagen? (Technische basics, api, twitch how does it communicate)
   2. Conflicts: Telegames (I have: no fail condition, lower laterncy, social chaos is feature)
   3. Potential (Motivation an konkreten beispielen) (konkrete beispiele vlt hier und mein Wunsch/pipe dream oben in konzeption)
5. Hardware Implementation
   1. Camera Setup
   2. Lighting and Set
   3. Marble Tracks
   4. *Expansion*: Configuring a custom Daughterboard for a tightly integrated Set
6. OR Technical Implementation
   1. Physical Set
   2. Core Software Framework (Unity + 2 PyScripts)
   3. Unity
      1. Video Input
      2. Camera Array (Video-Mixer?)
      3. More interactivity: Sfx
   4. Light-Up Switches
7. Evaluation
8. Fazit & Ausblick
   1. *Ausblick*: Virtual Particle Systems in 3D Space
   2. *Ausblick2*: Clickmaps
   3. *Ausblick3*: Development of an independent Webapp

Motivation:

Livestreams have been getting more and more popular over the last ten years.

The internet has made way for probably the most abrupt push towards democratization of media creation. At times compared with the invention of the printing press this modern technology has made it so easy for people to spread ideas and opinions, truly anyone could participate. While this freedom does not come without risks, it has enabled even such complex media as videos to become user-generated easily nowadays. While clips shared online are already quite powerful, traditional offline media has kept monopoly over one particular aspect – the real time nature of a live-TV broadcast has not been reached by online video.

So simply by being created in realtime, Livestreams can be much more engaging and interactive as user generated content. Streamers can directly respond to comments the second posted [to build a deeper connection to their fanbase], and react to complaints the moment they are raised. Therefore Streamers can not only change their style of commentary on the spot but even go as far as to adapt the content itself to the stimmungsbild of viewers.

This novel kind of interactivity is interesting to me. I would like to build upon this interactivity and experiment how to further expand it in hopes of giving streamers a bigger toolkit to rely on in the future.  
On the other hand watching livestreams can feel a little arduous/less curated or engaging at times, since it captures downtime that would get cut out from long or short form video content (such as YouTube-videos, gameplay highlight reels, Shorts or TikToks etc.). By giving viewers/chatters something to play around with I hope to bridge said downtime more effectively.  
Furthermore do I wish to even improve the highs during a stream if I give viewers a way to influence the content directly.  
# Since Livestreams can feel a

On one hand they are the logical next step for user generated content without needing to be produced by a team of professionals

On the other hand they replace conventions – or at least their tedious convention part. (Example E3)

Livestreams are filling an interesting hole.

Goal:

It is not about producing the prettiest/best *(no shipping-ready)* possible product but exploring various different Options, strategies and tools – and evaluating those. We are researching possibilities for improving interactivity and engagement in the respective fields we’re going to touch upon or dive into. With the goal of giving future projects a wider variety of vantage points and what to expect, giving inspiration and a different look on livestreams.

In terms of the 4D-creative process (further discussed in chapter XY) I plan to leave off after the completion of stage 3, Develop. While I will give insights into results and learnings from user testing and development the prototype will not be reduced down to a deliverable complete product but rather released as a tool inviting interested creatives to experiment with its features.

3.0/3.1

Before arriving at a streaming concept there were other prototypes defined and pitched to Ravensburger. The following chapter outlines the creative process leading up to the final pitch.  
To that end I took a look at different creative tools and processes, with the goal of finding ideas that could build upon the concepts and mechanics that are at work in Gravitrax and give more value to an already working system.

To gain an understanding for the toy and brainstorm for expansion possibilities I used a model popular in Ludology to identify the main motivators for interacting with Gravitrax. This model does not only explain what type of players engage with Gravitrax but could show, which extensions can Gravitrax benefit from. Even though it is originally derived from video games, it can give some insight into boardgames and toys as well.