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
        i. 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
        i. Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)
    5. Streaming Idea: Interactive Livestream
        i. (Basic Concept)
        ii. Inspiration and History of interactive live Video Feeds
        iii. Multicam
        iv. Moderation?
            1. A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…
        v. 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
        i. Video Input
        ii. Camera Array (Video-Mixer?)
        iii. More interactivity: Sfx
        iv.
    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?
    2.

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
        - *i. 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
        - *i. Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
    5. Streaming Idea: Interactive Livestream
        - *i. (Basic Concept)*
        - *ii.*
        - iii. Inspiration and History of interactive live Video Feeds
        - iv. Multicam
        - v. Moderation?
            1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
        - vi. 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
    i. Video Input
    ii. Camera Array (Video-Mixer?)
    iii. More interactivity: Sfx
    iv.
4. Light-Up Switches
5. *Ausblick*: Virtual Particle Systems in 3D Space
6. *Ausblick2*: Clickmaps
7. *Ausblick3*: Development of an independent Webapp

Manu ver.

7. *Abstract*
    1. *Tags and Buzzwords! (unity3d, 3dPrint, CAD-Software)*
    2. *Ergebnisse (coole latenz…*
8. Introduction
    1. Motivation *(kurz halten)*
        - i. *WICHTIGSTER:Streams immer cooler, interaktive noch cooler! -> immer mehr Firmen (z.B. Product launches)*
        - ii. *(Digitale testversionen easy) Physische produkte testen und ausprobieren*
        - iii. *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)*
9. Conception: Finding Ideas and creative processes *[taking a step back here]*
    1. {Extraneous Circumstances and }initial request
    2. Creative Process Paradigm: 4D-Phases
        - i. *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
        - i. *Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
    5. Streaming Idea: Interactive Livestream
        - i. *(Basic Concept)*
        - ii. UI and 3 Wireframes
        - iii. Inspiration and History of interactive live Video Feeds
        - iv. Multicam
        - v. Moderation?
            1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
        - vi. Streamers
            1. Interacting with a built track
            2. Building a track with feedback
        - vii. //Gibt keinen Programmablauf, nonlinear, come in and have fun…oder eben Phasen

        viii. *ANforderungsanalyse? (Ziele optimierung visual quality vs latency/performance) (vielleicht später)*

10. 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)
11. Hardware Implementation
    1. Camera Setup
    2. Lighting and Set
    3. Marble Tracks
    4. *Expansion*: Configuring a custom Daughterboard for a tightly integrated Set
12. OR Technical Implementation
    1. Physical Set
    2. Core Software Framework (Unity + 2 PyScripts)
    3. Unity
        i. Video Input
        ii. Camera Array (Video-Mixer?)
        iii. More interactivity: Sfx
        iv.
    4. Light-Up Switches
13. Evaluation
14. 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)*
        - i. *WICHTIGSTER:Streams immer cooler, interaktive noch cooler! -> immer mehr Firmen (z.B. Product launches)*
        - ii. *(Digitale testversionen easy) Physische produkte testen und ausprobieren*
        - iii. *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
        - i. *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
        - i. *Player Types: Found via Identifying Achievers as not targeted audience (sandbox-y, therefore free, but non-linear)*
    5. Streaming Idea: Interactive Livestream
        - i. *(Basic Concept)*
        - ii. UI and 3 Wireframes
        - iii. Inspiration and History of interactive live Video Feeds
        - iv. Multicam
        - v. Moderation?
            1. *A complexity I excluded, for focusing on all options for integrating technology – so you can sample them all…*
        - vi. Streamers
            1. Interacting with a built track
            2. Building a track with feedback
        - vii. //Gibt keinen Programmablauf, nonlinear, come in and have fun…oder eben Phasen

        viii. *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. Software Implementation and Custom Solutions
    1. //Physical Set
    2. Core Software Framework (Unity + 2 PyScripts)
    3. Unity
        - i. Video Input
        - ii. Camera Array (Video-Mixer?)
        - iii. Supporting features: Sfx, Animations & more
        - iv.
    4. Light-Up Switches
7. Evaluation
8. Conclusion & Outlook
    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.

We will look at the brainstorming phase through the lens of one of the creative processes used, the 4D P. This common process dictates, that the development of a product consists of 4 phases, all beginning with a "D".
In the 1st phase, "Discover", Designers are encouraged to think as creatively as they can. Ideas get collected no matter their ease of implementation or relevance on the market, to encourage producing more out-of-the-box concepts. This is what I'll be looking at in this chapter.

Ravensburger Quest here.

This request by Ravensburger gives this project another dimension, requiring me to conceptualize the project such that it adds considerable value to the Gravitrax-ecosytem, in one way or another.
Ehile this requirement did step more and more in the background as the project progressed, the concept needed to be planned in a way that both improves

upon the common live streaming experience and simultaneously enhances Gravitrax as an interactive experience.

So the second dimension of “Motivation” consists…
2. It’s always difficult to give potential buyers a taste of a physical toy. While it is easy to give out Demo versions of

Most of the digital entertainment industry likes to give out samples/testers of their products. Movies publish trailers, podcasts release entire episodes or shows from their catalogue on demand, and even video-games offer demo versions of their final product before launch for download – free of charge. All of the examples give consumers a way to experience in much the same way as the full product.
The closest things to testers Board games and toys offer are promotional material like photographs, illustrations, renders and videos. All of those play in a different medium than the full product, crucially missing any way of interaction. In some rare cases demo products are displayed in some of the bigger branches of the bigger national retailers, but even so shelf space is limited, and space for demo installations are at a premium. As a result, far from every interested buyer has the chance to test out a toy for themselves, regardless of who they are shopping for.

3. Motivation – Digital Boardgames have had a hard time arriving on the market. Ravensburger themselves have launched two of these in the past, archieving mixed success.
After asking boardgame-enthusiasts, they came to a consensus: Enthusiasts themselves say they don’t feel appealed by digital boardgames, since they identify with their hobby through its analogue and haptic nature, while fearing the games becoming unplayable in the future via losing compatibility, like companion apps “explainiation” with future OSes, or the apps vanishing from the internet and app stores entirely. This situation clashes with most board game enthusiasts identifying themselves as collectors.
At the same time digital games seem to appear more complex and are thus less likely to be recommended to, or picked up by, beginners.
Therefore a secondary goal/motivation was to look out for some ways to incorporate technology into the marble tracks, without interfering with the underlying game and gameplay loop.

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.

Hardware: Light up brick

…so I built a frame resembling a Gravitrax' brick, able to house my LEDs. Instead of cutting out perfectly spaced holes for each LED and lining them up one by one, I instead got ahold of some transparent Filament. It's some PLA "EXPLAIN" without any added pigments, resulting in it being being almost perfectly see through.
(Addendum: It has a very high light transmission coefficient of up to 90%, comparable to glass. Furthermore I've found out this figure is really consistent for PLA across the entire spectrum of visible light, meaning it barely distorts colors shining through it. A surprisingly good candidate for my application. - https://www.researchgate.net/figure/UV-and-visible-light-transmission-of-PLA-and-PLA-BHT-films_fig2_232382940)
Ideally I'd want to use some transparent PET-G, which is regarded as one of the best transparent 3D-Printing filaments. With lower virtuosity at its higher printing temperatures air gaps are minimized and print flow rate is increased, resulting in a more consistent lighting, and prettier look.

This lets me use the frame not only as a mount for the LEDs and a face plate to hide the unattractive wiring – but more importantly as a diffusor, by mouting the LEDs directly behind the solid wall of the part.

This feature is very welcome since I used a sub-optimal kind of led for this prototype on purpose. Ideally I wanted to use so-called "Filament Style LEDs", a relatively newer kind of LED, at least in the hobbyist space. These LEDs produce a thin, homogenous line of light, much like the more high-end drop-in replacements for light bulbs, trying to emulate the glow of an oldschool tungsten wire. While the produced lighting effect would have been very pleasant and sleek, I could only find them in a monochrome-colored design, without individually adressible segments, without a driver, at a higher price. This was for me personally too little flexibility for this iteration, so I opted for the highest-density strips of WS2812 RGB-LED-Modules. These have been tried and tested in prototyping, due to their easy 5V operation voltage, individual per-led addressability and relatively low cost. I could find them in thin, 166 Modules per meter strips, a resolution that should prove to be plenty for this application.

After some test prints I've also experimented with larger wall thickness and even a hollow wall, to abuse the properties of refration, the principle visually enlarging text placed behind a glass of water. PLA has an even higher refractive Index (greater 1.45) than water (around 1.33), which makes this effect not negligible. So by hollowing out every wall we have in total four changes in medium, altering the resulting visual noticably.

Top to bottom: 1.6 mm wall strength, solid; 2.4mm wall thickness, solid; 2.4 mm thick wall, hollow, coming out to effectively 2 times 0.4mm thick lines per wall (with 1.6mm of air in between).

The light appears much brighter and clearer head-on on the part with the hollowed out walls. But from an isometric perspective above the light looks suddenly much fainter than its counterparts. When tilting the part along the x-Axis you can even notice a break-point when the refraction changes the appearance of the light abruptly, roughly around 30° from horizontal. Which is sadly no good for a multi cam recording setup, so I went with one of the solid walls.
As an aside: The refractive effect is virtually destroyed on the solid prints, since imperfections, air gaps between every line and layer, and impurities in every single line disrupt any discernible pattern, resulting in a milky impression resembling frosted glass. Light now simply gets diffused and softened as it passes through.

Camera Setup

Not all views have the same requirements.
Some cameras may need to provide an overview of the marble run, giving viewers an outline of the path, while others provide a close-up view of higher density and higher importance segments, often those that require user interaction.

The closeups are harder to realise since a higher focal length is the only solution to avoid a disorienting distortion in its picture, create focus through depth of field and keep the camera body outside of the marble course and the other cameras views. Thus I chose DSLMs, mirrorless cameras for their strengths in video recording (as opposed to DSLRs, designed to excell at taking stills) and naturally their mount for interchangeable lenses.

Using Mirrorless Cameras for every angle would undoubtably be the best solution. But since funds for this project are not unlimited, enacting this wish would overstep the budget – in multiple ways. I would have to buy/borrow multiple Cameras with additional lenses, that is self-explanatory. But additionally these mirrorless cameras are - for all intents and purposes –

designed to record video and save it onto a memory card. Not for directly passing that video feed on – in real time. That is why their only direct video output is an HDMI port meant to connect to a monitor, wether mounted onto the camera or somewhere on set, not to a computer for streaming into the internet.

Because computers cannot natively handle an incoming video stream, that is meant for direct display, we need a capture device. There are different form factors with differing? interconnects on the market, but the best and most stable experience comes from an internal capture card, mounted directly into a PCI-Express slot on your workstation. That is mainly because the data does not have to take a detour through a (potentially bandwidth-saturated) USB-Chipset otherwise, and manufacturers can allocate and use as many PCIE-Lanes as they require for their implementation.

Therefore the simple decision to opt for more DSLM-Cameras requires not only extra Video capture cards to be purchased, but also a PC with a case that is big enough, a power supply with enough headroom, a plattform (meaning motherboard-chipset + CPU combination) with enough PCIE lanes to spare (so most likely a workstation or HEDT-class system) and of course enough unoccupied PCIE-slots. This simple change in requirements bumps the price for such a setup tremendously in one go.

As a result of this chain of consequences I limited myself to two DSLMs, one of which I already owned. The other two cameras are simply webcams, their capabilities are plenty for basic wide angle shots and they can be crucially connected via a stable USB 3.2 Gen1 interface (colloquially known under its former name as USB 3.0). Therefore a simple USB hub with USB 3.2 Gen2 (having double the theoretical speeds, bidirectionally) capabilities should be capable of running both of the webcams simultaneously. That leaves us with two capture cards for the two DSLMs. I was able to bring this number down even further by opting for a special kind of capture card. The 2 in 1 capture card "Live Gamer DUO" from AverMedia features 2 HDMI inputs, being able to record 1080p video at 60 frames per second, plenty for our purposes. This is a relatively modern card, with earliest mentions in 2020, enabling me to reduce cost further and open up the possibility for a setup with all DSLMs for later, [previously very difficult without the need for custom equipment] if anyone wishes to translate these findings into a full production set – without the need for any custom made equipment.

On the note of custom equipment -
From researching how to best hook up all of the recording equipment to the streaming pc, here is the way I would deem the best: Graphics card docks were quite popular in the mid- to late 2010s. Hooked up via a “Thunderbolt” cable, they were used hook a laptop up with a discrete, full size GPU. This enabled consumers to harness almost the full Graphics rendering power of a desktop GPU for rendering or gaming workloads when using their laptop stationarily. While they have become less readily available with GPU prices on the rise, nothing prevents us from using docks in a different manner. Instead of occupying the PCIE slot in the GPU-Dock with a graphics card, we can instead put the video capure card in there.

This would allow me to place the dock underneath or next to the set and hook up all of the cameras directly to it. It functions as a kind of “daughterboard” or wiring panel, consolidating all of the cabling to and from the pc into potentially a single long optical thunderbolt cable. This works because there are docks that feature usb ports in addition to their PCIe/GPU slot. Alternatively there exist rare, semi-custom docs (like this one by starTech) with two PCIe slots, tan can be outfitted with one capture card each (equaling up to 4 HDMI inputs) or a combination of one capture card and one USB expansion card (up to 2 HDMI inputs and up to 8 USB 3.0 ports).

The latter option is arguably the best solution I was able to find, giving both flexibility and a way more organized and stable set. This is highly advisable for a full production as running 4 plus high speed connections all the way from a set to a fully outfitted and manned streaming workstation is risky for safety and stability reasons, especially if the main marble run is supposed to be changed, maintained and updated frequently for smooth operation and to keep the contents of the live stream fresh and interesting for viewers.
Sadly I did not relize this option, since purchasing a dock, usb card and retrofitting my streaming station with a thunderbolt host card would have again exceeded budget. So this will stand as a suggestion for expansion.

6.2

Now, instead of using an approach based on solely using a software like OBS or Xsplit to decode the video signal and directly putting it out to a live broadcast, as it is common (as established in “state of the art”), I have built my own middle-ware.

This set of tools mainly [consists of?] a central application built with unity, a set of python scripts controlling peripherals and gathering data, and still using OBS. The main difference is that OBS is only used for encoding the live stream and setting up the stream configuration for Twitch. It is not used for managing scenes and layouting viewports. This task is taken over by the Unity application, that can resize and swap viewports as requested. It reads the camera feeds directly without intermediates and arranges them as textures on quads in Unitys virtual world(???). Sadly there is currently no way of setting exposure and focus natively with unitys [package name for cam capture] for the webcams as obs can. This makes the dslms more advantageous from this point of view, since their output images are configured on the camera bodies physically.

This stiteched image is directly recorded by obs as a screen capure… That has not only the advantage that we do not have to automate switching our camera views within OBS, but this setup has gained the ability to display any overlays or animations – really anything I desire or deem useful. With the full computational power and flexibility of Unity at my disposal I plan to turn this Scene Manager into a smart Interface between hardware layer (marble tracks) and the live stream, that can react and adjust to the viewers and potentially the state of the connected marble tracks. //
With its combined computational power and flexibility Unity promised to turn this Scene Manager into a smart Interface between hardware layer (marble tracks) and the live stream, that can react and adjust to the viewers and potentially the state of the connected marble tracks.