Zoom it like I walk it

Eine implizit interaktive Bildergalerie

| Matthias Pawlitzek  # 2315565  Hamburg, Germany  [matthias.pawlitzek@haw-hamburg.de](mailto:christian.schroeder@haw-hamburg.de)  Technische Implementation | Max Bauer  # 2323129  Hamburg, Germany  [max.bauer@haw-hamburg.de](mailto:max.bauer@haw-hamburg.de)  Design |
| --- | --- |

# ABSTRACT

Zoom it like I walk it ist eine digital erfahrbare Bildergalerie, welche es dem Benutzer ermöglicht, Einblicke in die zerstörerischen Ausmaße menschlicher Aktivitäten in Äthiopien zu erlangen. Die Galerie wird mithilfe eines Beamers auf eine große weiße Wand projiziert. Dabei wird immer nur ein Bild zur gleichen Zeit groß und mittig angezeigt.

Inhalt der Bilder sind Satellitenaufnahmen kleiner, kreisförmiger Kirchen, deren direkte Umfelder nicht wirtschaftlich ausgebeutet werden, was zu wunderschönen grünen Oasen inmitten einer kahlen und zerstörten Umgebung führt. Um sich die Bilder genauer anschauen zu können, muss der Benutzer nichts weiter tun, als vor ebenjener Wand zu stehen und vor-, rück- oder seitwärts zu gehen. Durch diese Bewegungen wird die Perspektive auf das aktuelle Bils dahingehend verändert, dass ein Gefühl beim Benutzer aufkommt, eine reale Galerie zu besuchen. Entfernt sich der Nutzer aus dem vordefinierten Spielbereich, wird ein Infotext eingeblendet, welcher Hintergrundinformationen zur Galerie liefert. Des weiteren ist es möglich, zu einem anderen Bild zu wechseln, indem man mit einer Hand natürlich Streichbewegungen zur Seite ausführt. Dies ermöglicht es dem Benutzer, sich die komplette Galerie anschauen zu können, die insgesamt sechs Bilder umfasst.

# INTRODUCTION

Zoom it like I walk it wurde entwickelt, um Benutzern das Konzept der impliziten Human-Computer-Interaction (*HDI*) demonstrieren zu können.

Man kann zwischen expliziter und impliziter HDI unterscheiden. Explizite Interaktionen erfordern eine bewusste (explizite) Interaktion des Benutzers. Um etwas zu erreichen, weiß der Benutzer, welchen Input er dem System geben muss. Ein gutes Beispiel für eine explizite Interaktion ist die Computermaus als Eingabegerät. Um den Cursor auf einem Bildschirm bewegen zu können, muss der Benutzer die Maus bewegen. Man steuert also bewusst einen Vorgang.

Implizite Interaktion hingegen stellt das genaue Gegenteil dar. Sie findet dann statt, wenn man als Benutzer unbewusst die Steuerung über ein technisches Gerät übernimmt. Als Beispiel hierfür kann man eine elektrische Tür nehmen, die sich öffnet, wenn der Benutzer sich ihr nähert. Der Benutzer muss also nicht mehr machen als physisch sich der Türe zu nähern. Die Tür als Hindernis wird ohne weitere Interaktion überwunden.

Was das Projekt betrifft, findet implizite Interaktion dann statt, wenn der Benutzer sich vor der Wand physisch bewegt. Dies ist eine natürliche Bewegung und keine direkte Interaktion mit einem Computersystem. Aufgrund eines Näherungssensors, der unterhalb der projizierten Fläche an der Wand angebracht wurde, kann man die Bewegungen des Benutzers erfassen und die Daten auswerten (siehe Abschnitt **Transferfunktion**). Eine weitere implizite Interaktion wird über Rotation ausgelöst. Ab einer gewissen Distanz zur Wand wird die Kopfrotation des Benutzers erfasst, welche wiederum die Perspektive auf das Bild auf einer Rotationsachse leicht verändert. In anderen Worten: Wenn man den Kopf nach oben neigt, neigt sich auch die Perspektive auf das aktuelle Bild nach oben und andersherum.

Eine explizite Interaktion musste dennoch eingebunden werden, um den Bilderwechsel zu ermöglichen. Die obengenannte Streichbewegung mit der Hand ist eindeutig den expliziten Interaktionen zuzuordnen, da man bewusst etwas steuern möchte.

## Interface Technologie & Setup of devices

Für das Projekt wird folgendes Setup benötigt:

An Hardware braucht man eine große, weiße Wand, einen Beamer, einen Computer sowie einen Microsoft Kinect Sensor.

Der Beamer sollte so eingestellt sein, dass er auf die Wand projizieren kann. Der Kinect Sensor wird unterhalb der projizierten Fläche angebracht, bestenfalls auf Augenhöhe des Benutzers. Der Computer wird mit dem Sensor und dem Beamer verbunden und das Programm gestartet. Daraufhin fängt der Sensor an, das Rauminnere nach Personen abzuscannen. Dabei sollte der verfügbare Platz mindestens 5m x 2.5m betragen. Sobald eine Person erkannt (*getrackt*) wurde, wird die Entfernung ihres Kopfes zum Sensor bestimmt. Nun kann der Benutzer anfangen, die Bildergalerie zu untersuchen.

# CONCEPT

Always having had the vernissage’s motto on our minds during development we wanted to create something new; something for what the new technology of Mixed Reality is perfect for. As we have already seen some examples of last semester’s projects, we wanted to go further and implement more than just one interaction by placing a physical marker into the digital world. That’s why we first invented the idea of a cuboid, which the player had to walk around in order to follow a game character.

We thought about building a cuboid with a roughly 30cmx30cm base area, reaching about 1 meter into the air. Putting the game design into a skyscraper-runner context we already had quite a nice but simple concept. Our game idea included almost the same gameplay elements as players can find them in the final version now. Setting the skyscraper’s roof on fire in the digital world would result in a game in which a game character had to run down the stairwell in order to survive. Same as in the final version we wanted to map the stairwell on the four long sides of the physical skyscraper. The player’s objective would have been to place platforms with a physical marker in his hands at corresponding spots so the game character neither falls down the entire skyscraper nor gets burned alive if the player does not react in time.

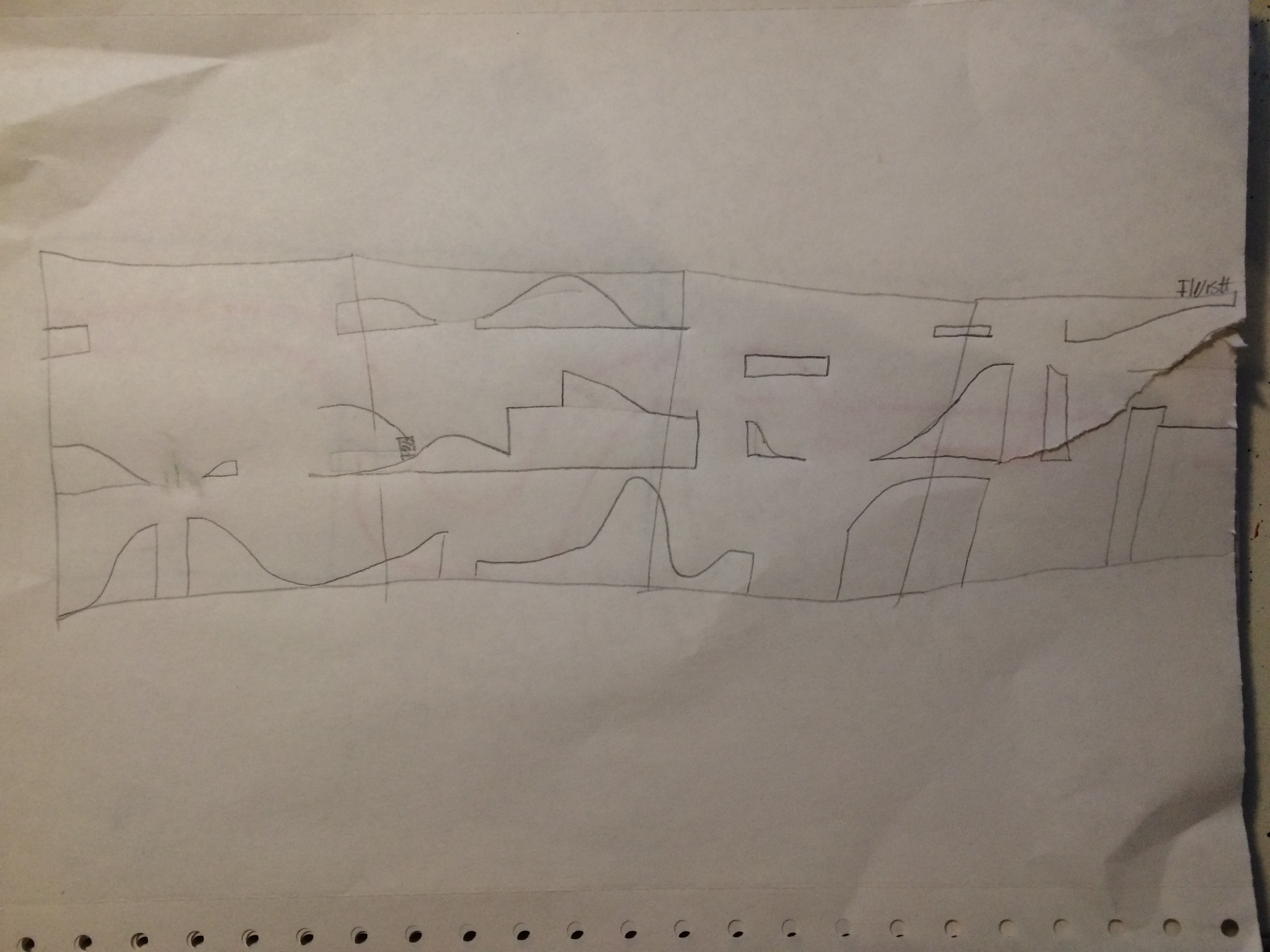
Pretty far in development already we realized that this idea had a single but grave problem. Our concept of players constantly walking around the skyscraper would result in heavy Motion Sickness. The first time we tried a Samsung Gear VR (the hardware that was going to be used for the vernissage) we could not even make three steps forward without feeling dizzy. Not to mention how players might feel if they had to circle around an object for about two minutes. Our idea was just ahead of technique at that time so the idea got changed.

Let the cuboid become a cube with roughly 30cm height, length and width and let it be rotatable on its z-axis having the players sitting on a chair. In fact, players shall only look straight forward and use both hands for interaction - their stronger hand for putting the platform in correct position and the remaining one for turning the cube on its axis. Since our cube did not really resemble the shape of a skyscraper anymore our game idea also had to be changed. After several days of brainstorming the final idea came up. A cave system shall be rendered on the cube’s four sides. Holes in the level ground will become obstacles that players must overcome while rising water will take care of needed tense.

Referring to the vernissage’s motto we were wondering how he interaction actually might look like. Our concept already included a cool feature by rotating the cube at its z-axis but how were players interact with the level itself? Were they just putting platforms at several spots in order to reach the exit? That idea seemed easy but still a little too *passionless*. Not satisfied we agreed to make more out of this basic mechanic by integrating obstacles where the stone marker had to be placed but also moved. There are various obstacles in the final game that are horizontally or vertically too big for the stone marker. In order to master those obstacles the player has to let the game character walk onto the stone platform and move it with the game character’s speed to the end of the gap. Given that the game became simple, varying but also challenging.

Having set the context to a cave game, the next question was how our level would look like. We chose between two ideas. The first one was about creating a maze with many dead ends and a trial and error game principle. The problem was that we would have had to implement a third game mechanic in order to make the player turn around. Since both hands were already occupied we could not find an elegant way to implement it. Consequently we stuck to our second idea that provided a game with a railway based gameplay meaning the game character is walking constantly from left to right with a certain speed. Because of rising water it was necessary to put the level end somewhere at the top of the level since the game character was not supposed to swim. In order to reach the exit we had to design the level in such way that the game character is not only constantly moving from left to right but also gaining height. Further the game character had to respawn at the corresponding level layer in the first level segment when touching the level right border. That is why we came up with the idea to implement three layers – all connected with each other – leading to the exit. At the end of the third layer shall be the level end. In total, the game character was about to circle the cube’s Z-axis three times before reaching the level end.

Our last step in concept stage was about what the game character was going to look like. Was it going to be a simple stickman figure? No, that would just not fit into the main idea. The game character had to fit into a cave based game and therefore we wanted to implement an ancient looking beast. At first we thought about a yeti but that idea got replaced soon by a fictional character that was going to be something between a bear and a yeti called “Troglo”. We named him like this because ancient cave inhabitants are called troglodytes and our created short form fits to a helpless beast trapped in a cave. We decided to put a fictional being into the game rather than a human because for us it seemed more motivating to save a lovely looking unknown species.



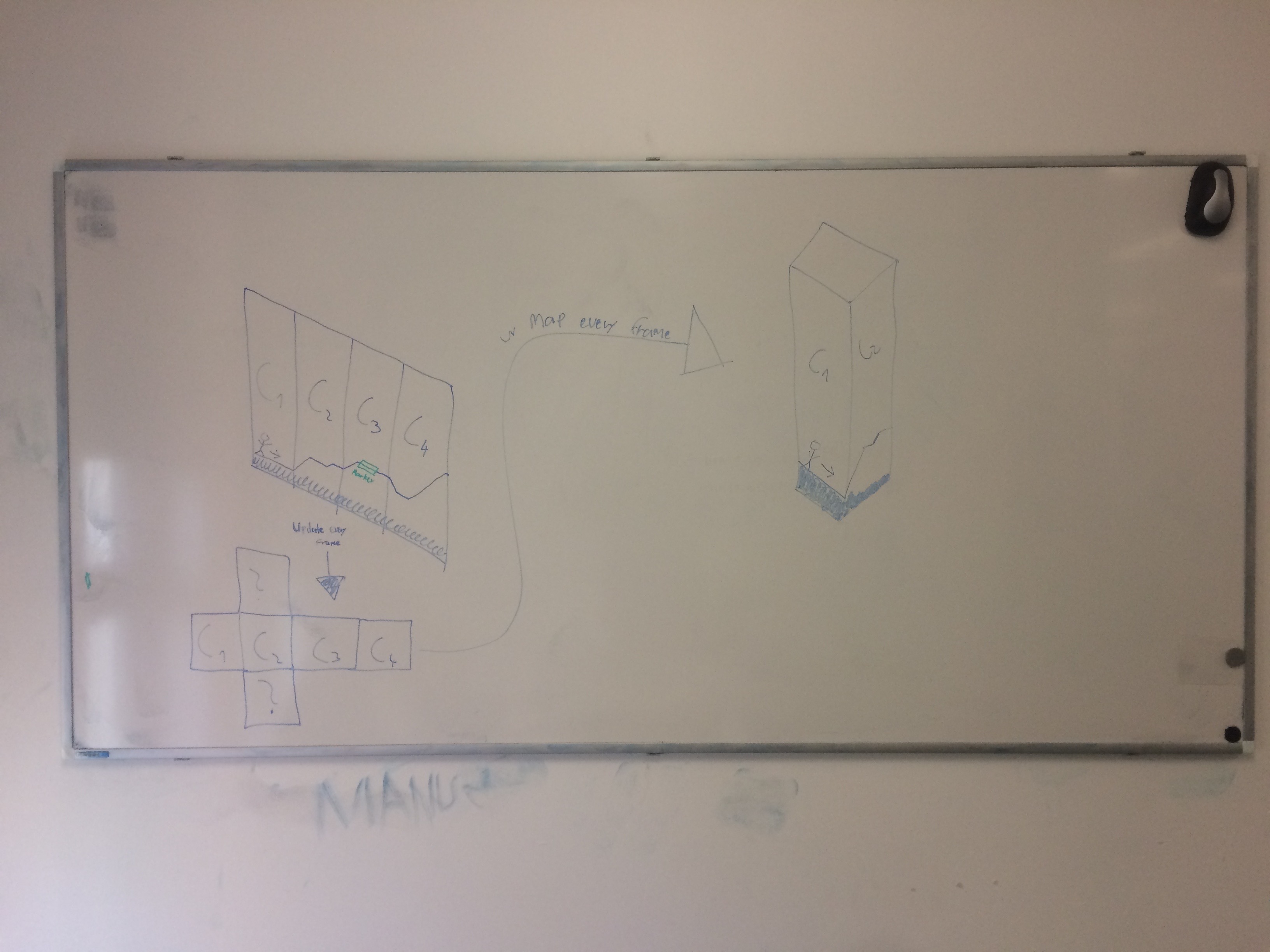
### First level concept

# Technical implementation

## Rendering the level

The project consists of two markers. One being the cube itself with the game level printed on and the second one being the stone marker that has to be put onto the cube in order for platforms to become visible. Before that we had to deal with another challenge: How were we going to render a 2D side scroller game onto the four sides of the cube? Since the images that had to be rendered were dynamic we had to update them every frame. But how could we achieve that?

At first we thought about using UV Mapping. Dividing the game level into 4 same sized parts, filming each of these parts with a camera, stripping down the cube and giving each side a new image very second using UV Mapping would have worked but was rather difficult to implement since Unity does not provide intern UV Mapping. It had to be implemented manually via code.



*UV Mapping idea*

After several days of research on the internet we stumbled across a YouTube video that explained how a security camera in a game could be realized in Unity. The purpose itself was completely different but the technique used in that video was exactly what we were looking for (see references).

It works as follows: For each side of the cube we created a new material and a new render texture. The material’s shader became the render texture that in turn got the image captured by its suitable camera every frame. Having this we already had the material updated every frame. In order to make the material become visible on the cube once Vuforia recognizes it we just added the material to the suitable side of the cube to its Mesh Renderer.

Using a technique like this saved plenty of time while resulting in same effect.

## Vuforia Integration

Troubles with Vuforia occurred twice during development. The first one was about the rendering of the stone marker. Even though it was recognized by Vuforia it sometimes disappeared when rotated. We fixed that by locking the marker’s z-axis so it was always rendered with the desired rotation. Further we wanted the stone marker to be rendered only when being close to the cube. In order to achieve that we attached a box collider to each cube side as well as to the stone marker so if they collided the marker should only then become visible.

More mysterious than the first problem was an error which we ran into several times when the game suddenly crashed without leaving any message in the debug section. We were also not able to trigger it; it seemed to appear just randomly. Our approach to the solution was a screenshot that we took just when the game crashed (see below). Obviously something went wrong with the rendering of the cube’s side and it looked like this error only occurred when the stone marker concealed the cube in such way that Vuforia did not recognize the cube’s marker anymore. When recognizing it once again the game crashed. As usual the hardest problem gets solved with an easy solution. For us that meant every our cube’s sides in Unity were attached two meshes; probably by us somewhen earlier by mistake. Perhaps that required too much resources resulting in a crash. By removing one attached mesh the mystery was not completely solved but at least the error did not occur once again.



*Screenshot when crashed (white cube only for development purposes)*

## Interaction

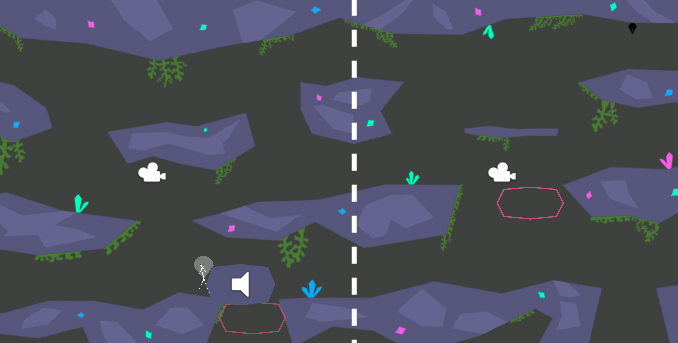
As mentioned above two interactions take place in gameplay. The interaction between stone marker and the digital world was realized by making use of Unity’s Box Colliders while rotating the cube was nothing but a handcraft manner.

## Exhibition summary

We almost only got very positive feedback during exhibition. But let us focus first on the things we could have done better.

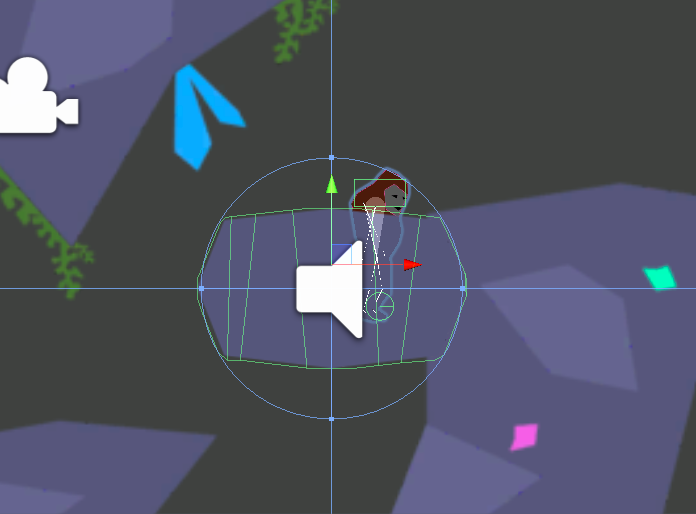
The players had to deal with three major problems in order to accomplish the challenge. One problem we were already expecting but were not able to fix in time was that players often covered the front side of the stone marker with their hand causing Vuforia to lose tracking of the image target. That happened because the physical marker was not visible for the players unless it got tracked by Vuforia. Even though we introduced players into the game by telling them the story and the goal as well as placing the stone marker into their stronger hands in that way they did not conceal the front side of the marker, many players in stress situations suddenly held the marker in a non-Vuforia compliant way. Lost tracking of the image target sometimes resulted in the players’ death and frustration, making them leave the game. This issue could have been solved by attaching the stone marker at one end of a stick and place the other stick’s end into the players’ hand likewise a selfie stick. As mentioned above there was no time left.

Another confusing element was about our in-game tutorial system. Below a screenshot of two sides of the cube is visible. As you can see the tutorials (red polygons) signalize the player to place the stone marker in the gaps. The upper tutorial is loopily moving from the left level platform to the right one, thus disappearing there. When players managed to master the very first gap (bottom mid, left from the dashed white line), then rotating the cube 90 degrees to follow Troglo to the next cube side they often thought the next spot where they have to place the stone marker was the one with the upper tutorial. Unfortunately they did not just follow Troglo’s way and look out for his next danger. Ignoring Troglo still walking on the bottom level ground the players hoped to save Troglo by placing the stone marker where the upper tutorial was. Frankly the next gap that had to be bridged (bottom right) by the stone marker was not given a tutorial by us because we only wanted to display a tutorial when a modified gameplay mechanic was ahead. So eventually the upper tutorial made various deaths occur. A simple solution for that problem is making the upper tutorial only appear once Troglo has entered the second level.



*Screenshot of level, left and right parts being mapped on different sides of cube*

The last and biggest problem our players were facing was about the magnetic haptic feedback players received when placing the stone marker into a suitable spot. This problem only affects the gaps where the stone marker was not big enough to cover the entire gap, forcing the players to move it along the way. The gameplay mechanic itself is not the problem here because of effective tutorials. Having only a few round magnets available we had to install them in the mentioned gaps in such way they unhappily could not form a continuous line. That caused the stone marker to jump from one magnet to the next one at these particular gaps when not held quite strongly. Because Unity’s Colliders are not designed for such jumpy behavior many times Troglo either got stuck in the stone marker’s digital rendered platform image or fell through it completely. Happening to many players (around every second one on their first try) this was the main cause for Troglo’s death.



*Troglo being stuck in the platform after the stone marker was moved too quickly because of two neighboring magnets*

Especially the last referred to problem was responsible for only a few players being actually able to achieve the game’s goal. But to our astonishment almost every single player (even the ones who did not even master the bottom level ground) was amazed by the idea, the concept, the gameplay mechanics, the design, the sound and the way we realized the idea of having the digital world playing with the analogue world. Further what we heard many times was that our game actually was *fun.* It did not stop at being a finished project complying the rules; beyond it was literally a game.

Often players told us they liked every project they experienced before but ours surprised them for being unique, new, simple and fascinating. Apart from that players dying because of buggy gameplay were even more motivated to reach the cave’s exit. The little problems above seemed tiny compared to what the players experienced. We were proud to have finished our project but we did not expect such amazing feedback.



*Photo taken at exhibition. The player places the marker in his right hand into a gap in order to make the only digital visible game character Troglo survive.*

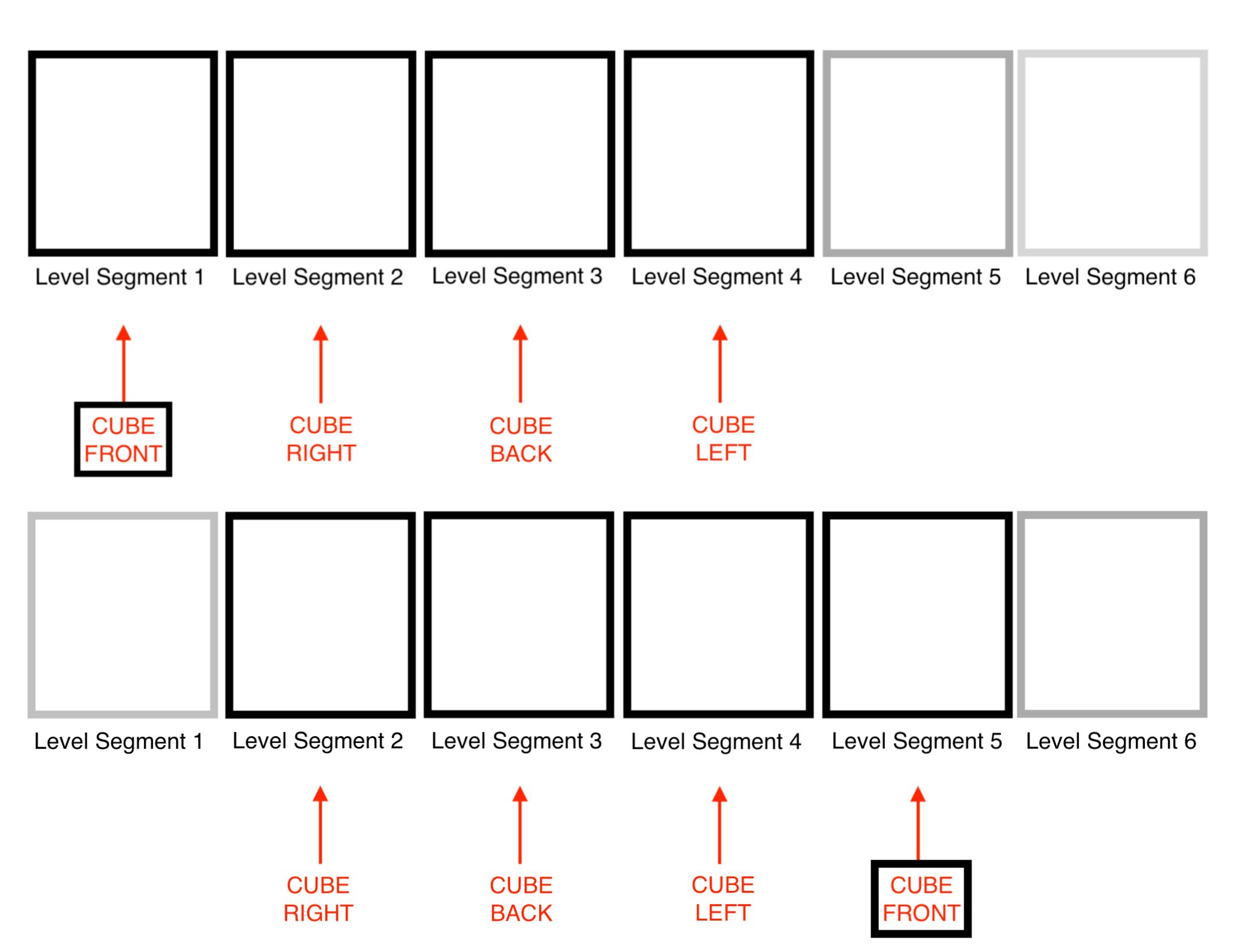
## Future Work

*CAVE* is a functional, fun and ready-to-play game. But someone might argue we just set the cornerstone for a much bigger project. We also think about this in case we want to develop the current state. It is unknown if someday this is going to happen but these are the features we would improve or bring in.

Primarily before introducing new features we will focus on players’ critics and feedback to maximize the current state’s quality. To do so a lot more people have to test and play our game to gain as much opinions as possible. This certainly includes mending the problems mentioned above that occurred during exhibition. Their fixes were already explained. Yet some more problems or even bugs might arise that we were not able to discover since developers likely seem to miss basic defects because they only focus on specific details.

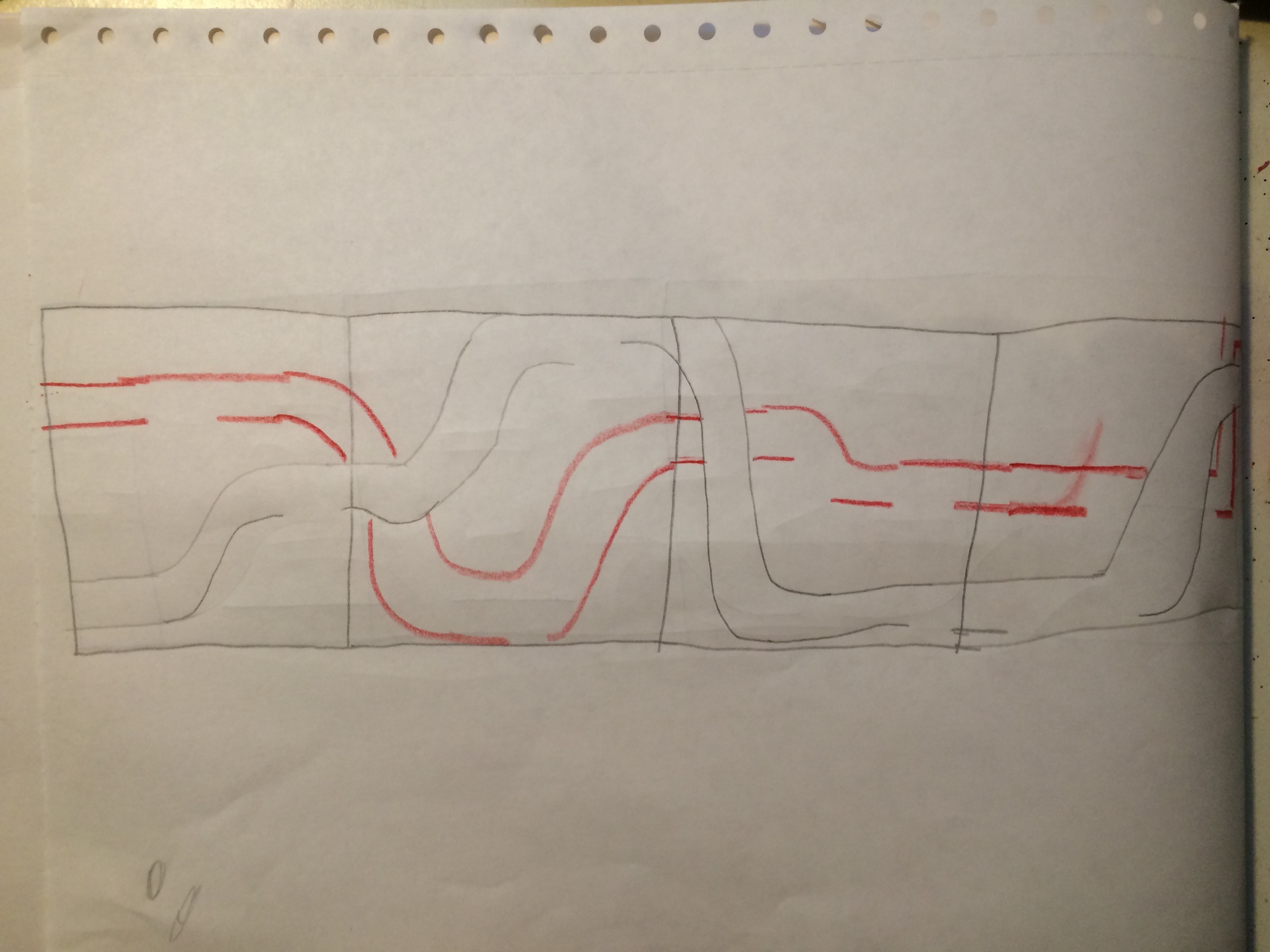
Next a small but quite important thing we would have to do is cleaning up the Unity project since it still includes some unnecessary files as well as it needs to be tidied up. We were not able to handle that in time.

Once these rather inconvenient but significant tasks are done we might finally focus on implementing new features. At first new levels will be designed. We pursue the idea that Troglo starts in Level 1 which is the level of the exhibition. Level 2 will take place at the surface of earth with new graphics, sounds and gameplay elements. Instead of evading rising water Troglo this time has to run from ancient humans hunting him. The interaction changes in such way that players still rotate the cube in order to follow Troglo but the level is much longer. That means more than four level segments as in Level 1 are going to be mapped onto the cube altogether by still having only four segments mapped at the same time. For example when Troglo enters Level 2 in level segment 1 he still starts walking to the right. When walking out of the first level segment he appears in the second one and so on. It differs from Level 1 as soon as Troglo exits level segment four. He does not respawn in level segment one as in the Level 1 but he goes on to segment five that then becomes mapped on the cube’s front side and replaces segment one. So once players will have rotated the cube for a whole 360 degrees the same level segment is not visible anymore. It is a new segment which Troglo moves along. The level design changes from a vertical to a horizontal one. Have a look at the graphic below.



*This graphic shows what happens in Level 2 when Troglo exits level segment 4 and enters level segment 5. The cube’s front side’s mapped-on level segment changes from segment 1 to segment 5. By entering segment 6 the cube’s right side’s mapped-on segment changes from segment 2 to segment 6. This goes on until the last level segment is entered.*

Another idea for a new level design could be a pipe-type level. Instead of extending the level horizontally still only four level segments will be designed. This concept includes playing with depth and perception. Our idea is as follows: We design several pipes that Troglo all has to run through. He starts in the grey pipe’s left end as in the concept drawing below and makes his way through it. Meanwhile the red pipe is put in the background – barely but still visible. Once Troglo exits the grey tube he gets respawned in level segment 1 again (as in Level 1) but this time he has to go through the red pipe; making the grey pipe become barely but still visible. The eye-catching element might be that the red pipe still seems to lay *behind* the grey pipe giving the players a feeling of moving into depth. Alternatively we can also start playing with the players’ perception by interlacing the pipes. Making the grey pipe overlay the red pipe at first meet up and later overlaying the grey pipe with the red one also looks exciting.



*Old level concept that could be realized in future*

Interesting is also the idea of changing the physical object. While the first two Levels take place on a cube, why not mapping the third Level onto a cylinder? A nice level design might be a tower that has to be climbed.

Additional several Vuforia Markers might be implemented and with them consequently new gameplay mechanics. We think about a marker that acts as a weapon to defeat enemies or a marker that changes Troglo’s walking direction (when a maze-like level is going to be designed for instance). Of course the way players rotate the physical object has to be changed in order to be able to hold two markers at the same time. This could be realized by a pedal below the table. By pressing the pedal with their feet players could rotate the cube slowly.

# REFERENCES

1. <https://www.youtube.com/watch?v=NYRmBkPOdo4> by JediNikola: <https://www.youtube.com/channel/UCfRHT3iU-v_ydxJtJtldU-A>

published on March 14, 2015, visited on December 15, 2018

1. <https://www.youtube.com/watch?v=E72yDpAfrgY> by Sword Coast Soundscapes: <https://www.youtube.com/channel/UCUGy8GD5oY4EX9awX4FSqBw>

published on August 16, 2016, visited on January 06, 2019

1. <https://www.youtube.com/watch?v=NWaNW-aYhVY> by FesliyanStudios Background Music: <https://www.youtube.com/channel/UCdEEFrPBkT5WLeRCl2qsEJQ>

published on October 18, 2013, visited on January 06, 2019

1. <https://www.youtube.com/watch?v=6L23P07uIJE> at 00:44 by All Sounds: <https://www.youtube.com/channel/UCFnmEGz7dcAwYflQYfhQccA>

published on July 20, 2018, visited on January 04, 2019

1. <https://www.youtube.com/watch?v=SOnUCNb1pgA> by Angrybullet2: <https://www.youtube.com/channel/UC3H-l1sAH9arRK5ZDcEaBbA>

published on June 1, 2013, visited on December 20, 2018

1. <https://www.youtube.com/watch?v=Ew4wztL_rQM> at 00:10 by Sound Design & Tools: <https://www.youtube.com/channel/UCGy7ppDyzQ4JX9WsSfkN6ow>

published on December 26, 2012, visited on December 20, 2018

1. Platform placing, Platform removing and Troglo’s death scream are taken from Logic Pro X’ loop library by Apple under a royalty-free basis license
2. <https://docs.unity3d.com/ScriptReference/> by Unity Technologies, published and updated on release of Unity 2018.3, visited several times during development
3. <https://de.wikipedia.org/wiki/Troglodyten> under Creative Common License, latest updated on January 19, 2018, visited on January 12, 2019