h_da hochschule darmstadt

member of

eut+

Prevent various children from killing themselves before and during their play in a cozy 3D isometric daycare room with a lot of potential for chaos.

DEADCARE

PROJECT DOCUMENTATION

HAKON WIND
MATRIKELNR:
1117802

INHALT

INTRODUCTION

When I first pitched the project, I had a cozy, but chaotic game in mind in which the player gets distracted quickly.

With this in mind, there are many challenges such as readability of design, game feel and the technical challenge itself.

With this project, I wanted to practice an organized and modular workflow, where all team members profit from. As in the industry, assets are supposed to be reused to maximize the output of the invested workforce.

INDIVIDUAL CONTRIBUTIONS

My personal IPAM focus changed during the project to (**10% Design, TA / G-Dev 80 %, 10% Meth**)

(A+G DESIGN)

I had three roles in this project: I participated in this project as the Vision keeper, as a 3D artist and as a sound designer.

Besides the whiteboard and our Miro board I decided to set up a discord server with different channels to keep spamming and discussions in one place – and miro for "final" decisions only.

VISION KEEPING:

As a vision keeper, I made sure that our final product represents the original idea – or at least the iterated version of it. I didn't know my teammates well before so it was quite a challenge to identify human behavior patterns in a new group.

In my opinion, there are two ways how a vision can be kept: As an autocrat, with a lot of restrictions and enslaved employees, or as a communicative person who encourages people to follow the idea and listens to the feedback and experience of the Team.

I went for the first one.

Just kidding, I wanted to have a productive and motivated team working on "my" pitch. We took the first week to iterate the pitch into a core loop and communicated as much as possible together, so everybody knows what we were heading for. With the whiteboard, miro and discord we never had the problem of discussing a topic twice. If we iterated a design choice we didn't start at the beginning since the design choices itself were already made and at least tested.

I learned that humor itself is not only a surprise reaction by the human brain to compensate the first seconds of confusion – it's also important where the surprise comes from and how it is communicated. As an example – one of my suggestions was a child eating from a dogbowl containing cookies with doglike behavior.

During a discussion, we figured out that the idea was funny, but it degraded the child from a child making questionable decisions to a character doing what the interacted object anticipates.
So, what we instead did was to remove the dogbowl and use a simple plate. With that, we had a better surprise effect.

Besides that, we decided not to make a game with gore and tons of blood. We will not ignore this effect for possible future interactions, but for now, the combination of sound and animation achieved enough to make it work. We will save this for a moment when we need to "go a step further" to make it funny – or surprising.

ARTIST:

I wanted to be the artist in the project to prove myself that I can work long term on a consistent art style and use art to express my vision.

So, I did a lot of research and compared different workflow approaches regarding our needs and usability.

MISCELLANIOUS:

Since I have some experience in Unity, I was able to give some helpful advice during the project. I won`t call any of my teammates out here – but I felt like some members did not accept some hints and tried to do everything from scratch on their own – and after some time went in the direction I

pointed them to. I'm not sure If this unnecessary work was caused by my rusty social skills or personal dislikeness – but I think this is something I can try to change in the upcoming projects.

But what helped was my approach of leading discussions and force the team to document decisions in some way, even if we might iterate on it later on. I had the feeling that we made great progress at the start and saved the time we needed later to create a solid project.

Sound Design:

As a Sound Designer, I couldn't do a lot since only one of 4 Sound designers responded. I created the Sound pitch, researched for cozy, but playful background music and iterated a few soundfiles with our Sound Designer.

Game Design:

I didn't do much after the first planning stage in the Game Design section, since my other roles took too much time.

TECH ART

ART:

Personal challenge, what i wanted to do right "this time", problematic zones, can I do it without concept art (later a lot of shape tests on paper), time to force myself to draw.

Tried sculpting for some things, unnecessary – painter provides everything I need and its easier to adjust instead of prebaked floaters. Only hp are subdivision workflow

PERSPECTIVE:

To achieve the isometric and cozy feel I had to research existing artworks and analyzing them. What I found out is that the mathematical definition of isometric and the translation into a camera is very strict. But when it comes to artworks of cozy and isometric rooms there is a variation in the camera angle settings. I chose 2 of the most interesting and extreme perspectives in the mood board and recreated it in Unity with a simple greyboxing/blockout of the room.
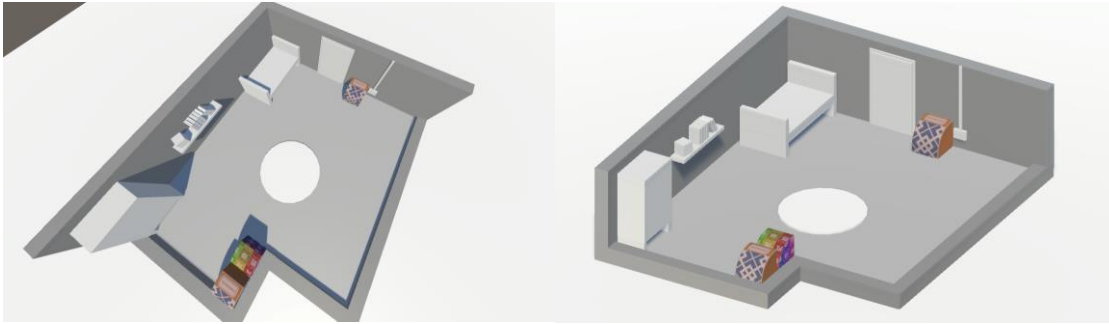
After iterating, I went for this perspective (figure 2):

Since we want the player to observe multiple objects the sight must be clear and needs to be dynamic enough to enjoy the result long term. To achieve coziness, I decided to implement as soon as possible light since the lighting and shape language seem to influence the cozy appeal of artworks the most. Especially warm colors and soft shadows are important.

I added some simple shapes and the default-colored checker map by Blender to track the influence of light on colors.

FIGURE 4 - ITERATING SHAPES (UNITY, SCREENSHOTS, 19.04.2023

Before our animator Zoey Goldschmidt could start the concept art, the shape needs to work. For this I added basic character 3d silhouettes provided by Zoey and discussed it with the team.
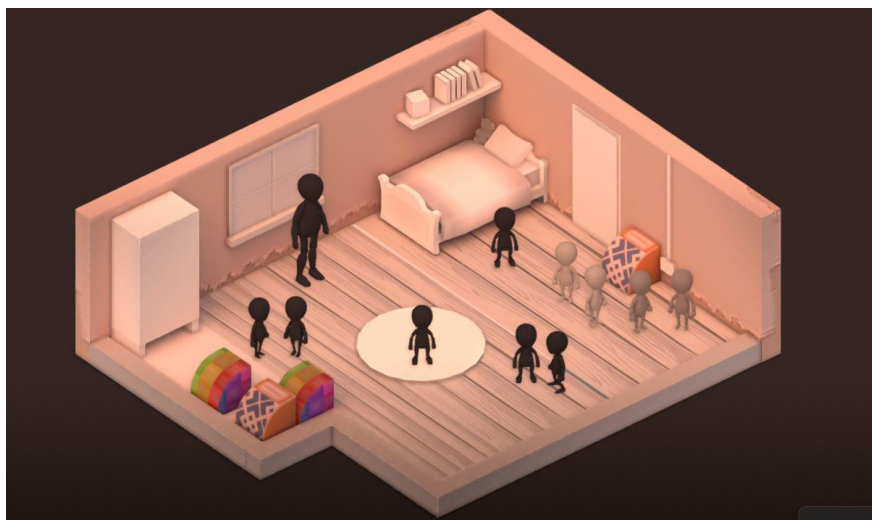

FIGURE 5 - ITERATING SHAPES 2 (UNITY, SCREENSHOTS, 25.04.2023)

Later, I added simple textures and shapes to experiment with the overall look. Every department had its own scene as a Sandbox to avoid merge conflicts or similar problems.

In this Room, I was able to track the consistency of every asset before using them in the actual levels.

FIGURE 6 ARTSCENE

TEXTURING – TRIM SHEETS

Based on these results, I started working on the art bible and texturing the biggest and most used assets. In this case, wood, and the bed.

Since the wood will be used a lot by different types of furniture, it doesn't make sense to create individual textures for every asset while they could just share one material as well. This technique is used in almost every AAA game with big, texture heavy environments.

➔ https://www.youtube.com/watch?v=DipfrjCgYW8&ab_channel=PolygonAcademy
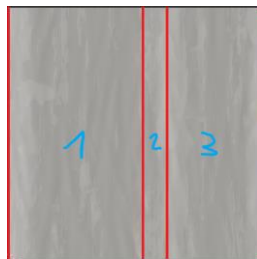
So, I came up with this layout:

FIGURE 7 - TRIM LAYOUT (PAINT 3D, 02.07.2023)

Since every furniture works in 3D like bended planks, I decided to split the area in 2 areas (1,3) and left one (2) for the edges. Later, when I UV new meshes, I only need to lay it above this texture and whenever I update this trim, it will still fit for every existing asset which uses this approach.

This saved me a lot of time since I don`t need to texture every asset individually – even If I had to rework the texture during the project a few times.

Another challenge of the production were the small props. They are too small to validate their own texture as long as I don't want to have dozens of 32px textures flying around.

On the other hand, they are important for the chaotic touch and environmental storytelling.

And – most importantly – they support the balance of the colorpalette.
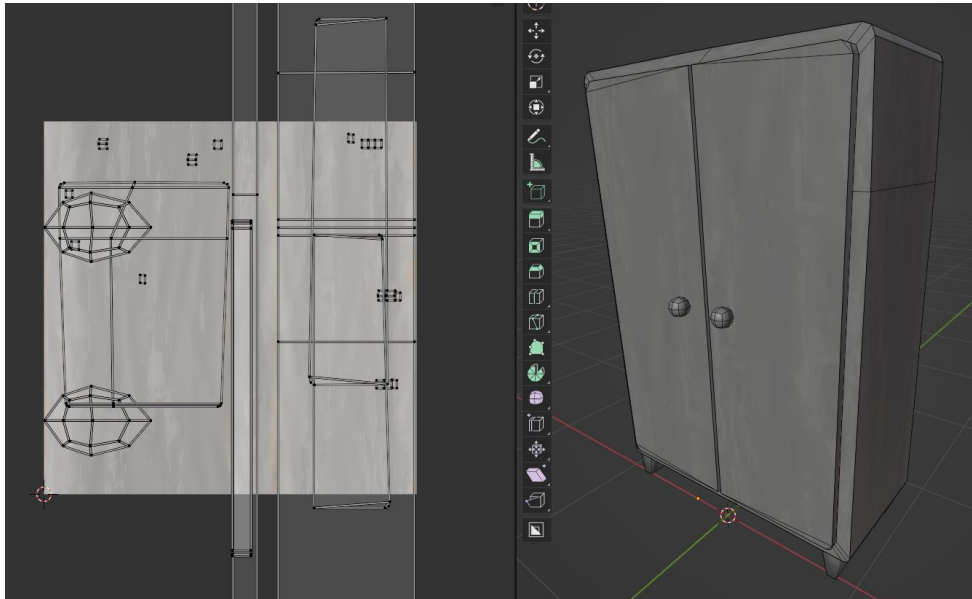
Example:

TEXTURING – GRADIENT MAPPING

After some research I discovered the technique of gradient mapping. Like trim sheets, they idea is to reuse textures for multiple assets but instead of covering a specific material it covers as many colors as possible in one texture. Overall, I have seen two approaches: using several gradients between predefined colors in a planned palette (Figure 7) or using a more general texture. I went for the second approach since I didn't know back then how I want to color the later levels. If I would have used the first approach, I had rework the UV (and texture) for every color which is not covered by the texture yet. But it would create better results since the texture has more gradient information. If I know the final color palette earlier in future projects, will go for the first approach.
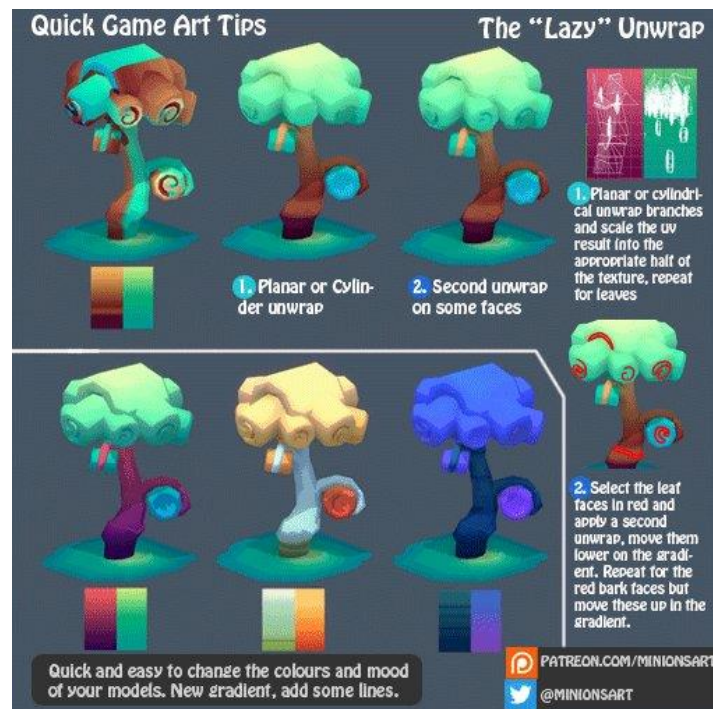
FIGURE 9 - QUICK GAME ART TIPS (MINONSART, TWITTER, 17.04.2017) !!!!!

WORKFLOW

In General, I use Blender for modelling, Substance Painter for texturing and Unity for masking (-> Chapter: Implementation in Unity).

SUBSTANCE PAINTER

In General, I´ve created every texture from scratch. I started inside Substance Painter with Simple Color Layers, which I combined with different noise patterns and generators to create blendable textures.

Since we go for a very stylized art style, it would have been an absolute overkill to work with sculpting techniques (except specific hero assets), so I generated everything from my black and white pattern.
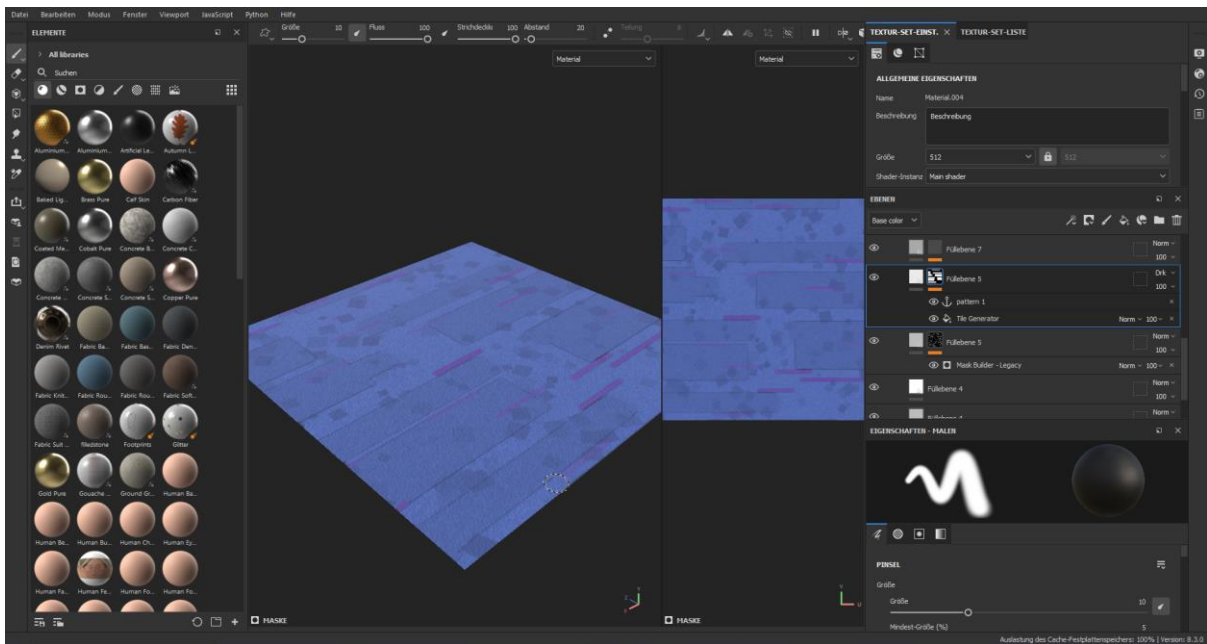
FIGURE 10 SUBSTANCE PAINTER (ROUNDED CARPET MATERIAL)

PROCESS:

During the project, I had an active asset list which covered every modular asset I defined as existential for the vision. This list was updated during the level design process.
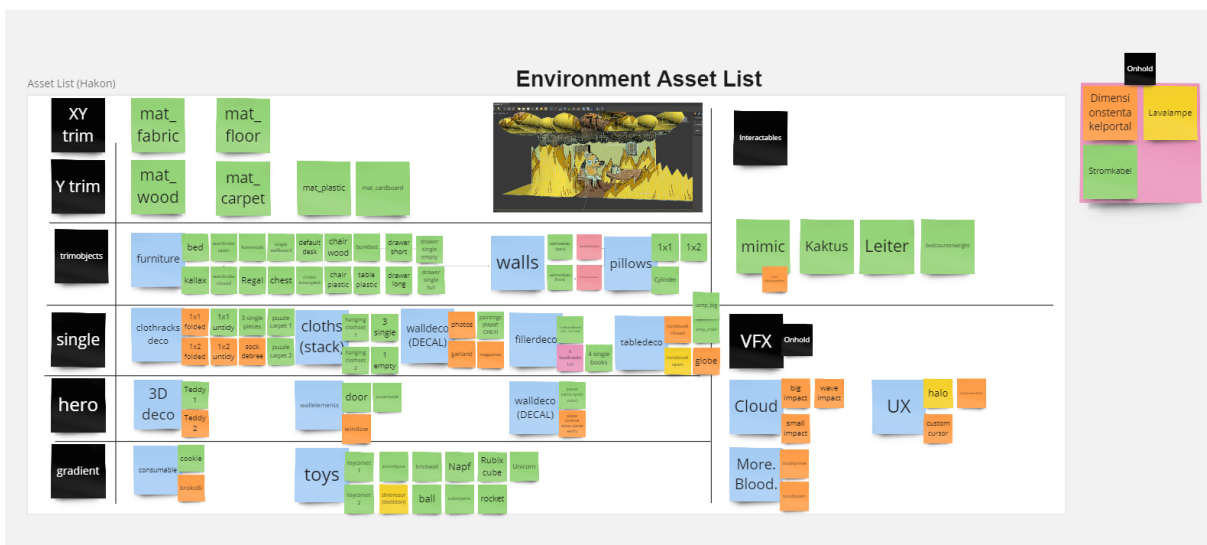

FIGURE 11 - ASSET LIST (MIRO.COM, DEAD CARE BOARD, 2023)

During the production, I had to skip the VFX area temporarily after we noticed that it required additional work on the coding side to implement it. This was mostly a communicative problem, since we preplanned everything on miro and discord, **but I shouldn't rely on personal assumptions and I will make it more clearly in the future.**

Normally, I would have calculated the needed time for the assets, but a lot of the important (and workheavy) tickets were already in work, so I took the freedom to rely on my experience as a 3D artist. Besides that, the remaining assets were mostly small modelling tasks like gradient mapping which can be done in a few minutes. For example, the toy rocket was done in 9 Minutes, from the idea to the final model. In this case, the freedom to do something relaxing after other tickets was worth trading for a 100% accurate schedule.

But as I said, this was more a concession to my personal workflow and I made sure that it had no negative influence on others.
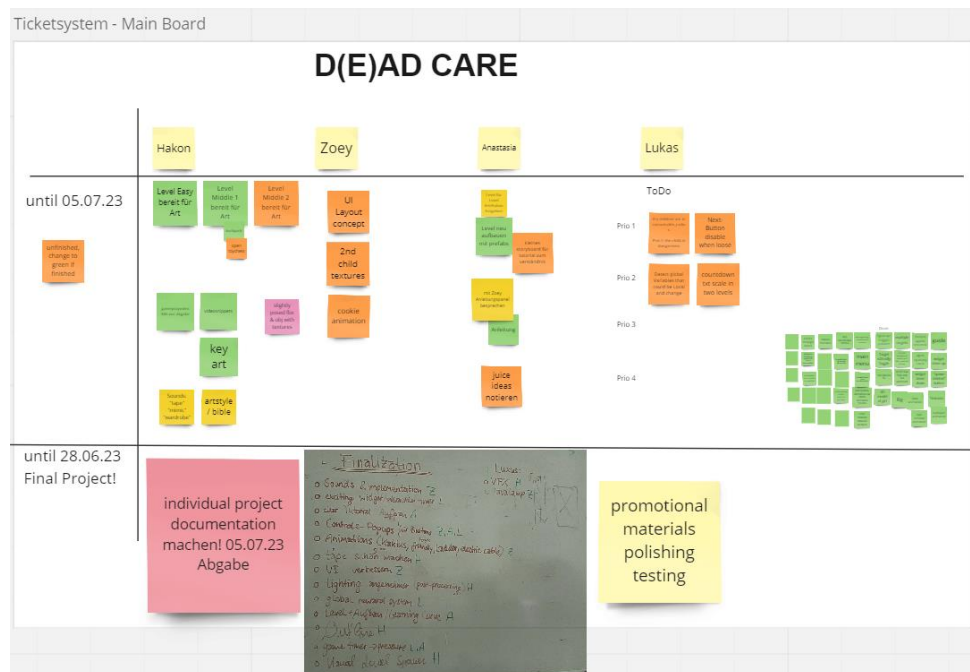

FIGURE 12 WIP TICKETS EXAMPLE (MIRO.COM, DEAD CARE BOARD, 03.07.2023)

During the process we communicated important tasks on our ticket system, which was implemented by our project manager Anastasia Schleicher. With this board we had a good overview of upcoming tasks and deadlines and spontaneous tickets such as "iterating asset x" or "rework texture Y" - which do not fit into the asset list but needed to be done.

This was a great side effect of prepared materials, especially trim textures. Even bigger furniture tickets were done quite quickly and reused textures needed only one iteration to adjust things. I could focus more on art tests inside of unity – like researching decals, getting in touch with shadergraph and several lightmap… anomalies.

IMPLEMENTATION IN UNITY

I decided to use the URP since we don't go for PBR materials and potential fluids, or other exceptions will refer to cartoon. Besides that, the HDRP might offer additional postprocessing and light settings, but I learned in the past that new fancy tools can evolve to timetraps during production. With this in mind, the URP seems to fulfill all of our needs for this project, especially because of updates like decals in 2021 (to which I will come later).
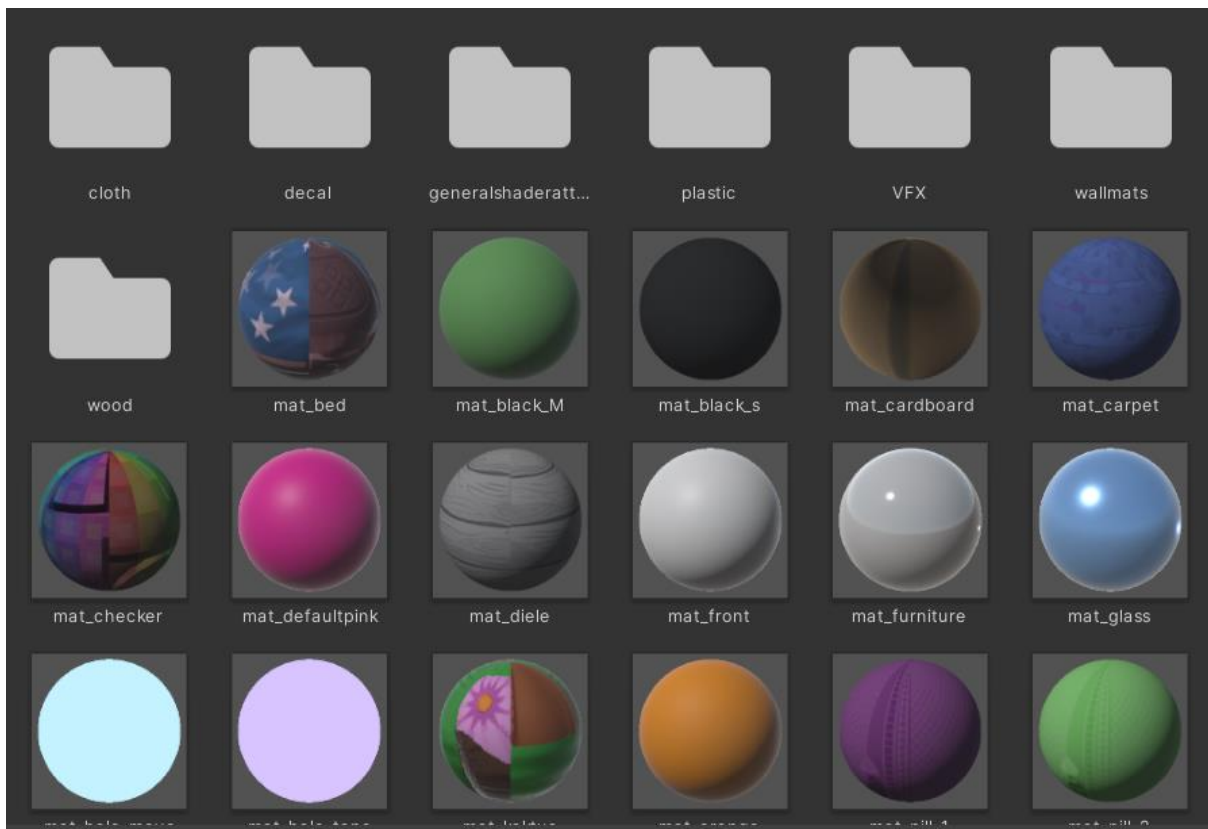
MATERIALS:

FIGURE 13 MATERIAL FOLDER (DEADCARE PROJECT, UNITY)

Originally, I wanted to reduce the amount of materials. That didn't work – at least not in the way I had in mind. I used material instances which shared the same texture but with different colors on the objects. In this way, I still had the advantage of reusing materials but as long as I wanted a new tint I had to create another material. I think in the future I might have to work with a new approach - but this is a topic for the post mortem.
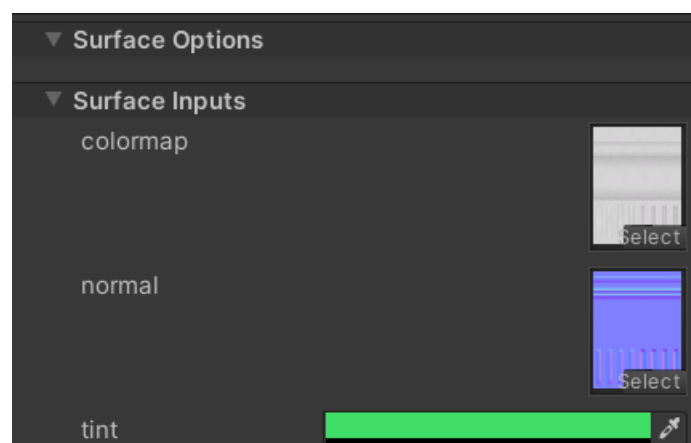

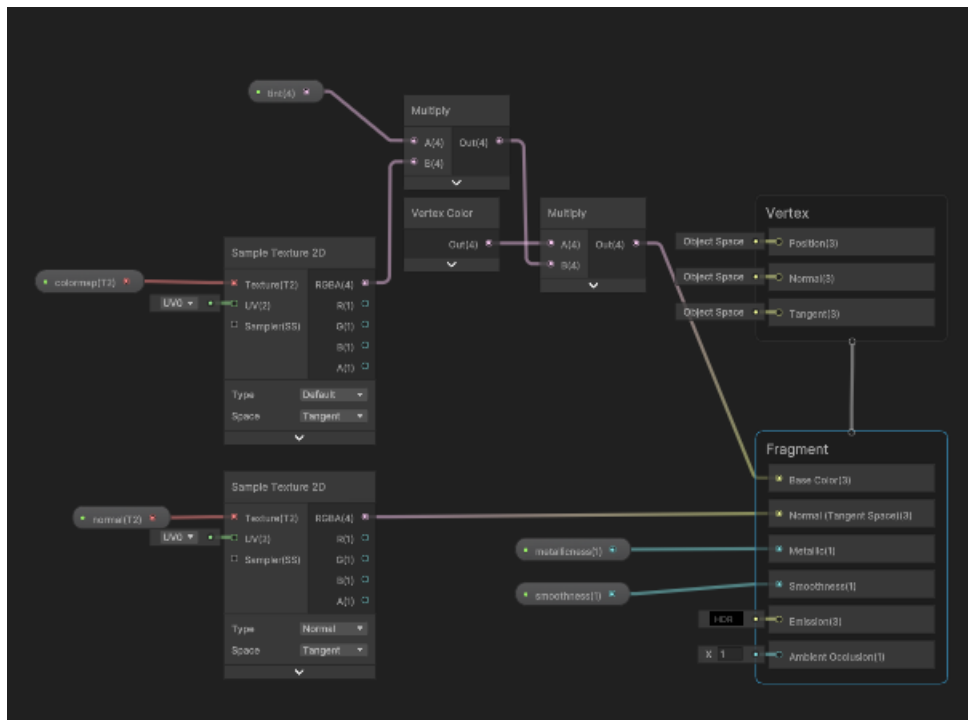FIGURE 14: PLASTIC MATERIAL SETUP EXAMPLE

FIGURE 15: BASEMATERIAL IN SHADERGRAPH

The shadergraph is a visual scripting tool by Unity to create custom materials. It can cover simple blending tasks up to more complex tasks like manipulating vertex position, animating values or procedural behaviors.

The default materials don't offer masking or custom detailed adjustments, which is why I went for it. Therefore, it is a nice introduction to get into shaders which is my is my professional secondary goal as an environment 3d artist in the industry.
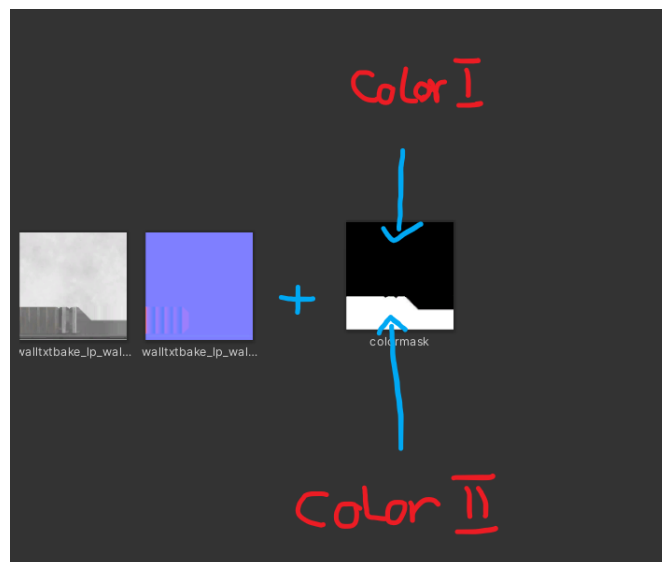

FIGURE 16: BLENDING COLORS WITH MASKS

And since textures can hold more than black and white, I can make use of the other channels as well if I need more distinguishable areas.
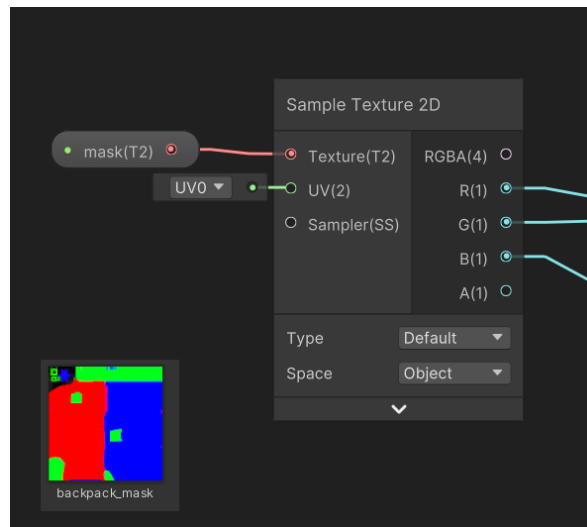
13

FIGURE 17: USING THE COLOR CHANNELS R G B OF THE MASK TEXTURE

As mentioned above, I tried to use as little textures as possible for even more complex materials. One challenge was the bed which I treated as a hero asset with its own texture set first.
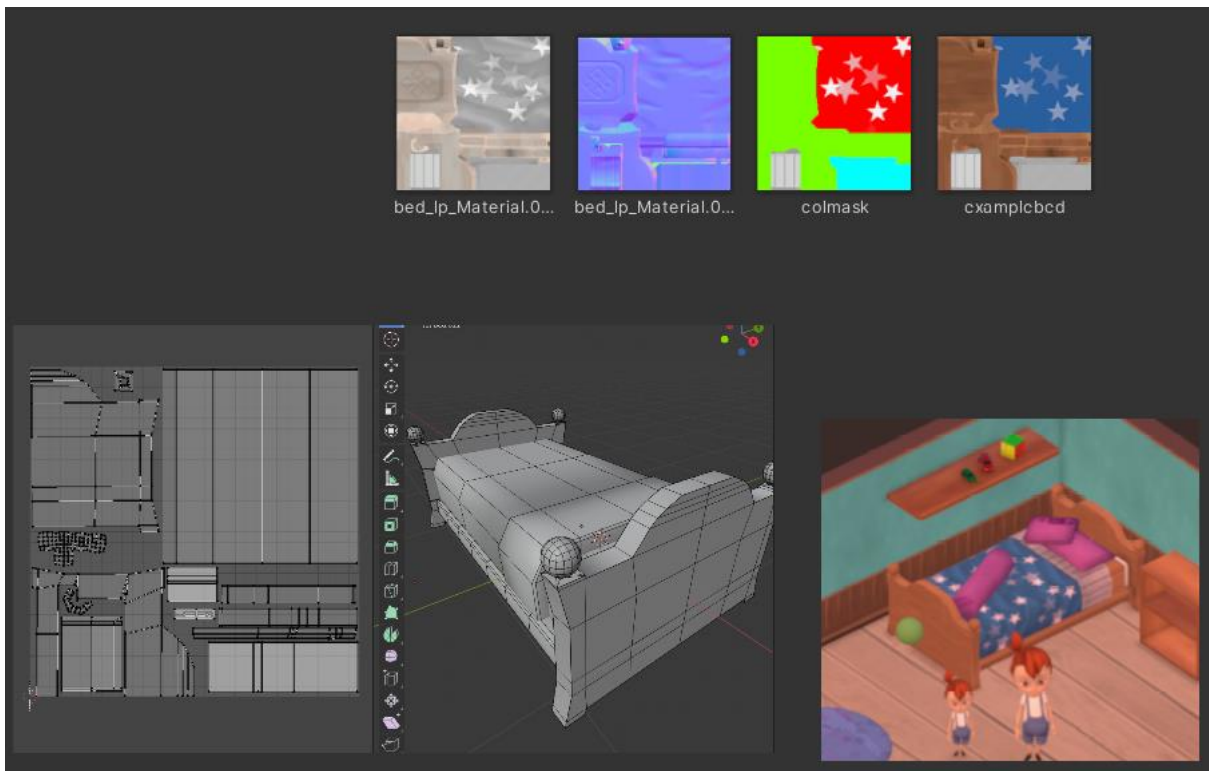

FIGURE 18 FIRST BED ITERATION

First, I got rid of the wood since I already got a trimtexture and other tileable textures. So I used different material slots on the mesh and adjusted the UVs.
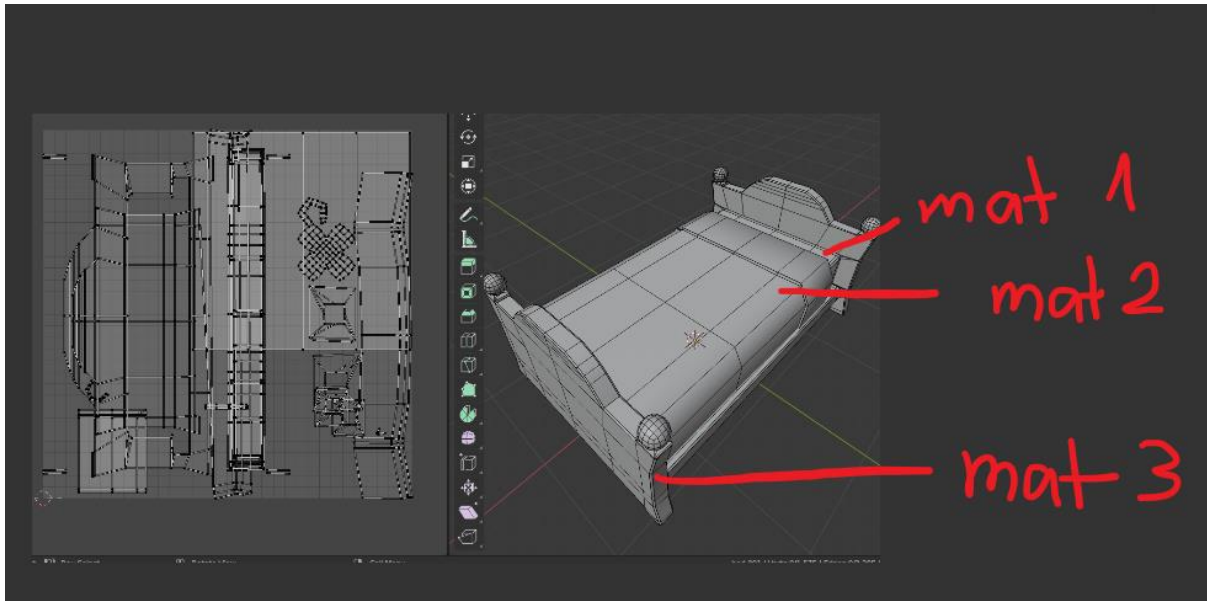
FIGURE 19: PERFORMANT BED

This approach has two downsides. First, it needs more topology for a better control of shading and UVs. Second, I can't use costum details such as bedfolds.

For the second problem, I had the following solution:

With decals, I can project additional information on top of textures. So technically I should be able to project custom patterns and bedfolds on it. Especially because decals support normals.

After some frustrating hours I noticed that this approach won`t work.

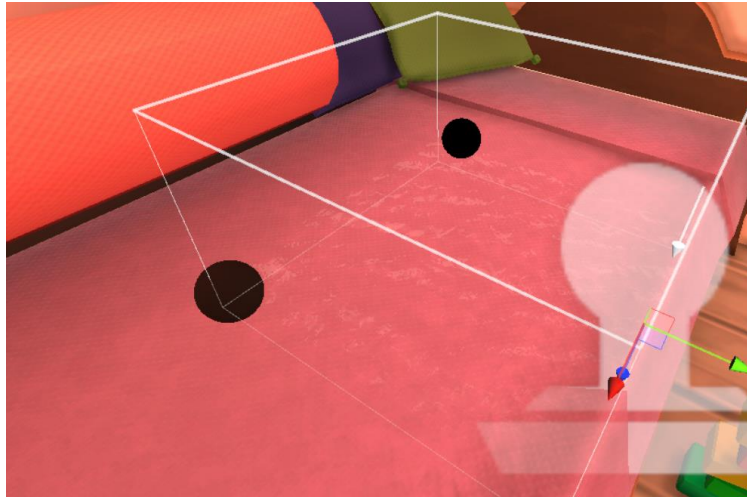**FIGURE 20 TEXTURE INPUTS FOR DECALS**

The information of normal maps got displayed, but it seemed that it changed only the color value than actual normal while projecting. After playing around with the decal settings, my own decal materials in the shadergraph editor and some research, I accepted that I don't know enough about the unity normal pipeline and I will research this problem in the semester break on my own with decal and trimsheets.

In addition, the current result without the decalsetup was sufficient enough, so it was not only a mental health decision but also for my productivity progress.

The blending with masks and color only worked well and I used that for some sketches in the project, so there would have been some workaround possibilities for additional detail if needed.

VFX TESTS:

Even though I've mentioned that we have put this on hold, I still did some research and things I want to document for the sake of completeness:

- Dust cloud effects to support the impact of animations (jumping on the bed, wardrobe interaction)
- Electric spark placeholder

-

FIGURE 21 UNITY ELECTRICITY PLACEHOLDER

Sadly, I had to throw the particles out since they caused errors in the project even though if they were not used in the scene. I guess this is caused by an installation error by the package manager. After we decided we won`t go further here, I didn't investigate there.
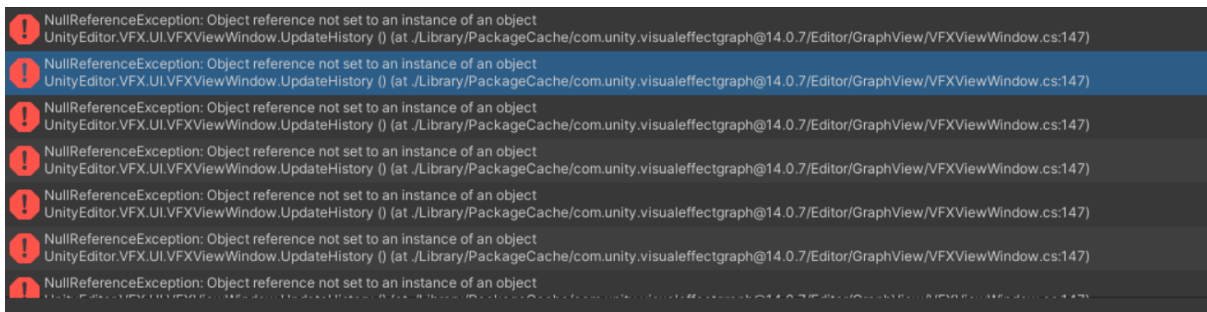
I did them in the visual effect graph in Unity and planned to get in touch with the different nodes while working on these effects.

I won't go too far into detail here but with some tutorials and a script I found online, I have created a custom interactable fluid shader for a lava lamp to save some time in the animation process later.
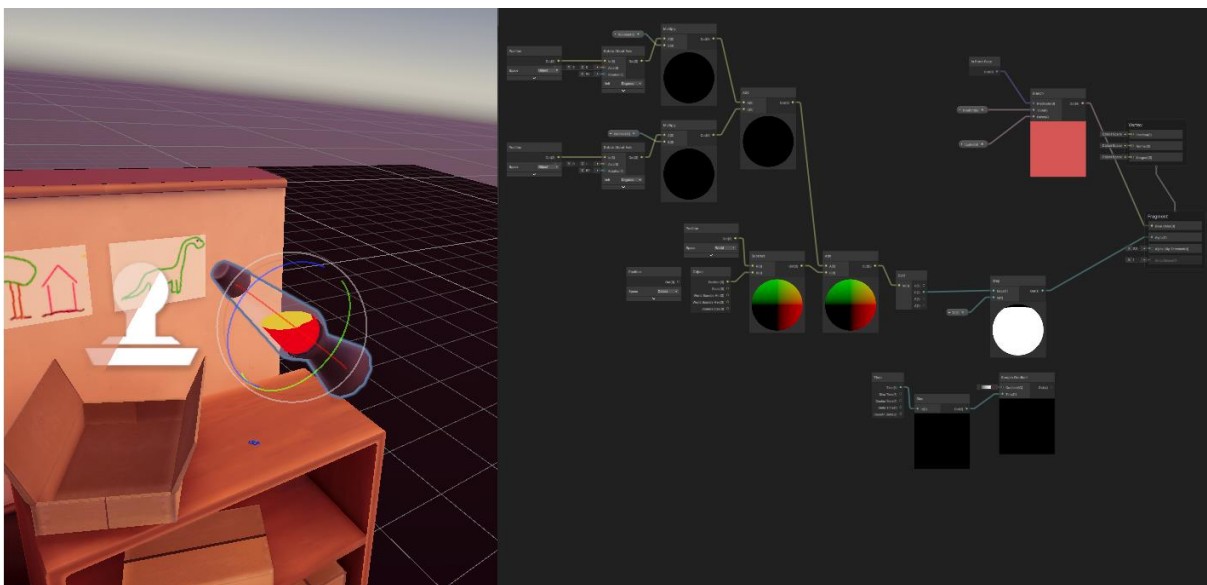

FIGURE 23 UNITY - FLUID MATERIAL

Unfortunately, it was during the cancelling decision so I stopped working on it but it already worked on a mechanical level and is missing currently an animated value from our code and some textures.

LIGHT

I´ve set up a URP asset which uses screen space ambient occlusion (SSAO), but the downside of it is that is calculated in real time and might influence the performance. If we would run into troubles there, I would change it to a baked version. For now, it is more suitable for us since we got moving objects in the scene.

Additionally, I used light probes, since we got a few moving objects in the scene and I don`t need technically real time illumination, but if the light differs during the level, the effect might be needed, and otherwise it is only a little moment of more light bake time.

During the process of adding props to a scene, I started losing control over specific shadow areas. For example, if a corner wasn't dark enough, I need to make the other shadows darker as well.

To fix that, I decided to use decals with a simple black / alpha gradient on top of the floor to exaggerate some areas.

FIGURE 24 UNITY - DECAL COMPARISON

To tweak details, I've added a postprocessing profile which I used to color shadows and their intensity to exaggerate the soft and cozy feel from the first light tests mentioned above.
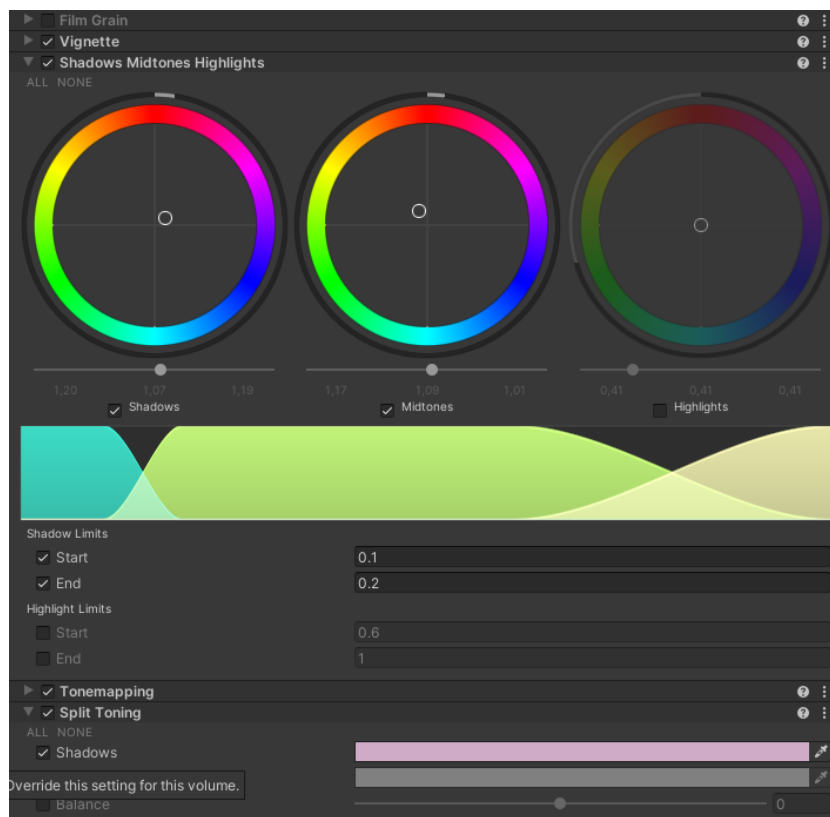

FIGURE 25: UNITY - GENERAL POST PROCESSING PROFILE

FIGURE 26 UNITY - POST PROCESSING COMPARISON

While baking occurred a Problem: Especially between the wall modules (which did not intersect) had Unity a hard time calculating an even result. This led to light bake artifacts. I could fix most of them by tweaking the lightmap resolution and changing the Lighting mode from shadow mask to baked indirect. I guess that the shadow mask is only calculating shadows while the indirect baking method is also adding highlights – but this is something I can only answer after some tests in a clean environment.

FIGURE 27 UNITY - LIGHTBAKE EDGE PROBLEM

DECALS:

Some problems, like artifacts can occur by intersecting geometry, so I exchanged some pictures with decal sheets. One important rule while creating those (of course in paint 3D) was to use frameable images, otherwise the cropped decal would display neighboring sketches, which are not wanted. One sheet was created by Melanie Werner and Rasmus Robertsson during bullshittery chatting on discord.

FIGURE 28 DECAL SKETCHATLAS (BY MYSELF)

FIGURE 29 DECAL SKETCHATLAS (BY 3RD PARTY)

It's basically just a miniature texture atlas, but for decals only.

GAME DEVELOPMENT

Due to our programmers request not to add or change anything in his code, I was unfortunately only able to write a small amount of code.

Timer:

I researched how to access the variable fillamount to access it with code. Besides that, I added the early version of our timer script which got deleted later, so I sadly have no prove for that anymore.

Tape effect:

I`ve created a script which animates a float. Since I was supposed not to change anything, I decoupled this script by using the isactive state as a trigger.


```
Tapespeed (Tape Settings)

Script                              tapeSO
Fillspeed                           0.3
```

```csharp
public class tapematerial : MonoBehaviour
{
    public TapeSettings tapesettings;
    public List<GameObject> go;
    public List<Material> tapemats;
    [SerializeField] ObjectSFX sfx;
    private float fillspeed;

    private void OnEnable() {
        if(sfx != null){
            sfx.InteractionAddTape();
        }
        fillspeed = tapesettings.fillspeed;
        go = new List<GameObject>();
        tapemats = new List<Material>();

        foreach (Transform child in this.transform)
        {
            go.Add(child.gameObject);
            tapemats.Add(child.GetComponent<MeshRenderer>().material);

        }
    }

    void Start()
    {
        foreach (Material mat in tapemats)
        {
            mat.DOFloat(1, "_Fillamount", fillspeed);
        }
    }

    void Update()
    {

    }

    private void OnDisable() {
        if(sfx != null){
            sfx.InteractionRemoveTape();
        }
    }
}
```
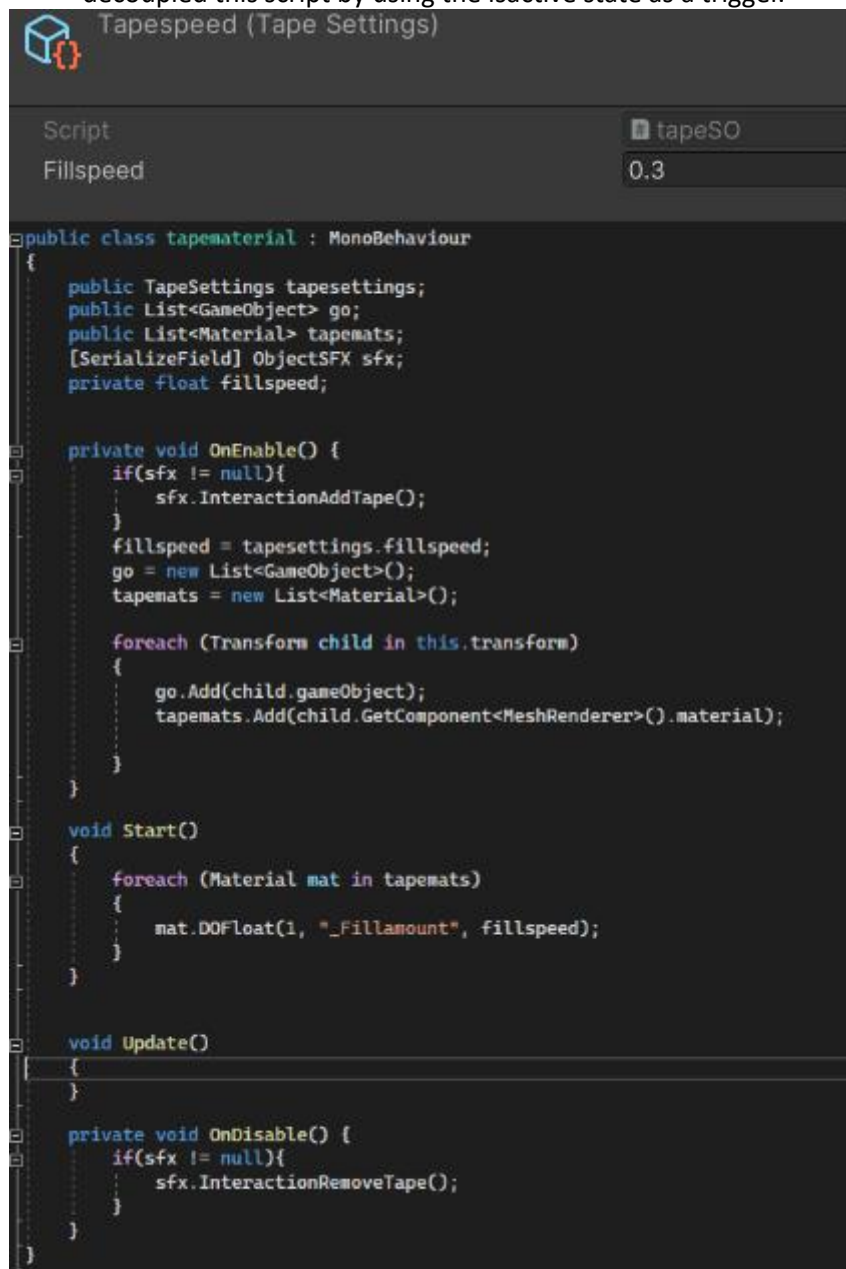
FIGURE 30 TAPE SCRIPT & SCRIPTABLE OBJECT

To have a global accessible variable to tweak the speed, I also added a scriptable object.

Dotween package:

I also researched and introduced dotween to my team, to have a quick and easy implementable solution for adding game juice to our project. Dotween is well documented and comes up with its own functions to animate code.

Outline shader:

Besides that, I researched and started implementing a third-party outline shader script using a solution for our current outline situation (Quick outline effect by Chris Nolet, 2018)

21

Since this approach didn't work immediately (material wasn't shown in the editor / play mode) and I don't want to cause conflicts in our team morale, I decided to do a workaround by parenting a scaled copy of the mesh in the prefab with an backface only material on it and I will give it another shot after the current stress level is decreased.
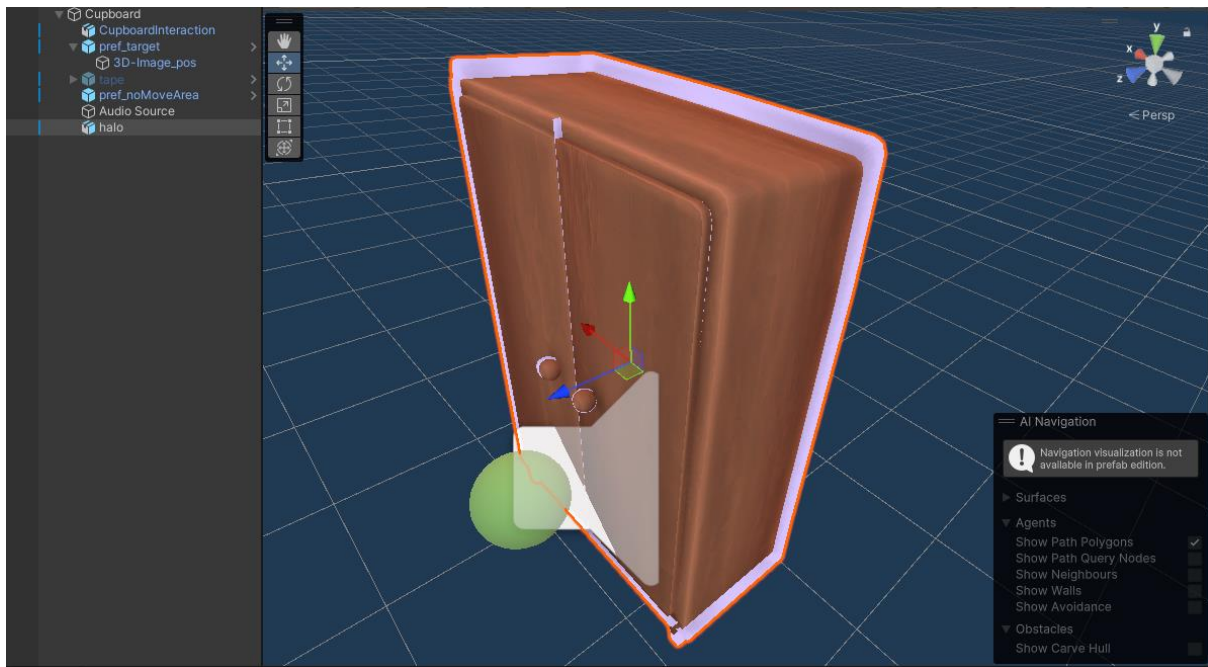

FIGURE 31 UNITY - OUTLINE WORKARROUND

Since the implementation of tech art does also fall in the content of the Game Dev for Producers lectures, I could argue that it is technically part of Game Dev, but it felt misplaced not mentioning it in tech art.

METHODOLOGY

Instead of the synthesis paper this documentation is my submission.
Besides that, the already implemented research approaches in the chapters above should fulfill the requirements.

ANALYSIS / POSTMORTEM

All in all, I am happy with the outcome of this project. Overall, the team communication and motivation were consistent, we had a good feedback and discussion behavior and – most importantly – we did not procrastinate. What I really liked was that we had respect for each other's roles. Everyone kept the responsibility which felt like a good work environment.

My vision was kept and the only thing I am missing is a reliable game flow, I think we have more potential in the game design section, but it is still on a solid and working level.

Zoey did a great job balancing between needs of the project (/my opinion) and her own wishes as an artist. Instead of arguing with each other, we discussed and convinced ourselves with the best solution, which was a very nice experience.

Anastasia did a great job as a project manager and kept an eye on the general chaos. I never had any pressure to keep track of any deadlines, since she reminded us in time before we could come into a crunch situation.

And Lukas did well programming his very first own game. This is a hard task, and I am surprised how well it went.

As a Artist, I am okayish with the result. I originally thought of a slightly different art style, but I think I would have achieved this with more knowledge in shader programming.

As a Sound Designer, I am a bit sad about organizational situation of the university, but I am happy that we got at least Julius, who offered to work in his freetime.

But there were also downsides I can't ignore too:

Sadly, I noticed that there are interhuman conflicts which I couldn't solve during the project, and I decided for myself that my personal tolerance threshold is reached.

I don't think that this is fitting in a postmortem, but what I learned Is that I need to start accepting that if another person is responsible for something, that I don't need to bother about it, now matter it went.

At least for the sake of my own mental health and my productivity.

I need to accept If a person just refuses advice, even if it might not be logical for myself.

Another thing is, that some details were forgotten during the discussions, also by myself. Our consistent way of communicating and documenting fixed these questions, but this could have gone worse. I will keep an eye in the next project on this, but I think it's a logical result by the amount of information compared to the short timespan we had.

CONCLUSION AND FUTURE WORK

I really wish that there will be a project where I can go for luxurious fields like VFX, but I think this is something I need to work on in my own projects. Speaking of those, I got now several ideas what I could do in the semester break instead of "recovering" … :D

And as I mentioned before, I also noticed for myself that I will try to avoid specific personal traits in the future, but this is more a new lesson for my social skills than a fulfillable target.

REFERENCES

APPENDICES

3D Assets:

Roomlayout:

- wallmodules (background)
- wallmodules (front)
- window
- door

Furniture:

- bunkbed
- drawer long
- drawer short
- single drawer element
- „Kommode"
- child desk
- table (wood)
- table (plastic)
- chair (wood)
- chair (plastic)
- shelf

floor:

- carpet (round)
- carpet puzzle (squared)
- carpet puzzle (rounded)

interactable:

- mimic chest
- wardrobe
- cupboard
- cactus
- electric cable
- bed
- cookie plate

decorative:

- backpack
- book set
- toy brickwall
- cardboardbox set
- clothset flat
- clothset angled
- dinosaur skeleton [WIP]
- football
- lamp short
- lamp long
- lavalamp [WIP]
- dogbowl
- sketchbook
- pillowsets
- toycar low resolution (red & green)

- toycar high resolution (blue)
- toy rubix cube (solved)
- toy rubix cube (unsolved)
- penset
- actionfigure
- teddy
- rocket
- several iterations of greyboxing and models

Textures:

Tileable:

- floor texture
- cardboard trim sheet
- carpet texture (rounded)
- cloth texture
- plastic trim sheet
- tape texture
- wood trim sheet

Hero (individual textureset):

- backpack texture
- electric cable texture
- cactus texture
- lavalamp texture
- cloth texture
- mimic texture
- teddy texture
- wall texture

Msc:

- gradient texture
- decal atlas sheet 1
- decal atlas sheet "rainbow"

THIRD PARTY ELEMENTS

Textures:

- decal atlas sheet 2 (Melanie Werner / Rasmus Robertsson)
- colored checker texture (Blender)

Packages (Art):

- URP (Unity)
- Shader Graph (Unity)
- Visual Effect Graph (Unity)
- Dotween
- Simple outline

Liquid Effect Tutorial:

https://www.youtube.com/watch?v=tI3USKIbnh0&ab_channel=GabrielAguiarProd.

Sound(background):

https://www.youtube.com/watch?v=p0MKy6CENbY&ab_channel=%40backgroundmusicforvideos

Sound Design (Julius Braun)

- mimic interaction
- wardrobe interaction

[not implemented yet]

- footsteps
- timer sound effect