Digital Game Overview

Game Concept

Initially, the main goal was to develop a game in which the player controls a character that is sliding through a water slide and needs to avoid obstacles, like certain levels of Sonic Forces, a video game from 2017 that I played on Nintendo Switch.

But then I remembered a game mechanic that I thought about some months ago – a mechanic related to colors. Depending on which color the player would collect, he could pass through some obstacles or be hit by them.

To supplement the mechanism shortly described in the last paragraph, I also had the idea to make some sort of points system, based on the obstacles that the player could pass through or could not so the goal was to achieve the end of the game with a certain minimum of points – take the game Detroit: Become Human as an example, when Connor, an android inspector, needs to interrogate other androids without making them feel too much or too less stress.

These are the main characteristics that I wanted to be present and crucial to the game. Later they will be expanded for better understanding.

Synopsis

It was a dreadful day for Nanail. A guy fully in love with his girl, finds out that she is cheating on him!

Heartbroken and desperate to fall asleep, he searches for medicine to make him go to sleep. Until he finds some strange gum-shaped pills that look like candies and gives them a try.

He will regret that.

Hallucinations or not, reality or fantasy, a new battle of survival in a neon-psychedelic world starts now!!

Title Meaning

This game’s main title, Magic Jelly Beans, has two main objectives:

* Present the game’s main plot – a guy that takes drugs and ends up having hallucinations while sleeping;
* Present the game’s main mechanic – these “magic jelly beans” are the collectibles of this game and allow the color system mechanic to exist.

The word “magic” is a double reference – “magic” because they allow the protagonist to “imagine” things and enter into a new “magic” world, and is linked to “magic mushrooms”, an existent drug known for creating hallucinations as a side-effect of taking them.

The words “jelly beans” are a reference to candies with a bean shape.

A picture containing sweet, colorful

Description automatically generated

Protagonist Name Meaning

Nanail is a variation of the name “Daniel”, which has Hebrew origins and is related to a prophet cited in the Bible that can interpret dreams. It also contains the Portuguese word “nana”, which is a verb form from the slang verb “nanar”, which means “to sleep”. <https://momlovesbest.com/names-that-mean-dream>

Target Audience

The target audience for this game should be people older than 16 years old. The main reasons for this target audience are the presence of sexual content, which can be mild or explicit, and an ironic mild reference to illegal drugs (the magic jelly beans that the protagonist takes). This decision was based on the PEGI 16 description.

Game Genre

The genre for this video game will be action (game) since it consists of a fast-paced game that requires quick reflexes from the player. More specifically, this game will be an auto-run platformer, a subgenre of platform games in which the character is forced to run forward and needs to stay alive for as long as possible while avoiding obstacles or other challenges that come their way. The opposite of the endless runner games, where the game is infinite, this game would consist of multiple levels, each one focusing on different mechanics and challenges, while the main goal remains the same – to survive and reach the end of each level. Basically – each night, Nanail takes the pills to sleep and then he gets the deadly hallucinations and needs to survive.

The perspective adopted for this game is the third person one, the common one used on this type of games (such as Boson X) that also allows the player to have more control about whether it collides or not with a certain object – it’s more visual than it could be with a first-person camera, at least, because you see the character moving.

Game Engine

Since this game was developed by only one person in a small amount of time, the only two options that were considered were Unity and Unreal Engine.

Firstly, only Unity was considered to be the chosen game engine, mostly because of the already previously acquired knowledge taken by me. Unity is also well suited to developing 3D games and has a huge community always available to help others. The online resources are endless too. So, for all this, adding to the fact that I have such a limited time to make this game and would not have time to learn to work with a new game engine, the decision to pick Unity seemed very predictable.

But then I realized there was some news online about the Unity company merging with IronSource, a company that developed a malware installer. Starting to think my game development would be compromised, I thought about switching to Unreal Engine.

Unreal is one of the most popular game engines nowadays, often used to make AAA video games. Although you could need to write code in C++, which can be hard since it’s a low-level programming language, Unreal also offers the possibility to write code visually in the editor itself, using the Blueprint functionality. Very promising.

In the end, it doesn’t even matter. I ended up picking Unity again, because of my previous knowledge.

Unique Selling Points

* Explore an “out-of-the-box” mechanic that is easy to understand – colors!
* Play an unusual platform game – with a premise that involves cheating and horns!
* Experience a platform game in which every decision you make can make the difference between reaching or not the end!
* Submerge into a neon-colorful-psych world of hallucinations!
* A game where you could end up losing so that you can win!

Game Play and Game Mechanics

Conceptual Map

Rules and goals

Rules are an important component of game mechanics since, according to the operational rules defined by David Parlett, are essentially the actions that the player is allowed to do to play the game.

This game’s rules are:

* + The player can move the character to the left and the right;
  + The player is able to dash forward;
  + The player can collect special objects – collectibles – that have a gum shape (look like candies) to change the character’s color;
  + The player can pass through obstacles if the character and obstacle have the same color;
  + The player is not able to pass through obstacles if the character and obstacle colors are not the same;
  + The character falls if he steps into a hole;
  + The character also falls if his color and the slide’s color are the same;
  + The character’s stress level increases when he cannot pass through an obstacle;
  + The character’s stress level decreases when he can pass through an obstacle;
  + The character is pushed back after he collides with an obstacle he cannot pass through, and then he can trespass after colliding for the second time;
  + When colliding for the second time with an obstacle with a different color, the character’s stress level is not increased;
  + The character dies if his stress level goes out of range;
  + Falling and dying have the same meaning – after that, the game continues from the last picked checkpoint;
  + The player may need to restart the level if he made the wrong decisions (picked the wrong collectibles at the wrong time) and then cannot end the level.

This game’s main goal is to make Nanail survive until the end of each level, which each level represents a night in which he took the pills, and for that, he needs to avoid obstacles or be immune to them.

Game mechanics

Game world

Technology

* Technical Implementation

In this section there will be discussed what was going to be implemented, most aspects related to the reasons behind the modeled meshes, and why they ended up being the way they are. In short, I sustain the decisions I made.

Nanail – the man protagonist of this game

I wanted to make a skinny-mature male protagonist. Although my main inspiration was the character Vincent from the game Catherine: Full Body, who is a man in his thirties but completely irresponsible and aimless, I wanted a man that was also in his thirties, but that had goals. To have a wife, to make a family. I tried to make some hints about this during this game’s intro, when I refer he ends up depressed after being cheated to the point that he needs to take drugs to forget the trauma – because he truly loved her, and I tried to give him some emo vibes.

First, since Nanail is a “man in his thirties”, I wanted to make him hairy, to give him a more mature personality and despite, at first glance, people could think he is lazy and careless about his looks (he does not shave), I thought of this differently – it’s because he attained a certain state of mind in his life and at his age that he embraced who he really is.

Second, I wanted him to be shirtless – again, to make him look comfortable with who he is (and to be able to show his hairy torso, back, and armpits), and to connect with the main game narrative. Remember that he wants to sleep, it’s bedtime. Most men turn out to be shirtless when they go to sleep, just like what happens also with Vincent. Consequently, he only wears pants and socks.

Third, I wanted him to have horns – and I think this one is easy to get it. He got cheated by his girl, and in Portugal, we have a slang expression that is “corneou-o”, which means “cuckold”, and “to give horns”. I did not even know that there was also an expression for this in English. We, humans, keep learning things every day. For the horns’ design, I made multiple sketches the check out which one was better, and I ended up with one that looked more like the ox’s horns. Initially, it was supposed to come out through the forehead, but then I realized that, even though my main goal was never going to make a realistic-looking guy, it was going to be hard to make an acceptable hair. That’s why I ended up making some sort of helmet that is stuck to his head.

Fourth – talking about the pants. I wanted them to be large since it’s supposed to be pajama pants and they use to be comfortable, and I added decorative beans to make a reference to the game’s main title and to what has put him on that hallucination-nightmare state – taking some strange gum-shaped pills, some magic jelly beans!

Collectibles

The idea was to make all collectibles based on a well-known candy-gum shape, each one with a main colour. Take as an example the banana gums – they are yellow – or the gummy bears – each one with a different colour! This had two purposes – to show to the player the “magic jelly beans” that the man had taken (so, relating to the narrative and game concept) and to turn the reference to (illegal) drugs milder. It’s cooler to tell that, although they can be (potentially) illegal, they seem just colourful candies that make you change your colour! That’s way cooler.

For this reason, five gummies were selected: banana, red gummy, pink gummy, spiral, and bear. These are all gummies that are even sold by *Haribo*.







Source all of them: https://www.haribo.com/pt-pt/produtos/haribo/favoritos-classic

Obstacles

Initially, I only thought about ordinary obstacles that can be found in every game of the same genre and/or objective. And what do I mean by ordinary obstacles? Walls. So I started brainstorming about what kinds of walls I could make to add to my game. The first planned walls were the wall slices (only part is obstructed), complete walls, moving walls, wrecking balls (yes, just like Miley Cyrus’ one), and grids. The purpose was to make multiple grids with different hole thicknesses so that in some cases the player can pass through the hole, but in others, that is just impossible. After deliberating a little more, I concluded that a grid with a smaller hole thickness would do the same effect as a complete wall, so this last one was discarded.

However, I remembered that I needed to make more relations and references between the gameplay and the narrative itself – so that no one would think “ok, that’s a good starting narrative but the gameplay doesn't need it anyway, this game could be placed on everything else, it doesn’t matter”. I started imagining good starting points to develop fresh-new obstacles that could integrate the player better into the narrative.

I focused on the fact that he got betrayed. Why did this happen? He seemed a nice guy. The problem should be around some notion that his wife needed and that he was not able to give her. Maybe she wanted better sex… And that’s how I ended up giving more depth to the narrative without really telling something about it, just living it to the player’s creativity and imagination. Although this one was easy – she did not like his penis size as it was already presented in the intro – “Size does matter after all”. So, why not the wife lover's huge penis as an obstacle? Scary as hell.

More ideas. What if, during the hallucinations, he starts thinking that his wife got pregnant? Of course, he would suspect if the child is someone else’s. What if the hallucination itself told him that, in reality, HE IS NOT THE FATHER! That’s so creepy.

More. What if even his wife’s so beloved soft breasts turn against him? And that’s how I eventually added these three obstacles into the game. I also thought about adding a clock, just to the fact that what he was going through was not real and was like a dreamy world, but I ended up discarding it.

Checkpoint

To make the checkpoint, I wanted, again, something that could be related to the main story as well. I tried to think of some subject that he could want (just like a person wants to reach a new checkpoint to stop playing because it’s bedtime). His wife back? Having romantic-not-only-fleshy sex again with the person he loves the most, that he could die for her? Perfect. Sad, but perfect.

This culminated in me wanting to make a vulva. However, I realized it was something too explicit and sexual to be present at a game. So I censured it and made it inside a cut peach, where its pit is usually located.

Ground

There comes the tricky part. In the first instance, I ought to make a complex slide, similar to the ones that exist in aquatic parks, with plenty of bends.

Later, I came to the conclusion that it would be too hard to make the game programming logic work fine with such a complex water slide shape, in such a small amount of time. From now on, all slides were horizontal.

* 3D Design

During this game’s development, Blender was the most used tool, since it is one of the main tools used nowadays for modeling, and because of some previous knowledge – Blender was used during one of the last semester’s classes. Also, the other well-known modeling software, Maya, is not free (although it can be for students) and does not have such a huge community as Blender has. Furthermore, the choice seemed obvious.

In the beginning, Adobe Substance 3D Painter was also used – in this case, it was for painting the first sketch of the character model – texturing basically. I used layers and masks to paint each different colour and ended up baking the recently-created textures. But then I stopped using this, and I’ll explain why later.

Character’s first model

Nanail’s first model was the first thing I modeled during this project. At the beginning of the development, the goal was to make a low-poly character to give a more *retro* look, because of the shader that was going to be made for the slides (where there are a bunch of squares – for more details, check the Shaders section), and because of lack of modeling experience (I had some classes about it, that’s fair, but my background is programming, ok? Don’t tell me to make some realistic character. And talking about realism – the purpose was to make an unrealistic-neon-visual game, that can also look a little bit childish sometimes. Because, for me, games don’t need to be realistic to be fun. Do you recognize that it is a human being? Ok, fine.

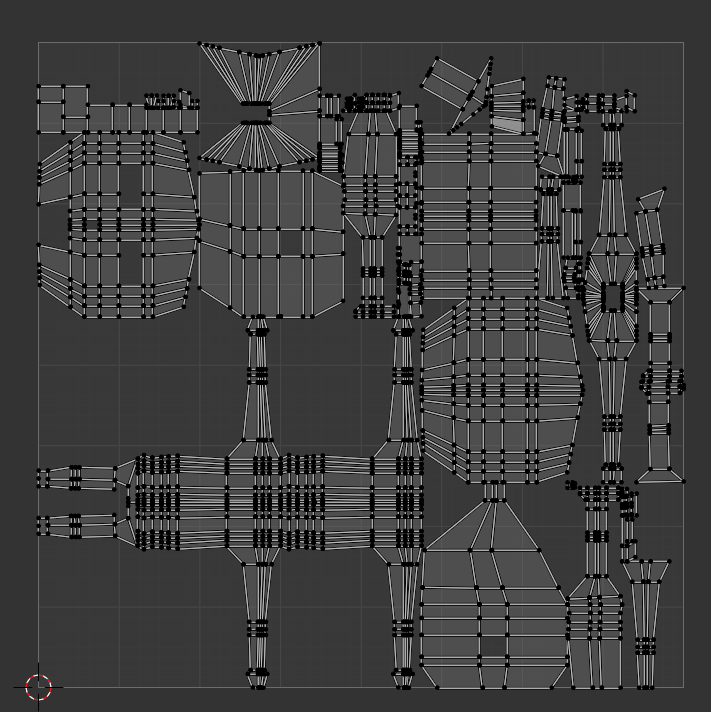
So, while I was looking for good ways of making low-poly characters, I discovered an add-on called Auto-mirror – that “automatically” was making the mirror of my character, so that I could check in real-time both sides of them while only modeling one. I later discovered that I could only enable the Mirror Modifier on Blender early and continue the modeling that it would be the same thing (that’s what the add-on does), but I really enjoyed it, it was faster and good for me.

The first character’s model was made of mostly loop cuts, extrudes, and scales while making attention to making an overall human-looking model.

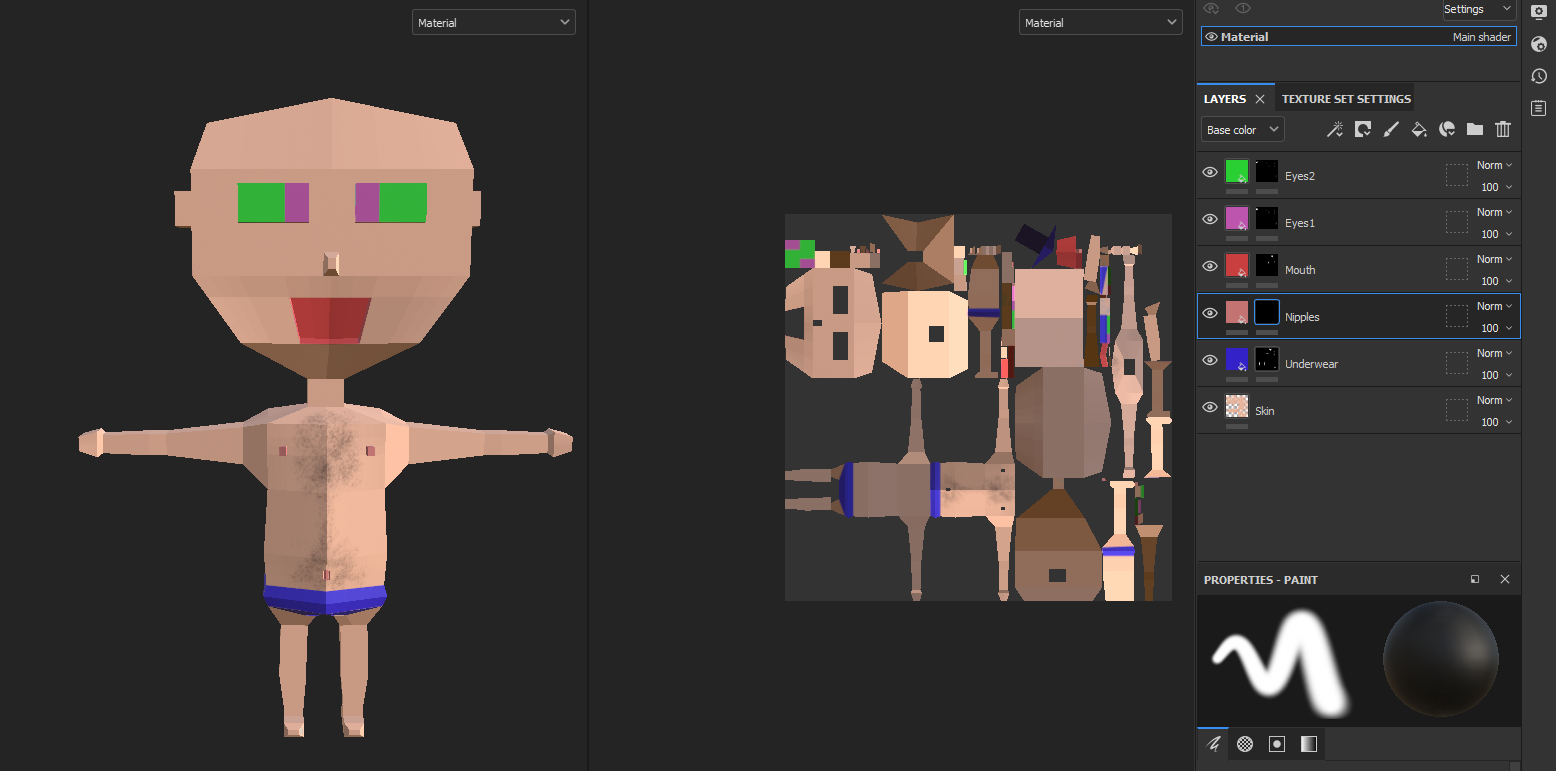
A picture containing text

Description automatically generated

Finished the modeling, I made the UV mapping to later be used on Substance Painter.



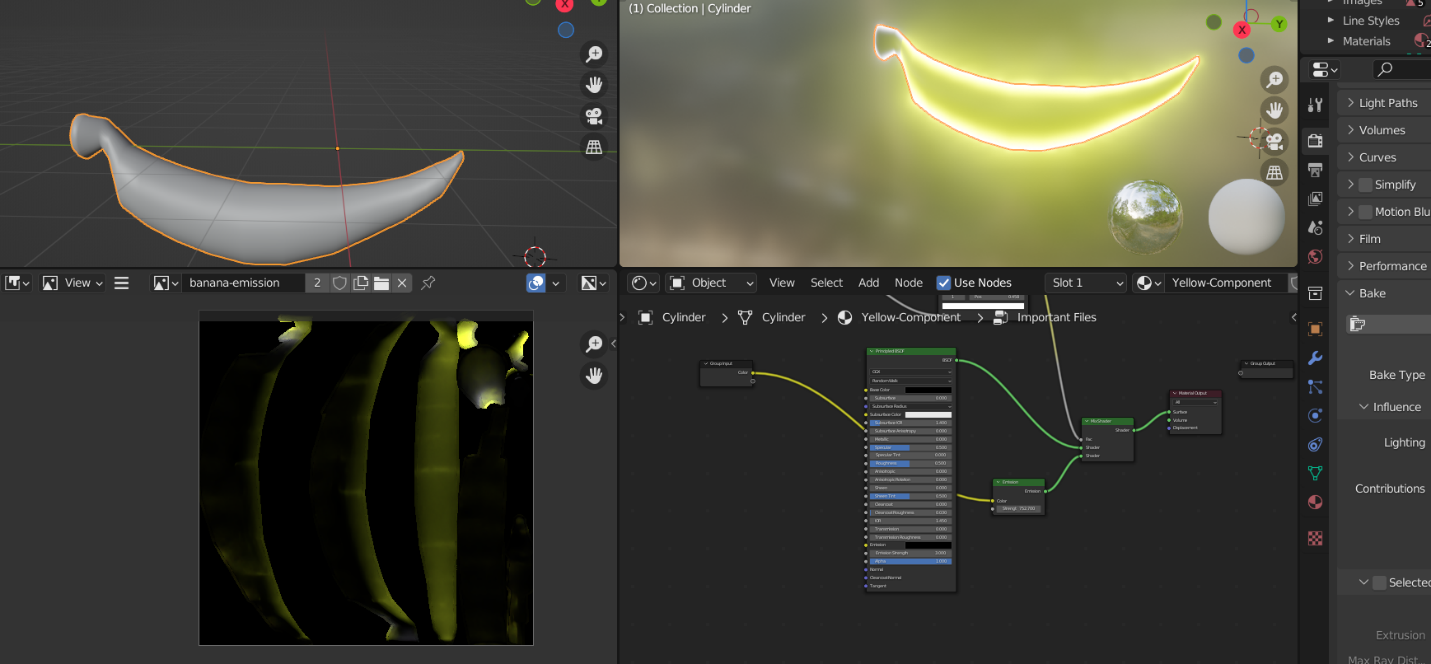
Finally, I started texturing on Substance Painter and baked the textures, so that I could use them on Unity and create a material for the character himself.



This approach was later discarded – I ended up realizing that it would be better for my case to assign multiple materials to a mesh on Unity, instead of using other software to create and bake all the textures and then create only one material for a mesh.

Since a big component of this game is the constant neon-glowing visuals, it was important for me that each part of each mesh would be distinguishable and have the glow effect at the same time, which was not happening if I baked the textures – the meshes on the scene that only had one material (and applied a material with glowing effect, using a shader), with the post-processing enabled, was making the material based on textures not visible at all!

Another thing that happened was that, even if I made a shader on Blender and then baked the textures to create the material on Unity, the result was underwhelming. Below there is an example of how the resulting texture after baking an object with a glowing effect.



Collectibles

* Banana Gummy

To make the banana shape, I first started with a cylinder, and then I added loop cuts and scaled some edges - all this while the proportional editing: smooth was enabled. To make the corners, I used the not-so-fancy technique of making an extrude and scaling it. I then made correspondence between each one of the two components of this banana mesh and a material slot; to help me select these face groups, I enabled the Toggle X-Ray.

Chart

Description automatically generated

Chart

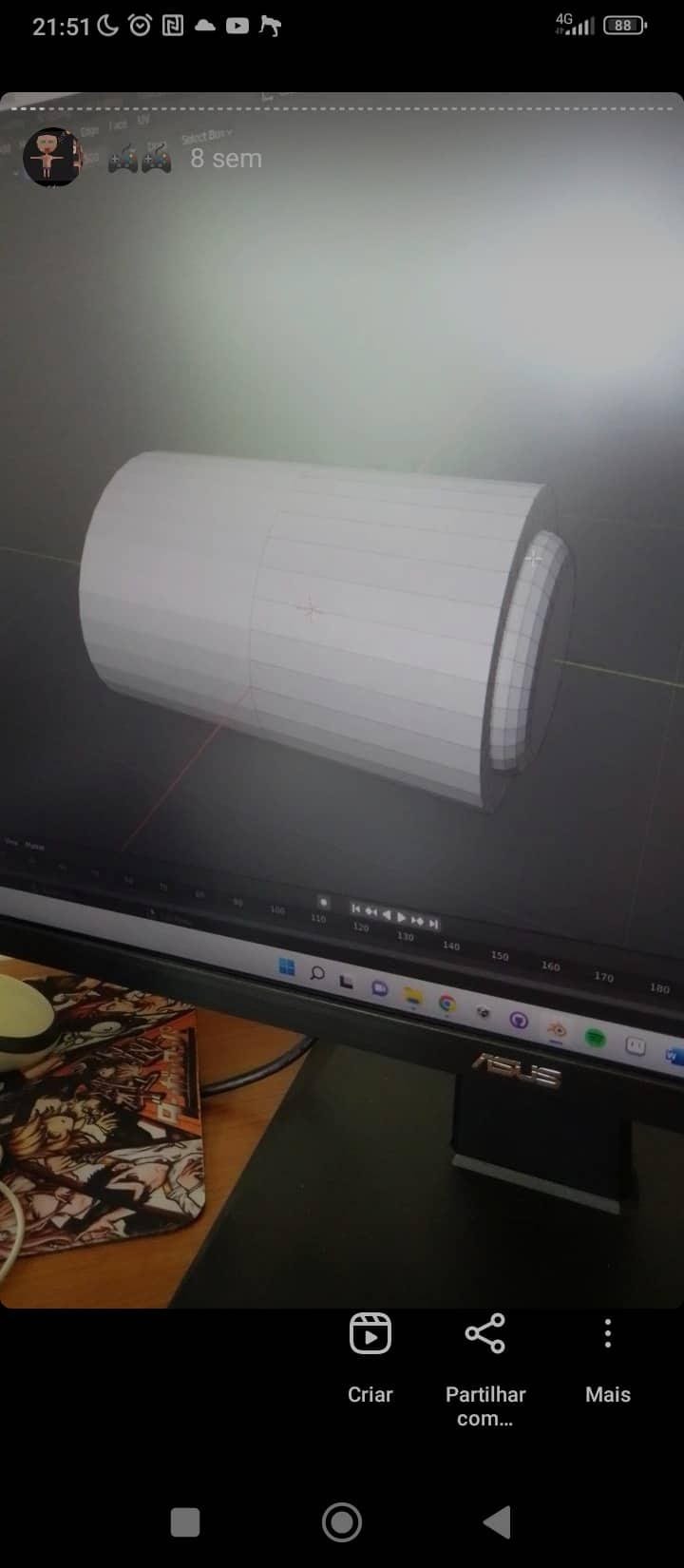
Description automatically generated

Graphical user interface, application

Description automatically generated

* Red Gummy

In this case, I used again a cylinder as a base mesh and scaled it to make it look more flat. After that, I enabled the Auto-mirror add-on and recurred to the Inset Faces Blender functionality at the base of the cylinder, followed by an extrude to the newly created face. In the end, I made a bevel to give it a more rounder appearance. To distinguished parts of this model were assigned a new material slot – one for the red component and another for the white component (top and base parts).



A picture containing text, indoor

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

* Bear Gummy

Almost sure it was the collectible that gave me more hard work, although most of it is successions of loop cuts, extrudes, and scales, with bevels to the ears edges, feet, and lateral edges as well. A lot of edges were also edited by hand. It was also a great help to use the Subdivision Surface modifier before modeling this mesh, to help make the most detailed parts, such as the face. In the end, the Decimate modifier was applied to reduce the number of total faces.

A screenshot of a cell phone

Description automatically generated with low confidence

A picture containing indoor

Description automatically generated

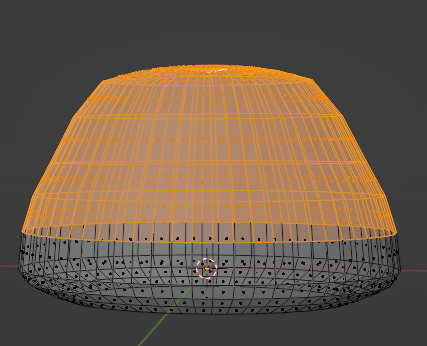
* Pink Gummy

For this model, I recurred to a sphere base mesh at the beginning of the process. I turned on the option Toggle X-Ray to select entire (round) sections of this mesh and scaled them, while the proportional editing mode: smooth was enabled as well, and scaled up at the bottom and scaled down at the top.

In the end, I assigned two different face groups to a material slot each – the top component would have a pink-coloured material and the bottom one a white-coloured material.

A picture containing tent, outdoor object, dome

Description automatically generated



A screenshot of a computer

Description automatically generated with low confidence

* Spiral Gummy

To make this mesh, I used a plane that was firstly scaled on the Y axis. Then, I made a bunch of loop cuts and turned on the proportional editing mode: smooth so that, while rotating the first vertice at the top left of the plane, and while controlling the proportional editing range by wheel scrolling, I could start to have a spiral shape. Finally, I applied a Solidify modifier to give it some depth.

Chart, scatter chart, box and whisker chart

Description automatically generated A screenshot of a computer

Description automatically generated with low confidence

A picture containing cable, outdoor object, wire

Description automatically generated