
Game Design Document
For
MOONATEES

Table Of Contents

.Revision Notes

0.01 - Made the first version of the first part of the document.

0.02 - Added some more things.

0.03 - Added some flow for the menus and described them, did something else

0.04 – Had a small writing session with Sami to clarify some sound/graphics issues among other things

0.05 – Minor corrections to the text

0.06 – Had a meeting with the team and a coaching session with Pekka, added the meeting notes here to expand on them tomorrow.

0.07 – Removed mentions of survival and some minor things.

0.08 – Added a reason to use wind to the physics section.

0.09 – Added Item Properties and Electricity into Mechanics

0.1 – Added Wind into Mechanics

0.2 – Added Table of Contents and Page numbering, rearranged chapters, expanded controls

0.3 – Added core loop, a game flowboard, made the settings flowchart a little simpler, removed main menu flowchart since it's in the game flowboard, removed mentions of a store.

0.4 – Added first steps of design, added a 'philosophy' for level design, fixed settings flowchart

0.5 – Merged overview into the introduction section, added chapter numbering, updated some segments with minor stuff as I saw them while reading the document again.

0.6 – Moved in-game HUD under UI. Removed mentions of tablets.

0.7 – Made some clarifications to game progression and the effects of player choices.

0.8 – Updated Core Loop and Game Flowboard.

0.9 – Described the ship a little bit in the environment part.

1.0 Finished for Gate 2

1.1 **GAME PATH: Marked OUTDATED on clearly outdated stuff, might have missed some.**

1.1.1 **GAME PATH: Updated In-Game HUD & Menus**

▪

▪

▪

.1. Introduction for a new member/ Overview

WELCOME TO TEAM MOONATEES!

We are extremely happy to have you here developing this game that we really feel passionately about. We thought it might be useful to shortly introduce ourselves and the game itself in an introduction section.

Features of the game include:

First Person Zero Gravity Puzzler with separately controlled arms. So we use a first person camera to control our character and it's arms in a 3D environment. The reason we say 'it's' is because our main character is a moonatee, a manatee from a moon. The moonatee will attempt to solve problems that occur on the spaceship, so that's where the puzzles come in.

Comic Style visuals. We aim to have a comic book style for the game. The challenge here is to bring that feeling of comic books into a 3D spaceship environment, but so far we think we have an idea on how to do it.

Unexpected solutions. The puzzles should have many solutions. Some of these solutions need to be something that the player does not expect to actually work... BUT IT DOES! WTF!

Complex yet fun mechanism includes things like wind mechanisms for propelling objects into different places, moonatees included. Electricity based puzzles that are based on getting electricity from point A to B by forming links between objects.

A lighthearted story of industrious moonatees trying their best to deliver their cargo of precious buckets to a moonatee colony far, far away.

Design Goals:

Unexpected funny business/Chaos/'Härväys' A favorite child has many names, the main feeling we want to go for with the game is a feeling of *'Anything might happen, and wow it's actually happening'*. Unexpected events occur and we make the player feel that they might have caused it with their tomfoolery with the arms.

Freedom. This might seem like i'm repeating things, but it's very important that the player does not feel pigeonholed into a solution. He really should feel that there are always options available to him. The player should feel free in his environment, he should feel that he can always solve the puzzle in a way that makes some sense to him.

80's comic book. We aim to have a feeling that the player is in a 80's comic book. It's very silly, but makes some sense, the visuals don't have the same gradient as they do in the old papers but the feeling is there. Audio and visuals should mostly aim for this.

Level Design philosophy. Described in detail in the levels section of the document, but basically the solutions that have to exist for puzzles are: By the book solution, Unorthodox solution, Wrong but works solution. With the possibility of an added Chaos Factor.

Mika Simula

I am Mika Simula, a Communication professional who understands technology. My roles are Producer and Marketing Manager. I have been previously mentored by Juhani Lassila, the Head of Communications at Jolla, so I am ready to flap my flippers like crazy.

Antti Pikkuaho

Hello! I'm a student at the University of Applied Sciences of Oulu, third year studying information processing sciences. Before that I was in the university studying the same thing, but it was too impractical for my tastes. I've been an avid gamer for at least 15 years. My main area of expertise is game design, programming and 3d art. I'm also a little bit of a weeaboo who likes visiting anime conventions and reads manga, but true to my chaotic nature i'm also interested in rock climbing and football and stuff like that.

Sami Räbinä

Hi! I'm third year OUAS student, studying degree programme of Business Information Systems, line option of Internet Services and Digital Media. I was introduced in gaming when I was 4 years old, but still I prefer myself as a casual player, since usually my platform of choice are consoles rather than PC. I enjoy creating concept arts and illustrations for games but I'm capable as a sound designer and musician/composer. Also I have some experience in social media marketing and web designing as well. Otherwise I'm just usually regarded as a odd creature, guitar freak, photography enthusiast and highly obsessed with all moustache related stuff.

Tiina Pyörälä

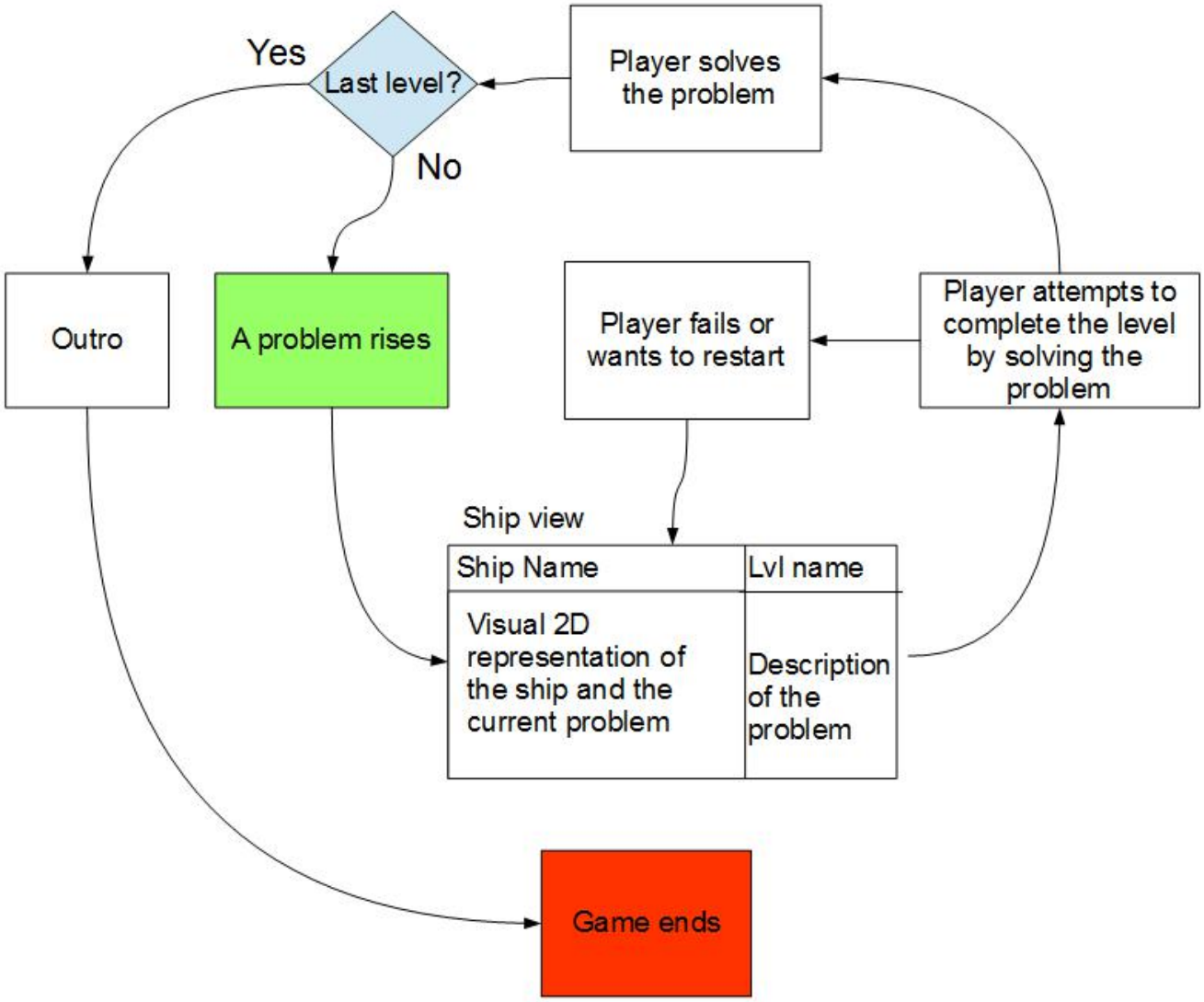
Hiyah! I'm graduate of BBA (Business Administration) degree programme of Business Information Systems. I'm specialized in assets creation, UV mapping and currently I am improving my knowledge in the field of character rigs and animations. I'm a fan of the Zelda series and I love reading books.

Long Hoang:

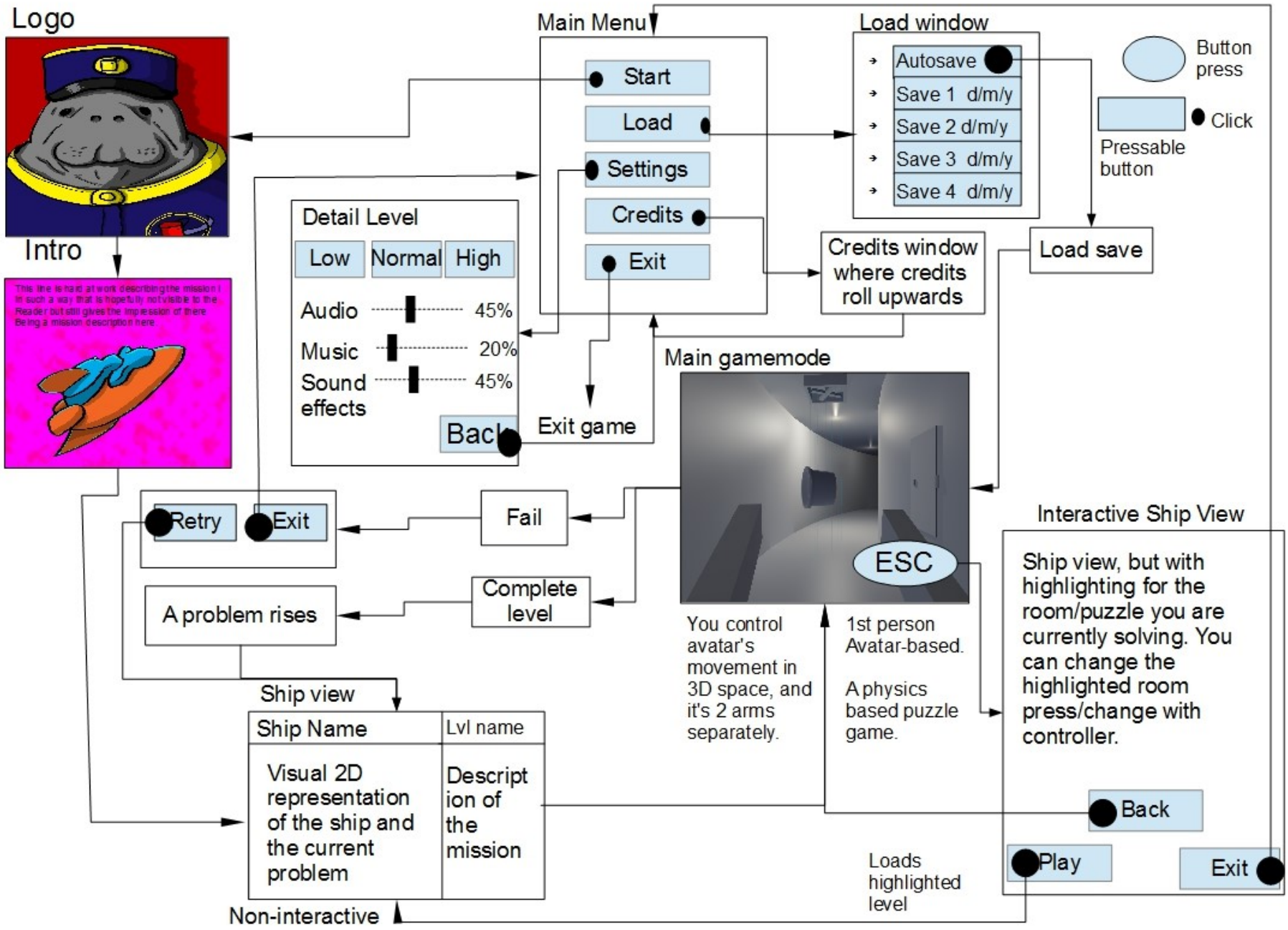
Hello! I'm third year student from Oulu University of Applied Sciences. I start playing games from high school and I want to make a game as a developer when I come to university.

1.1 Game Flow

Core Loop:



Game flowboard is on the next page.


Logo
Intro
This line is hard at work describing the mission I in such a way that is hopefully not visible to the Reader but still gives the impression of there Being a mission description here
Main Menu
Start
Load
Settings
Credits
Exit
Load window
Autosave
Save 1 d/m/y
Save 2 d/m/y
Save 3 d/m/y
Save 4 d/m/y
Button press
Click
Pressable button
Load save
Detail Level
Low
Normal
High
Audio
45%
Music
20%
Sound effects
45%
Back
Exit game
Credits window where credits roll upwards
Main gamemode
ESC
You control avatar's movement in 3D space, and it's 2 arms separately.
1st person Avatar-based.
A physics based puzzle game.
Retry
Exit
Fail
Complete level
A problem rises
Ship view
Ship Name
Lvl name
Visual 2D representation of the ship and the current problem
Description of the mission
Non-interactive
Interactive Ship View
Ship view, but with highlighting for the room/puzzle you are currently solving. You can change the highlighted room press/change with controller.
Back
Play
Exit
Loads highlighted level

.1.2 Target Platform / Minimum Hardware

PC/Consoles//Maybe even VR.

The game should be playable with even a low-end PC, featuring low-polycount models (At least via settings).

The game is recommended to be played with a gamepad/controller, but can be played by other means as well dependent on the system the player is using.

.2. Gameplay

.2.1 Progression

The game progresses in a sort of day by day style. Each day brings on different challenges for the player to complete. The problems range from smaller onesm that don't need to adhere to the puzzle solution philosophy, to larger ones that do need to adhere to the puzzle solution philosophy. At least one larger puzzle must exist in a day.

The game environment stays the same regardless of the day that is currently being completed, it's always the whole ship and basically an open world. Only the problems change day by day. Some solutions that the player comes up with will damage the ship, but it has no bearing on whether the player wins the game or not. The effects of the players damaging solutions exist in later stages as well. For exampe a broken fan will stay broken.

When the player has already completed a day he can replay that day, but this does not change the effect of his solution that he did the first time. This is because the player just probably wants to replay the day for fun, there is no real point in him wanting to change the whole outcome. This can be changed if the feedback contradicts this.

.2.1.1 Winning

The game will always be won in the end, there is no possibility of losing the game completely. Only failure the game has is temporary and will lead the player to restart the day from the start of the last puzzle that was completed. The game is won when all the days are cleared. Storywise this means that the moonatees have reached their destination with their precious cargo of buckets and everyone is happy.

When the game ends, we show what effects the players solutions had on the ship. We show in what condition the ship is in, this can be played in the background while the credits roll. This is to give the player a feeling of having an effect on the journey and to see that his actions did something permanent in the game world.

.2.1 Mechanics

Moving around in zero gravity while using your two hands to manipulate your surroundings. The arms would be controlled separately. Moonatees don't have fingers so the only way to actually grab things that are floating around would be to bring your arms together around the object. The game features physics for every object that is not attached to a surface, every object can be manipulated using your arms by grabbing them, hitting them, or by crashing your torso into them.

You can use the objects by either grabbing them and using them like you would in real life. For example to use a toilet plunger you grab it, and shove it into the toilet. You can also manipulate things by hitting them, so you can for instance press a button or hit something so hard that it flies to where you want it to fly.

Sensitivity values for movement, handling of the arms, how hard a hit affects another object, looking,etc will have to be found out by testing. These values should be public variables in the script so they can be edited straight from the editor.

.2.1.1 Physics

Objects float around in zero gravity. The game uses basic unity physics meaning that the objects interact with each other via colliders. The only way to manipulate the objects therefore is to either hit them with something, grab and throw, or activating something that pushes them along some axis, in this case the wind.

The reason to use the wind as an aid to move objects is that the heavier the object is, the slower the moonatee becomes when he is floating away from walls/floor/roof. Near those things the moonatee moves normally no matter what since he can use his tail as leverage. The item might be so heavy, that the moonatee cannot move by floating at all.

Grabbing:

When both arms hit an object, that object is grabbed. It's position changes according to the players hand movements. The objects rotation stays the same as when it was when grabbed. This needs to be hard-coded as no physics like this exist in Unity.

Throwing/Letting go:

When the player moves his arms in separate directions the object that was being grabbed is let go, the movement energy of the arms at that point is transferred into the object. The direction of the throw is dependant on the movement of the arms, and the position of letting go.

.2.1.2 Item Properties [OUTDATED] (Things have been added in scripting)

Different kinds of items have different kinds of properties. Some of them come straight from the unity engine like **collider size**, rigidbody properties like **mass**, object **position/rotation** etc... But we also plan to have our own set of properties for items:

- **Electricity Based**
    - Can the item conduct electricity? **Boolean Conductivity**
    - Is the item conducting electricity right now? **Enum ConductingStatus Conducting/NotConducting/ConstatlyConducting**
    - How far does the item look for when looking for other conducting items? **Double ConductorColliderSize**
- **Wind Based**
    - Is the item a source of 'Wind'? **Boolean Propeller**
    - Is the item in the area of a wind effect? **Boolean Propelled**

        - NOTE: Only if the effect we are looking for cannot be done with normal physics forces. This can be activated by a trigger area for example.
    - How fast does the wind travel from this object? **Double WindSpeed**
    - Where is the end point of the wind that is traveling from here? **Vector3 EndPoint**
- **General**
    - Can the item destruct or make other items malfunction? **Boolean Destructive**

.2.1.3 Electricity

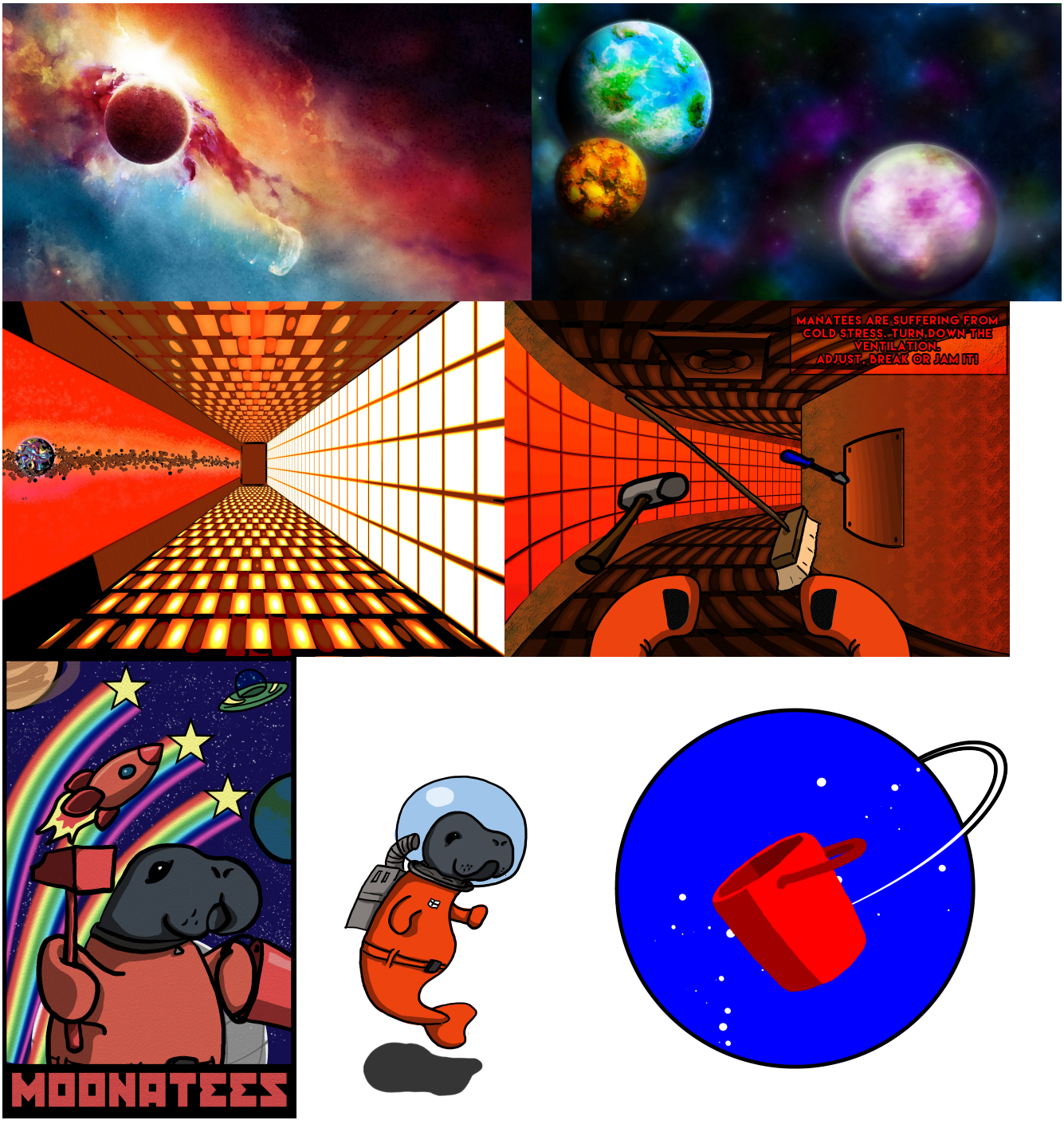
The game has electricity based puzzles that for example work like this: A wall has a generator that is constantly generating electricity (**ConductingStatus ConstantlyConducting**), this item has a range as a double for how far it can project electricity (Reference: Tesla Coil). This range is converted into a **collider** that acts as a **trigger**, it does not have physical propertier or rigidbodies. When two colliders meet we check for item properties that we define in an itemscript. If both objects have **Conductivity** True, we check if either of them is **Conducting/ConstantlyConducting** and if true we form a visual link between these two objects, changing the other objects **ConductingStatus** into **Conducting** as well. The point of this is to lead the electricity somewhere like another generator, that might result in opening a door for example. When the two linked colliders exit each others ranges; the one that is not **ConstantlyConducting** is changed back into **NotConducting.**

.2.1.4 Wind

There are items that work like fans constantly propelling (**Propeller True)** an object that enters their propelling area until it reaches the **EndPoint** that has been defined for that object. This force is constant from the propeller to the **EndPoint** and doesn't weaken with distance from the object. The force is defined by **WindSpeed**. In-case this cannot be achieved by unity physics forces, it can be done by making a collider that extend from the propeller until the **EndPoint** activating a boolean (**Propelled True)** in the itemscript of the object, and getting the values from the fan for the movement behavior.

.3. Visual Style

The visual style of the game would be based on simple cell-shading technology with heavy inspiration drawn from Soviet-style space propaganda and 80's comic books. The game would feature strong cartoon-like colors and high contrast. There could be sprites with texts like “POW”, “PLONK”, etc appearing along with the sound effects when the player does something. This sprite could have a yellow background with red outlines, the text would be with the games own font. One of example game that has a little bit of a similar style is XIII.



.4. Audio Style

Inspiration from “Tron”, other scifi-movies like “Blade Runner” and electronic music such as Daft Punk, ELO (Time, 1981), Muse, Kenny Loggings, Jan Hammer, Massive Attack, Depeche Mode. Influences from genres like Synthwave, spacerock, 80's soundtracks, scifirock, ambient electro.

Sort of expansive sounds that feel like you would be in open space. Slow-paced, but happy.

There should be very few (to none) situations that would fit with fast music, the player should not feel anxious beyond the sometimes frustrating game elements.

Sound effects should be comedic, sort of “plonky” sounds.

The game would have dialogue and grunting moonatees.

.5. Starting Out

.5.1 Game Start – Main Menu

The player would be greeted by a space scenery background with the moonatees doing repairs on the ship. The feeling we want to achieve here is *'Well this looks nice'.* The scenery is such that the player might stare into a little while and get lost in it and the background music. He is also presented with the menu options on how to proceed. The player might first proceed into **Settings** to see if they have been configured correctly automatically, *which we should aim for*, and if not; changes the settings to suit his hardware. Then he proceeds to press **Start** to start playing the game.

.5.2 Game Start and Intro [OUTDATED] (Refer to DAY 1 document)

The player would be greeted by the opening cinematic*(Described in the cinematics part). The player should feel either confused, in a funny way, or that he is laughing at the intro. This was the first time he saw the moonatees.* Then the game cuts into the cockpit, a 3D environment, where a loudspeaker is yelling about the breach in the hull, this is how the player is taken into his first mission. His first mission is to plug the opening left by the now deceased moonatee, how the player does it is up to him.

The player should have a feeling of 'What now?' and look around the cockpit a bit, enjoying what he sees and then be reminded about his mission. The main feeling in this part is wonderment/amazement.

The early part of the game is used to familiarize the player with the mechanics of the game going from the simplest to the hardest. This familiarization would be done one mechanic at the time. First we guide the player on how he should move the moonatee by guiding him to move from the cockpit into the corridors and into the scene of the accident. One of the screens in the cockpit should have the non-interactive ship view(*Detailed in the UI section*) opened and the camera should be aimed at that screen at the start of the game.

The ship’s automatic controls and mechanisms are turned off by the hull breach and the ship flies towards a meteor storm while the player is fixing up the hull breach, this meteor shower causes more problems and/or the player has to try to avoid the meteors using the ships controls.

.6. First steps of development

.6.1 Prototype

This is mostly for programmers, but has some work for 3D artists as well.

We make a prototype of the game that is used for testing out values and the mechanics of the game. We have a simple layout of a starting room with random floating objects to test out the

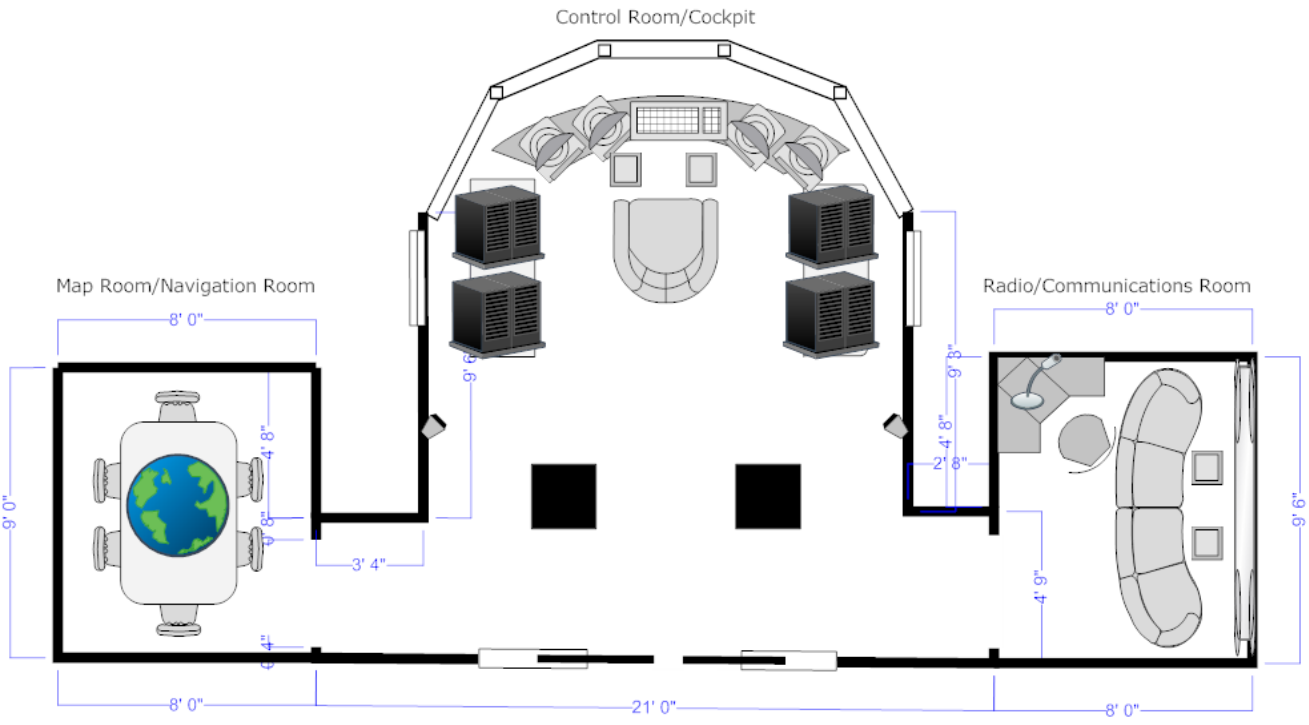
characters collisions with the objects. This room leads into a corridor from which you can enter rooms designed for the sole purpose of testing out a particular mechanic.

1. A recycling area where in the first room you need to separate one particular piece of garbage from the others using wind. The wind apparatus is such that can rotate around it's own axis and around the garbage as well. In the second room the goal is to separate the two different kinds of garbage from each other and succesfully recycle them.
2. Room that is based on a puzzle of transporting electricity. When you enter this room the door closes behind you, on the wall next to the door is an inactive electric generator, on the other side of the room there is an active electric generator. The room has conductive items like buckets, screwdrivers, metal rods, etc that you then form an electric conducting link with, the distance has to be changeable in the editor.
3. Room for testing smashing objects into some specific place. This room has a chest where you have to gather the stuff that has been flown around from that chest.
4. A rotating room. This room is formed like a torus, it spins around it's center point. The room activates it's spinning after the player enters. The player must proceed one way, since the other is blocked. The player must reach the other end of the room to press a button that is also unreachable by the player character itself, but you can hit it with other objects. The way there has some obstacles that impede the player's progress, some might even be fatal obstacles like unprotected fans. The rotating room is meant to cause slight nausea, it might be easier to achieve the feeling of rotation by having a window to the side and some visible objects in space.

NOTE: Come up with rooms when you come up with new mechanics.

.6.2 Other

This is something that the other team members can work on while the prototype is being developed, and if they have nothing to contribute to that.



Control Room. The player will start the game in the captain's control room. The assets it needs are listed in the asset list. The room layout, this can be changed to suit any visions the artists' might have, is on the previous page.

Concept art for the environment/rooms/style of the ship/the ship itself/assets.

Game Menus Making the main menu's background, the buttons' visual style. Making the settings menus'/other menus' backgrounds. Layout concepts can also be helpful. The ship view and it's holographic look in-game can be worked on, especially after the control room is in a Unity scene.

Level Design Come up with more funny puzzles for our rooms. Design them concretely so that they include the possible solutions as well as the steps the player could do in-between the problem and the solution itself. We have a '*philosophy*' for level design, refer to Levels/Puzzles section to find it.

.7. UI

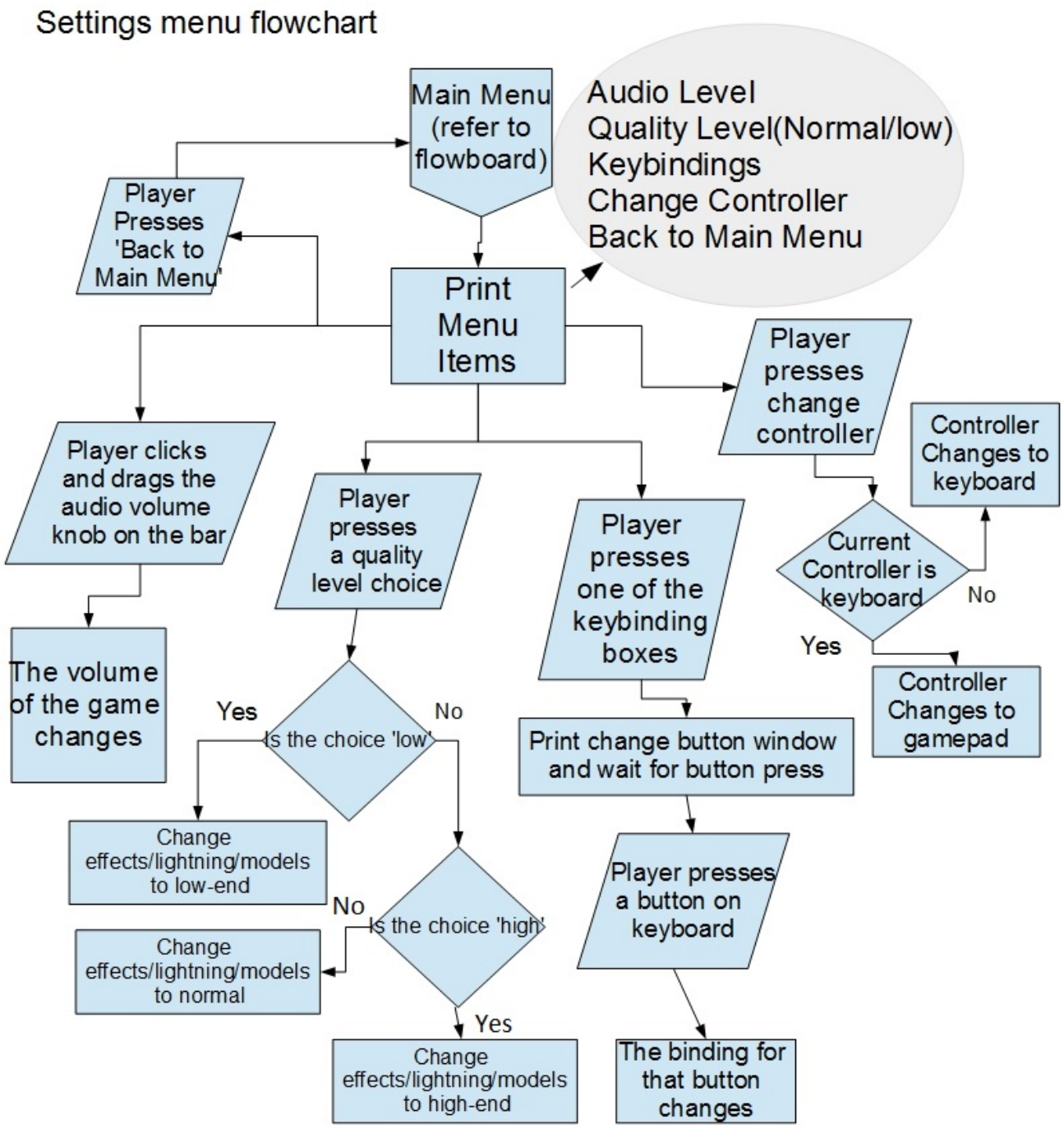
.7.1 Main Menu

Menu would be quite simple but at least the background should be visually beautiful and in accordance with the rest of the games style. *A sample idea was presented previously*. The colors of the menu items would be done with bright and 'happy' colors while being slightly transparent. **On the PC t**he player clicks on the menu items with his mouse, or uses his arrow keys to navigate the menu items while using enter to choose. **On the console** he would use his arrow keys to navigate the menu and A or equivalent key to select his choice.

When the player makes the choice the button should look like it has been pressed, maybe doing a little denture to the sides of the buttons and turning off the transparency, making it solidly colored. **Note: If there is time, we can come up with a more visually appealing way of indicating choice, like perhaps moonatees coming and ferrying the players choice off-screen. Not a priority.**

- Start
    - Where the player would press when he wishes to start the game, this would immediately cut into the starting cutscene
- Load
    - When the player starts the game for the first time, this would be greyed out. This option would be for players who want to continue from the point where they made a save.
- Settings
    - Detailed below in detail.
- Credits
    - This is where we claim all the glory. Just the basic scroll down credits, maybe with funny moonatee versions of ourselves as accompanying pictures.
- Quit

.7.2 Game Settings



.7.3 In-Game HUD & Menus

The HUD would be extremely minimalistic, displaying perhaps even nothing in the main view of the game.

The game's menus would be accessible via the cinema scene. The cinema scene is the first scene of the game which shows the opening cinematics in a 3D environment (*More info in day 1 doc.)*. Later on the player would be able to press 'ESC' to enter the cinema, and the screen has the ship view and the option to see the settings menus and things like that.

Ship View: The main function of this window would be to choose a day, if the player wants to restart or redo a day, and to describe what is currently happening on the ship, the current level that is. The window is separated into **two parts:**

The **map window** that shows the ship in all its glory, and has icons to describe the problems in each of the rooms. For example, a fire would have a fire type icon. This icon could be situated in a torpedo room for example. The days that have not been passed are greyed out and are unelectable. The currently selected room, and thus dag, is highlighted with an outline around the room.

The **description window** that briefly describes the problem. Using the example from above, this window explains why a fire in the torpedo room is extremely dangerous and what does failure to correspond really entail for the ship and its crew.

Below these would be buttons to enter the game settings, save/load menu. These could also be situated in separate screens to the left and right respectively.

A non-interactive version of this view with no-highlighting or greying out is visible on some screens inside the game, for example in the cockpit/control room.

.Highlighting

All the games items have a black outline. We can highlight the items that we want by changing the outlines color [TBD, but a bright shock color that's not widely in use otherwise]. When there is a critical thing that the player needs to do to progress in the game, we can make the outline change between two different colors to a 'blinking' effect.

.Dialogue bubbles

The game's dialogue comes up in bubbles. These will only appear when the player is close enough and is looking at the thing that is supposed to be talking. There should be a dialogue state machine so that the game knows which message to show and when. When the player is looking at the dialogue box, there should be a circular progress bar at the bottom of the speech bubble slowly filling. After that is filled, the dialogue is marked as seen, and won't be shown again.

.7.4 Game Over Screen

Lost: Have a moonatee grunting noises to the tune of 'SNAKE? SNAKE? SNAAAAAAAKE!' from the Metal Gear series of games. The player would have an option to 'Restart' or to 'Quit'.

If the player fails violently (For example; flies into a fan) play sound effects like "Wah Wah

Wah Waaah" with perhaps some snide remarks like "so sad"

Won: The game finishes with an outro of the moonatees arriving to the colony where everyone is glad to finally receive their precious buckets. Then we roll the credits. Described in detail in the cinematics session.

.7.5 Level Selection [OUTDATED] (I need to update)

This is done via the interactive ship window that can be accessed in-game by pressing 'ESC'/start on the gamepad, where every level that is not open yet(Not passed, not in progress) is greyed out at first. The levels that have been passed or are in progress have been colored. More on this window in section 7.3

Levels are shown as the rooms of the ship in the ship view. Level equals a day in the game. The one that is currently in progress has a icon or visuals indicating what sort of disaster is in progress. The player can choose another level or the current one by pressing a room with the mouse, or changing the highlighted room with the gamepad's stick/arrows. The description window changes to match the highlighted rooms level details. The player can then press 'play' to load that level.

.7.6 In-Game Settings

In the ship view there is a button to access the settings, these settings are identical to the settings accessed from the main menu. Refer to 7.2

.8. Controls

.8.1 Mouse and Keyboard

The player would have the option to change his keybindings.

.8.1.1 Default Bindings

- W to move forward
- A to move to the left
- S to move backwards
- D to move to the right
- Mouse to look around, except for when the player presses mouse button 1 or mouse button 2
- When the player presses mouse button 1, he would move the left arm around by moving his mouse
- When the player presses mouse button 2, he would move the right arm around by moving his mouse.

- When the player presses both mouse buttons down, the left arm moves normally, but the right arm would have the right/left axis inversed. For example, the player moves his mouse to the right while pressing both buttons bringing both arms closer to each other until a collision is noticed.

.8.1.2 Alternative

Note: Sami's idea of controls here

.8.2 Gamepad

.8.2.1 Default

- Left Stick to move around
- Right Stick to look around
- When left trigger is pressed down, the player can move his left arm around by using the left stick
- When right trigger is pressed down, the player can move his right arm aroubnd by using the right stick.

.8.2.2 Alternative

Note: Sami's control idea here.

.9. Levels/Puzzles

.9.1 Philosophy

We have a '*Philosophy'* for level design, it means that every level has to have these kinds of solutions:

By-the-book: This is what the engineer/no-fun type of player could come up with. The sort of obvious solution. A fan is causing problems by making a room too cold? Turn it off using the electric switch ofcourse...

Unorthodox: This is what a player might do if he is the curious type of player, it's right to do this, but it's not obvious. It's something that feels a little bit wrong but causes no damage, this is the solution that makes the player go '*HUH!? THAT WORKED WTF!?'* For our previous example this could include stuff like making small controlled fires in the room, changing the room into a fridge, etc.

Wrong, but works: I would call this the lazy solution for the instant gratification type player. It's something the player can do and it does solve the current problem, but it probably does also cause problems further down the line. In our previous example it would be breaking the fan in any way, like sticking a broom into it. The player's feelings here should be of enjoyment in making chaos.

In addition to this we need to keep in mind the chaotic elements of the game, the puzzles should be funny in *some* way, and there needs to be a possibility for unexpected events.

For added chaos/difficulty/funniness a level can have a;

Chaos Factor: An added difficulty factor for the level, for instance sudden bursts of wind, sudden explosions, sudden emergence of objects. Rotating room stops for a little while and then rotates double the speed for a little while, etc.

.9.2 Rooms in the ship:

- Control Room
  - Cockpit
  - Radio room
  - Navigation/map room
- Bedrooms
  - A corridor in the middle, but both sides of the room have a small pool filled with water with either floating airmattresses for beds or some underwater vegetation/ice.
- Storages
- Hold
- Engine Room
- Escape pods
- Medical room
- Maintenance corridors
- Hyper-sleep chamber
- Generator room
- The rotating rooms around the generator (These double up as the thing that actually generates the electricity, so they interact with the generator)
- Ventilation shafts, the most important one is the huge one somewhere near the middle of the ship
- A possible Torpedo Room
- Canteen
  - Kitchen
  - Food Storage
  - Eating area
  - Entertainment area
- Bathroom(s)
- Break room with a table tennis table and pads and other stuff

.9.3 Puzzle types

NOTE: Of course a puzzle might be a combination of all these, but still at least one level for just one type of puzzle should be a good idea so that we can familiarize the player with types of problems.

- Transferring electricity from one conduit to another
- Plugging holes
- Unplugging toilets
- Repairing ventilation etc.
- Wind based puzzles (**NOTE: This will most likely be the artificial gravity thing as well**)
- 'How to activate that button' type puzzles.
- Rotating rooms can be used for multiple types of puzzles like
  - The player needs to go through a confusing set of rotating rooms
  - The player needs to hit something in the rotating room, to make it harder the player cannot reach it by himself.
- The player needs to hit an object, guiding it somewhere, with a thrown object.
- Control room 'avoid the meteors' puzzle.
- **Others as well. Included in the days' or rooms' documents**

.9.4 Full level list [OUTDATED] (Just refer to room and day documents)

- Generator and it's surrounding rotating rooms.
- The ventilation shaft with it's wind puzzles
- The control room where communication with other races comes into play as well as avoiding space debris. This location can be used for certain to see where the problems are.
  - Avoiding space debris/asteroids
- Torpedo room with a fire.

.10. Assets [OUTDATED] (Up-to-date version in google drive.)

Asset List/Rooms	Control Room	Ventilation Shaft	Prototype	Puzzle Scene					**Animated**	**Concept Done**	**3D Done**	**Animation Done**
Captain's Seat	X									X		
Board of Controls	X								X			
Pillar with a hologram on top	X								X			

Loudspeaker	X					X			
Computers	X								
Screens	X								
Bucket		X		X			X	X	
Screwdriver		X		X			X	X	
Wall-Embedded Fan		X	X	X		X	X	X	
Huge Fan		X				X	X		
Broom		X		X			X		
Toilet		X							
Moonatee		X				X	X	X	
Moonatee flippers	X	X	X	X		X	X	X	X
Wind effect		X	X	X					X
Maintenance Walkway		X						X	
Electricity Effect			X	X					X
Chest with a separated lid.			X				X		
Generator			X	X			X		
Lever			X			X			

Assets not yet assigned to any particular scene:

- Toilet plunger, hammer, etc tools
- A visual representation of the ship for the **Ship Window**.
- Icons for: Fire, Hull breach, Electric failure, etc.
- The Ship from the outside
- Airmattresses

.10.1 Equipment and Tools

Casual tools and objects like hammers, screw drivers, wrenches, tape, glue, brooms which the player is able to use to solve different puzzles.

.10.2 Environment [Slightly OUTDATED]

Tidy spaceship, since it's originally designed for humans. For an over 500kg moonatee the rooms and corridors are somewhat small to operate comfortable. For their own comfort the moonatees have made some customizations.

The ship is roughly divided into 3 different areas:

1. Top level: Maintenance area with the ventilation shaft, electric wirings, etc. This area is in good condition but slightly messy.
2. Moonatee living area, this is the main area of the ship and includes all the rooms that the moonatees normally use when operating a spaceship. It's a well kept area, but there are signs of life.
3. Bottom level: This area has rooms that are rarely used. Rooms like storages and the torpedo room. This area has quite a lot of wear'n'tear since it's been mostly abandoned after the moonatees bought the ship. There's more rust and dirt here.

.10.4 Audio

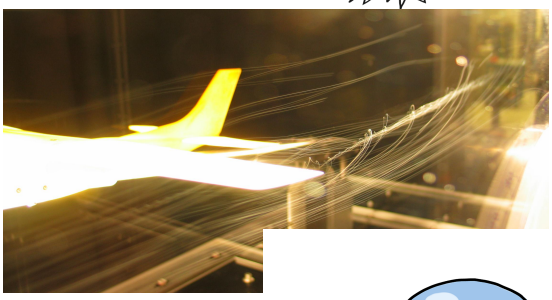
Music at least for menu, intro and ending credits, ambient noises and sounds, characters speaking, screaming grunting, etc.

.10.5 Effects

Sound effects showing up as cartoon sprites.

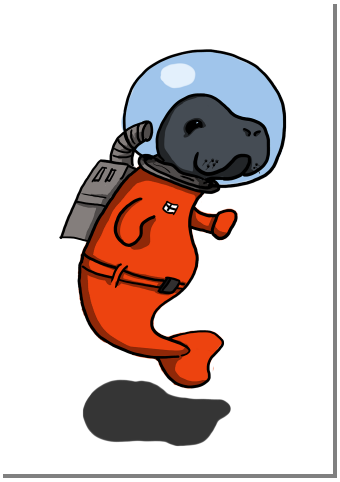
Wind having a visual effect as well.

Electricity effect between two conducting objects.



.11. Characters

Moonatees



Manatees are large, fully aquatic, mostly herbivorous marine mammals sometimes known as sea cows. Moonatees share same features as their sea living cousins, in other words they are 4 meters tall and weight over 500 kilograms, but main difference is that they live… in space. Moonatees are kind giants, usually working as delivery boys, since they are able to carry heavy cargo and don’t own really bad habits beside their clumsiness and simple minds, unless cargo is sea weed which is their main source of nutrients.

Humans

Usually source of all despair and accidents facing poor moonatees, since moonatees don’t have any natural enemies they are only left to contend with the space waste left behind by the selfish humans.

Other species?

Examples include Navy Seals

.12. Story

This is a lighthearted story about Moonatees in a space ship, trying to reach their destination so they can deliver their cargo, which is buckets. So far there is the beginning of the story, described in the intro, and a general idea of the ending which is the moonatees reaching their destination while everyone is happy.

.12.1 Cinematics

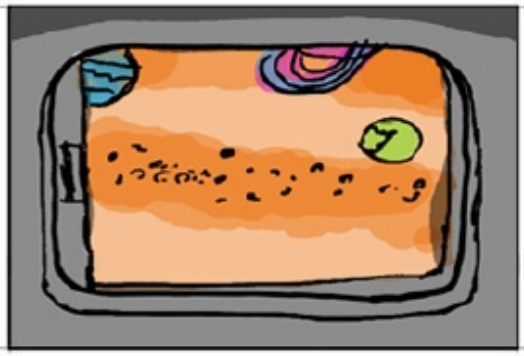
.12.1.1 Intro [OUTDATED]

Moonatee captain slowly approaching the camera, as he reaches the “peak” some inspirational jingle and the captain lifting his arm up, his crew cheers and the captain shakes. Show the logo of the game above the captain, then cut to the ship traveling in space and describe the mission either by speech or very short text. Cut to a moonatee crew member operating a switch, that controls the anti-gravity’s direction and power status. He then flies towards the wall making a hole in it. Cut back to the captain, show a loudspeaker behind the captain doing an announcement of a hull breach somewhere, show debris flying by the captains window along with the unfortunate moonatee crew member. It's time to save the ship once again!

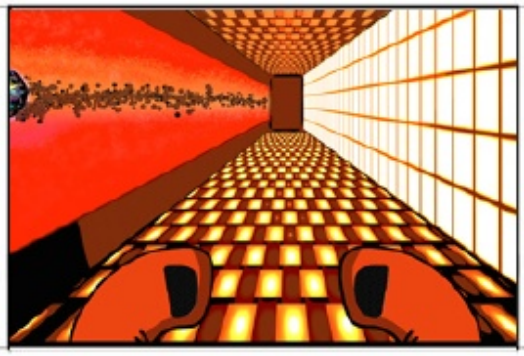
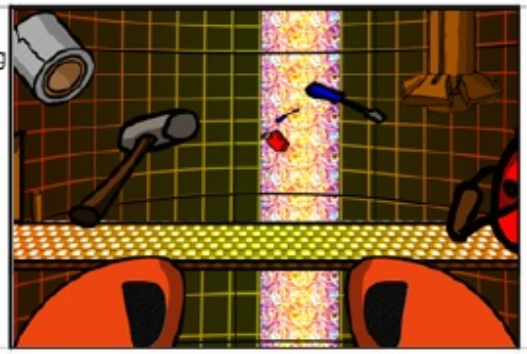
.12.1.2 Teaser

Storyboard begins on the next page.

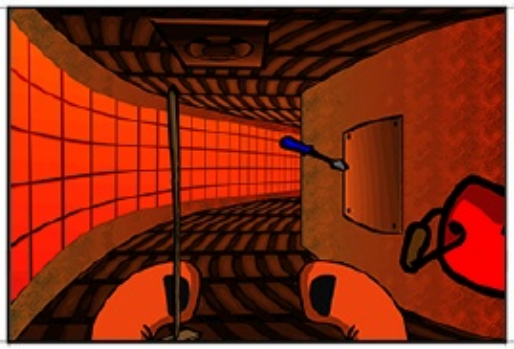
BOOK Moonatees teaser CHAPTER PAGE 1

DIALOGUE	FRAME
Blank black screen	
Windowview of empty but colourful space. Some objects like planets, junk and asteroids flying through the space. Childish music starts playing. Slow, maybe classical? Music box?	
Show a moonatee floating past the window looking quite helpless and afraid with debris around him. Bucket, broom, tools, teddy bear for example.	

BOOK Moonatees teaser CHAPTER PAGE 2

DIALOGUE	FRAME
Bright background and moonatees logo shows up with some fancy text animations. Lens flare?	
Show first person view of a moonatee going around the ships corridors. Uplifting and childish even chaotic music playing background	
Show the big ventilation shaft with stuff like tools and etc flying around with air current made by the fans. Manatee trying to pick up objects, but in the end is only able to push them beyond his reach.	

BOOK Moonatees teaser CHAPTER _____ PAGE 3

DIALOGUE	FRAME
Moonatee solving ventilation puzzle. First given simple instructions and for a moment moonatee just stands looking around for different flying objects.	
Moonatee grabs a broom and looks thoughtfully at the broom and ventilation system.	
Moonatee shoves broom up in the ventilation/fan causing huge GASHUNKI sound and comic like sound effect sprite popping up. It's broken, but moonatee just grunts approvingly. Music stops and screen fades to black.