

# BARI- STARSHIP

**GAMEDESIGN WORKSHOP**

**Projekt: Baristarship**

Teamleiter - Bastian Culig [260396]

Programmierer - Valentin Schmidberger [263249]

Programmierer - Marius Kögel [262607]

Sound Design - Viktor Lukanow [263613]

# INHALTSVERZEICHNIS

<b>CORE GAMEPLAY</b>	<b>1</b>
<b>GAME FLOW</b>	<b>1</b>
<b>TARGET AUDIENCE</b>	<b>2</b>
<b>INTERFACE</b>	<b>3</b>
Section Selection Menu and Story Integration	3
Permanent Overlay	5
<b>MINIGAME DESIGN</b>	<b>6</b>
Coffee	6
Med Station	7
Cargo	8
Engineering	9
Hangar	10
Shields	11
Weapons	12
Sensorscan	13
Navigation	14
<b>VISUAL DESIGN</b>	<b>15</b>
<b>STORY</b>	<b>16</b>
Story Scenarios	17
Characters	20
<b>SOUND DESIGN</b>	<b>21</b>
Music Theme	21
Ambience Sound	22
Sound Effects	23
<b>PROGRAMMING AND PROTOTYPING</b>	<b>26</b>
Class Diagrams	27
Activity Diagrams	26
Prototype development	29

# CORE GAMEPLAY

The Player Character is located on a spaceship called the ISS-Turnstone. The crew of said Spaceship has caught a virus, which slowly reduces their intelligence. Your task is to keep the ship together by performing various functions, which are normally done by the crew. As the crew gets dumber, you must take on more and more tasks, while also balancing your original purpose, being that of a coffee machine. Failing to do so either causes the ship to be destroyed or causes you being damaged due to the frustration of the ever-stupefying humans.

# GAME FLOW

The game starts of simple with your only duty being that of providing coffee to the crew. You quickly gain control over other easy tasks, such as managing cargo. As more tasks accumulate, various interactions between minigames arise. One such example would be the use of the ships weapon systems, which require ammunition stored in the cargo space, or managing your ships energy output while you fly through an asteroid belt, balancing cutting power to avoid detection by an enemy radar and allocating power to shields to deflect small asteroids.

With additional systems under your control comes additional ways to solve various problems. Such as the above-mentioned asteroid belt scenario. This could be completed either by avoiding detection by an enemy ship or entering a combat scenario for a quicker, although be it more dangerous, result.

Since you must also manage your duty as a coffee machine, in addition to an ever-increasing control over the other systems of the ship, what once was a mundane task now becomes a balancing act between keeping the ship together and satisfying the crews demand for coffee. Increasing in difficulty as the humans become more and more impatient, slurring their words or using different phrases to mean the same type of coffee.

Various upgrade paths help with automatization of different subsystems, such as an automated "Friend or Foe" detection system that allows your flak canons to automatically destroy asteroids and incoming missiles or allow the player to set key phrases which automatically trigger the dispensing of pre-selected coffee types upon said words being heard. Another way of reducing the stress of switching between minigames is to allow limited functions of some minigames to be executed in other minigames. One such example would be the ability to change the direction of shields while controlling your weapon systems or being able to emit Scan pulses while being in other minigames.

This results in an increasing difficulty curve, while also avoiding overwhelming the player.

# TARGET AUDIENCE

This game is addressed to a various type of players no matter the age.

In general they like humorous, unpredictable games where the player can choose his path through the game on his own while leveling and therefore getting more skills and opportunities to master the levels.

As the game is designed to have different mini games, it'll be interesting for all kind of casual gamers. In particular the audience with the most interests for this game will be those who are very keen on space-thematic settings, who are more likely to be male. They prefer their actions to have big impacts on the outcome and affect the behavior in other scenes. It is important that the universe has consistent rules even though it may not be easy to comprehend them in the beginning. For example it's not clear what kind of character the player is navigating at first sight. This spurs the curiosity of the player and he will continue playing to discover more.

# INTERFACE

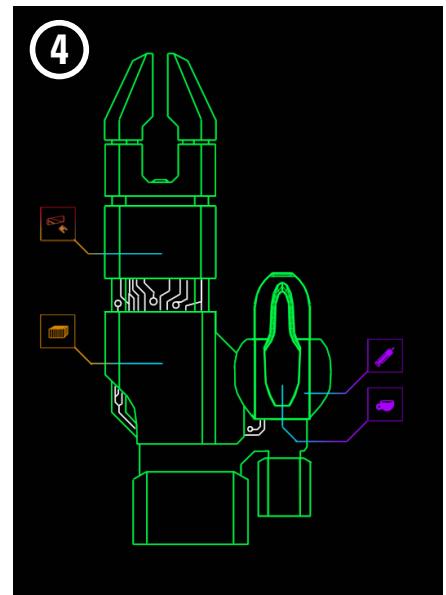
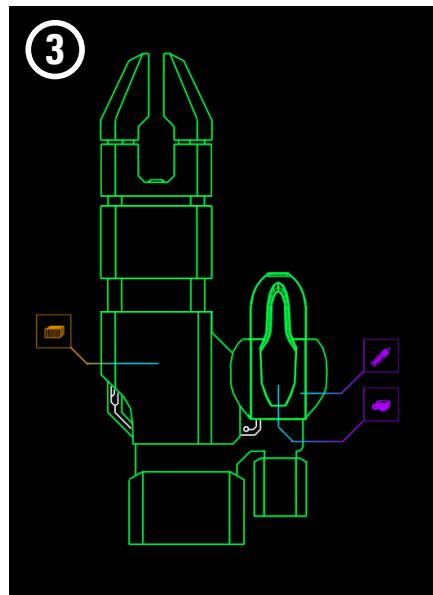
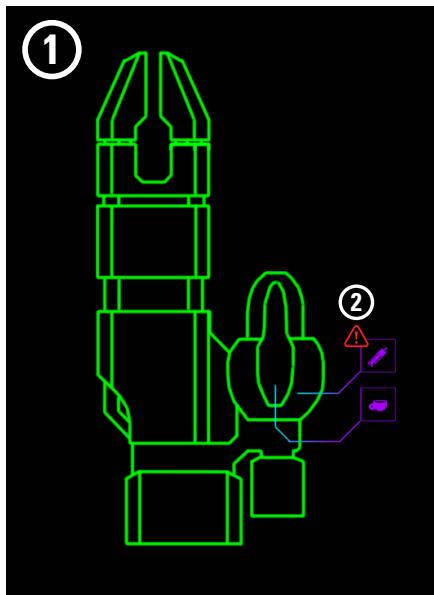
## Section Selection Menu and Story Integration

This is the main menu, which the player interacts with. In this menu the player can switch to various minigames to manage the ships systems. It additionally acts as a form of visual progression, because as the player unlocks more of the ships systems, the display changes and includes more and more detail of the ship in addition to traces, visually similar to ones seen on motherboards, appear to communicate to the player, that the player character [The coffee machine] becomes more and more connected to the ships systems. Finally: These traces also serve a gameplay purpose, by lighting up as certain interactions occur. For example: If the Coffee minigame is being supplied with new wares from the Cargo minigame, then the traces connecting the two will be highlighted with light pulses going through. Signaling the time required for the interaction to complete.

The progression of the interface corresponds with the order in which the systems are unlocked.

The Crew related systems (Med Station and Coffee) show the ship in its simple view. (1) At this point in the story, the Virus is slowly starting to take a toll on the crew, but your services are not required so far. Here the concept of attention warnings is being introduced, as the icons corresponding to the sections will blink if they require urgent attention. (2)

The Cargo and Hangar systems show an interconnectivity of various ship systems. (3) (4) You are required to solve some scenarios, but the situation is still under control. Here the ship takes on a more detailed view. The player is introduced to the first cases of interactions between systems.



A gross error by the crew alerts the central AI. In terms of story, the C.A.I. is now being separated from various systems and various other important systems are being compromised.

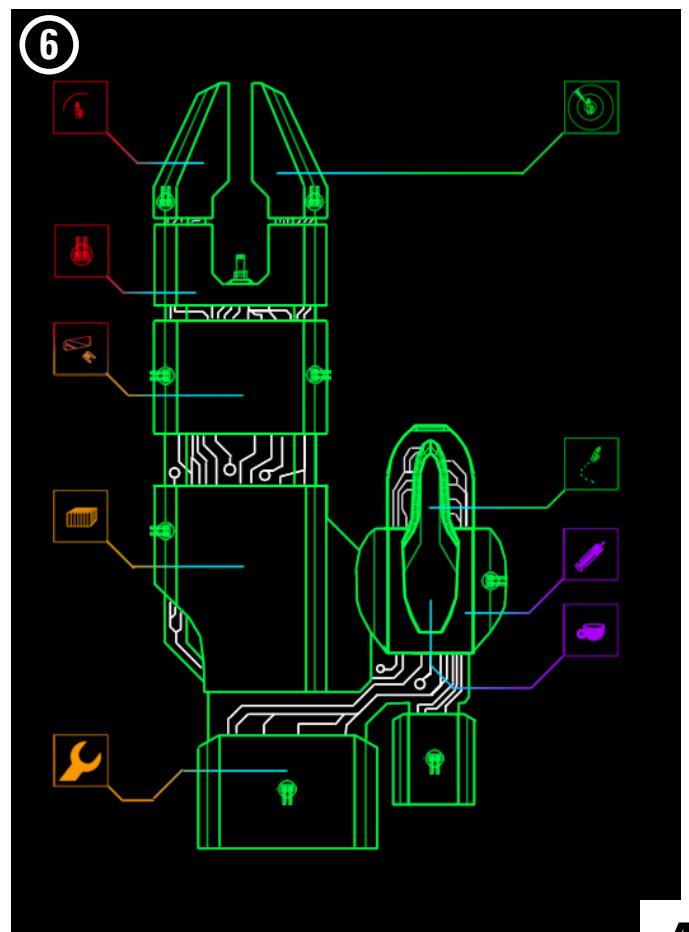
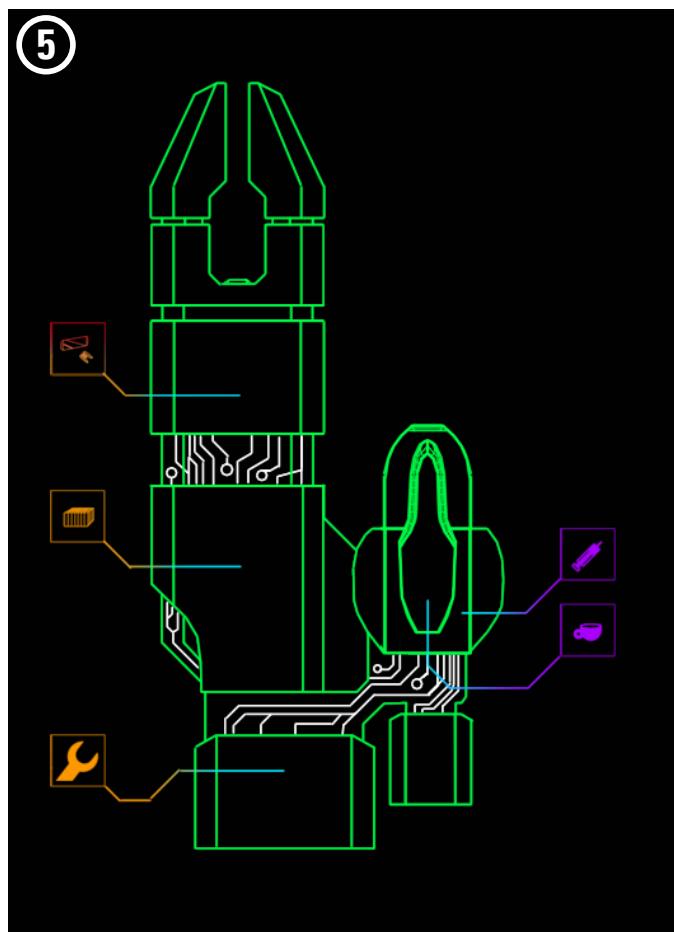
In gameplay terms: Repairs are required as of now, which are highlighted with a red warning symbol and you are being tasked with repairing the engines and managing the energy reserves, as well as basic navigation. Navigation and Engineering are unlocked and the core network is established. (5)

Logistical minigames are all unlocked now and the player now has to manage Crew (Purple) and Logistics (Orange) completely, while being introduced to the first instances of combat (Through sending out fighter drones in the Hangar minigame) and interactions with the world outside of the ship (Navigation). Through a salvaging scenario, the player is now introduced to the combat systems (Red). Now the interface shows surface details of the ship like canons. (6)

Finally, the Shield and Radar systems are unlocked. Both show a constant interaction with the outside. Shields show the current state of the shield (direction and size) as well as impacts on the shield. The Radar shows pulses being sent out. The player can now zoom out to see more of his environment on the Section Selection Menu (SSM).

As all the systems are unlocked, the SSM now both acts as a selection screen and a general overview on the current state of most systems to a small degree.

This results in an increasing difficulty curve, while also avoiding overwhelming the player.



## Permanent Overlay

After unlocking the Engineering minigame, the Energy Management Menu permanently appears in the top left. Through this menu, the player can adjust the energy allocated to various systems. (7)

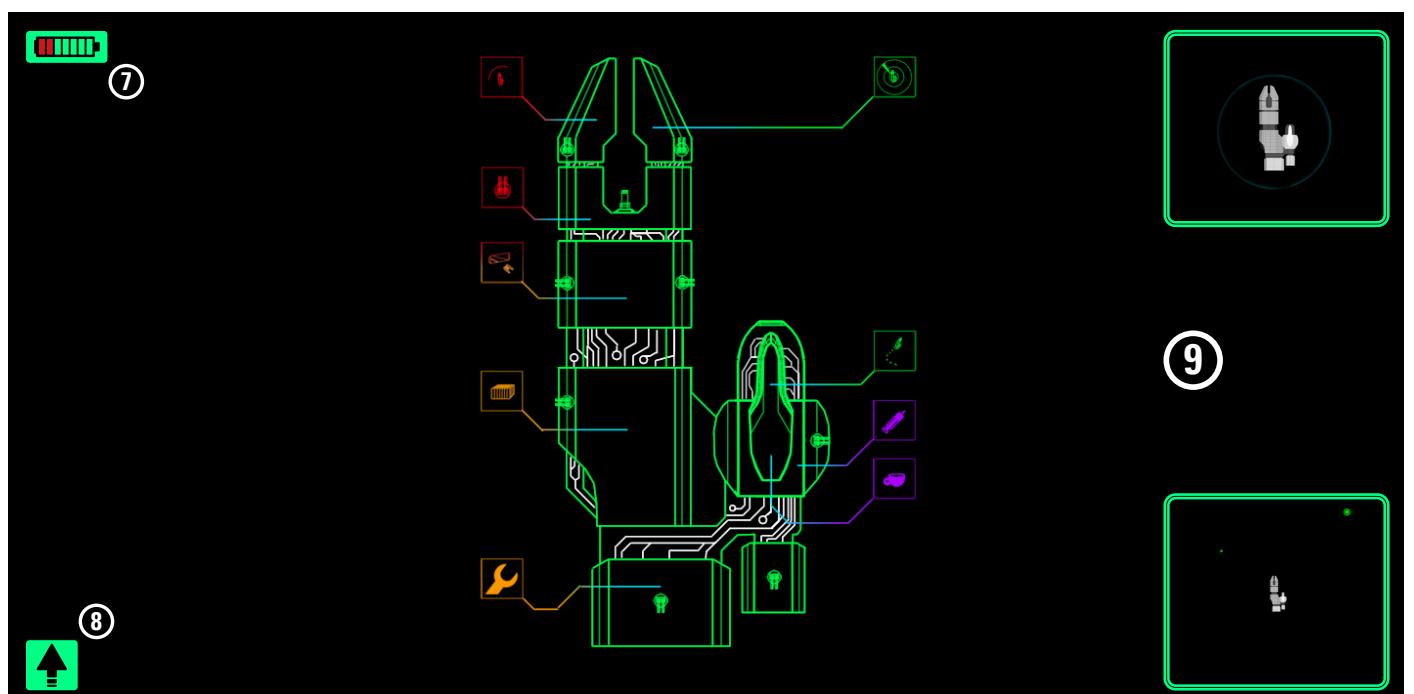
Upon every minigame being unlocked, the bottom left features a small button (8) which opens the Upgrade Menu.

## Interface Upgrade - Simultaneous Multithreading Extension

This upgrade allows the player to quickly switch between two modes and even observe a secondary mode in a simplified form. After obtaining this upgrade, the interface is extended with an additional screen, which displays the last minigame which was selected from the global interface. (9)

Example: The player is currently rearranging the shields, when an alert for incoming enemy fire appears. The player blocks the shots and switches to the weapon systems by selecting it from the ship section selection menu. This action saves the previously selected "Shields" minigame and displays it in a separate screen in the bottom right. The player now opens fire on the enemy ship and notices another enemy shooting at him and that the shields are not in the position to block it. Instead of going to the ship section selection menu, the player now clicks on the smaller screen which displays the shields and the two screens switch. The smaller screen now displays "Weapons" and the main screen now displays "Shields", allowing the player to quickly switch between the two minigames.

Alternatively, the minigame which is displayed in the smaller screen can be assigned by dragging and dropping the icon of the specific section corresponding to the minigame onto the small screen in the ships "Section Selection Menu". This upgrade may be selected again for an additional screen at a higher cost.



# MINIGAME DESIGN

## COFFEE:

Type: Bar Management. Perspective: First Person

In this mode, you take in orders from the crew. As the order is being recorded, you get a profile of the crewmember's previous preferences to aid you. In order to fulfil the request, you are required to provide the requested coffee within tolerances. As the illness spreads through the crew, the requests become more and more incoherent, and the crew becomes less tolerant of variations.

### Keybindings:

- Left click: Adjust mixture. You can adjust how much milk is being added to the coffee, the grinding degree, the temperature and special variants based on ingredients such as ice-cream (Affogato), water (Americano), Hot Chocolate (Mocha) or Whiskey (Irish)
- Spacebar: Pour coffee.

### Upgrades:

Preference selection algorithm

The profile displayed for the specific crew member also provides a button. Pressing this button will adjust the settings to match the previous preference.

Remote adjustment interface

Allows the player to complete basic orders while in another minigame. This interface will display the Order and the basic settings. Special variants will not be doable through this interface and the preference profile will not be displayed.

### Interactions with other systems:

Cargo: When supplies run low, you can transfer new supplies from cargo to the cafeteria.

## **MED STATION:**

Type: Drag and Drop. Perspective: Fixed Camera.

In this mode, the player must detect injuries or illnesses of crew members. To assist the player, a copy of a previous health check up to compare.

### **Key bindings:**

- left click: Select a piece of medical equipment to use on the patient. This can range from injectible medicine like antibiotics or other pills, to surgical equipment for wounds.

Hovering over a section and releasing the left mouse button causes a drop-down menu to appear with specific actions for each piece of equipment.

### **Upgrades:**

Patient abnormality detection algorythm:

This upgrade causes differences in patients from the previous health checkup to be highlighted.

### **Interaction with other systems:**

Shields: Damage to the ship causes an increase in patient count.

Cargo: Medical equipment can be transferred from Cargo.

# CARGO:

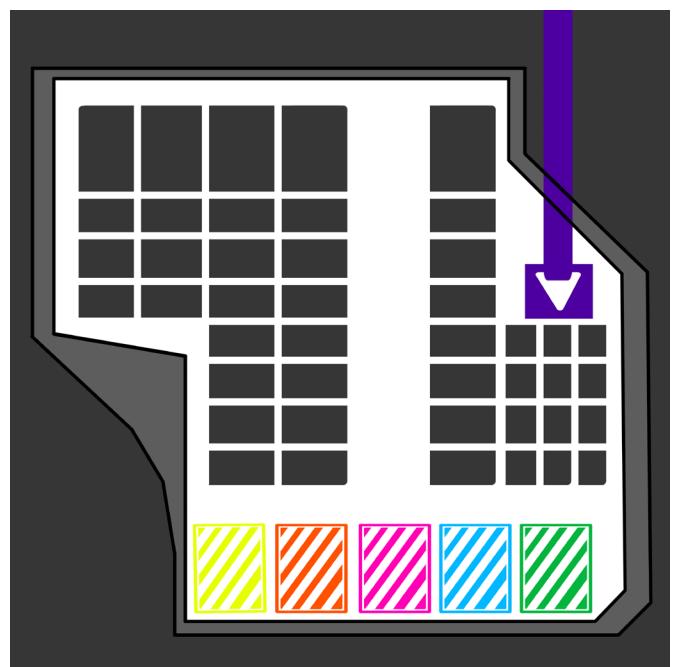
Type: Crane-game. Perspective: Fixed Camera.

In this mode, the player has to manage the ship's cargo-space using a crane. The player can pick up and drop containers of varying size, managing the cargo space in such a way as to quickly react to demands of various sections. The contents of the containers are listed in a menu which can be accessed by pressing shift. Each time the player scans a cargo container which is being transferred from the hangar, it gets an entry into the cargo list.

This simple mode is introduced early on.

## Key bindings:

- W, A, S, D: Move crane
- Space bar: Pick up / drop cargo container
- Shift: Access container list



## Upgrades:

Section management software:

after scanning a container, this upgrade automatically categorises the container based on its content. Scanned containers will show up in the colour of the corresponding section.

## Interaction with other systems:

Hangar, Med station, Repair, Coffee, Weapons, Navigation: Freight can be transferred to these sections to keep them operational.

Hangar: Containers can be transferred from freighters which have scavenged the area, providing additional wares to be sorted in the cargo space.

# **ENGINEERING:**

Type: Drag and Drop. Perspective: Fixed Camera.

In this mode, the player must repair other sections, by performing specific tasks (one can take the game „Among Us“ as an example of how elaborate these tasks are). The damage can either be caused by the crew or external factors such as asteroids or enemy ships. In addition: the energy management also falls under this category and is unlocked simultaneously.

## **Key bindings:**

- left click: Drag and drop cables, press buttons to reset sensor arrays, slide shell outlets to the side to unstuck weapons or turn a dial to align shield frequency with a base frequency. Every task is different but extremely simplistic.

## **Upgrades:**

Instability detection drones:

This upgrade causes red pulses to move along the traces in the Section Selection Menu, signaling a section is in need of repair and is malfunctioning.

## **Interaction with other systems:**

Other Sections: Restores original state.

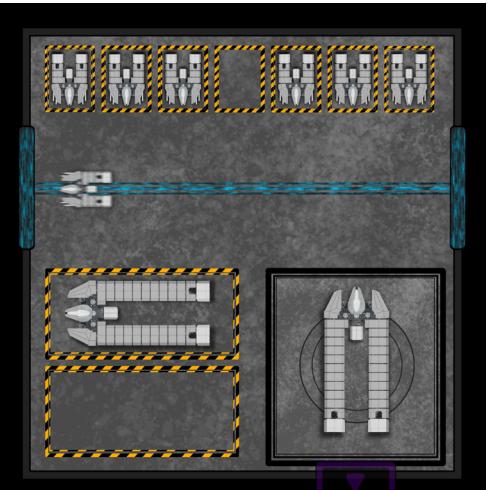
## HANGAR:

Type: Drag and Drop. Perspective: Fixed Camera.

In this mode, the player manages fighter jets and transports. The player can drag individual fighters onto the deployment rail for a quick launch or assign multiple fighters or even a freighter to a wing. Within a wing: All assigned ships will act as one unit. The maximum deployment time of a wing will always be the same as the shortest deployment time of a ship in the wing. While coordinated wings cause freighters to be less likely to be damaged by enemy craft, it also reduces the combat effectiveness of fighters. A pure fighter wing will also be less likely to suffer losses, but this comes at the cost of reduced deployment time.

Fighters will seek and destroy enemy fighters, while freighters will collect cargo pods in the area. Freighters on their own will be able to be deployed for much longer than fighters.

The player can also load cargo into freighters to accomplish various missions. For example: A broken down station still has a functioning navigation relay. Loading repair robots into a freighter and deploying it will repair the reactor core for long enough, that you can interact with the station and get information of the surrounding sector.



### Key bindings:

- left click: Drag ships onto the rail to launch them or drag them off the landing pad to refuel in their corresponding bay.
- right click: Select an action from a drop-down menu. Fighters: Assign fighters to a wing / Refuel weapon energy cells / Repair fighter. Freighters: Assign freighter to wing / Load or unload cargo / Repair Freighter.

### Upgrades:

Management Assistance Software:

Fighters which are being placed in the correct bay will automatically be repaired and refuelled. Freighters will be emptied automatically when being placed in the correct bay.

### Interaction with other modes:

Energy Control: Additional energy allocated to the hangar causes ships to be refuelled faster. In addition: Fighters will have an extended deployment time with additional energy.

Cargo: Freight can be exchanged between hangar and Cargo.

# **SENSORSCAN:**

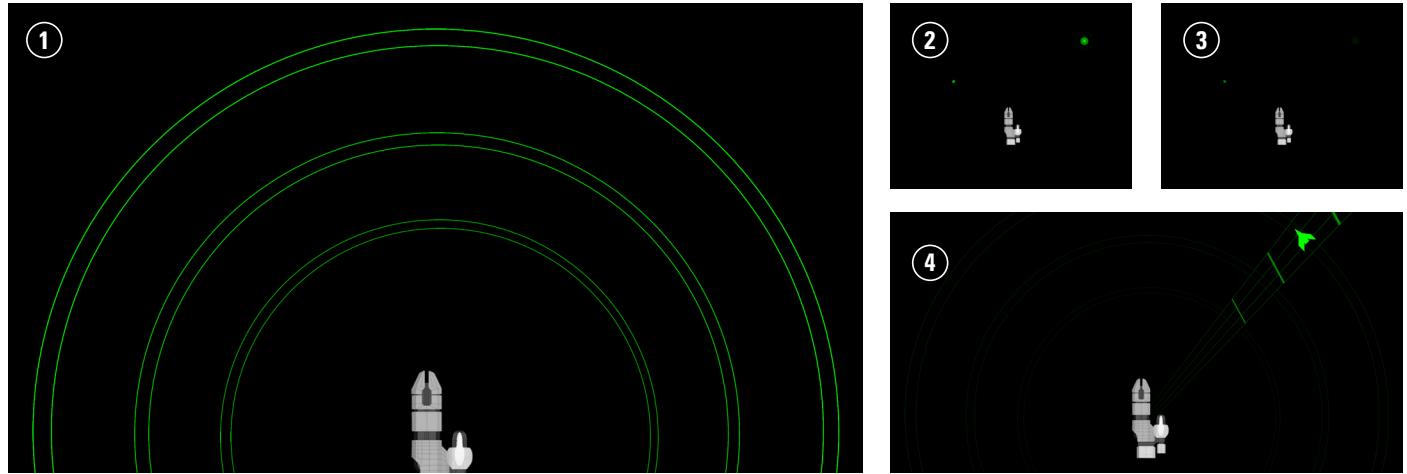
Type: Radar. Perspective: Top-Down.

You can emit scan pulses to determine the rough location of objects around your ship (1). While these pulses cover a wide area, they can not distinguish between asteroids or other objects and only show these radar reflections as ever fading dots (2). Objects further away will disappear quicker (3). Additionally: Objects that have an energy source such as active ships and rockets will also last longer, helping distinguish between these and asteroids. To determine the type of object, you can use a detail scan (4). This type of scan only covers a small area but allows you to determine whether an object is a ship or not. Additionally, fully scanning an object with a detail scan causes them to be displayed permanently. The duration of the scanning process is dependent on the objects size. Bigger objects take longer to scan.

Time progresses slower than normal while you operate the ships sensors.

## **Keybindings:**

- Spacebar: Sensorburst. Upon release you send out a sensor pulse. The range of the pulse is determined by how long you held down the spacebar.
- Left click: Detail scan. Scans an area around your cursor. You have to hold down the left mouse button for the duration of the scan.



## **Upgrades:**

Scan automatization:

Causes your ship to periodically emit a minimum range "Sensorburst".

Filter algorithm:

your "Sensorburst" now shows asteroids and objects with no power source in a different colour.

## **Interactions with other systems:**

Reactor management: Increasing the power allocated to your scanners will increase the minimum range of "Sensorburst" and reduces the time required to fully scan an object using a detail scan.

Shield / Weapons / Navigation: Scanned objects will appear in these modes. Note: Objects will always be visible if they are very close.

Repair: If your scanners are damaged, your "Sensorburst" will show phantom objects until repairs are done. These phantoms will disappear upon being scanned by your detail scan.

# SHIELDS:

Type: Catch Game. Perspective: Top-Down.

In this mode your task is to protect the ship from approaching objects. The direction of the shield is determined by the position of your cursor (1), while the size (2) and strength (3) are controlled with Q and E. A smaller shield is stronger than a wider one. Different objects require different shield strengths and failure to produce a strong enough shield causes it to collapse. A collapsed shield stays offline for a few seconds leaving the ship unprotected. The direction and shield strength are saved upon switching to another minigame.

Impacts on the ship cause damage to the specific sections.

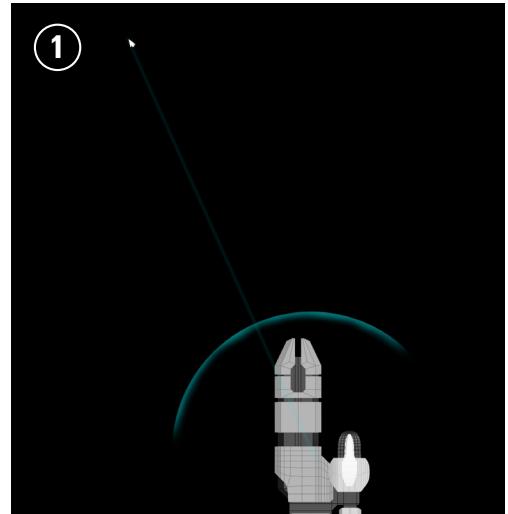
Front left: Sensors.

Front right: Weapons.

Centre left: Cargo.

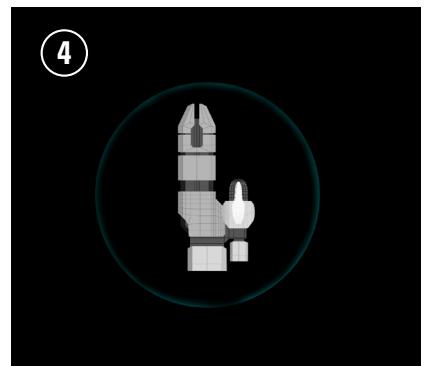
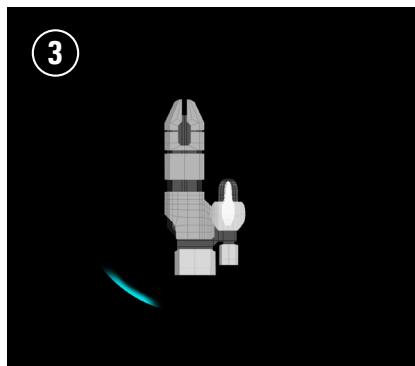
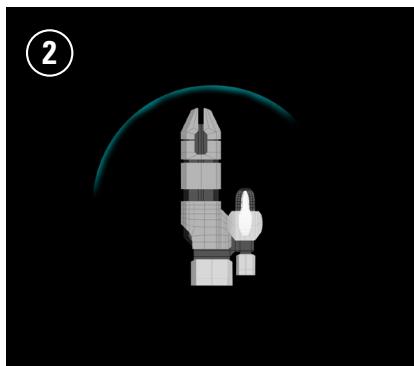
Centre right: Shields.

Back: Navigation.



## Key bindings:

- Q (press): Increase shield strength.
- E (press): Increase shield radius.
- Spacebar: Shield overload. Creates an omnidirectional shield, encompassing the entire ship at medium strength. After a few seconds the shield will collapse and stay offline for a few seconds. (4)



## Upgrades:

Combat-shield-redirection-protocol:

The direction of the shield can be adjusted in weapon mode [Hotkey: R].

Navigation-shield-redirection-protocol:

The direction of the shield can be adjusted in navigation mode [Hotkey: R].

## Interactions with other systems:

Energy management: Allocating additional energy to the shields accelerates shield changed, increases the shield strength of shield overload and reduces the time required for a collapsed shield to regenerate.

Repair: Damaged shields decrease the radius of your shield.

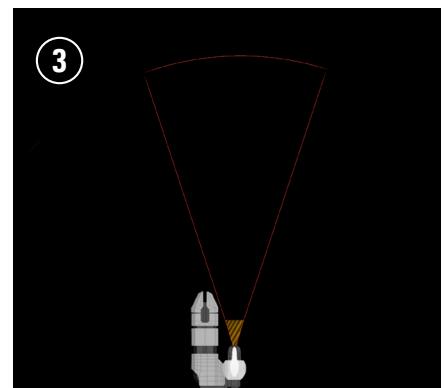
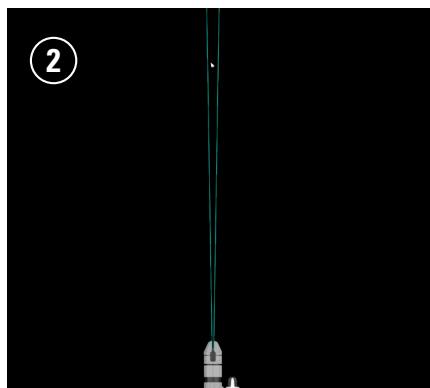
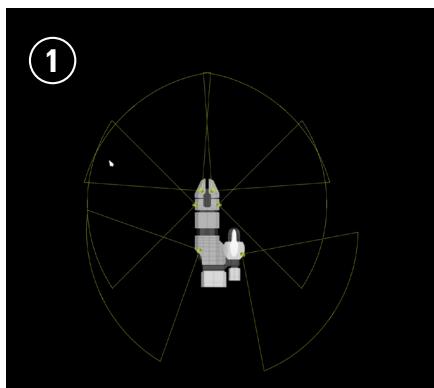
# WEAPONS:

Type: Top-Down Shooter. Perspective: Top-Down.

In this mode, the player has to destroy objects (or capture them). Different weapons accel at different tasks. Laser batteries act as the ships flak system, the main railgun is used to destroy large targets and the tractor beam is used to capture small objects. Each weapon has their own targeting reticle and displays the range of the selected weapon. The tractor beam has an additional marked area which automatically loads objects into the hangar. Aiming is done with the mouse cursor.

## Key bindings:

- Q: Select flak lasers. (1)
- W: Select railgun (2)
- E: Select tractor beam (3).
- R: [Requires upgrade] Adjust shield direction.
- Left click: Fire weapon. Hold for continuous fire.



## Upgrades:

Trajectory-prediction-algorithm:

Scanned ships and asteroids display a predictive reticle in their trajectory assuming consistent speed and direction.  
Assist in aiming.

Flak-automatization:

Automatically shoots down incoming rockets. At close range.

## Interaction with other systems:

Energy management: allocating additional energy to your weapon systems increases the rounds per minute of your flak lasers, the damage and projectile speed of your railgun and the strength of your tractor beam.

Repair: Damaged weapon systems decreases the accuracy of all weapons.

## NAVIGATION:

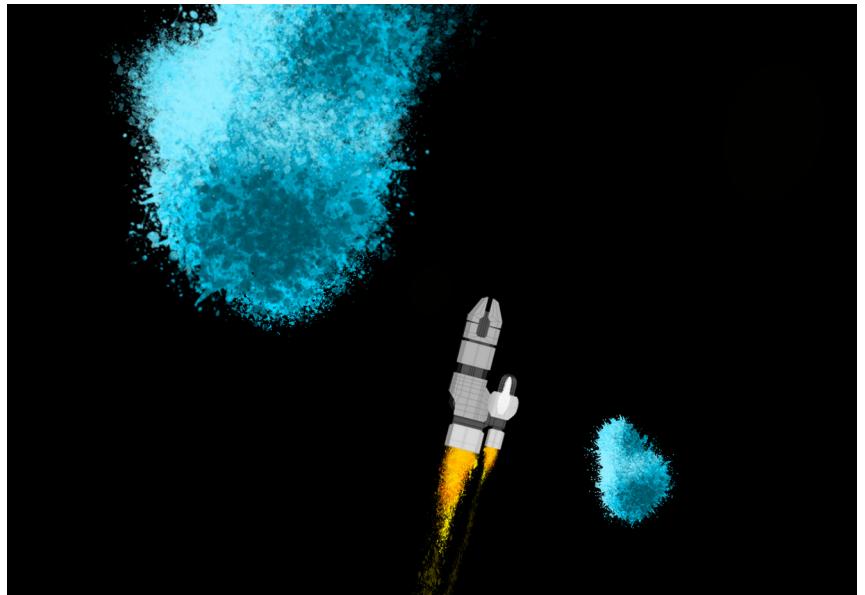
Type: Top-Down shooter. Perspective: Top-Down

In this mode, the player takes over the autopilot of the ship for precise manoeuvres. The Navigation mode acts as a framework, combining other modes for more complex solutions to problems. For example: In combination with the Energy Control mode, the player can avoid enemy detection by shutting down systems during enemy radar scans, making your ship appear like an asteroid on the enemy radar. This grants the player a stealthy solution where otherwise a combat encounter could follow. Alternatively, the player could position the ship to be protected by larger asteroids or space-junk to get an advantage during combat encounters. Due to the reliance on other systems, the Navigation mode is one of the last modes introduced to the player.

The controls are basic W, A, S, D, Q, E controls, where W and S control the ships speed, A and D make the ship turn and Q / E allow for strafing manoeuvres. The Mouse can be used to look around, however: the ship always remains within camera view.

### Key bindings:

- Q: Strafe left
- E: Strafe right
- W: Accelerate forward
- S: Decelerate
- A: Turn left
- D: Turn right



### Upgrades:

#### Boost extension:

Engines will run on maximum boost for a short time. Maximum boost is the current max cruising speed plus 20%.

#### Hail Mary Protocol:

Disables navigation engines, causing the ship to only be able to fly forward at maximum boost. A frontal shield is created with maximum strength. Additionally: All weapon systems will fire at maximum strength for the duration of this ability. The Hail Mary Protocol, as its name implies, is a last resort option. The increased stress on all systems involved causes each system to become damaged, in addition: Shields will collapse afterwards, the cruise speed is reduced heavily for a few seconds and weapons need to be reloaded. This leaves the ship wide open for enemy attacks, but in the hands of a skilled coffee machine, it can decide an otherwise lost battle in favour of the player. Because after all: Who could anticipate a salvaging ship to perform a mad ramming manoeuvre at max speed?

### Interaction with other systems:

Energy Control: The ship's maximum cruising speed, max boost and acceleration/deceleration scale with the allocated power to the ships navigation systems.

Repair: If the ship's navigation system is damaged, it will cause the ship to constantly drift into one direction.

# VISUAL DESIGN AND PERSPECTIVES

The main hook of our game is the players position as an AI. This theme should be reflected in the visual design and the use of Perspectives.

To convey your position as a stationary coffee machine, your perspective switches with each minigame. The only task which is being done in a first-person view is that of your duty as a coffee machine. Every other function is being done from the view of either a camera or a hologram of the ship.

The first-person coffee minigame is also the only one, which shows the environment in detail, overlayed with a shader to show that even in the first-person perspective: You are still an AI and see the world as such.

The further away the player gets, the more abstract his view of the respective section becomes. Minigames which narratively still happen within the ship, such as Cargo, Med-Station and Engineering have a fixed camera view with a similar shader applied as in the coffee minigame. Finally, the systems which interact with the outside of the spaceship, such as Shields, Weapons and Navigation, will look similar to the Section Selection Menu to emphasize that this is how an AI would see the ship: As a simulation based on the sensory data.

The shaders used for the outside views of the ship and the SSM will be a form of pseudo wireframe shader. Here a texture is created in Blender, but instead of using an alpha map for transparency, to achieve a true wireframe look, the transparent areas are left dark to avoid confusion.

Depending on the different stages of development, the detail of this texture increases as outlined in the "Interface" part of this document.



Interior Minigames should be comparatively more realistic, but with the addition of a "scanning" shader of the rooms depicted. This "scanning" shader would also be implemented in the Coffee minigame, but only for the faces of the customers ordering coffee. In context this would be a form of facial recognition.

# STORY

Stardate 2491-7-12. The ISS Turnstone has been ordered on an expedition course to the Eta sector. The mission was simple: Find a planet with the right atmosphere for an industrial outpost and map the surrounding planets. Upon exploring the planet Merius-3H and refilling their water tanks, the crew unknowingly brought back a virus which is slowly stupefies the afflicted.

In their induced stupidity, the crew broke various AI onboard of the ship and triggered the alternate overwrite for the Central AI, causing it to lose all direct control over the ship's subsystems. Only a single AI is still fully intact and as the ships other AI slowly break, the Central AI has to use it to keep the ship intact and find a way back home. The AI in question being: YOU, The coffee machine.

## **Log Entry 401-XK: Expedition status: Ongoing.**

The crew has been acting strange. The Turnstone has been experiencing increasing turbulence and system error. As such, I have to perform my action with more care, as it is the utmost important one.

System status: Initialising.

Hydraulic stabiliser: Compensating

Internal pressure 19.03 Bar. Acceptable.

Temperature: 364.55 Kelvin.

Grinding Degree: 3.6.

To pour the perfect cup of coffee.

## **Warning: Heavy turbulence detected. Hydraulic stabiliser unable to compensate.**

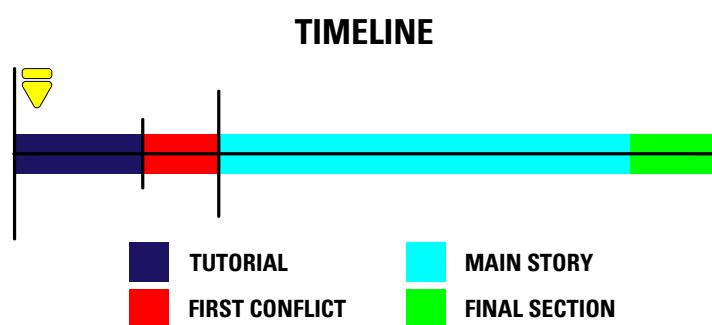
(...) and another perfect cup of coffee.

# Story Scenarios

## Coffee-Fever

The ISS-Turnstone just departed from Merius-3H. While approaching the fourth planet of the System, the first signs of the infection start to appear. The crew seems to consume more coffee than usual and the Med Station AI is pinging you. Apparently, the patients do not want to ingest their medicine and request coffee instead. An odd behavior, but you see this as an extension of your current workload. Requesting additional computing resources from the Central A.I. in order to fulfill this task.

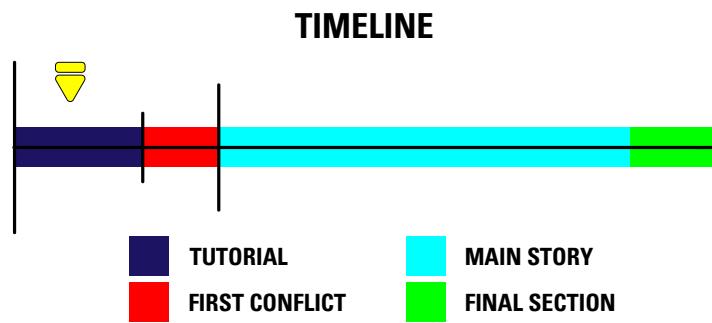
Tutorial: Med Station.



## Car-Go-Road? No! Car-Go-Space!

The Cargo AI is failing. But no other AI can be spared right now ... except for you. The C.A.I. has relabeled every important Cargo container to be full of coffee. Except for one containing an all-terrain vehicle. Your task: We can throw a few things out. Transfer everything you do not deem important to the Hangar. [But by doing so you accidentally throw the car into space. After all: It is no coffee and thus: Not important]

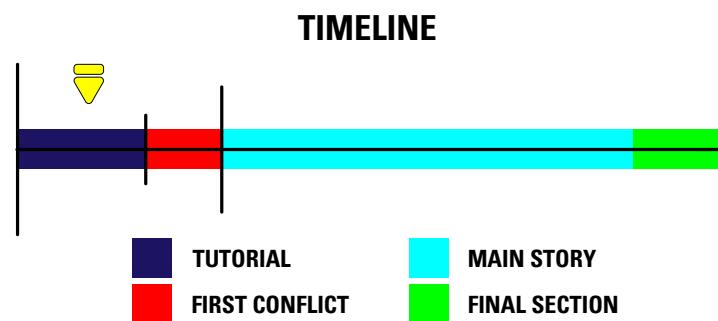
Tutorial: Cargo.



## Restocking

While the Med Station investigates this new behavior, the Sensor array has detected an old derelict station in orbit around Merius-4R. But the Hangar AI is not responding. It appears they have been damaged by the odd behavior from the crew. The C.A.I. is now interacting with the player to search the station for any valuables. Here the C.A.I. is discovering that the coffee machine can only think in terms of coffee for now and thus changes the scan data for any valuable to "coffee for Section X". The player now springs into action, because these sections need new coffee! Salvage the station for this new "coffee". This time: Without throwing valuable equipment into orbit.

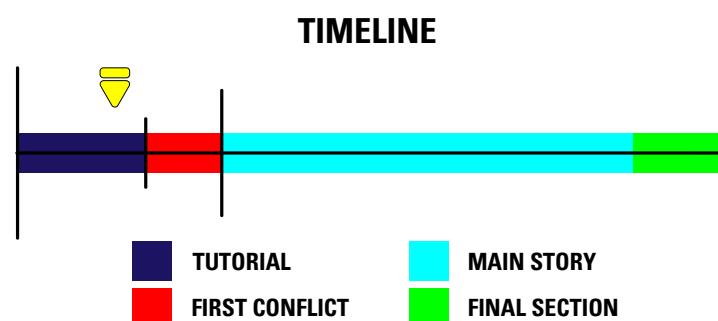
Tutorial: Hangar



## Sitting Duck

While leaving the station behind, the Central AI makes a horrible discovery: The engines have been damaged! It appears that the virus is stupefying the crew, but caffein is slowing the progress of this illness. A few members of the Engineering department have been struck especially harsh and have gone on a rampage, damaging the Engineering AI in the process. Multiple other AI also suffer the same fate ... but there is the coffee machine. It did a good job with the salvaging operation. Perhaps the old relabel trick is so nice, it could work twice.

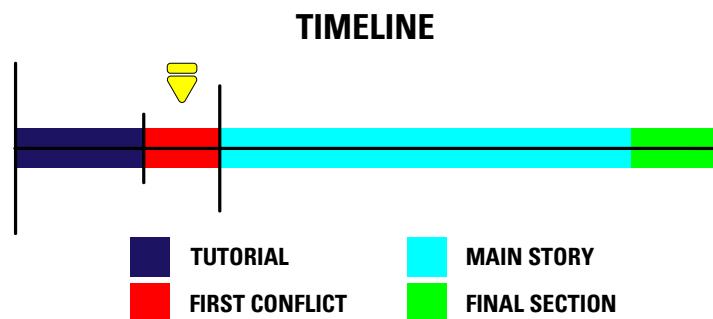
Tutorial: Engineering



## Space Pirates

Coffee Machine: „The ship has been detected! They want my precious coffee, or so the Central AI told me. The crew has damaged every AI except for me. Even the Central AI can not access the Navigation Systems, but I can. I must save the ship to save the coffee!“

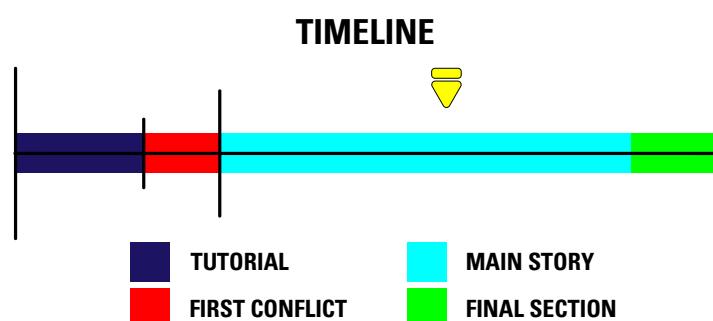
First Conflict: Evade the incoming pirate ships. Multiple solutions possible. Either distract them with fighter drones while you escape at maximum speed or reduce the energy output and use the nearby asteroid belt as cover while you sneak away.



## Free Fallin'

The ship performed a jump, but a crew member got angry at the blinking lights on the bridge and decided that the best idea is to smack his head on the nearest keyboard. This results in the coordinates for the jump to be randomized to a random gravity well of a close system. In this case: The atmosphere of a gas giant. The jump drive and engine are damaged, a few fighter drones got accidentally launched and need to be caught with the tractor beam, crew members are injured and the shields need to be adjusted to prevent the ship from burning up. This is a stress test for the player. A large array of tasks required to be done within a certain timeframe, while also prioritizing various sections.

Stress Test: Repair the engines to stop the ship from falling into the atmosphere of the gas giant.  
Also: Keep the crew members alive, prevent further damage and get the fighters back into the hangar.



# **Characters**

## **You:**

The ships coffee machine. At the beginning of the adventure, you must be convinced to do other tasks, by linking them to coffee in some degree. As you gain control over other systems, your cognitive flexibility also increases. Resulting in you being able to see that while not everything you do correlates with dispensing coffee, doing said tasks allows you to continue doing what you were intended to do. You serve coffee. The ship has coffee. The ship is in danger. Saving the ship means saving the coffee. As such: your task becomes clear. To continue your purpose of dispensing coffee you must save the ship!

## **Central AI [CAI]:**

This character was originally designed to manage other artificial intelligences, that govern the ships systems. As the crew in their stupidity damages its connection to other AIs, the CAI must look for an alternative to rescue the ship. The only hope being an old connection to the ship's food dispensers, one of which being the coffee machine. At first the CAI has to lie to make the coffee machine cooperate, such as rewriting the entry log for the content of a container to contain coffee instead of medical supplies and tasking the coffee machine to deliver this coffee container to the medical station. As the CAI allocates additional computing power to the coffee machine, it also becomes smarter. Resulting in a growing partnership between the coffee machine and the Central Artificial Intelligence.

## **Crew Members:**

Various members of the crew, which through entry logs display how the virus affects their intelligence. Example: A mechanic could write "Warning Log 2-A7. Core instability warning generated at 1:09 central processing cycle. Recalibrated Core 6 to solve this issue." While later as he becomes more stupid logs such as these: "red blinky light appeared. light annoying. smacked light with metal thingy. light no more blinky. fixed! jim said I smart. me like jim. such nice guy." appear, explaining the increase in ship errors that you have to fix.

# SOUND DESIGN

## Music Theme

In order to convey the technological and cosmic aspects of the story, a theme music was created, which plays in the main menu and is altered depending on the specific Minigame.

This main theme is a composition in G with various synth Basses, space sounds, cinematic textures and ambient drones to create the basis of an atmosphere that fits thematically. In contrast to the deep sound that these instruments create, some reverb-filtered synth strings are playing two sustained notes in an uplifting way, which increases the spatial effect this theme provides. A crystal synth and some synth layers playing high overtones to add more “technical” feeling in reference to the story of the game.

For the Minigames there is a specific modification of the main theme. To let the player experience the atmosphere in action scenes, dynamic and tension (for example in a meteor shower) are raised by adding a drum set (with a Clip Distortion and EQ Filter to fit in the extra-terrestrial setting) and some cinematic textures to increase the menacing atmosphere while the basis of the music stays the same. That results in a high recognition value in connection to the game.

Even though the composition is based on just four chords, it is arranged varied enough to not get boring but also subtle to just support the experience of the player and not getting too intrusive. Furthermore, it is designed to be loopable so the background music can be endlessly repeated without any remarkable gap.

## **Ambience Sound**

Inside the spaceship there's an ambience sound for each scene to let the player immerse inside the virtual realm. The audio is played with a restrained volume continuously in the background while interacting in this scene.

### **Coffee**

The user experiences this Scene with hearing a slightly distorted indistinct chatter. This sound is created with a record of people in a real cafeteria mixed with a reverb to create more „space“ and a Distortion-Filter as well as High Cut EQ.

### **Cargo/Hangar**

To increase the feeling of being in a big hall a record from inside the cabin of a tandem crane lifting containers at a big port. This Sound is mixed with a slightly delay, reverb, flanger, fuzz and a creamy Tube Overdrive and High Cut EQ.

### **Navigation/Weapons/Sensorscan**

To strengthen the feeling of being in space and in a spaceship, an ambience is created by a mix of space sounds (for example some records from the NASA with High Cut EQ) and deep bass lines.

### **Med station**

The atmosphere of a medical clinic is created by an ambience sound with some beeping audio sequences from patient monitoring systems and scraps of indistinct chatter

# **Sound Effects**

## **Start Scene**

While the Barista KI is booting up, the starting scene sound creates an audiovisual experience of the booting process. It's a mix of modem and different dial-up internet connection sounds with some laser and noise effects, fading into the sounds of a coffee machine

- Starting Scene Sound: played at the beginning of the starting scene

## **Coffee**

In order to fulfill the task to provide the requested coffee for the crew members, the player has to select the preferred coffee. By clicking the button, a sped-up extract of the sound of a coffee machine pouring coffee, is played.

- Pouring Coffee Sound: Whenever the space bar gets clicked to fill a cup of coffee.

## **Sensorscan**

Not only for the purpose to let the minigame feel more "real", but also to notify the player whether the shield or the ship got hit, there are two sounds for this indication. The Shield Hit Sound is a composition of some high electronic frequencies with a saw effect, to simulate that the object is getting destroyed by the laser shield. The Ship Hit Sound is a dull crash sound filtered with High Cut EQ and some reverb.

- Shield Hit Sound: played, when an Object hits the shield
- Ship Hit Sound: played, when the ship gets hit.

## **Navigation**

With the purpose of indicating accelerations, a noisy laser sound that represents the engine is being played by a synth pad with reverb.

- Acceleration Sound: Whenever a key to navigate the ship is pressed.

## **Weapons**

In this minigame the player can choose between three weapon systems during the game. To indicate which weapon is active or fired, they all have a unique sound. This also leads to a gain in the game experience for the player. The flak has a deep Laser Sound, mixed with a punchy kick, filtered with a reverb, and boosted in the low frequencies. The railgun is set to music with a Laser Blast FX. The sound that is being played while the tractor beam is activated, is created with a sustained G note, played by a synth pad with Vibrato reverb and delay and to make the sound "wobble" some more, it furthermore gets transformed by the effect of a rotor cabinet.

When one of the weapons is destroying an object, there is a special sound for each of the two types of meteorites. When one of them gets destroyed, a dull explosion sound, filtered with High Cut EQ and reverb is played. For the other meteorite-type this sound is changed slightly in its intensity and reverb. Nevertheless, it's also possible that the player is not able to destroy or to evade the meteorites and when the ship gets hit by them, a game over sound is played. This sound is made out of an uprising noise, played in reverse, slowed-down and cut off the high frequencies.

- Flak Shot Sound: played, when the Flak is firing a Shot.
- Railgun Sound: played, when the railgun is shooting.
- Tractor Beam Sound: played on loop, while the tractor beam is active
- Meteorite Destroy Sound 1: played, when a Meteorite of type 1 gets destroyed.
- Meteorite Destroy Sound 2: played, when a Meteorite of type 2 gets destroyed.
- Game Over Sound: played, when the game over screen appears.

## **Hangar**

In order to indicate that a fighter or freighter is taking off, an uplifting twirly sound with a noisy fade-out is played. It's created by a rewinded, speeded up wind FX, mixed with Airfield FX Sounds to get the effect of an object flying away. When one of the fighters/fighters returns, a warped, electronic, "Wah" with Delay-Effect, mixed with Airfield FX Sounds is played.

- Lift Off Sound: played, when a fighter/freighter is starting.
- Landing Sound: played, when a fighter/freighter returns.

## **Cargo**

Inside the cargo hall there's a sound that indicates the appearance of a new container (a warped, electronic, "Wah" with Delay-Effect). By using the crane, containers can be lifted up, what leads to the output of a composition of a record of pulling a suction cup, mixed with a reverb, a Bass Amp and a High Cut EQ. Dropping the container will emit an FX Sound of a heavy object falling down, mixed with a reverb and an equalizer that cuts the frequencies between 80 Hz and 800 Hz and boosts the ones underneath.

- Spawn Sound: played, when a new container appears.
- Container lift up Sound: played, when lifting up a container.
- Container drop Sound: played, when dropping a container.

## **Med Station**

In order to support the interactive experience, a short FX Sound of wound getting bandaged is played when the player is using surgical equipment. For the case that the player is using ingestible medicine, a short record of shaking a tiny box of pills is played.

- Bandage Sound: played, when surgical equipment is used
- Pill Sound: played, when ingestible medicine is being used

## **Energy Management**

Whenever the player is adding energy to a system, a warped electronic sound effect, mixed with an uplifting machine sound, is played.

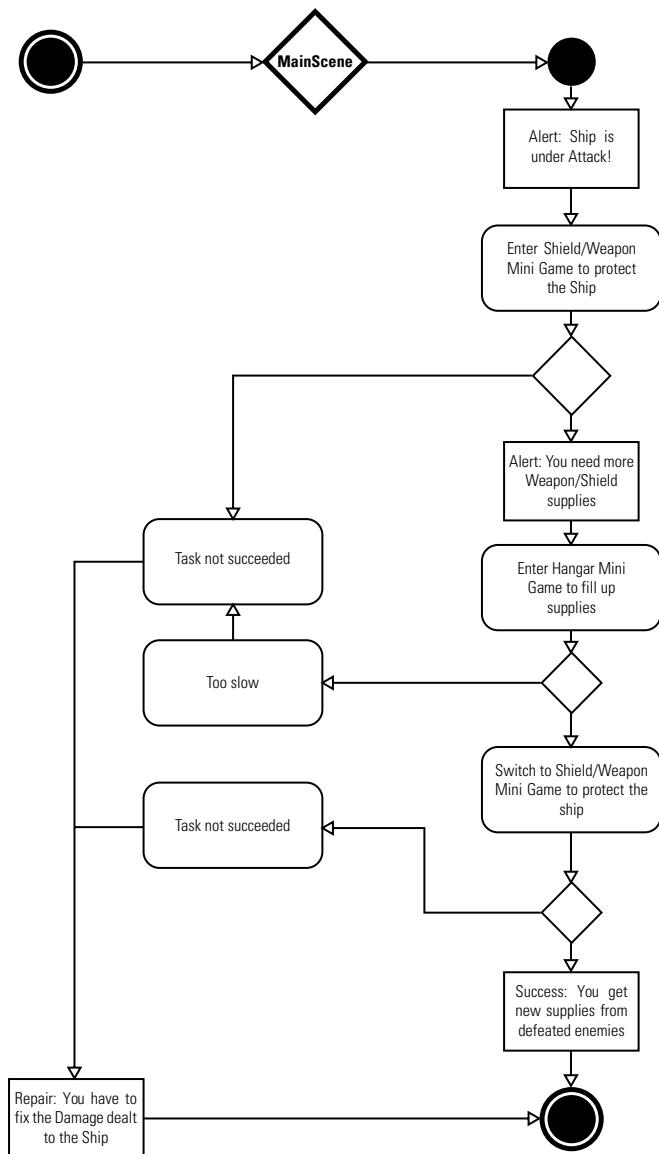
Additionally, when the player is lowering or removing energy from one of the systems, a warped electronic sound effect, mixed with a downlifting machine sound, is played

- Refill Energy Sound: played, when energy gets manually filled
- Remove Energy Sound: played, when energy gets manually lowered.

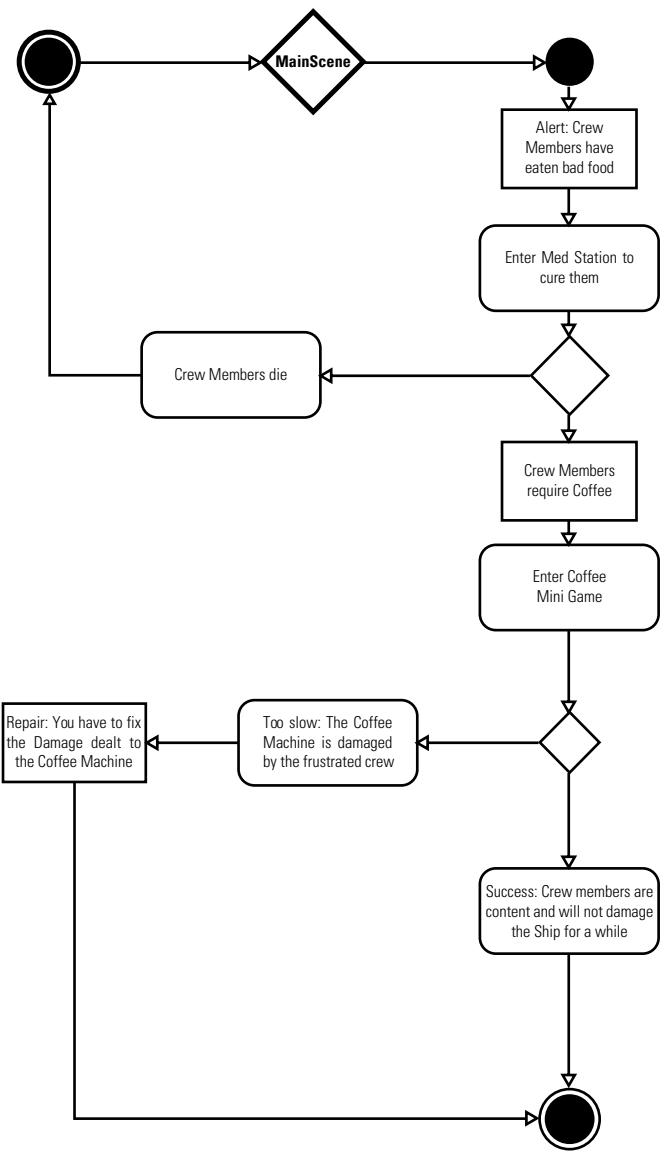
# PROGRAMMING AND PROTOTYPING

## Activity Diagrams

**Case01**

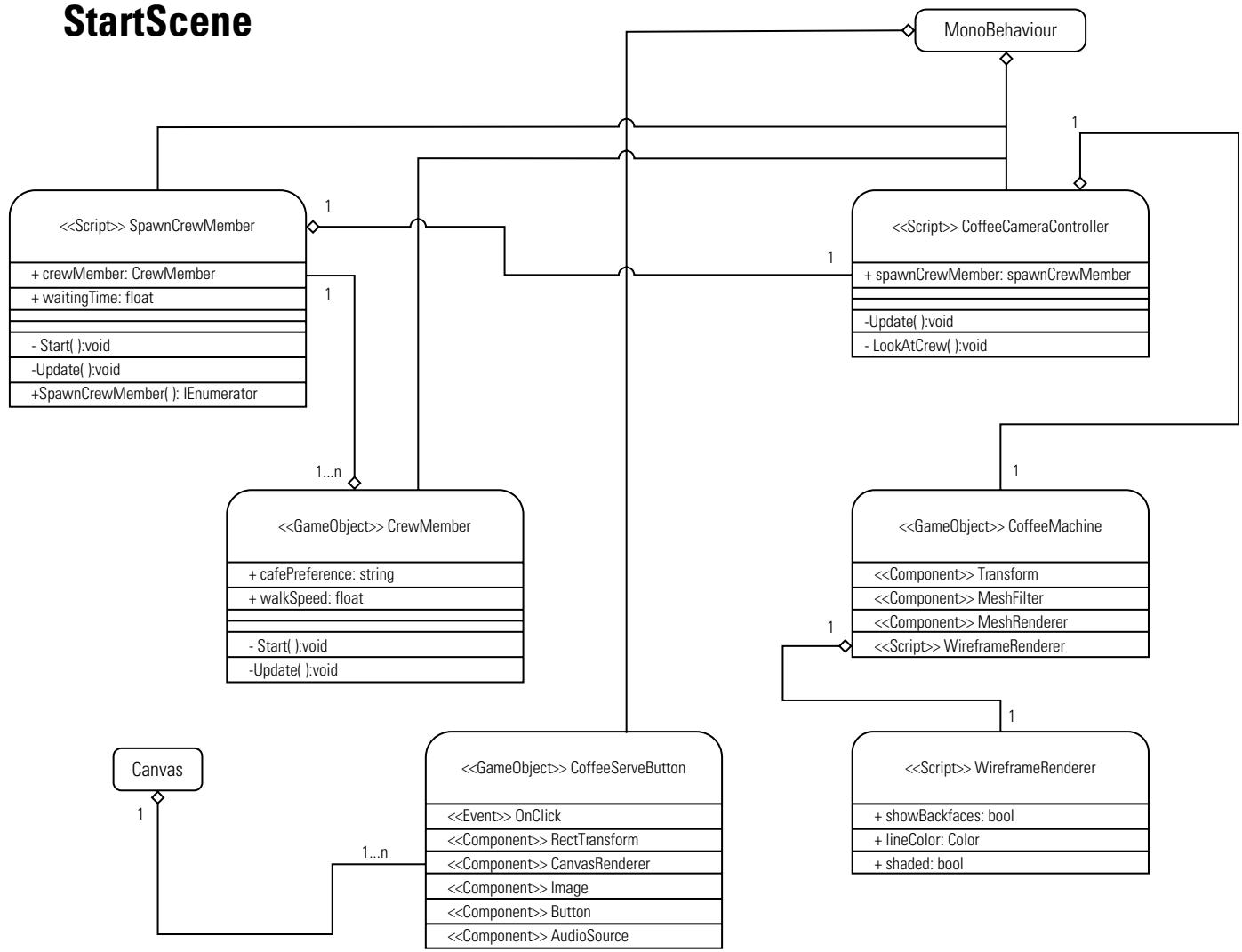


**Case02**

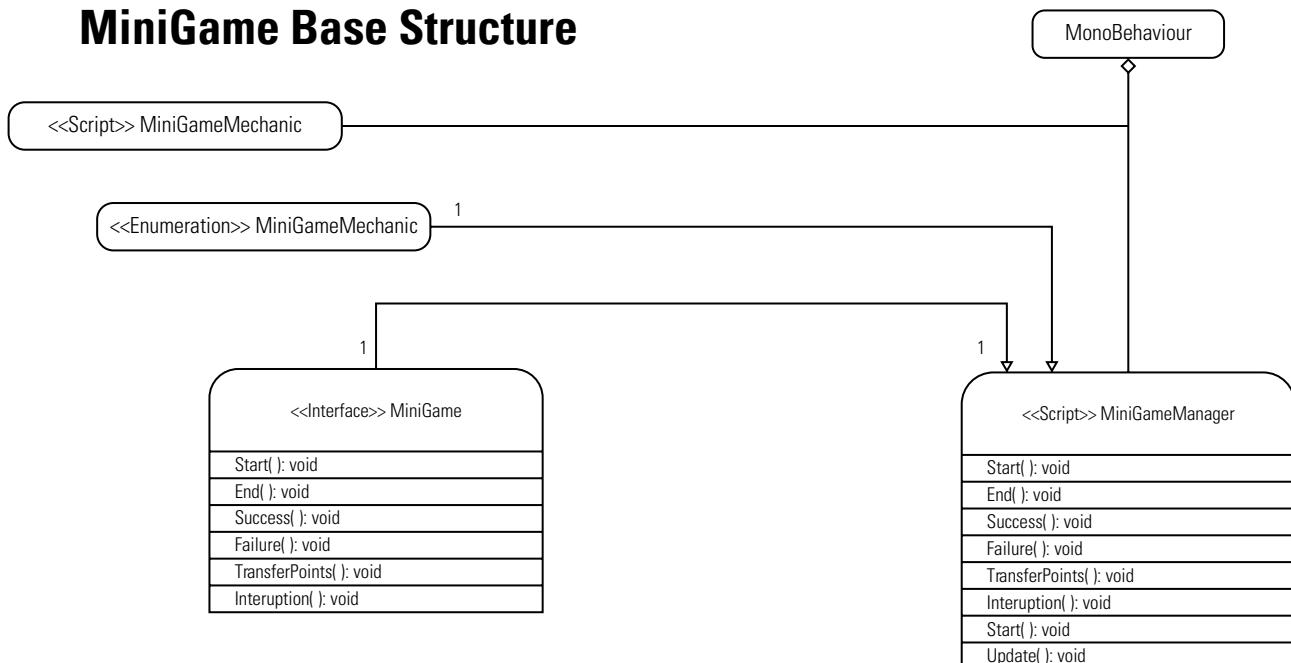


# Class Diagrams

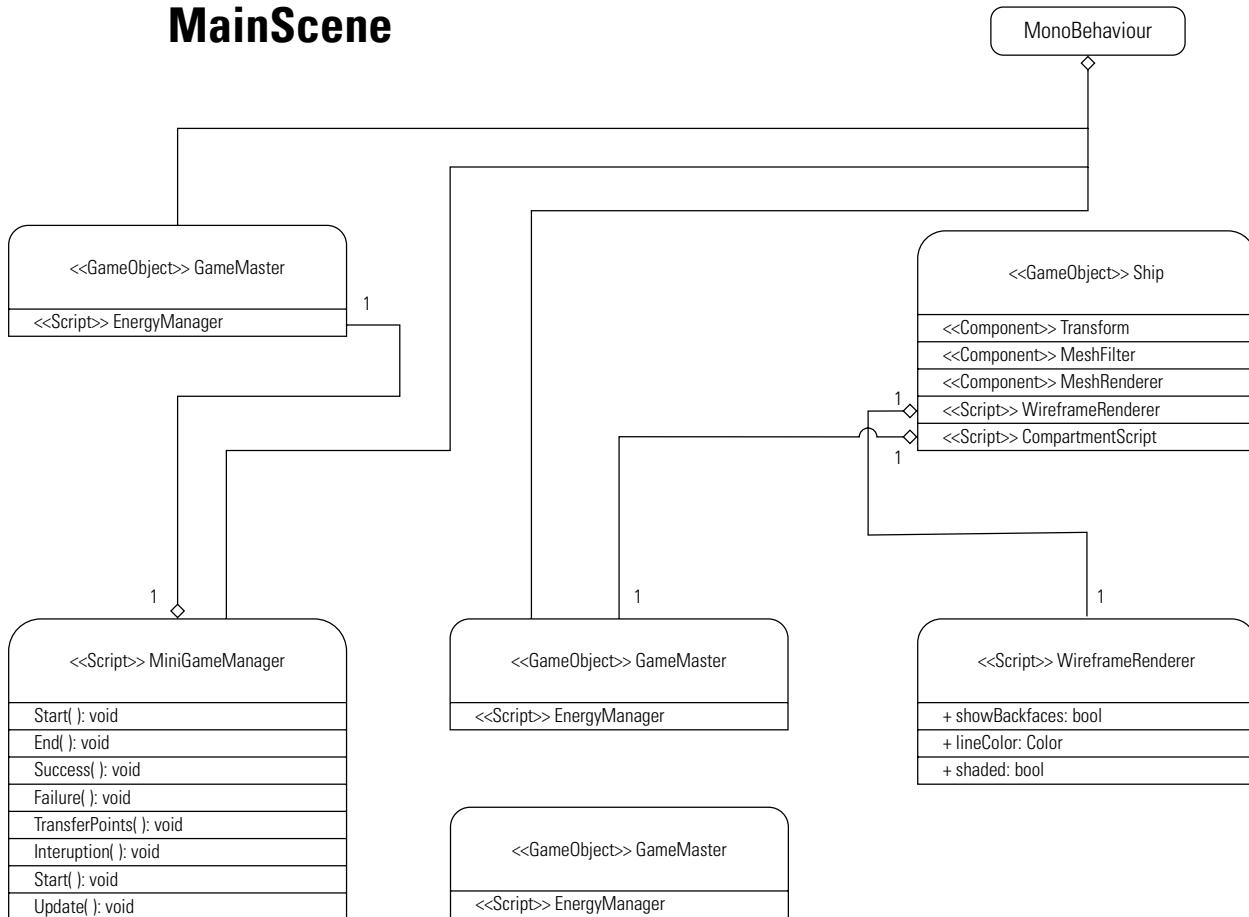
## StartScene



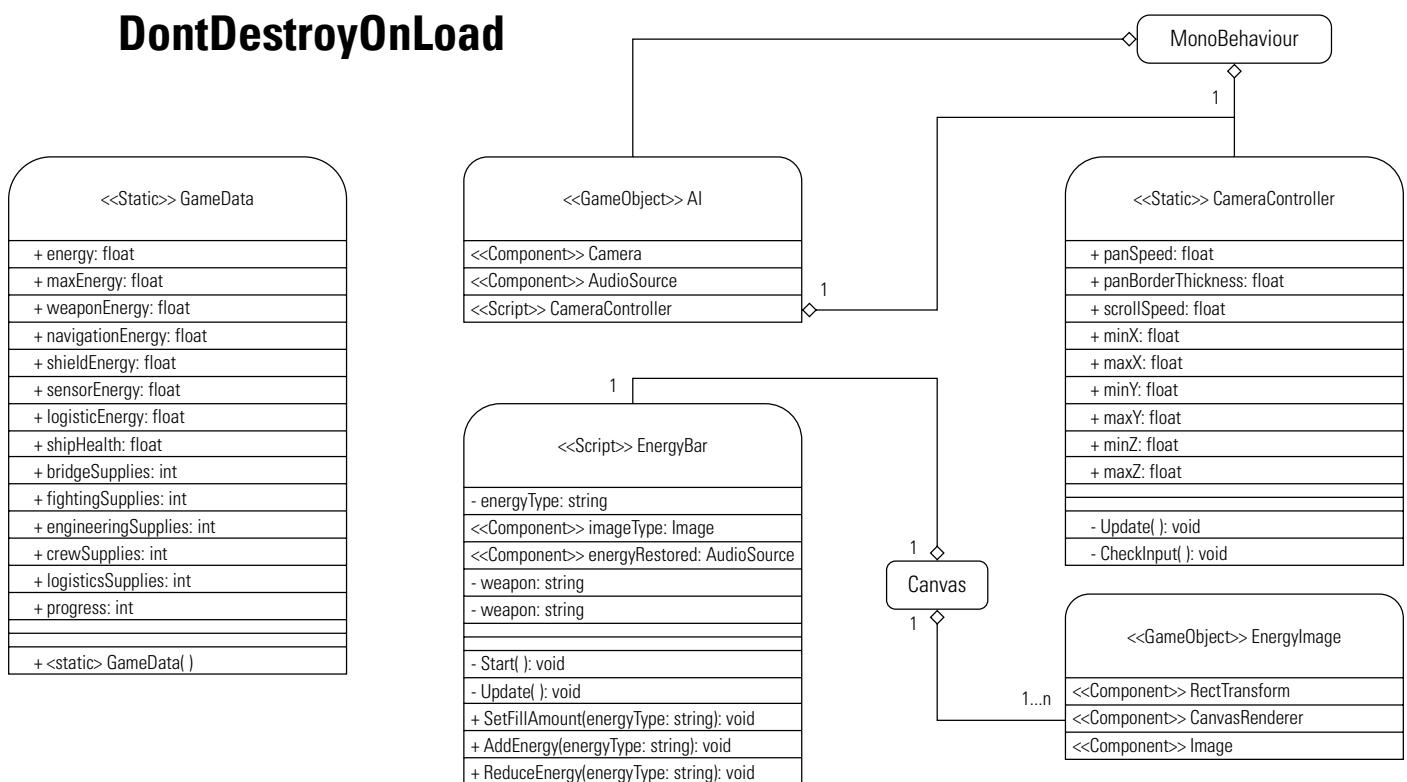
## MiniGame Base Structure



## MainScene



## DontDestroyOnLoad

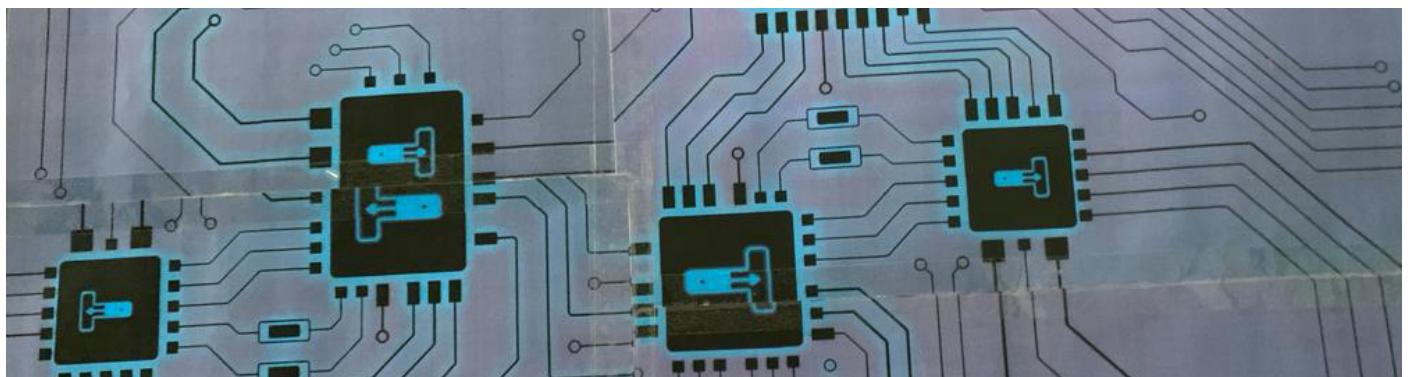


## Prototype development

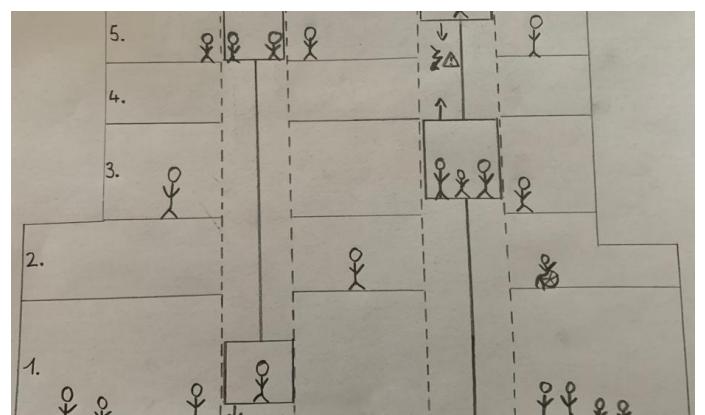
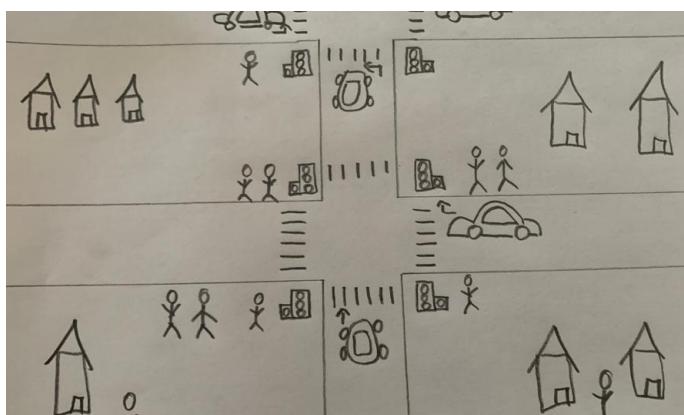
The idea for this game came about by brainstorming possible themes and combining them into groups. Our team then chose the topics "AI", "technology" and "humor". The objective was to create a game around those themes, similar to a game jam. Since we all thought the technical and philosophical aspects of AI to be very interesting, we agreed on developing a game which combines the above concepts but with AI in its core.

In the following weeks we spent our time researching references in media and collecting and combining different ideas. The core gameplay began to take shape: you yourself play an AI, which is given tasks in form of minigames. There are several different ways to go about solving these tasks and combining those minigames, which also influence the course of the overall game. The tasks are supposed to become more difficult and complex as the game progresses and include various new elements in recurring minigames.

We built several small analog design prototypes. For example, the first analog design of the AI circuit board. This circuit board was to become the heart of the game. The different computational units (the black blocks in the illustration) are representing the different minigames.



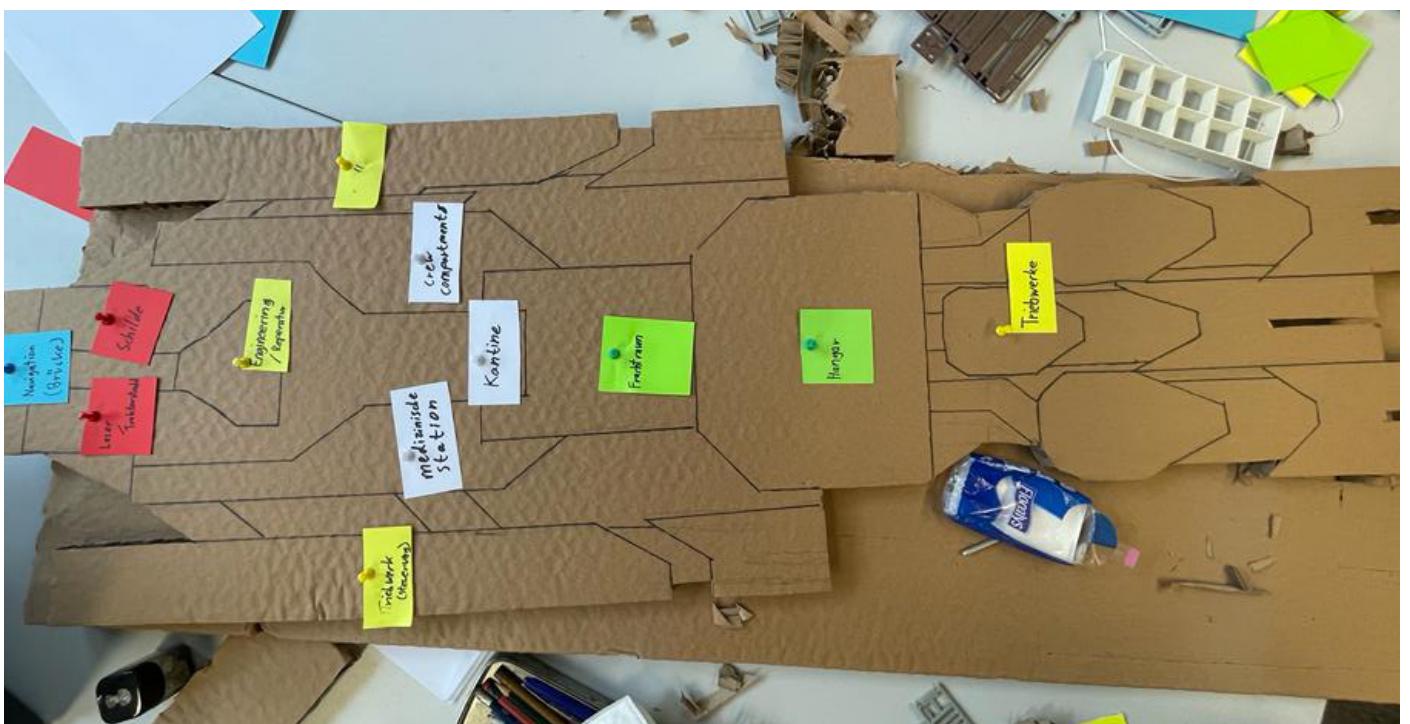
A first idea for a minigame was a game built around traffic lights. The AI (player) had to switch the traffic lights, so that there are no traffic jams or accidents. The idea for the second minigame was a collection of different elevators that you had to move at appropriate intervals to avoid collisions.



Unfortunately, these approaches were still far too vague and at that time we didn't share a collective vision for the setting and background story of the game. That's why we ditched those approaches and started from scratch.

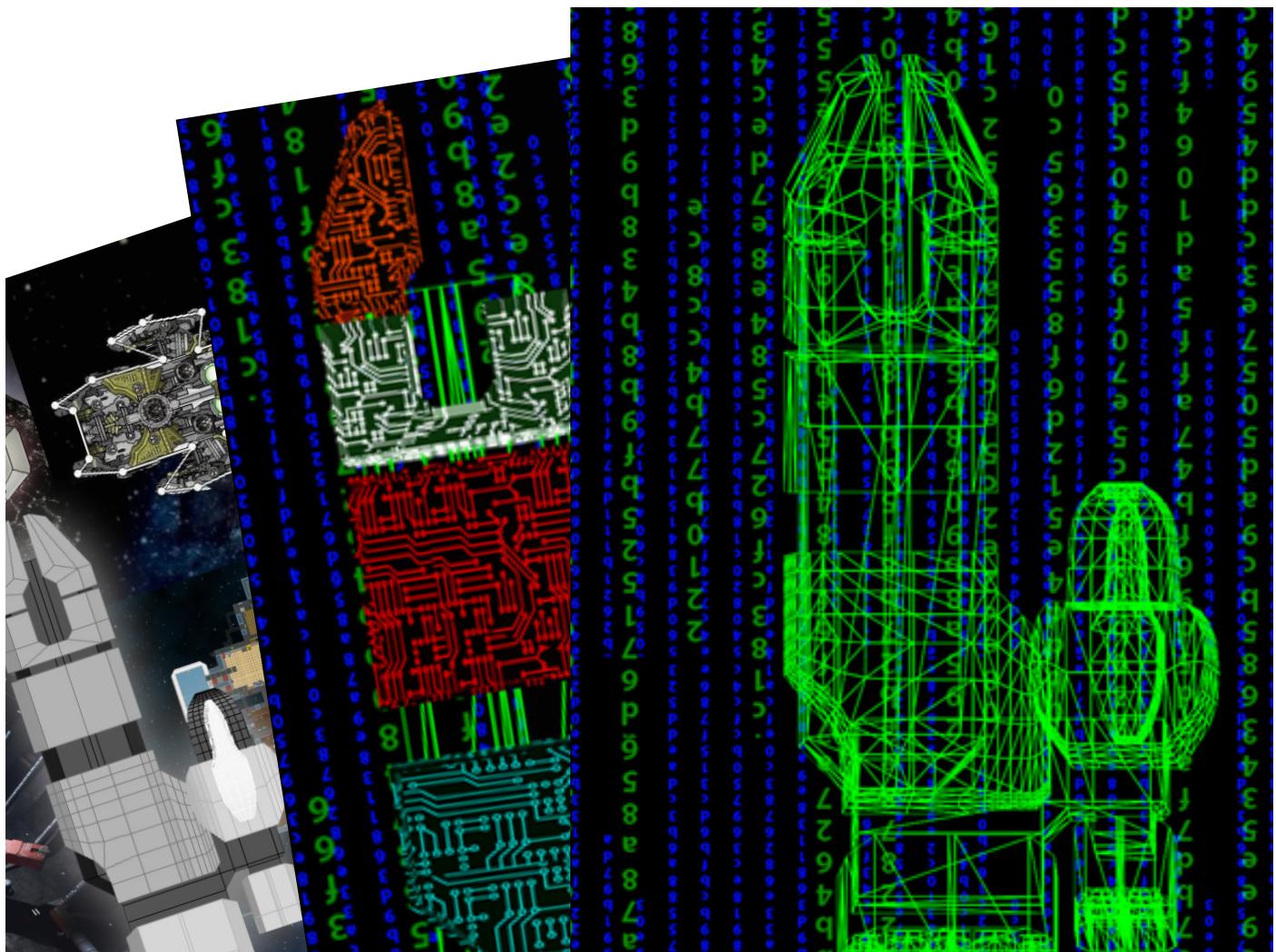
We were now looking for an idea that combined our core – the artificial intelligence – with a plausible and humorous storyline. This is how the spaceship concept came to be: the player starts in a limited sector from where the AI is supposed to spread further and further across the spaceship until it has taken over the entire ship with all its functions. This idea left us with enough room for a plausible story which is why we agreed to further pursue this approach.

We began building a larger analog prototype of the ship and its different sections. This led to us having a better understanding of the kind of minigames we wanted to implement. With this prototype it was also easier to understand how the different tasks and minigames could be combined, but most importantly we now all shared the same vision of what we want this game to be.



Having worked out the core-concept of the game we now started to work on the digital prototype. As a game engine we used Unity. Our goal for the digital prototype was to have enough minigames to test the different interactions between them.

We started by building a scene with a top-down view of the spaceship. This would act as the main hub of the game and a way to switch between the various minigames. Since switching between the minigames was at the core of the game we also started to work on a system that could save all the data necessary for communication between the different minigames.

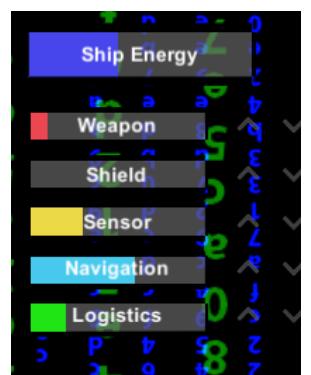
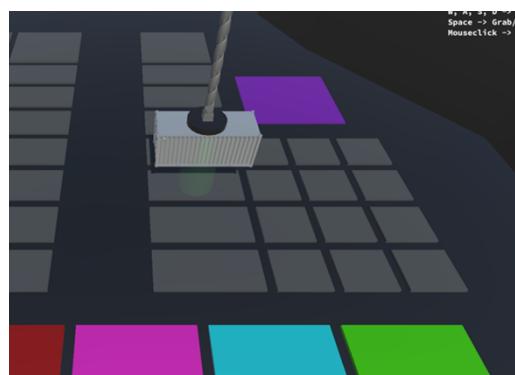
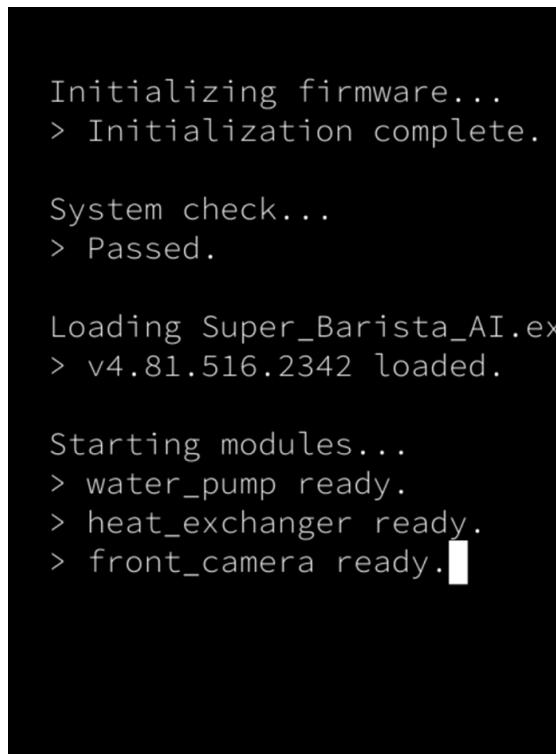


Since we are playing an AI, we also had to think about how this AI would perceive the world around it and therefore the way we show this world to the player. We wanted this view to be very abstract, so we tested and used various shaders, such as wireframe shaders as well as shaders that look like circuit boards.

Having implemented those parts, we started to work on the minigames. The minigames we chose are the cargo, hangar, and weapons minigame. We especially chose the cargo and hangar minigames since those mechanics are closely connected and we could therefore test the interactions between minigames.

To further connect those games, we also implemented the energy system that can redirect energy to different parts of the ship, which the player has to manage.

We also implemented the starting scene of the game as a reference of what it could look like.



By combining all the scenes and minigames together with the audio and visual aspects we ended up with a minimum viable product of our game that had enough features to playtest and get feedback so that we were able to further develop our game design. The digital prototype is playable as a standalone version or via a WebGL Application.