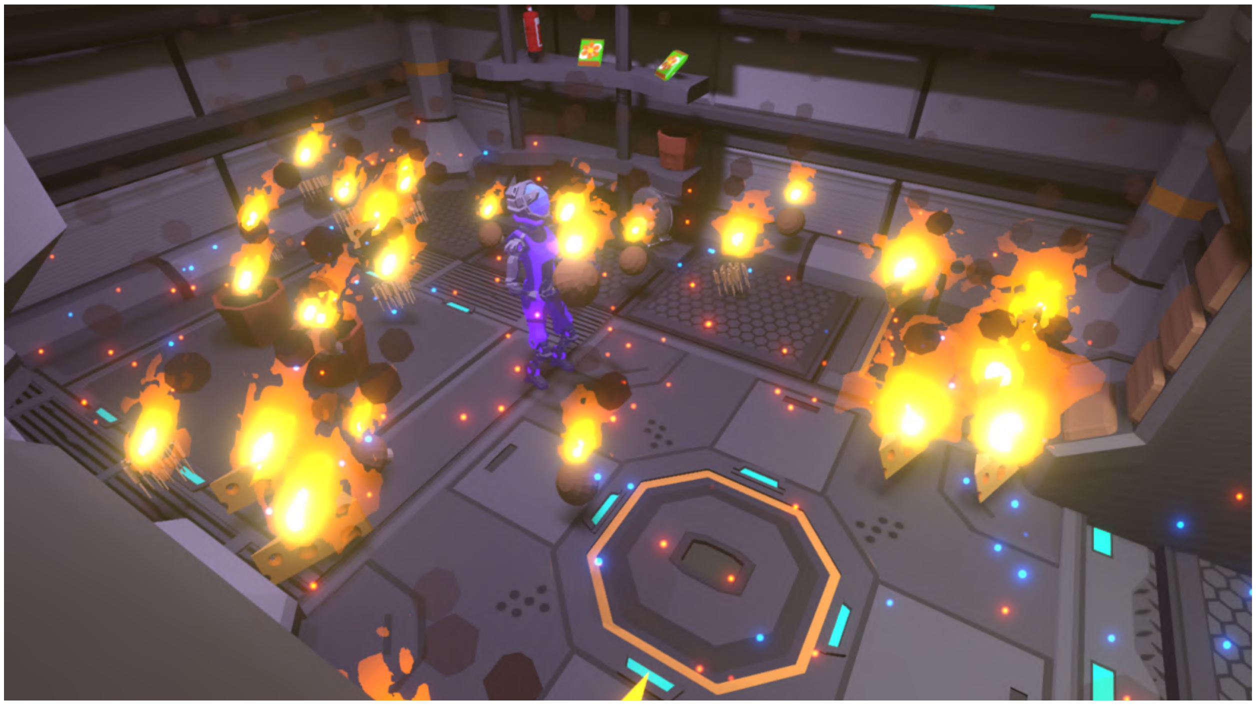


Final Release: Code Bread

19th of July 2020

Team *Callstack Overflow*$_2$

Maximilian Werhahn

Mark Pilgram

Min-Shan Luong

Felix Neumeyer

Game Summary

Code Bread is a cooperative top-down management game set on a spaceship in which two players try to bake and deliver as many pizzas as possible while dealing with various environmental factors.

Pizza orders continuously come in via the ship's antique telephone, which needs to be answered to accept an order. Orders only remain for a limited duration before the customer becomes impatient and cancels their order.

To make a pizza, the players must first gather the required ingredients. These currently include wheat, tomatoes, cheese and mushrooms. Wheat and tomato plants can be grown in plant pots, but require water to grow, which is obtained by catching and melting passing ice asteroids using a tractor beam and the heat from either an engine room or an oven. Mushrooms grow randomly around the ship, requiring players to search for them. Finally cheese is produced by the space ship's third member of the crew, the cow. Provided the cow stays fed and has oxygen to breathe, it will gradually produce cheese. Should players forget to feed the cow, it will automatically eat whatever edible items and plants it can find nearby as it roams around the ship.

Once these ingredients have been gathered, they can be brought to one of the ship's ovens to make a pizza. Pizzas take time to bake, but can also be left in the oven for too long and burn, so players will have to keep an eye out for when the pizzas they've started are ready. Having successfully removed an unburnt pizza from the oven, the pizza can be taken to the ship's pizza teleporter to be delivered. Players shouldn't wait too long though, as a pizza left out in the open will go cold and decrease in value.

The challenge of the game comes from frequent events that interrupt the pizza-baking process and pose a risk to a player's health or the sustainability of the pizza ingredient production pipeline. Players need oxygen to survive, and without plants to convert breathed air back into oxygen, rooms on the spaceship will quickly become uninhabitable. The ship is constantly bombarded by asteroids, resulting in hull breaches that suck the air out of any room connected to the breach via an open door. Most items can randomly catch fire and burn away, with the fire consuming precious oxygen and spreading to nearby burnable objects. As was already mentioned previously, the cow may also become a hazard to the players' plants if they forget to feed it. To combat all this players will need to repair hull breaches, put out fires using a fire extinguisher, keep the cow fed and plant additional plants to maintain a habitable environment inside the spaceship.

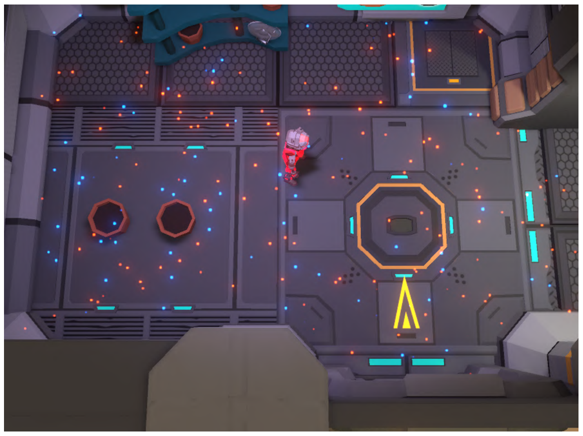
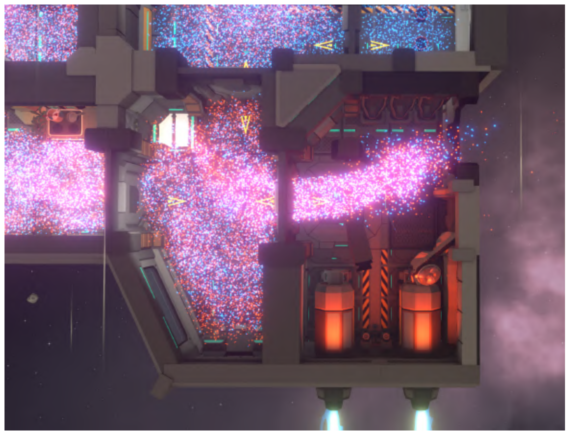
Should both players fall unconscious simultaneously (i.e. when one player suffocates and the other player is unable to revive them without suffering the same fate), the game is over.

Game Elements

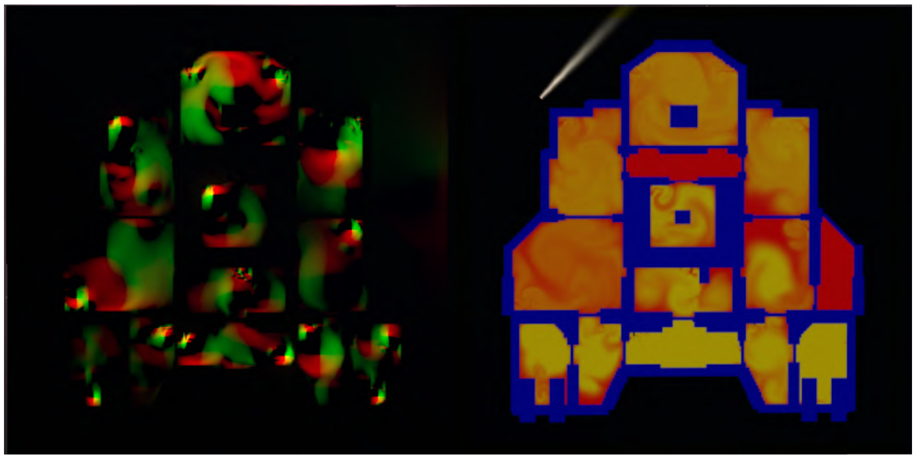
In the following, the core elements and mechanics of our game are listed and the descriptions are supported by various screenshots.

Fluid Simulation

Blue particles represent oxygen, red particles carbon dioxide. The air density can be measured by the relative particle density. The number of particles shown in the second image is exaggerated to highlight the flow of air more clearly. While both particle count settings run with near identical performance, the exaggerated mode reduces item visibility, making it impractical during actual gameplay

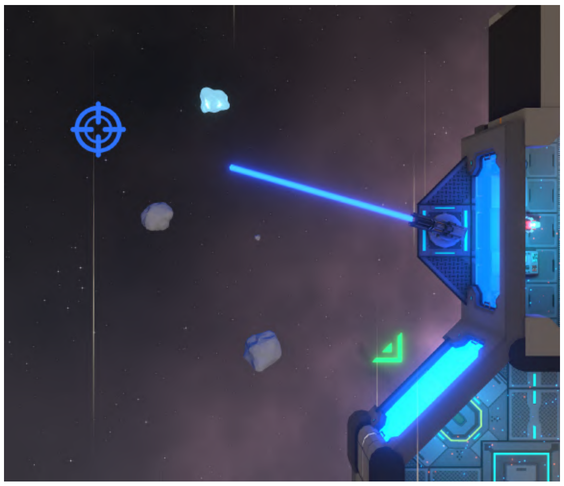



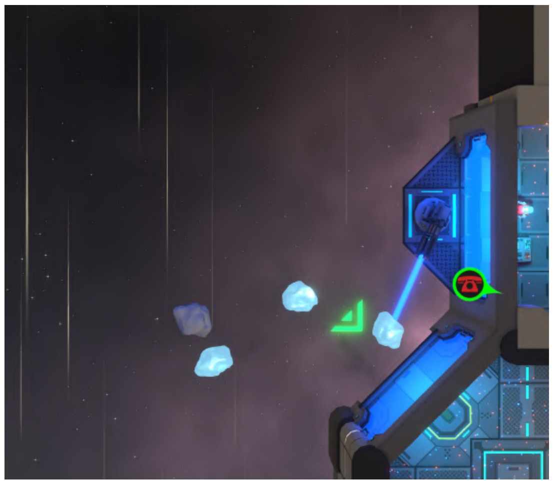
The left side of the image below shows the centered velocity grid of the fluid simulation. The velocity is colored green if its x-value is larger than 0 or red if its y-value is positive. Negative values are not represented. The right side depicts the oxygen and carbon dioxide concentration in the colors red and green respectively, combining to yellow in areas where both gasses are present. Blue pixels represent the walls/solids of the simulation.



Ice asteroid collecting minigame

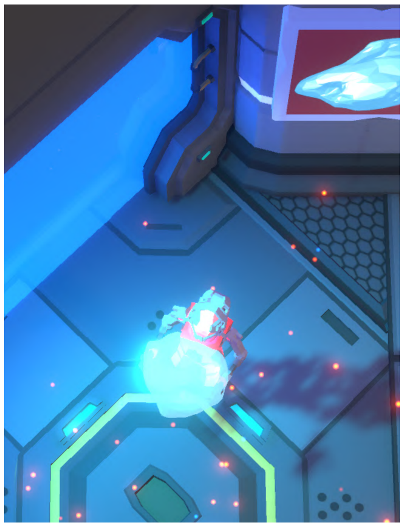
Ice asteroids can be collected by controlling the spaceship's tractorbeam. The beam requires players to lead their target and then navigate it to the glowing laser wall on the bottom right without getting the beam interrupted. Once an ice asteroid has been brought inside the ship, it can be picked up for further processing.

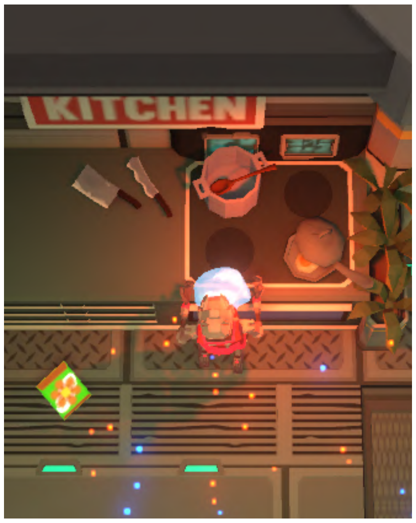


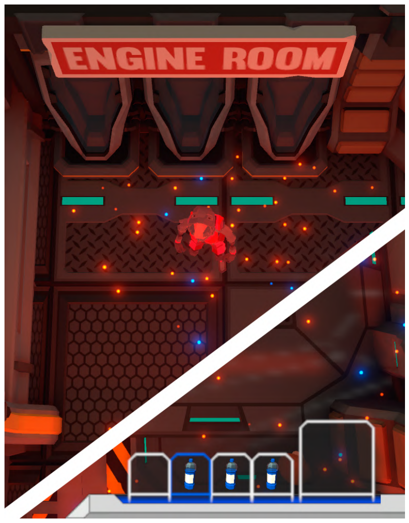


Ice melting / water bottles

Ice asteroids melt into two water bottles when near an oven or inside an engine room.

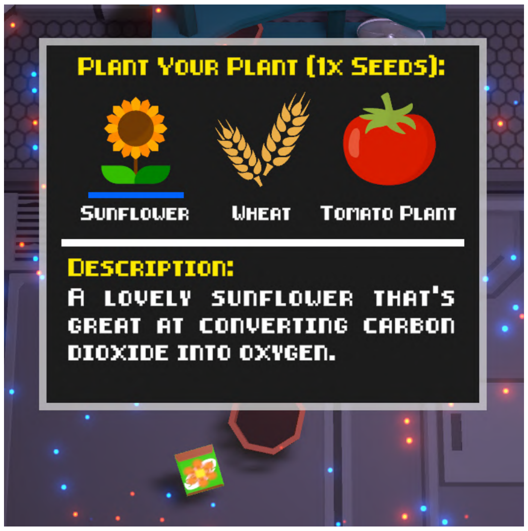




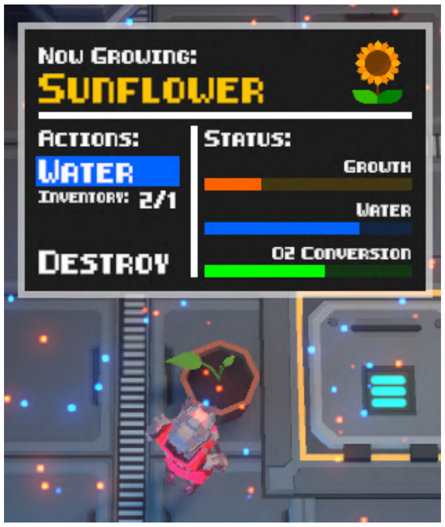


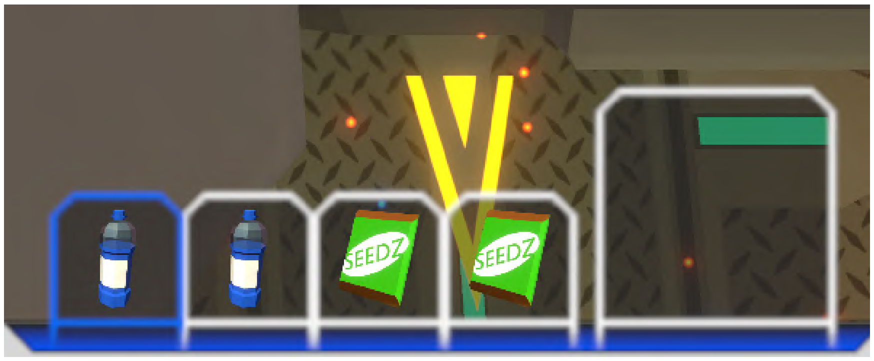
Growing plants (images of different states of the plant)

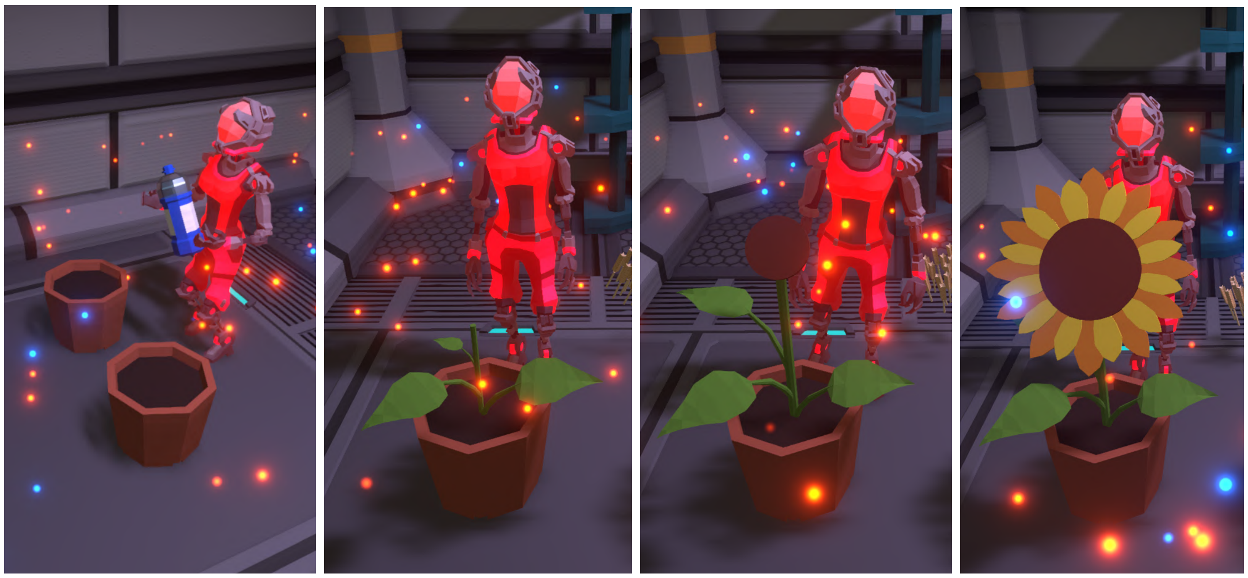
When interacting with a plant pot while carrying a seed package, the player can choose the type of plant to cultivate. Players need to water plants for them to grow.



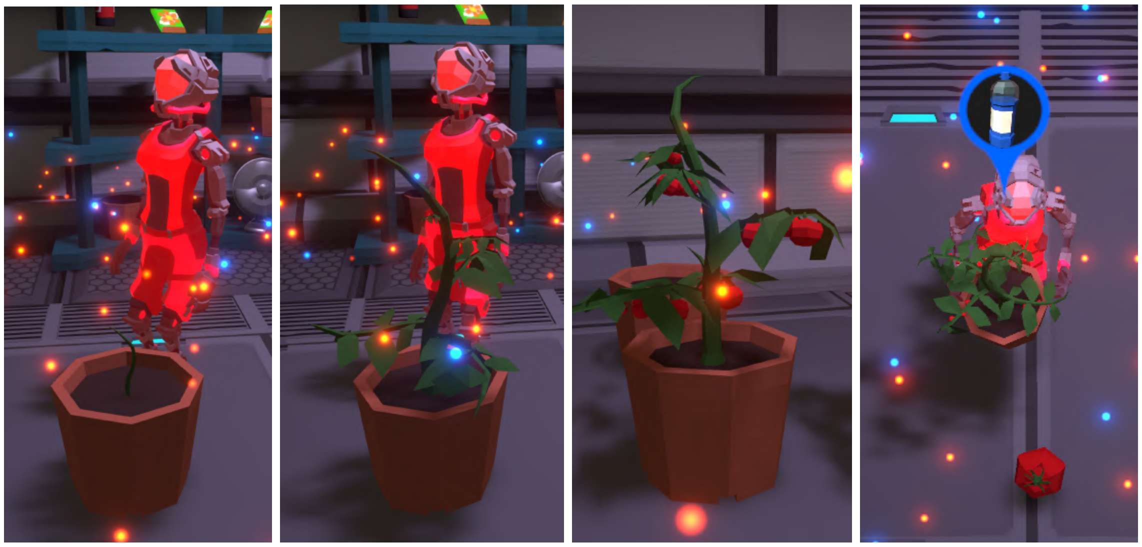
A LOVELY SUNFLOWER THAT'S GREAT AT CONVERTING CARBON DIOXIDE INTO OXYGEN.



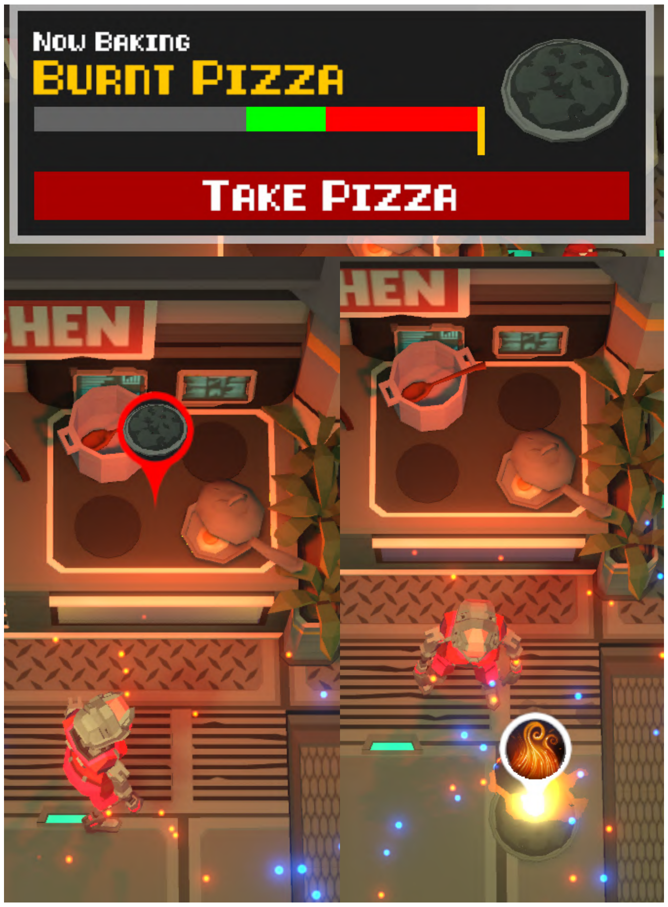




To harvest ingredients, e.g. from a tomato plant, it has to be watered repeatedly. Should the plant run out of water, an event pointer (shown in the image on the right) will notify the players of this requirement..



Fire and Fire Extinguisher



Fire can break out when the player forgets to take the pizza out of the oven in time.

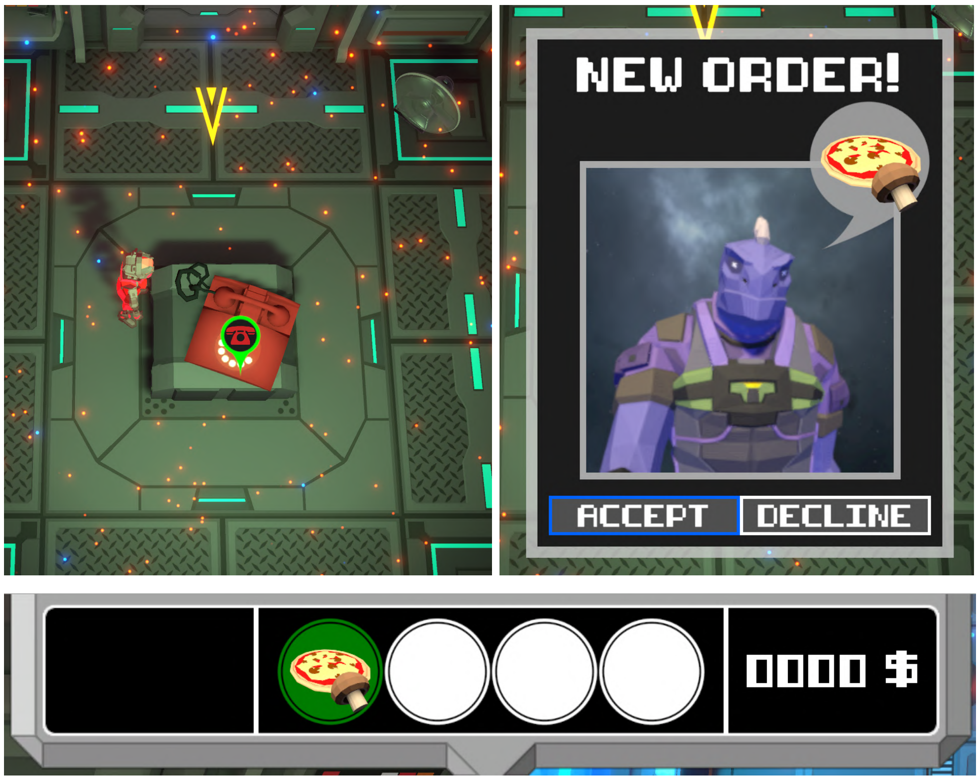
The burnt pizza will burn when taken out of the oven. The fire can spread over time to nearby items. Items will be destroyed if the fire is not extinguished within a certain item-specific time frame.

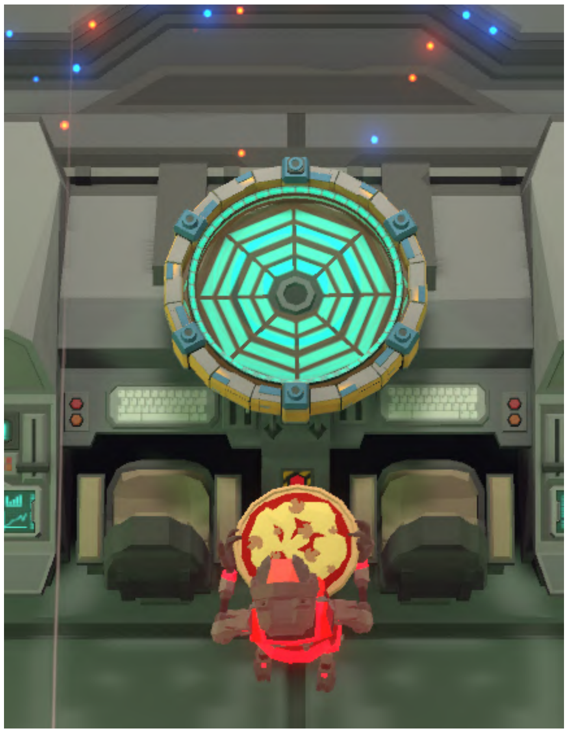
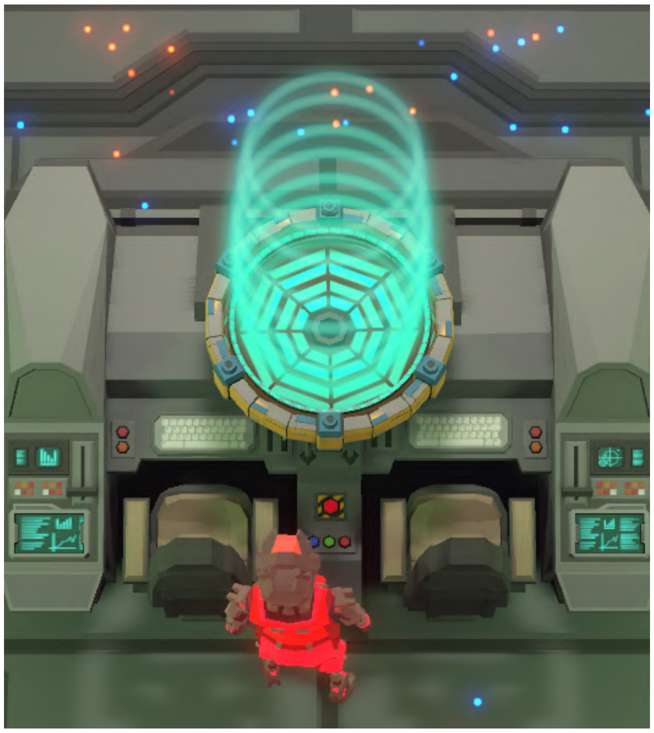
Fire can be extinguished by the player using a fire extinguisher. Fire needs oxygen to burn and turns oxygen into carbon dioxide. Therefore it will also go out if there's no more oxygen in the room.

Items can also catch fire randomly throughout the whole spaceship.

Telephone orders

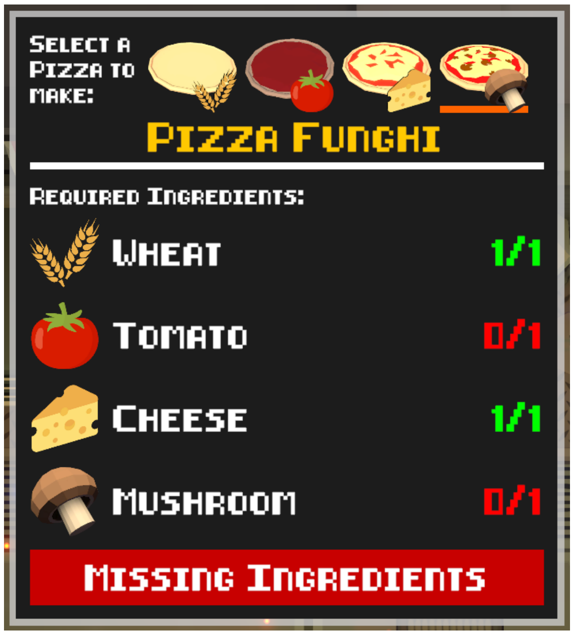
On a regular basis, customers call via telephone to place an order. The players can either accept or decline this order. Once accepted, an icon pops up at the top of the screen showing the ordered pizza and, via a colored background, indicating the time that is left until the order expires. As soon as the pizza is done, the players can deliver it at the pizza teleporter, which brings it directly to the respective customer. Depending on how much time it took to complete the order and whether the pizza was freshly baked, the players receive more or less money.



Pizza baking

Interacting with an oven opens the following UI menus:



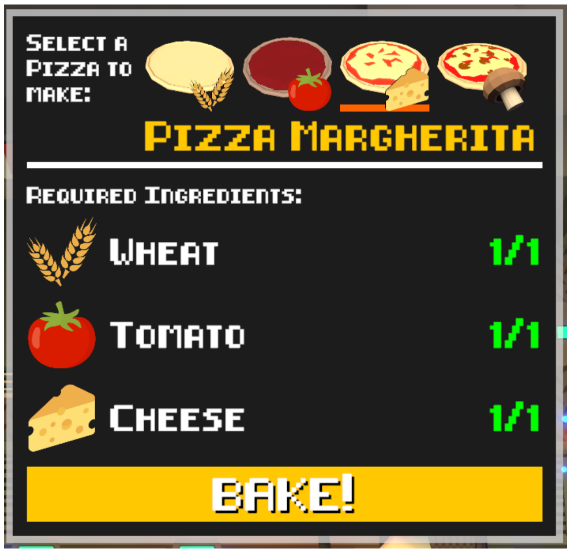
Select a pizza to make:

Pizza Funghi

Required Ingredients:

Wheat	1/1
Tomato	0/1
Cheese	1/1
Mushroom	0/1

Missing Ingredients



Select a pizza to make:

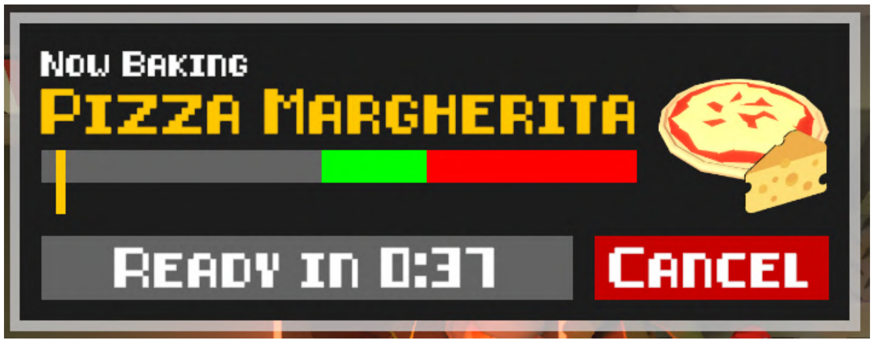
Pizza Margherita

Required Ingredients:

Wheat	1/1
Tomato	1/1
Cheese	1/1

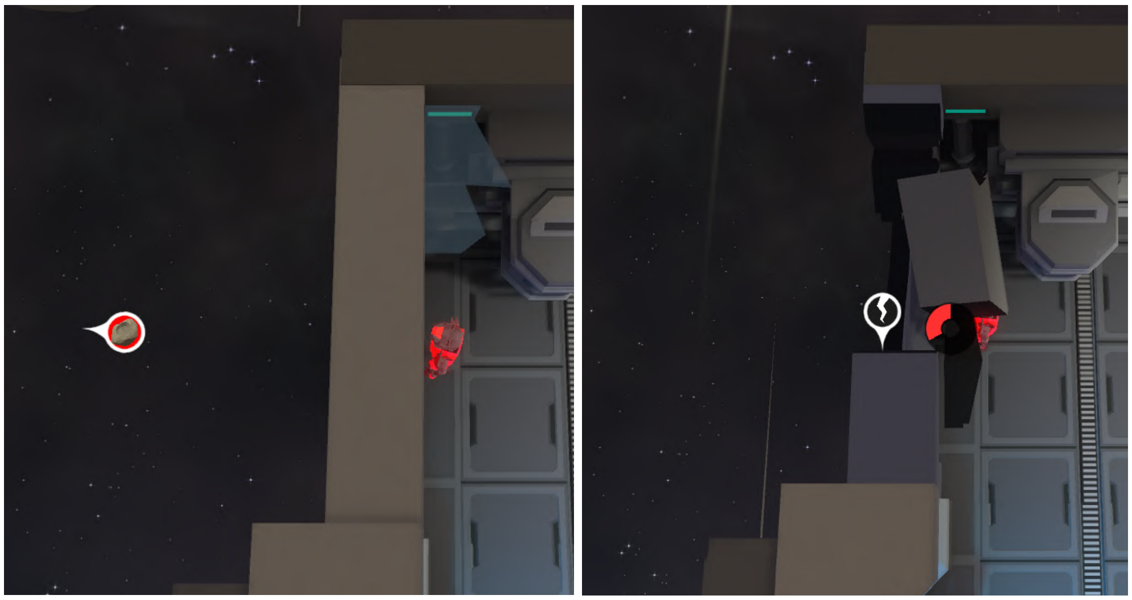
Bake!

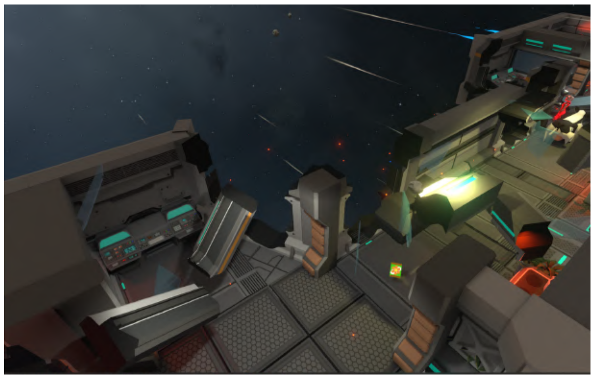
Players can choose a pizza recipe to bake. Each recipe has its own set of required ingredients, bake time and price. Once the baking process for a pizza has been started, the oven menu will switch to display the baking progress. Should the process be canceled during the gray phase of the progress bar, all ingredients used to make it are lost. In the green phase, the pizza can be removed from the oven in its finished state. Should players wait too long and the baking process reach the red stage, then the pizza has been burnt.

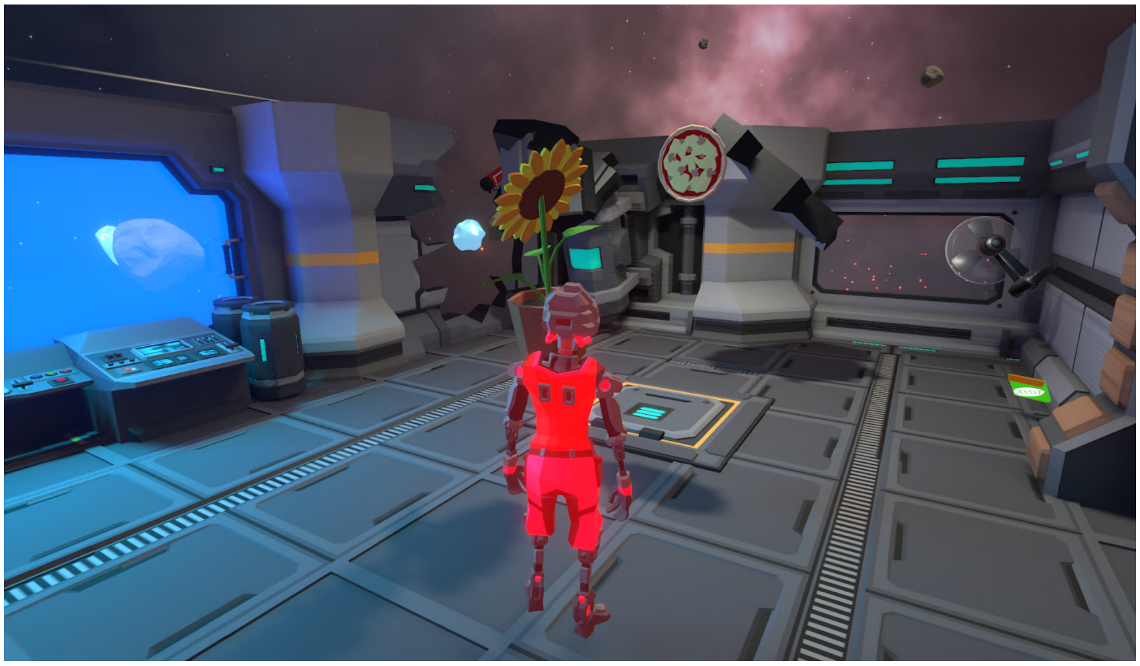


Breakable walls (+ repairing/gravity loss)

Players receive a warning event pointer just before an asteroid is about to hit the ship. This asteroid event pointer transforms into a broken wall event pointer once the impact has occurred. Broken walls can be repaired by having a player interact with them, the repair progress is displayed both visually by the state of the wall and by a red circular progress bar. As long as a room has at least one hull breach, rigidbodies within it are not influenced by gravity.

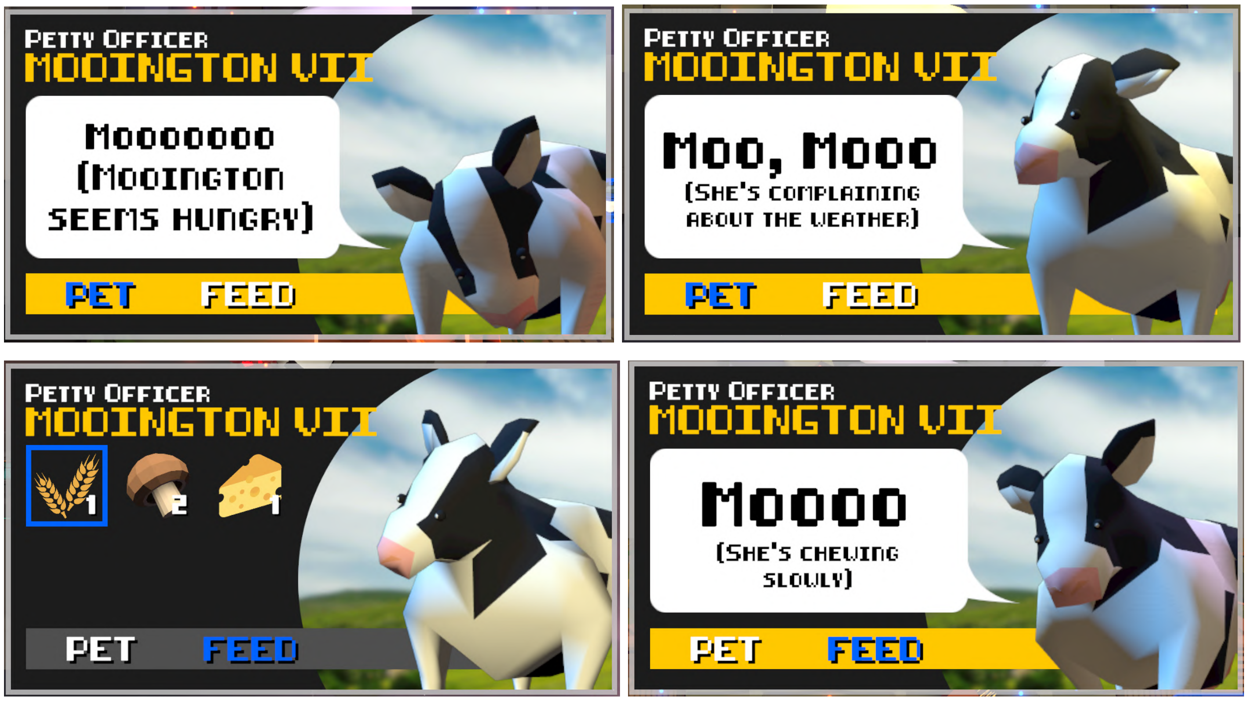


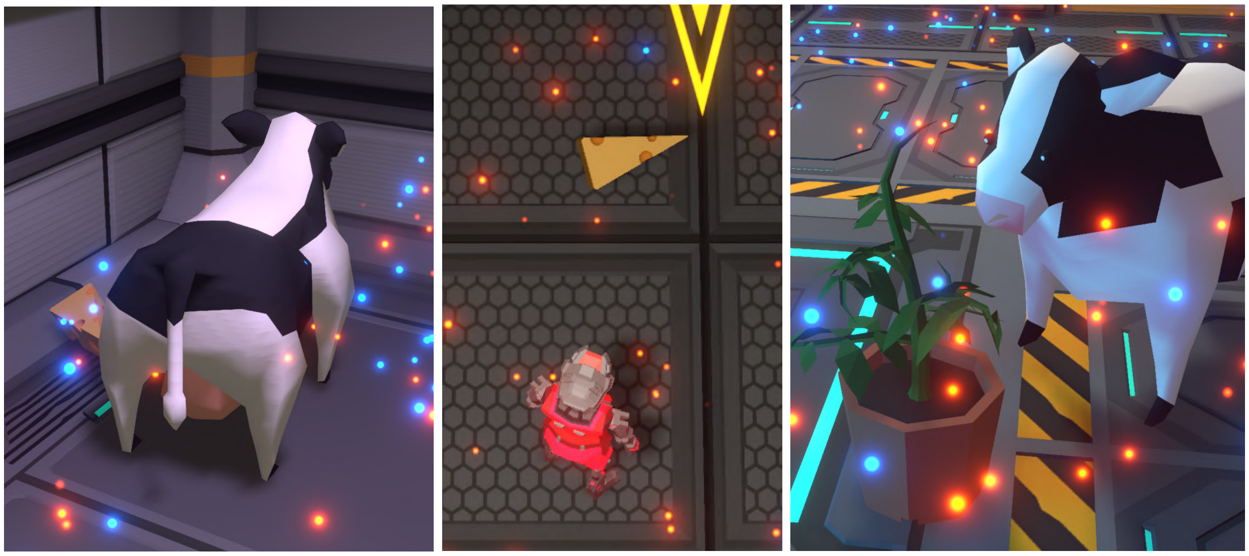




Cow

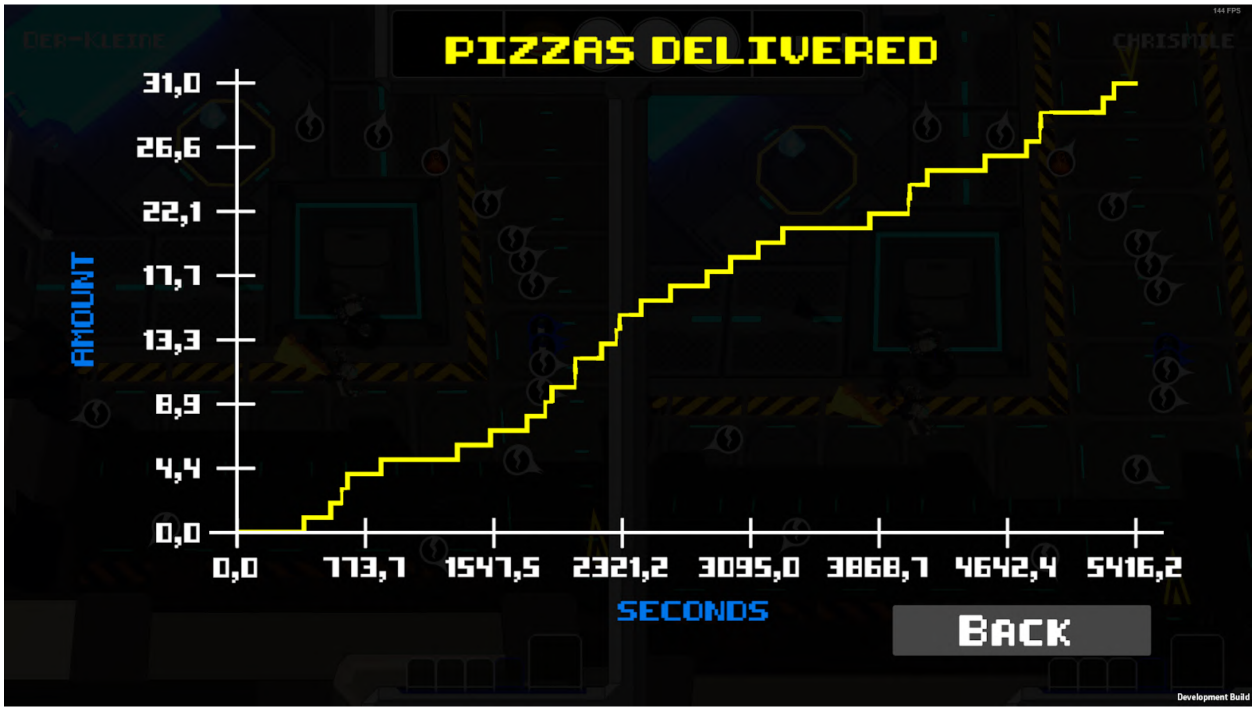
The cow roams around the ship automatically, gradually producing cheese when provided food and oxygen. players can interact with the cow and pet it to get vague information about its status, or feed it with any edible item found in their inventory. If the cow doesn't get fed and becomes hungry, it will automatically start eating plants and items.





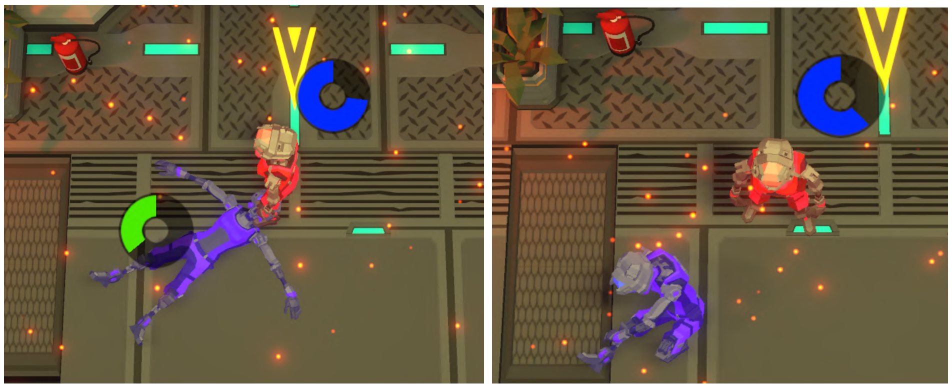
Statistics

During the game several events are tracked that can be viewed once the game has ended. These events include player deaths, asteroid impacts, number of pizzas delivered, money, cow pets, and many more.



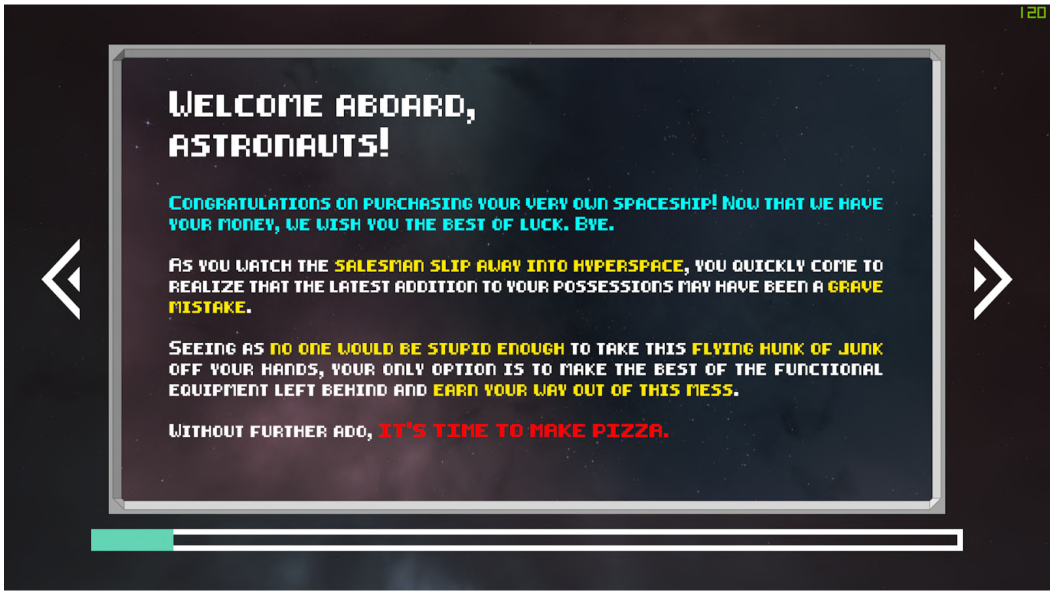
Oxygen Deprivation and Revival

If a player has passed out due to lack of oxygen, they can be revived by the second player. In the images below you can only see red air particles, namely carbon dioxide. There is no oxygen (blue particles) left in the room. The progress of revival is shown with the green circular bar. The blue circular bar indicates the time/oxygen left of the second player, before they will also pass out.



Tutorial

For now, we only added a tutorial in the form of descriptive slides. Each element in the game is explained in detail and their purpose or usage is depicted. The player controls are shown for controllers as well as keyboards for up to two players.



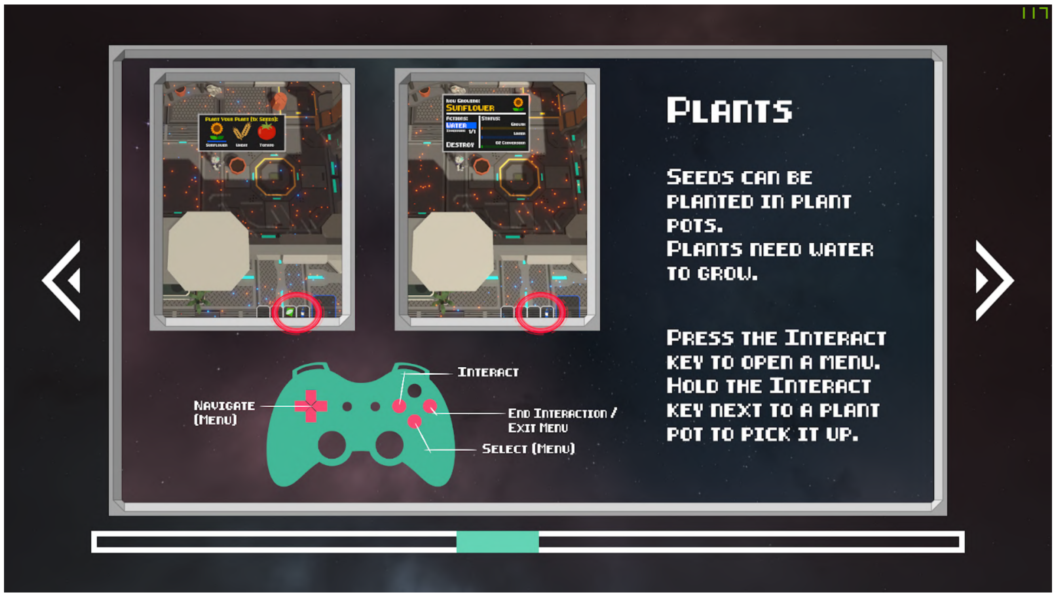
WELCOME ABOARD, ASTRONAUTS!

CONGRATULATIONS ON PURCHASING YOUR VERY OWN SPACESHIP! NOW THAT WE HAVE YOUR MONEY, WE WISH YOU THE BEST OF LUCK. BYE.

AS YOU WATCH THE SALESMAN SLIP AWAY INTO HYPERSPACE, YOU QUICKLY COME TO REALIZE THAT THE LATEST ADDITION TO YOUR POSSESSIONS MAY HAVE BEEN A GRAVE MISTAKE.

SEEING AS NO ONE WOULD BE STUPID ENOUGH TO TAKE THIS FLYING HUNK OF JUNK OFF YOUR HANDS, YOUR ONLY OPTION IS TO MAKE THE BEST OF THE FUNCTIONAL EQUIPMENT LEFT BEHIND AND EARN YOUR WAY OUT OF THIS MESS.

WITHOUT FURTHER ADO, IT'S TIME TO MAKE PIZZA.



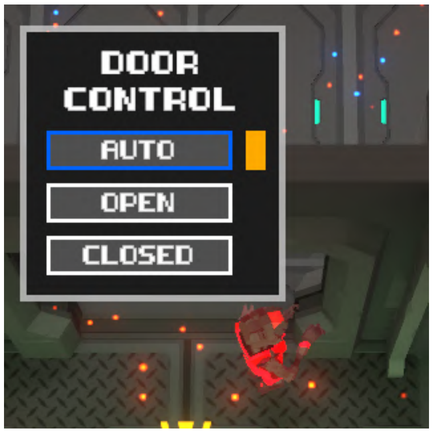
PLANTS

SEEDS CAN BE PLANTED IN PLANT POTS.
PLANTS NEED WATER TO GROW.

PRESS THE INTERACT KEY TO OPEN A MENU.
HOLD THE INTERACT KEY NEXT TO A PLANT POT TO PICK IT UP.

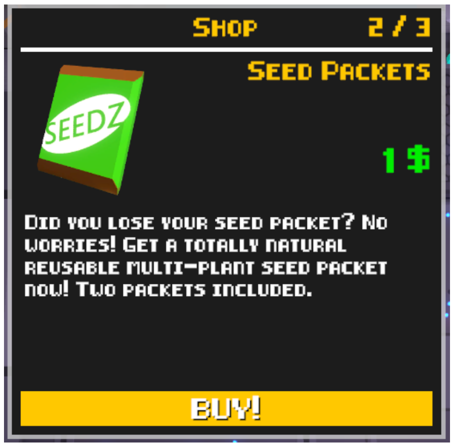
Door/UI menu

Doors can be set to open or close automatically when a player or animal approaches, or remain in a fixed state



Shop

The shop allows players to spend the money they earnt on additional plant pots, seeds and fire extinguishers.

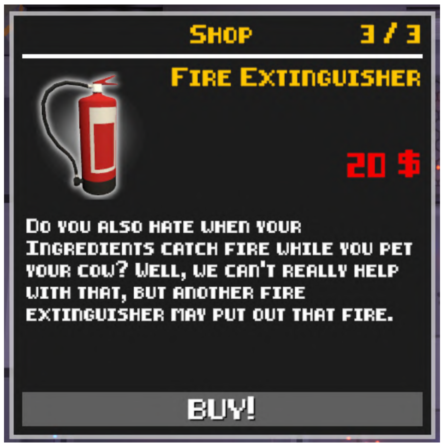

SHOP 2 / 3

SEED PACKETS

1 $

DID YOU LOSE YOUR SEED PACKET? NO WORRIES! GET A TOTALLY NATURAL REUSABLE MULTI-PLANT SEED PACKET NOW! TWO PACKETS INCLUDED.

BUY!


SHOP 3 / 3

FIRE EXTINGUISHER

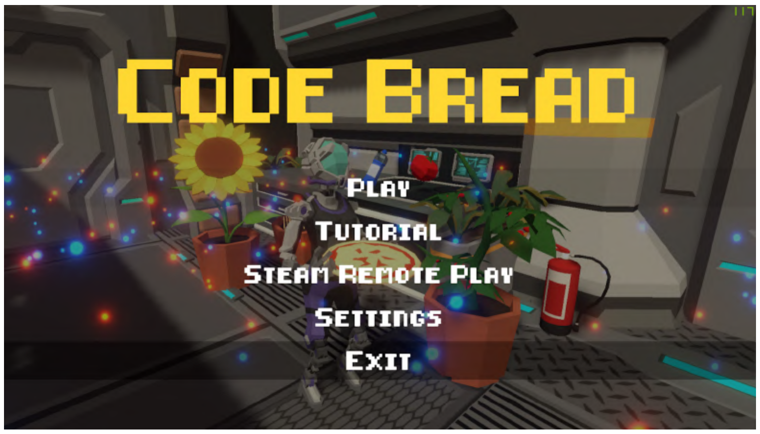
20 $

DO YOU ALSO HATE WHEN YOUR INGREDIENTS CATCH FIRE WHILE YOU PET YOUR COW? WELL, WE CAN'T REALLY HELP WITH THAT, BUT ANOTHER FIRE EXTINGUISHER MAY PUT OUT THAT FIRE.

BUY!

Start Menu

From the start menu the game can be started, the tutorial opened or settings such as resolution or audio volumes adjusted. To play the game online, other players can be invited via steam remote play, which only works if the game is started via steam. So far, only keyboard controls are supported for remote players and both characters could theoretically be controlled by either player using the keyboard.



Changes since Playtesting

(See Playtesting chapter for changes made between the alpha release report and playtesting report)

- Added dynamic and baked spot lights with color to every room to replace the directional light source
- Decorated rooms to visualize their respective function
    - Kitchen tools on top of ovens
    - Pictures/panels
    - Labels for special rooms
    - Advertisement for syn cola...
- Pizzas can become cold now -> less or no money earned
- Description for plants when opening the corresponding menu
- Plant pots automatically correct their orientation when knocked oversour
- Balancing
    - Wall repair times
    - Ice asteroid melt time
- Players now have different colors (red/blue)
- Fixed spelling mistakes
- Added a cheat menu and custom camera for creating the trailer
- Map of the spaceship

Our Experience

We stuck true to our big idea - we implemented our supporting feature, a functional 2D fluid simulation, to determine the oxygen or carbon dioxide concentration at discrete points in space. It is supporting our core gameplay, which is chaotic but cooperative, by providing additional tasks to the players. We are extremely satisfied with our result and after the end of this project we might work further on it.

At the start of this project we followed our development schedule quite closely. Apart from networking, we implemented all features listed until the first milestone, the interim report. The level was completed, the fluid simulation was running smoothly and we could already gather ingredients and bake pizzas. Objects can also randomly catch fire, which might spread further. After this checkpoint, we further improved the already implemented features and also decided to drop certain aspects such as the extras. However until the alpha release, we could finish almost all of our low, desired and high targets, as well as the functional minimum tasks while also adding novel content.

Although we completed all tasks we had planned, the development schedule was not really used as a guideline after a certain point of the project. Other stages of development were more helpful. For example, some results of the prototype were transferred directly to our game, namely all gathering or plant mechanics, ingredients and asteroid impacts. We only thought about adding uncontrollable life (the cow) to our spaceship after receiving feedback to our game idea and after starting to construct our prototype.

The playtesting phase proved to be important as well. We gained a lot of insights regarding the design and chosen features of the game and could improve it further in categories such as visual clarity, balancing and performance. Of course, several occuring bugs were also found and fixed.

Lastly, the questions after the presentations in class made as clarify and improve certain features further.

Personal Comments

1. What was the biggest technical difficulty during the project?

Learning URP.

Keeping the density of the quantities (oxygen/carbon dioxide) constant throughout one fluid simulation step was quite difficult. For this we had to compute the overall density per room by a simple reduction operation on the GPU. Therefore, the only way the density increases or decreases is by the influence of consuming or providing objects.

2. What was your impression of working with the theme?

From early on we set out to make a game that technically fits the theme, but not in the way that people would first expect when they think of climate change. This allowed us to quickly reach a stage where we didn't have to think too much about the theme. With a dynamically changing environment being the core element of our game, there was always a way to vaguely associate it with climate change, regardless of what the player's main objectives in the game may be.

3. What would you do differently in your next game project?

We would use Unity's High Definition Render Pipeline rather than the Universal Render Pipeline (URP). URP currently has restrictions e.g. no realtime shadows for point lights and limited light baking functionality. Implementing an interactive tutorial for the playtesters seems also important. Through this, they could learn and memorize the controls a lot easier.

4. Do you consider the project a success?

Absolutely. We even consider continuing the game after the course, as the feedback from our first playtesters was mostly positive, even though there were still many things to improve.

5. To what extent did you meet your project plan and milestones (not at all, partly, mostly, always)?

At the beginning, we mostly stuck to our project plan and milestones. However, this changed during the working phase for the Interim Report Milestone since we've made some changes based on the feedback we have received from the other groups and also later during playtesting.

6. What improvements would you suggest for the course organization?

- There were several areas where the project structure document didn’t seem quite up to date (e.g. unclear presentation time limit), though some of that can probably be excused due to the circumstances of this semester.
- Some parts of the project structure document and lecture slides weren’t as relevant for the development of a multiplayer/co-op game (e.g. playtesting)
- LRZ Meeting Tool’s screen sharing solution was less than ideal. Most of the development of our game was coordinated via Discord, where sharing your screen with other members of a call fluidly with audio and minimal latency wasn’t an issue.

7. Do you feel there wasn't enough time or that the schedule was too compressed?

For this project, we managed to implement most of our desired features and could fulfill our (adapted) targets of the corresponding milestones. However, the adaptations we made based on the playtesting could not be tested extensively again. An iterative process with a testing and improving cycle might take up more time than available but could be a good addition.