# ASTERS PLANETAI

## Game Design Document

**Authors:** Alvin Häger, Nathan Bhat, Tom Axblad, Ludvig Kindberg

**Date**: 6th Of December, 2019

**Version:** final version

## Table of Contents

# Game Overview

## Game Concept

The game is a 2D puzzle platformer in which the player takes control of a humanoid alien character searching for pieces of his starship to be able to escape from a planet that is going to explode. In keeping with the metroidvania style genre, you explore a large interconnected world map, solving puzzles to progress. The game has a dark, post-apocalyptic, sci-fi and futuristic theme, which is captured in traditional vintage style in pixelated graphics. The game combines exploration, challenging platforming, intriguing puzzles with a unique core mechanic that presents an exciting twist on traditional platformer games.

The gameplay revolves around the core mechanic of the game - this comes in the form of a  'time reset' ability. This ability allows your character wind back the clock to the start of the game, resetting time to $t_0$. You are then able to interact with previous iterations - 'time clones' - of yourself, which repeat the actions you previously made before using the ability.

You can change your physical location you go to when resetting time by activating checkpoints, which are scattered about the map. Intelligent use of these clones will allow the player to solve puzzles in different ways by working together with the time clones to progress through mysterious landscape of the planet in order to collect pieces of your starship. For example, in order to open a door that is unlocked when a button is pressed down, you could press the button, reset time, wait for your previous self to press the button, and then proceed to pass through the door.

The mechanic has an endless number of possible applications, which sets a solid foundation to build the game on and extend in the future. By exploring the map you are prompted by the environment to learn how to use your ability in new, creative ways, which in turn allow for more exploration as new areas become accessible to the player through these newly learned applications. The planet you explore is full of branching caves, which you can follow upwards to find your way to patches of the surface of the planet. In Asters Planetai, exploration is meant to be part of the reward - the final vision for the game is one where the beautiful scenery of an open world combined with dynamic, non-deterministic puzzles to entice the player into exploring more.

## Game flow summary

The player is placed on to a free-roaming map to explore. The player's goal is to collect the pieces of their starship so they can leave the planet. The different parts of the ship are scattered in different areas that the player has to find in order to collect them, and this can be done in any order the player wishes. You can read more about the story, setting and character design in the Story, Setting and Character section. The map contains exploration areas as well as 'levels' that are integrated into the map. These puzzles or exploration areas that need to be overcome by the player using the core time reset mechanic.  Once the player collects all the pieces of the starship, they have completed the primary objective of the game and can leave the planet, and the game will 'end'. However, upon reloading the player can still keep exploring the planet.

## Genre

The game is a metroidvania style puzzle platformer in an open world setting. It has undercurrents of different themes and gameplay inspired by games such as

- Gris
- Limbo
- Rain world
- Braid
- Ori and the blind forest
- Vvvvvv
- The Talos Principle

## Theme and game 'feel'

The game combines sci-fi, post-apocalyptic, alien and futuristic themes into a world that should feel mysterious, menacing, vibrant and beautiful. The feel of the game should therefore be dark and brooding, which should be reflected in the visuals and sounds of the game.

## Unique selling point

The unique selling point of the game is the fact that it presents a new twist on an old concept, combining exploration, puzzles and a unique core mechanic into the vintage pixel feel of traditional platformers. The time reset ability allows for dynamic gameplay, i.e. using them to solve puzzles in different ways or just cover the entire map with a stream of time clones, which can be used to hilarious effect.

## Target Audience

Ages ranging from 15 to 24, with a focus on teenagers and young adults. Although this is the age group we focus on, the graphics and the desired look and feel of the game are limited by time constraints in terms of the graphics that can be realistically achieved in this time. The target audience are gamers that are interested in indie games but the game should be easy to pick up and get into.

## Target Platform

Windows PCs, with keyboard or gamepad.

## Feature set

**Time reset and time clone mechanic**
The core mechanic of the game, which allows the player to reset time to the physical location of the most recently visited checkpoint, spawning non-controllable clones that perform the actions performed by the player before resetting time.

**Time freeze mechanic**
In addition to the time reset, a time freeze mechanic allows the player to freeze themselves in time, which when resetting time allows you to use the time clone as a platform to jump on and off of.

**Explorable open world demo**
The game should have an explorable open-world demo where the player feels encouraged to explore and progress through the map by solving puzzles 'levels', which are integrated naturally into the open world setting. The actual setting is that the cave and barren surface of the planet, with different branches leading into different areas.

**Custom 2D pixelated graphics for tilemap and game assets**
A custom tilemap using pixelated graphics as well as custom sprites for the main character and all the other game entities used in the game.

**Puzzle entities that the player can interact with to solve puzzles**
The game should contain many different game puzzle entities to interact with, see puzzle entities for the full list. But to mention a few

- buttons that connect to and can deactivate laser walls when they are pressed
- geysers that shoot out air currents that push the player upwards

## Future features to be implemented

**Intelligent enemies that roam the world map (optional feature)**
In the future, the game should contain enemies with lifelike AI that roam the world and interact with you in different ways. Rain world is an example of a game that does this in a similar way to what we envision, as intelligent enemies roam the map and act in lifelike, dynamic ways.

**Inventory and collection of pieces of the starship** **(core feature)**
in the final conception of the game, we want to be able to include the ability to collect pieces of your starship and display these in an inventory, so that you can feel like you are progressing, which also spurs the player on to explore. This is something that should be implemented in a final version, as it adds to the core to the idea of exploration by giving the player a sense of tangible progression and more reason to explore the map.

**Ability to speed up time for clones (quality of life update)**
In order to prevent the monotony of having to wait around for a particular clone to reach a critical point in its time cycle, for instance pressing a button that opens up an area for you, a future feature that would improve playability would be the ability to fast forward the movement of the clones. If enemies are implemented, one would have to find a way of combining these two in a way that feels natural, as enemies should also be able to attack and interact with clones.

**Teleportation between certain checkpoints** **(core feature)**
In the final conception of the game the large map would require that you can teleport between certain biomes or large areas from different checkpoints. Think of Skyrim for example; here you can teleport between checkpoints in large sections of the map.

# Project scope

## Demo

Due to the very limited time of the project we will implement only a small world map with 2 puzzle levels and one big exploratory area, with one extra level that simply introduces the core mechanics and is used in the demo. The majority of the core mechanics necessary to our conception of the gameplay have been implemented, leaving out some features that would have improved the game, and a few that would need to be implemented to complete our vision for the game. One feature that is definitely required is the ability to collect pieces of your ship and a visual representation of your progression in this goal, as this is a core part of our game. For the purpose of the demo, this feature will not be implemented.

We have implemented many puzzle entities in the current game, see puzzle entities for the full list of implemented entities and how they work.

## Final version

In the final version of the game, a lot more levels and content would need to be created, and the scenery in this game would also need to be improved upon. This would need to be done to justify the price tag of the game.

# Gameplay

## Game Progression

At the beginning of the game, the character is placed in an open area on our fictional planet, where the player can move in any direction to start exploring. The player can choose to wait to see what happens if they allow the planet to explode. The player will be able to move through a number of branches that lead into different interconnected areas, which consist of exploration areas as well as puzzles that need to be solved to progress. As they move they will encounter checkpoints placed in strategic areas, which are placed so that the player doesn't have to repeat several areas upon dying. As the player moves through the different branches, the puzzles in the room get progressively more complicated, combining puzzle entities and mechanics learned in previous areas.

## Objectives

The objective, as mentioned earlier, is to collect the pieces of your ship by solving puzzles and exploring the open world setting that is the planet you find yourself stranded on. The collection of the pieces of the ship is currently not implemented.

## Play Flow

As the game allows you reset time to create clones, the general play flow should be the player exploring the planet and progressing forwards through puzzles. In the puzzles, the player will reset time several times, trying out different strategies by trial and error to see what works as a solution to a particular puzzle. As the game should ideally have many branches, this should allow for replayability, prompting you to play the game again to try other sections of the map.

## Map structure

The open-world map where the game plays out can be broadly categorised into two main types of areas:

- exploratory levels
- puzzle levels

**exploratory levels** work to establish a sense of worldbuilding by fleshing out the story and theme of the game. This is done through visually by presenting the player with three types of areas: The surface of the planet, which have sandstorms, and the underground caves inside the planet, which are always blue for the purpose of the demo. An example of an exploratory level is the Exploration Geyser level.

**Puzzle levels** present a challenge for the player that combines puzzle entities in interesting ways. These should be integrated seamlessly into the environment and feel like they belong there. Example of this is the Balancing platforms level.

The map should feel 'open' and nonlinear in terms of choice, allowing the player to pick which branch of a cave they choose to follow or the direction they decide to go in.

# Typical puzzle structure

Each puzzle should feel unique and distinct from other puzzles, introducing different ways to apply the time reset clone mechanic to solve the puzzle. This should start out as simple, leading to more intricate puzzles. The puzzles will combine puzzle entities to create dynamic puzzles that should feel meaningful and rewarding. The game's theme and feel should reflected in the puzzles, in terms of the puzzle entities and how they fit into the world. See Levels to see examples of how the puzzle entities are used in levels.

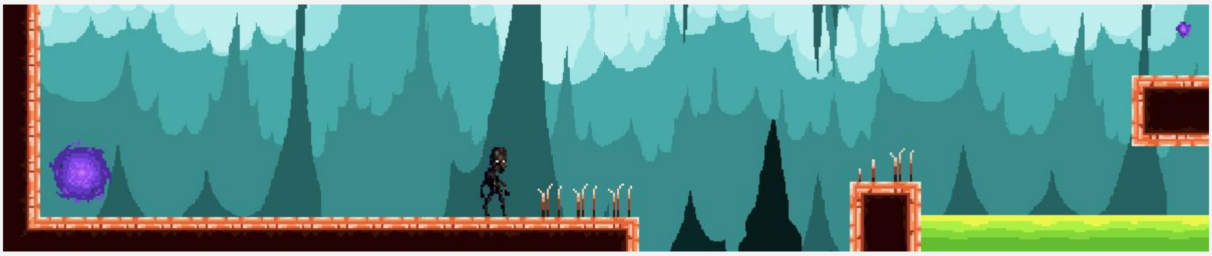
## Puzzle entities

The following entities are used to create puzzles by combining these elements in different ways to create puzzles. Each entity is described below, and the visuals for them are presented in the game art section.

### Checkpoint

Checkpoints have been mentioned before, they are the spawning point for the player and the clone and are activated by simply walking into them. They serve as an important part in the level to assist the player when solving different puzzles. This is because by placing checkpoints out with reasonable distance between them, players can avoid having to replay certain areas of the game upon dying. They also allow your clones to reset and get to where you located faster, as they always have to travel from the last visited checkpoint.
In the game, the checkpoints are represented by time portals, which allow the player to tether their physical form to a location when they move through time.

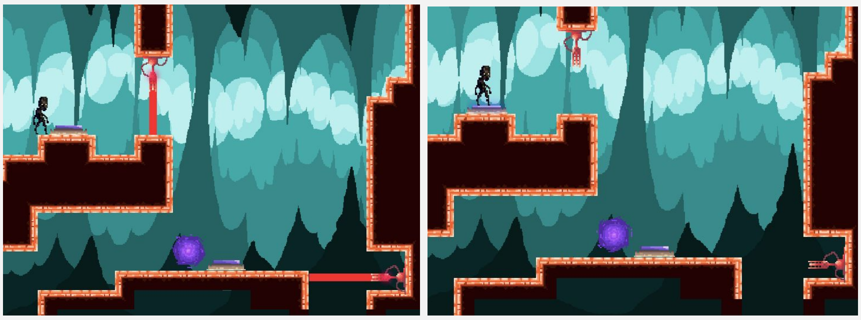
Here we see for example two checkpoints that separate a dangerous area where the player has to jump over a poison lake. After jumping over it, if the player dies after getting to the next checkpoint, they will not have to redo this area and respawn from the currently inactive checkpoint in the picture.



## Button switch

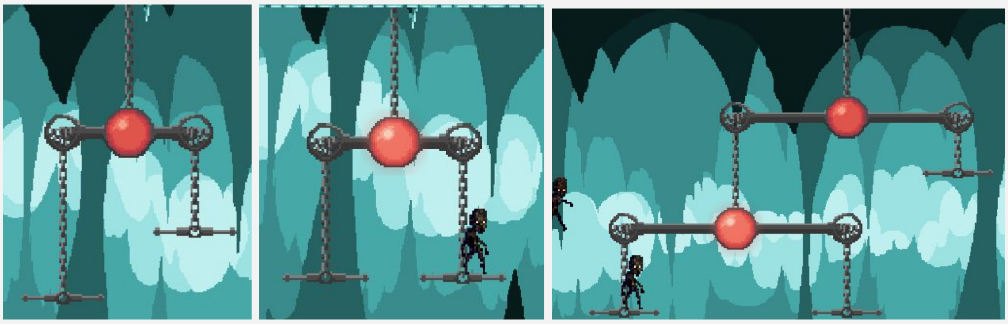
The button entity acts as a switch for other puzzle entities in the game. By switch we mean that they can activate, deactivate other puzzle elements. They can be activated by either the player or a clone standing on them, same results if one or twenty more are standing on the same button. For example, a button connected to a laser turret can turn it on or off.

Here we see a button switch that is connected to two laser turrets. In this case,when the player is on the button switch, the lasers are deactivated. Buttons can also be connected so that lasers fire when you are on the button.



## Balancing platforms

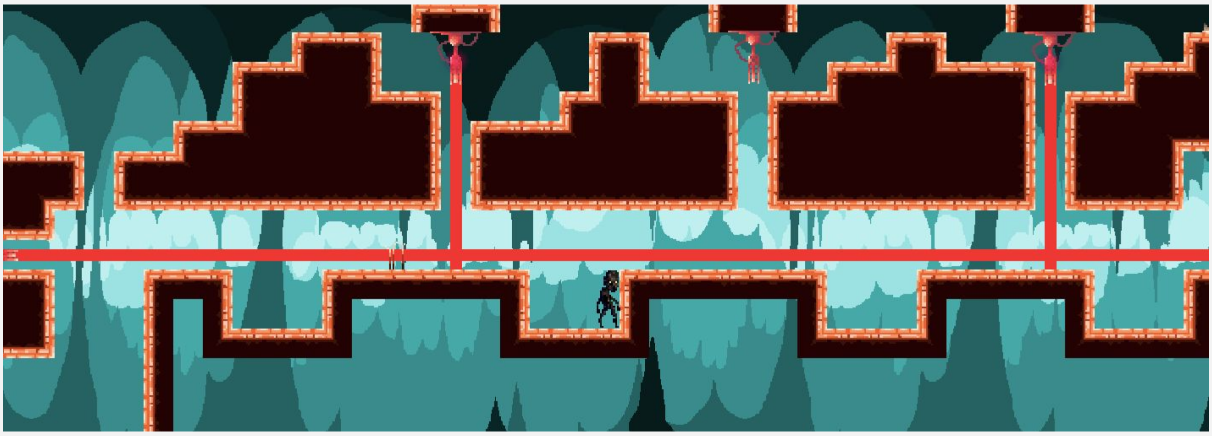
The balancing platform entity is about matching the weight of two scales.
This is done by placing enough clones on one of the platforms to even out the weight discrepancy. The balancing platforms can be connected like a tree, meaning that one platform can in fact be another set of balancing platforms, so you can have a branching set of platforms. The balancing platform can also be connected to other elements like the laser turret, allowing you to create puzzles that require you to solve the balancing problem. Below is an ingame picture of an active and inactive balancing platform, as well as a balancing platform with 2 layers.



## Laser turret

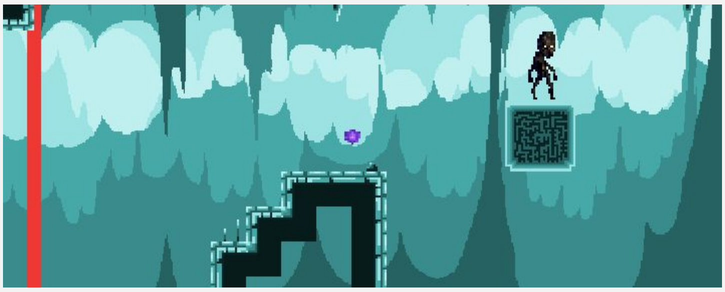
The laser turret entity can be used as a door that can be controlled by the button entity or the balancing platform entity in the game. Basically, a turret shoots a laser in a straight line that will kill the player upon contact. This basically creates an impassable  laser ‘wall’ that can be used in puzzles. The turret has two states: active and inactive. when the turret is active the laser is continuously firing, typically used to block off certain areas.

As an example, here we can see how the player has to navigate through a laser that fires periodically from the leftmost turrets, whilst also making sure to not hit lasers shooting from the top!



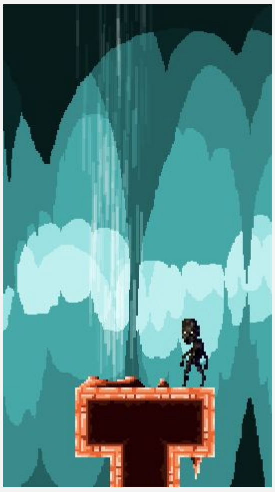
## Moving platform

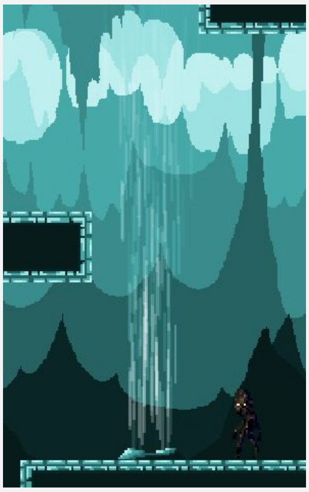
The moving platform is used to transport the player and/or the clones in any direction such as up, down, left or right. The platform can be controlled by the button entity or the balancing platform entity, either stopping the platform or moving it. Picture below shows how a player rides a moving platform.

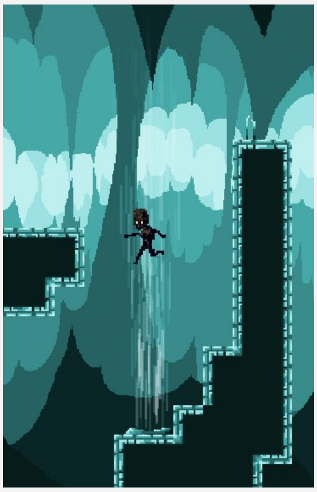


## Geyser

The geyser shoots out air currents will push the player upwards, resulting in the player being able to travel upwards much further than a jump. At the top of the air stream the player will bob up and down as the air acts as a buoyant force. The geyser can be controlled by the button entity or the balancing platform entity, which can be used to turn it on or off. The geyser comes in two different colors, as shown below.

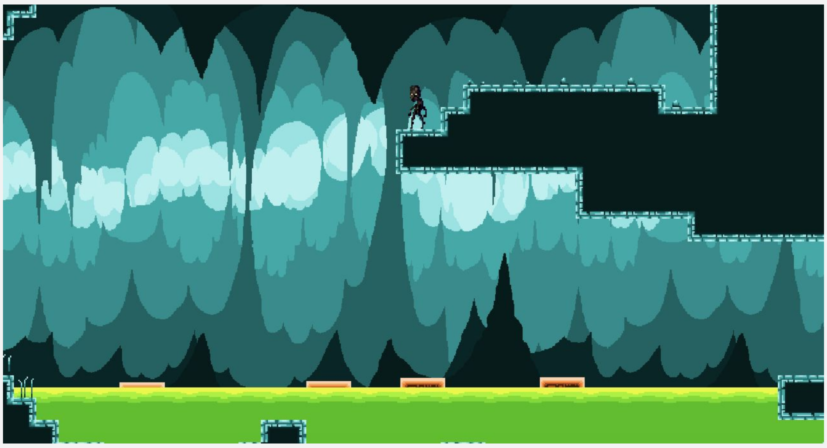






### Acid

The acid entity is basically a pool of acid that will kill the player or the clones upon contact. The acid cannot be controlled in any sort of way. This can be used to create for example acid lakes.



# Mechanics

This section describes all the game mechanics of the game - how they work and how they apply to the game in more detail. Brief example scenarios of how they can be used are also given. The controls for all mechanics can be found in the section *Interface - control system.*

## Physics

The physics of the game is synonymous with that of the real world, and should therefore reflect reality's physics, albeit in a 2D world. With some slight modification, due to the nature of the humanoid you are playing.

## Player movement

The player can move in three general directions left, right and jump. The left and right movement are determined by how long you are pressing the buttons resulting in a short presse will result in walking and longer running. The jump height is also determine on how long the button is pressed, a short press will result in a lower jump and a long press will result in a high jump, there is a predetermined maximum jump height. The player is NOT able to get on top of a frozen clone if it was frozen at the top of a max jump.

The physics of movement should be such that they feel natural and intuitive. Although the movement shouldn't feel slow-paced and restrictive, it should feel realistic in terms of movement speed and jump height.

Player movement is used to move the alien character around the map. All of the frames of the animations can be found the spritesheet of the game art section.

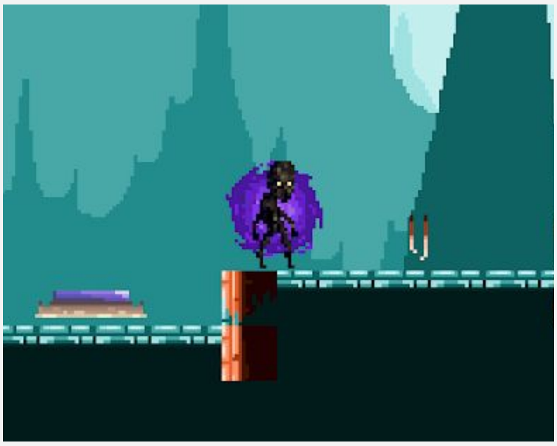
## Time reset mechanic

The time reset mechanic allows the player to reset their position to the last visited checkpoint on the map. It also creates permanent time clones, that play out the previous actions of the player before the reset button is pressed. These clones are permanent, unless destroyed by the player themselves or another hazard in the game, e.g. guns/lasers, and each time the reset button is pressed a new clone is created, while the others simply replay the same actions as before. When a clone reaches the end of the time cycle, where the player pressed reset, they stand in place.

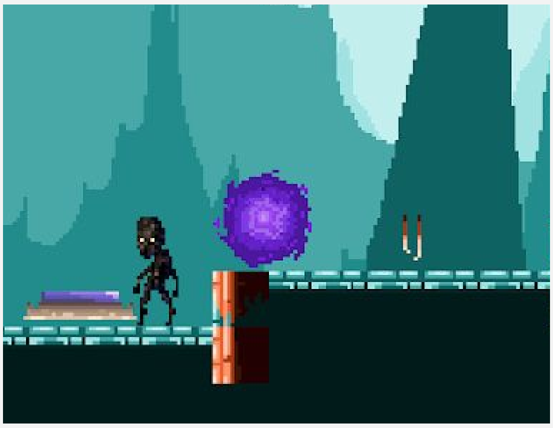
Pressing the reset button resets the player's position to the last visited checkpoint, whilst also resetting all the clones to their starting checkpoints, upon which they will follow their recorded paths until the end, upon which they will remain idle at their end positions.

This is the core mechanic of the game and is going to be used a lot, main use case is to get your clones to perform actions that helps the player solve puzzles.

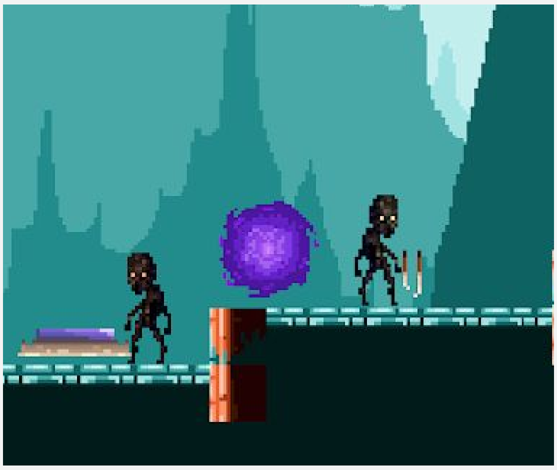
The technical challenges and implementation details can be found in Time mechanic - implementation details



The player standing at a checkpoint, first spawn at the checkpoint.



The player walks to the left and at this point hits the reset button. Picture taken just before the reset happens.

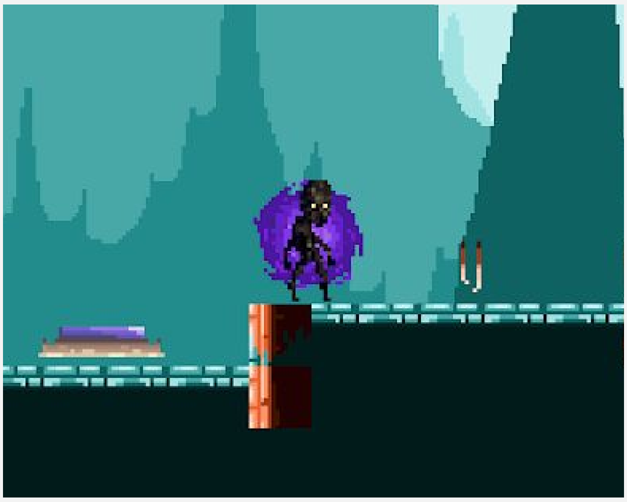


The player walks to the right and we can see the clone walk to the left, and stop and stand still at the point where the reset happened.

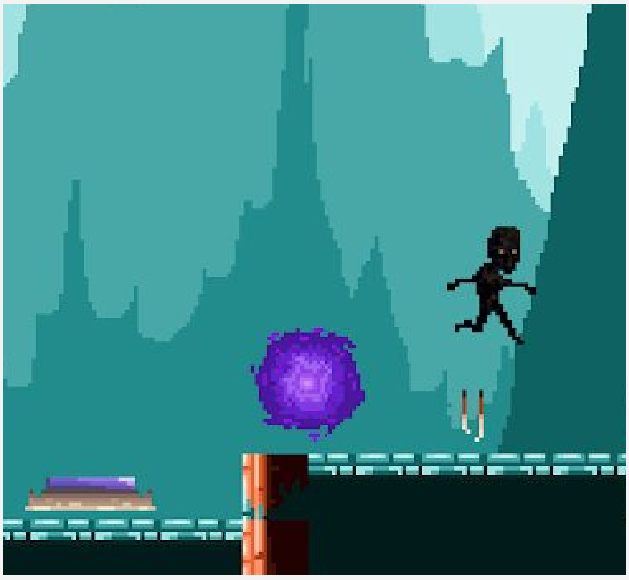
## Time freeze mechanic

The time freeze mechanic allows the player to freeze time at any time anywhere on the map. This results in that the player gets frozen in time and can not move any more, resulting it that it most reset time. The clone will, as expected due to the time reset mechanic, rerun all that the player done including freezing. Thus becoming a solid object and can be used by the player to stand and jump on.

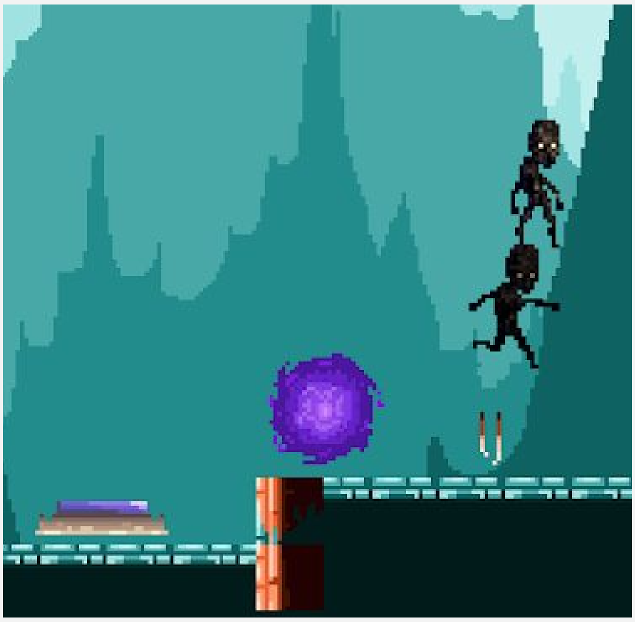
This can be used in many different ways, for example building stairs, getting to a higher ground to jump further.



The player standing at a checkpoint, first spawn at the checkpoint.



The player jumps and freezes the time mid air, the player is now frozen until the time reset mechanic is used.

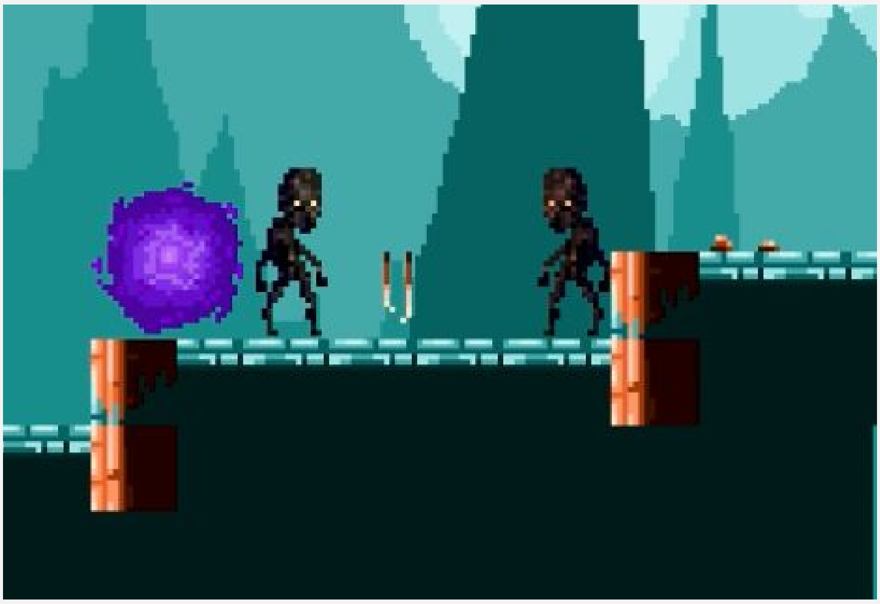


The player standing on top of the frozen clone.

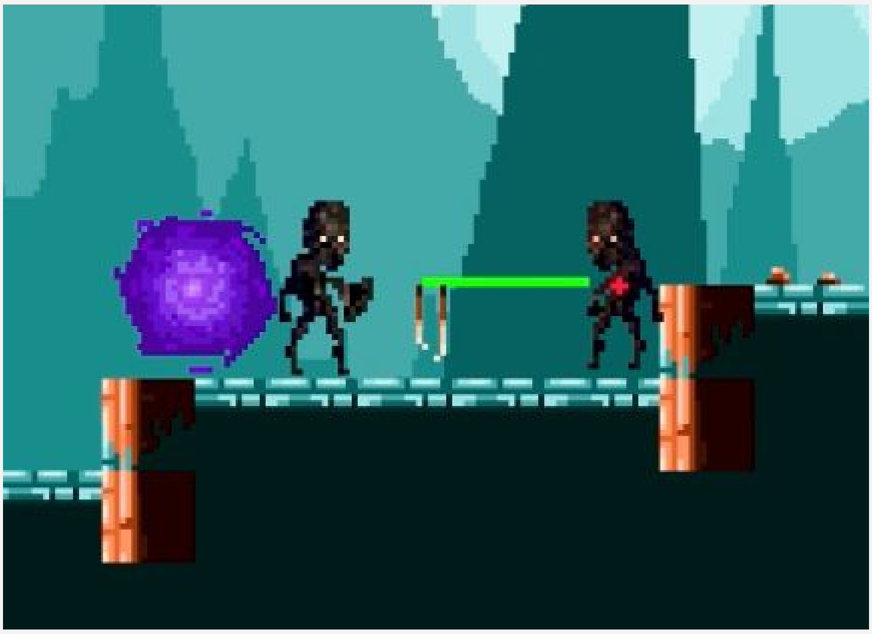
## Shooting

The shooting mechanic allows the player to shoot the clones, removing them from the timeline. Thus they cannot perform the sequence of actions that they were supposed to. Due to the time reset mechanic, both of the clones(the one shooting and the one getting shot) will be reset once the time is reset. If the player shoots a clone that will shoot another clone, the second clone will continue as if nothing has happened and perform the actions it is supposed to.

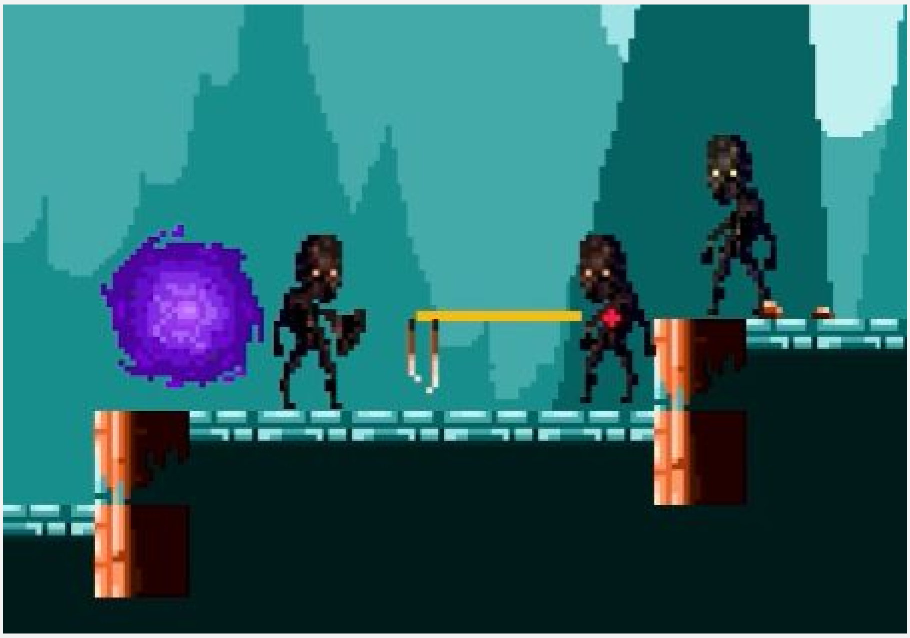
This can be used to remove a clone that is doing something it is no longer supposed to do, or to correct mistakes in the first attempts of completing a level or puzzle.



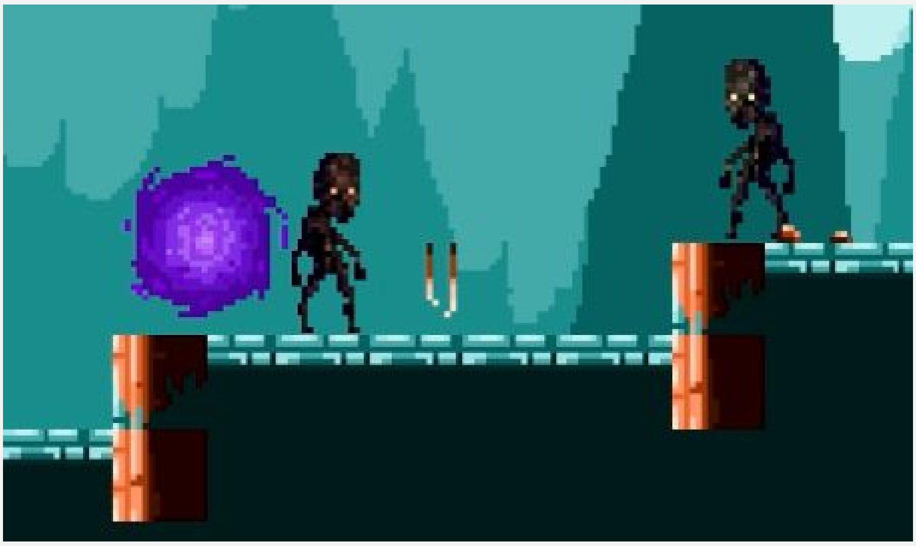
The player to the left having a staring competition with the clone to the right.



The player to the left shooting the clone to the right.



The player all the way to the right watches as the clone shoots the first clone.



The player and the second clone remain unscathed.

## Clearing clones

The clear clones allow the player to remove all of the clones that spawn from the currently active checkpoint. This can only be done when the player is at the checkpoint and will only as mentioned affect the currently active checkpoint due to it being the one that the player is standing at.

This can be used to get a clean start from a checkpoint to complete a level or part of a level due to that a lot of clones spawning from the same checkpoint can get confusing.

## Inventory and zoom out (Not implemented)

The inventory and zoom out mechanic allows the player to do two main things. The first being to check the progress of the game, by checking what parts of the starship have been collected and what remains to collect to rebuild the ship and escape the planet. This will display some sort of design with greyed out parts that are missing. The second thing this will allow you to do is zoom out to be able to view the map and get an overview.  Thus it will be made so that it looks like the player pulls out a map and the camera is zoomed out the get an overview of the current level/puzzle the player is trying to complete. This allows the player to get an overview of what is to come and how different parts of the puzzle can be related to each other as the levels get more complex. Furthermore, this also allows the player to see what the clones are doing as they wander the map.

## Fast forward (Not implemented)

The fast forward mechanic allows the player to fast forward time, thus removing the aspect of needing to wait for the clones to do certain task. This will be implemented in the way that the clones perform the same actions, just at X time the normal speed. During this time, the player(character) is highly focused and thus not able to move or do anything else. As mentioned the main reason behind this is to not have the player being stuck waiting for the clones to solve the puzzles and deactivate a certain turret to be able to move to the next level.

## Teleport (Not implemented)

The teleport mechanic allows the player to teleport between different checkpoints/time portals. Due to the exploratory part of the game, the larger the map grows the more convenient it might become to be able to teleport between key points between expansive areas. Teleporting will also reset time.

# Story, Setting and Character

## Story

The story behind the game is intentionally ambiguous, inviting the player to draw their own conclusions and interpret the world themselves. The story is in loose terms defined as such - you are an alien race in possession of a unique ability to reset time and you have crashed landed onto a planet close to a currently exploding sun. During entry into the planet's stratosphere, your starship has been torn apart into pieces that are scattered around the planet, leaving you to collect them to rebuild your ship and escape before before the sun's  destruction.

The passage of time will be presented visually by a sun moving left to right in the background before exploding. As the alien character's ability allows you to reset time, you can escape your demise by going back in time - this serves more as an interesting story element to add suspense and tension, leading to progressive changes to the visuals of the game. (i.e. making the screen look redder, i.e. screen shakes once it's nearing the explosion) What happens when the sun explodes? In a final version of the game, the player should be able to find out, by waiting it out to see how their doom will play out.

You could also theorise about the ability itself. Basically, it allows you to create an alternate timeline of yourself. Since you always reset to $t_0$ , the only thing you change when you reset time is the where; the physical location your form is tethered to. This is represented in game by checkpoints in the form of time portals. When you reset time, you will restart from the location of the most recently visited checkpoint, and the previous version of yourself will replay their actions again, whilst you represent the new timeline to be written and controlled.

## Setting

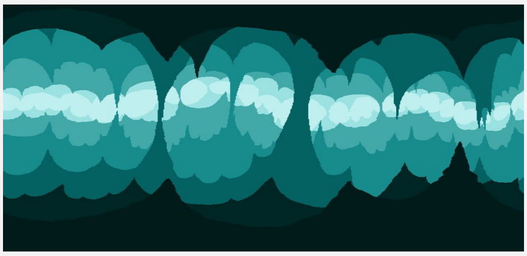
The setting of the game is the planet you are left stranded on after losing your starship. The setting has two main colour palettes which are used to create different types of areas, which are:

- caves
- surface of the planet

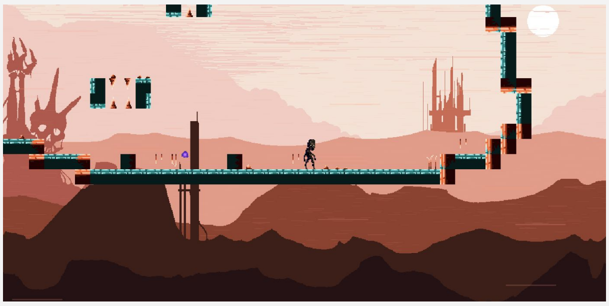
The environment come in two different visual palettes (in our current version of the game) The frosty caves are icy blue, in contrast to surface tileset,  which is fiery orange. The surface of the world is subject to sandstorms, which is reflected in game by particle effects and sound. During the game, the idea is that the player can venture up through the cave-like tunnels towards patches of the planet's surface.

The setting should convey a feeling of scale and mystery, with the surface looking more barren and post-apocalyptic, whilst the caves look very mysterious and vibrant to explore. In the current implementation, the surface uses the fiery cave tileset. See here for the tilemap used for the world's environment

These factors were taken into account when creating the design for the backgrounds to the platforming games, which switch between the surface of the planet and the cool caves below.



This picture shows the particle details used for the sandstorm effect above ground

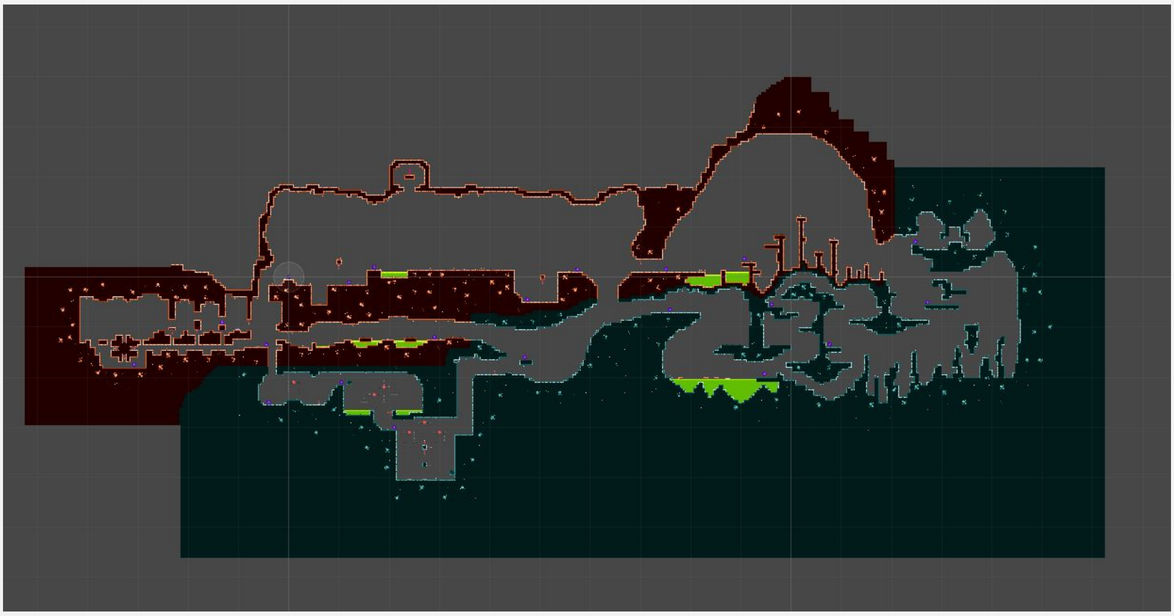


## Main character

The character you play is a bit of a mystery. he's clearly some kind of alien, but beyond that we don't know. The player himself is meant to interpret the story of the character. The visuals and sprites used reflect the mysterious nature of the alien character. The design was inspired by games like limbo and also by this picture.

## World Map

This is an overview of the world which combines all of the levels into a cohesive map to explore. The world map consists of  the introduction level, laser coordination, exploration Geyser and balance level. They are patched together  in a way that encourages exploration and allows the player to revisit areas, whilst also connecting them in a way that feels natural.

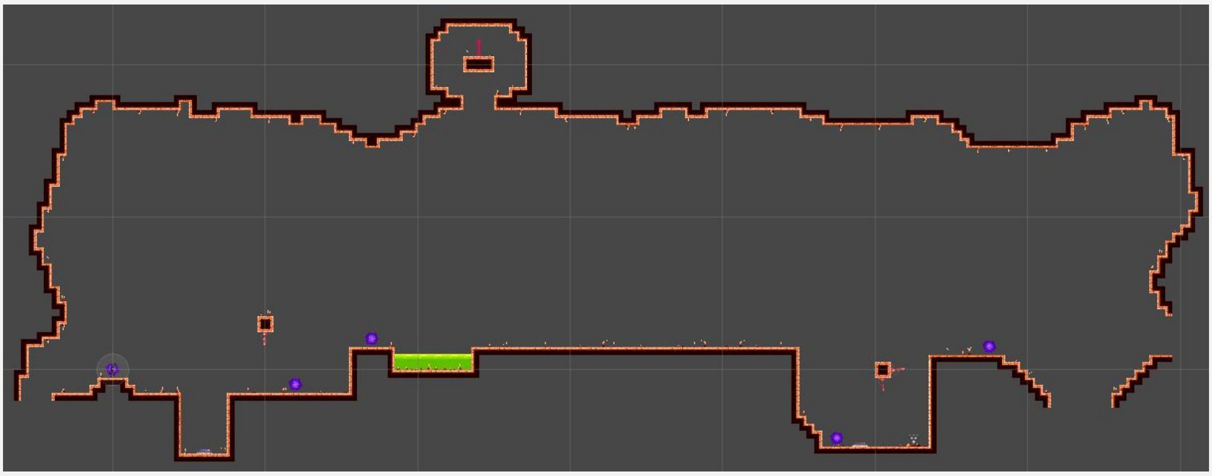


# Levels

This section describes the levels that combined to create the world map.  As mentioned earlier, levels can be exploratory levels that serve to establish world building or puzzle levels where the player has to solve a puzzle to move forwards.

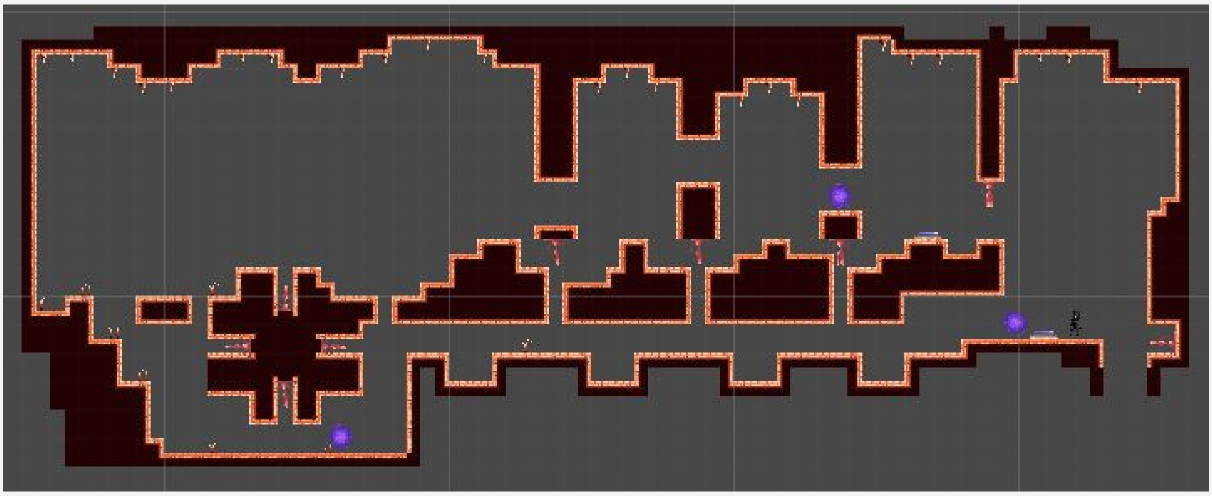
## Introduction level

The introduction level is located in the top middle part of the world, also acting as the starting point of the game. This level takes place over ground.
This level is used as a demo that introduces a lot of the different mechanics available in the game, and its purpose is to demonstrate many of the gameplay mechanics of the game so far. In the real version of the game, the mechanics will be introduced gradually in different areas, instead of all at once.



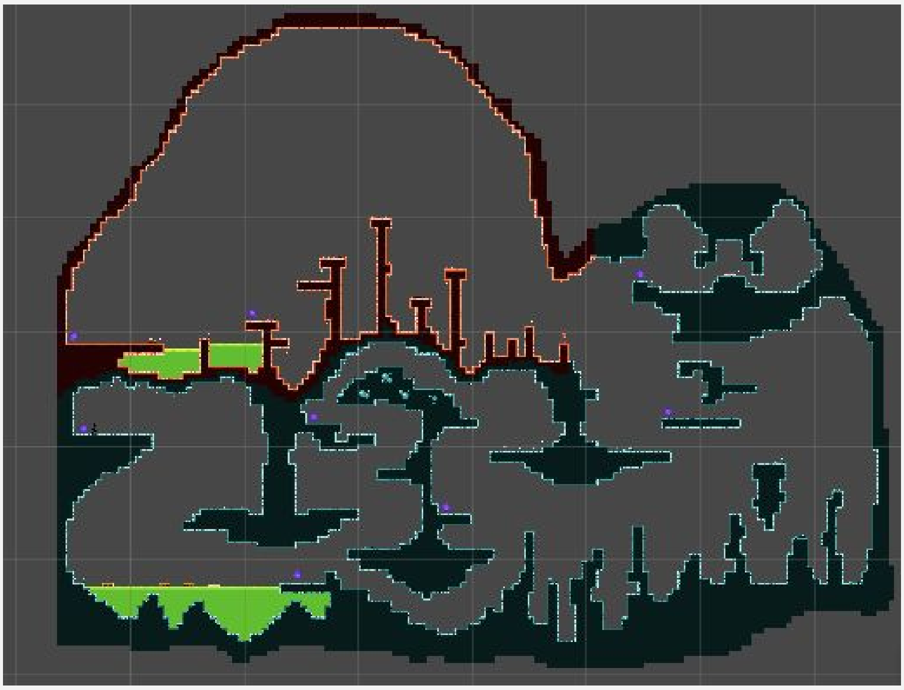
## Laser coordination level

The laser coordination level is a puzzle level where the player has to maneuver through dangerous lasers that shoot out of turrets in an enclosed, cavelike area. The player has to coordinate with their clone to turn lasers on and off by using button switches to make it through the level. The location of this level is to the furthest most left part of the map, and is taking place over ground.



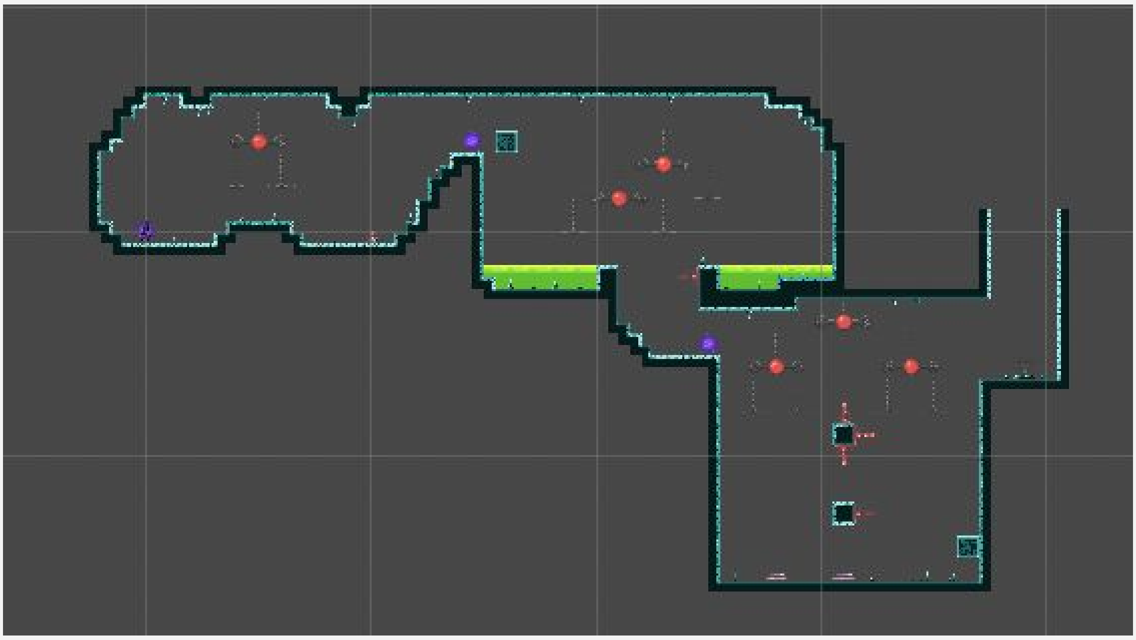
## Exploration Geyser level

The exploration Geyser level is an exploratory level that has two main areas, one with frosty cave tunnels and another cave area above ground, as well as other minor areas. It contains a poison lake where platforms bob around on the poison water for the player to jump on, a frosty cave area for the player to explore. Across the map there are pillars with geysers on them which shoot out air currents the player can ride and use to float around. This level contains some puzzles but is generally meant to be explored by floating around. The level is located to the furthest most right on the map, taking place both under and over ground.



## Balancing platforms level

The level is located to the middle bottom part of the map located directly below the Laser Coordination level, taking place underground. The purpose of this level is to introduce the balancing platforms mechanic, where you have to solve balancing platform problems to progress. This also utilises the moving platforms, acid, turrets and buttons. The general idea is that after the player plays this level they should have an understanding of how the balance platforms work and how to solve them.



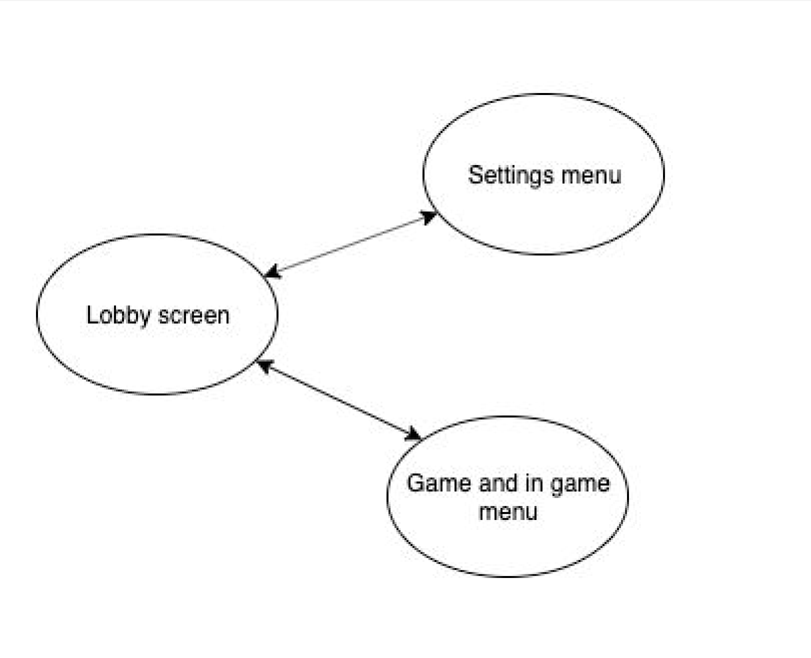
# Interface

## HUD

The game should have a very simple HUD when in game.The information that is displayed should be minimal. Due to the countdown of the planet being represented by the sun's movement from the left to the right of the screen, no additional indicator is required. Any other information is not needed for the gameplay, the inventory can be accessed through a button.

## Menus

The idea is to have very few menus, due to the few amount of things that can be changed, so the main settings can be accessed from the settings menu. Specific control settings such as the settings for the keyboard controls, screen settings, etc.  will be found by navigating through the in game menu.



## Camera

The camera system uses two main settings: One setting where the camera follows the player with a brief delay, and another where the camera uses a system of pivots that are placed on the map to smoothly move the camera between static positions as the player moves around the map. This pivot system is used to 'pivot' the camera over to a new position as the player moves around.

For example, if a player reaches a ledge that has poison underneath it, the camera will try to position itself so that the poison is in clear view, whilst also maintaining view of the player.

These two camera settings are used together in the game depending on what suits the particular scenario best, and the camera changes position and setting throughout a level.

## Control System

The default controls for the system are the following:

| Action | Keyboard control | Xbox controller |
|---|---|---|
| Jump | Spacebar | A |
| Walk/run left | A | Left joystick |
| Walk/run right | D | Left joystick |
| Freeze time | Q | B |
| Clear clones at checkpoint | E | Y |
| Shoot | C | RB |
| Restart time | R | X |

# Audio

All of the audio in the game were created by the team. This includes all sound effects and Music/Ambience.

## Sound Effects

### List of Sound Effects

- Jumping
- Shooting
- Dying
- Time Reset/clearing clones (This is the same sound effect at the moment, in the future we would like to distinguish between these two cases)
- Pressing/depressing a button switch
- Activating a checkpoint

The sound effects were created to sound 8-bit and retro, as it fits our pixelated artstyle. When creating the sound effects we felt that many of them were too cheerful for the darker themes of our game, so we decided to change some of the sound effects (like the jump sound).

### Future Sound Effects

We would like to implement more sound effects in the game, such as a sound that plays when you readjust a balance platform, and when you successfully balance a balance platform. We would also like to include a sound effect for when you freeze time. As mentioned earlier we would like different sound effects for resetting time and clearing clones from a checkpoint.

## Music/Ambience

- Main theme
- Wind/Sandstorm

The main theme was made to sound eerie, dark, desolate and alien. The inspirations for the theme came from synth sounds, the Metroid soundtrack, the Joker film soundtrack and the Blade Runner 2049 film soundtrack. The aim was to make the synths sound powerful, yet calm and slow due to the cold lifeless theme of our game.

To emphasize the time freezing mechanic we slow down the music to a halt when we time freeze, which creates an effect like a record stopping.

The sandstorm effect was created by blowing into a microphone and then bit crushing the result to make it fit the retro feel. This ambient noise only plays when the player is aboveground and the sandstorm particle effect is active.

### Future Music/Ambience

In the future we would like to incorporate more musical themes depending on the area the player is in, as well as iterate on the theme that we currently have. We would also like to add some more ambience to the different locations, as that is now missing from the underground areas.

# Technical Details

## Target Hardware

The game is targeted towards PC, but could possibly be extended to fit other consoles. The controls are built for a regular PC keyboard and the use of an xbox controller.

## Game Engine And scripting language

The game is being developed using the Unity game engine, using C# as the scripting language. The game is developed on the Unity version 2019.2.11f1.
All of the code can be found in this GitHub repository.

## Time mechanic - implementation details

The implementation of the time reset mechanic is based around optimized input recording and position adjustments. To get the clones to move like the playable character used to move, a script continuously check for each frame if any registered inputs just started being pressed, and if so, the time since reset is recorded for that specific input. Once that same button stops being pressed, a second time is recorded and a difference between the two times can be calculated. Using this information, a coroutine is added to the clone to be created at the next reset, which simulates the same input presses at the same times as the player. The reason we use coroutines is because the implementation is far more optimized compared to if the game would check if an input is pressed at all frames for all clones.

However, there is a problem with this implementation; the final position of the clones are almost never exactly the true position that the players was in at the time. The reason for this is because a replaying of inputs using coroutines are almost never at the exact time as the original movement, it doesn’t matter if the input is recorded using the normal update function, or the fixed update function. The solution to this new issue is what we internally have called the precision fix. The precision fix uses an infinitely repeating coroutine on the playable character which records position, velocity, and collision state, and saves these events also as separate coroutines to the next clone. these syncing events are applied to the clones at the same time as they were recorded on the player and ensures that the clone ends up at the exact same places roughly at the same times as the player did.

If the collision state of a clone is not the same as the players was after applying a syncing event, that clone is counted as “broken out of sync” and only button events will be applied going forward in the current time loop. This allows the clones to behave differently if some parts of the world have changed between resets. Thing such as toggleable walls opening or closing, lasers changing, platforms moving, or something else.

# Management

## Detailed Schedule

**For the course**
Due to the scope of the project and the course, the two major milestones are the first hello world demo as well as the final presentation on December 9th. No more detailed scheduling has been organised, as it wasn’t really necessary. We divided each week into sprints, with a sprint meeting each monday. During the meetings goals for the upcoming week were determined, so we had a roadmap for the coming week. Hence we have the ability to have many short and quick prototypes, resulting in a quick and efficient way to test new concepts and then tweak game concepts accordingly.

The size of the group (4 people) made for a really efficient team, where everybody could speak up and pull their weight of the project. The small size of the team has resulted and eliminated a lot of overhead, which might have otherwise been needed in a larger team. The team also quickly got through the first phases (forming, storming, norming, performing and adjourning.) that every new team go through, according to Tuckman’s theory.

**For the release**
To begin with, our methodology in terms of how we work will remain the same. We would however need a project owner for certain parts (Art, mechanic, level design etc). This would be done to shorten the time required  for final calls on decisions that need to be made throughout the process of bringing this game to the market.
From the current state of the game leading up to the planned release on the 31st of March a rough timeline would be as follows:

- End of January, 2020:  we would have all the mechanics implemented. The order of implementation would be determined if it is core to the game, quality of life update or a nice to have feature. During this time more levels would be designed and started to be implemented. Also the marketing would start to take place, first and foremost in terms of the team spending some time trying to get a intresse on social media.
- End of February, we would have the majority of the levels designed, story development, level development in how you can reach certain parts of the map and what order the levels should come (depending on how hard the levels are, should be in increasing order).
- During March, now the game should be mostly done and the focus should be on two things primarily. Namely testing and marketing, the testing to find bugs and if

necessary features that are missing. When it comes to the marketing, it should be a lot of time and money spent during this period (if we have any money that is to say).

- Last of March, the game would be released in all its glory. The game will first be released on steam and Itch.io.
- During April, with the game out on the market now the marketing will continue to make sure that it reaches it full potential through content made and distributed through the right social channels.

## Risk Analysis

The most obvious risks are those that we are aware of and know how to mitigate, that we can plan around and even prior to starting know how to handle. For example, the number of persons in the team requires that everybody pitches in and contributes with what they are good at and what they want to do. Mainly due to the hard time constraints on this game, six weeks from start to final delivery is a short amount of time. But this is a known risk factor and can be mitigated by prioritizing the most crucial core mechanics.

The known unknowns are the risks with creating something new in this short time frame, is the concept doable as well as is it enjoyable? This we only can get a clearer view after some rapid prototyping, but putting 1-2 weeks into finding this out can have devastating consequences on the final delivery due to the whole time frame is 6 weeks.

The unknown unknowns are problems that we don’t know about, that can be anything that we can not predict or try to mitigate in advance.

## Localization Plan

With the characteristics of our game we are planning to release to a broad audience at once. The reasoning behind this is due to the fact that the steam store is supported throughout the world, and Steam is planned to be our main distribution platform where the game will be released. We aim to develop and cater our game to the tastes of a global audience. Thus we are primarily aiming to let game do most of the communication through visual cues and styling of the game. Hence we do not need to spend time and money on voice overs in different languages. The main difference that will be between the different countries are the language of the menus as well as the instructions in the first level that introduce the player to the different mechanics in the game.

Thus we avoid the risk of wanting to expand to a larger market afterwards, and from the beginning knows if it will be a hit or not. Another reason is that the game does not utilize any local or area based features, thus limiting it to a local market first. Another reason why

## Test Plan

The strategy for testing out game is divided into two main categories - the user experience and the hardware testing.

The user experience tests will mostly be done pre release and early in the development process, by using friends and contacts as test subjects. The goal is that everybody in the team would play the game as development rolls out new features to find bugs and find areas that could be improved upon. We will also try to get user experience testing done by offering a small payment in the form of premium content for our game for people that fit the different targets group to test out the game. This will be done early in the process of development  as it’s easier to change direction.

When it comes to the testing of the hardware, we have as of now, not noticed any performance issues. This is also what we strive towards - the game should be able to run and function smoothly on almost every computer, laptops included. The reasoning behind this is that we want to attract and open up for a larger consumer market and include casual gamers. Therefore we need to perform testing of all different combinations of operating systems and possible hardware configurations. This is a particular task that we plan to  we are outsource this to a third party that will perform all of the testing for us.

Having an open demo version of the game is not something that we are going to do. The reason behind this is that it actually can hurt the sales, and games without a demo can actually tend to have more purchases, offering up a better profit. Presumably this has to do with the generation of intrigue and hype, and the fact that having tried it out makes you feel like you don’t need to buy it anymore. This removes the need to design a proper demo level that quickly hooks the player in. Today youtube and other streaming services offer up a similar experience if you want to get a quick review and first look at the gameplay of a game.

Source:
https://www.gamespot.com/articles/game-demos-can-hurt-sales-suggests-research/1100-6410863/

# Business plan

## Market analysis

We begin by giving an overview of the market in Sweden, Stockholm and the global market. There are about 38 game developer studios in Stockholm. To gain a foothold in the swedish game developers market alone is a major task, let alone trying to break through the global market.
For all of the studios mentioned above and many indie developers, the number of games that have been released on Steam since its founding  is around 30 000 games, with more than half of them released since 2017. (according to steam-spy)

The conclusion that can be made is that the market is saturated, with more games then a single gamer can have time to play. Thus we need to make our game as unique as possible to get some market shares and attract the games so the decides to invest their time into our game.

To know what you are competing against  avoid plagiarism and the repercussions, there is a need to know what games that exist out there that are similar to ours. In this way, we both analyze competitors to see what worked and what didn’t. Our competitors are games with a similar style and feel as well as a similar core mechanic. We have not found any game with the exact same core mechanic, but many with different spins on the idea of slowing down, resetting, and rewinding time.

Our core mechanic is an ability that resets time, so we cast out a wide net that captures games that have mechanics that relate to time in similar ways to our own. Some notable examples are the Talos Principle and Braid.

We also looked at some games that have a similar 2D based style and feel. In this case, there are many competitors in this area that have been sources of inspiration, which might also be competition for our game:  Axiom verge, Gris and Vvvvvv.

## Finance plan

The main source of revenue for our game will be cost of purchasing our  game, which will be priced at $15 USD.
The second source of revenue is from the ‘premium content’. This bundle will give you access to the music of the game as well as concept art, etc. One important thing to mention is that this will not give you access to more gameplay/content.

The reason behind our particular market pricing is that according to some sources the average indie game on steam sells for $9 USD, with an average number of 21,000 units sold units.

Another reason for the higher price is that a higher retail price will usually result in a lower number of units sold of the product.  But on the flip side, the number of units sold to break even is lower. To give us peace of mind,we need to know how many units to sell to break even. The other reasoning behind this is that according to sources listed the game market is different and setting a higher market price can yield a better total income then a lower price. The only thing that we can be really certain about is that the quality and the amount of gameplay is on par with the price.

https://www.pcgamer.com/are-indie-games-too-cheap/

https://www.gamesindustry.biz/articles/2017-08-23-indie-game-pricing-more-art-than-science

## Marketing plan

Marketing is about exposure and the costs involved. As an example, Electronic Arts (EA) spent 702 million U.S. dollars on marketing and sales related  activities between April 2018 to March 2019. The conclusion we can make from this is that marketing is going to be a huge cost. According to typical marketing strategies, the amount of time/money spent on marketing should correspond to the amount of time spent on developing. Hence as an indie developer we are aiming to spend somewhere between 25%-50% of our budget on development, with a ramp up on marketing towards our release date. We are aiming to release the game on the 31st of March. Thus we are probably going to spend somewhere around the ballpark of $50 000 USD on marketing. All things considered this is a very rough estimate on how much time and money will need to be spent in this endeavour, and will probably need to revised later.

The strategy will then follow the 25%-50% rule during the first three months of the following year. The budget spent on the first two months will be used to spur conversation and awareness about the game in different forums, gearing towards the market of ‘early adopters’. During the final months leading up to release, a lot of the budget will go towards creating hype around the game by buying up ad space online.

The first market slice we are trying to reach are the early adopters, in other words gamers that play similar games. This will be done by trying to build up some interest for the game prior to the release, more precisely by writing about the developing process as well as sharing exciting concept art for the game. We can capitalise on feedback we will receive on the game by collecting input through beta testing and user experience tests.

The second aspect of our marketing campaign is to create hype for our game through intelligent use of social media. This will be done through our own media accounts connected to the game.

The third part of the strategy to reach the market is to hand the game to famous youtubers and streamers to let them play the game. The second option here is to pay them to play the game.

The final part of the marketing strategy is post game release, now it is about maintaining the hype and keep making sure that the game does not fall below the radar to quickly. Due to most of the units sold is fairly short after the release. Hence we need to continue with producing content for social media as well as through ads. This is when we target the broad mass of non hard core gamers or indie lovers.

To summarise the market plan is  in this table.

| Time to release | Target group | Goal - Action | Resource |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Three months out | The real indie and puzzle game lovers that contribute to development process and follow developer blogs | In this phase we find online forums to post and get in touch with our consumer base and the broader gaming community. Running a blog. | Time |
| Two months out | Indie and puzzle lovers | Similar to the previous month, start buying ads or setting up ad space at popular conventions | Time and some money |
| One months out | People looking for upcoming indie releases. | Start producing content and spending resources on deployment of targeted ads and on ads that aren't catered towards particular groups | Time and money |
| Post release | All  gamers, with focus on casual gamers. | Continue with ads as well as producing content for the social media channels. | Time and money |

**Sources**

https://www.statista.com/statistics/672141/electronic-arts-marketing-and-sales-spending/

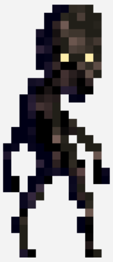
https://gameifyouare.com/2019/10/23/what-should-my-indie-game-marketing-budget-be/

https://www.gamasutra.com/blogs/JustinCarroll/20170327/294552/The_Realistic_Guide_to_Pricing_Indie_Game_Marketing.php

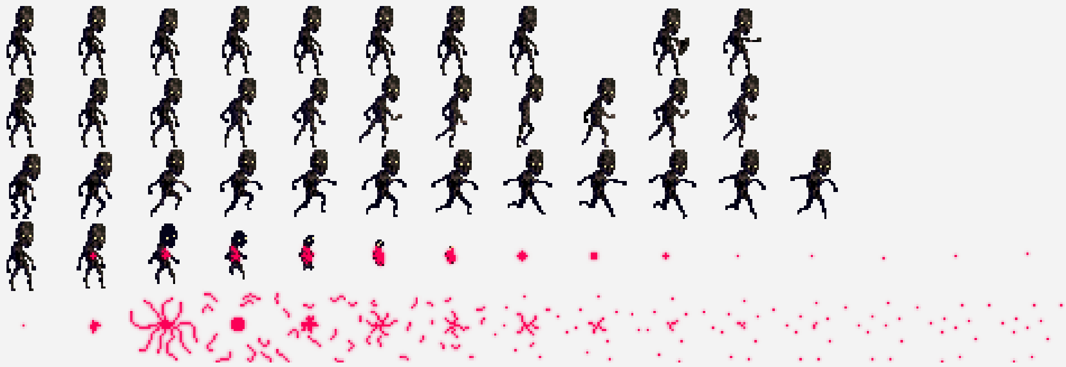
# Game Art

All of the art and tile maps as well as the animations are created by the team are presented here.  Some of these are also presented in relevant areas of the GDD sections.

**Main character design**
The character will be an alien humanoid figure, and the design is inspired by a similar image from unknown source,  See here in appendix

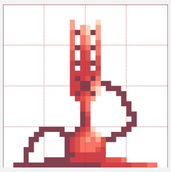


**Spritesheet for main character animations**
This contains animations for the character when
idle, shooting, running, jumping, and when dying, which uses a disintegrate animation - these appear in the spritesheet reading the rows left to right, from top to bottom.

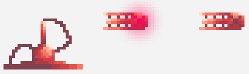


**Geyser**
The geyser comes in two color palettes, and also uses particle effects in unity to bring the air current to life

**Laser turret**

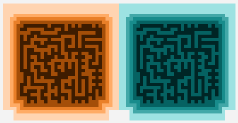


The laser turret consists of two parts,the main body and the turret, which has a red tinge when the laser is activated.

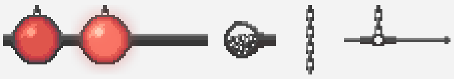


## Moving platforms

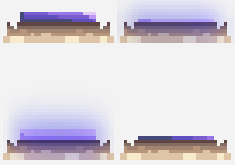
The moving platform comes in two different color palettes, blue and red.



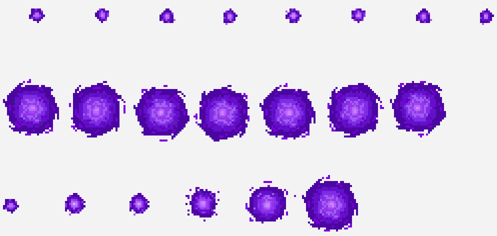
**Balance platform**
The spritesheet used for the balance platform.



**Button switch**
button that is pressed down vs not pressed down. the lights indicate that the connected entity is activated. e.g. laser is turned on.

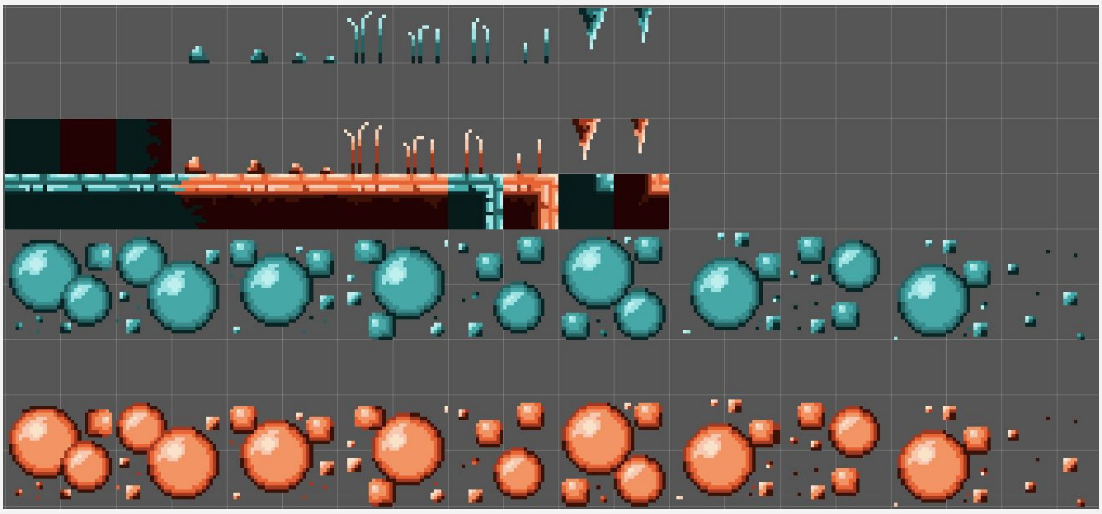


**Checkpoint**
inactive checkpoint, active checkpoint and the transition between them (top to bottom)



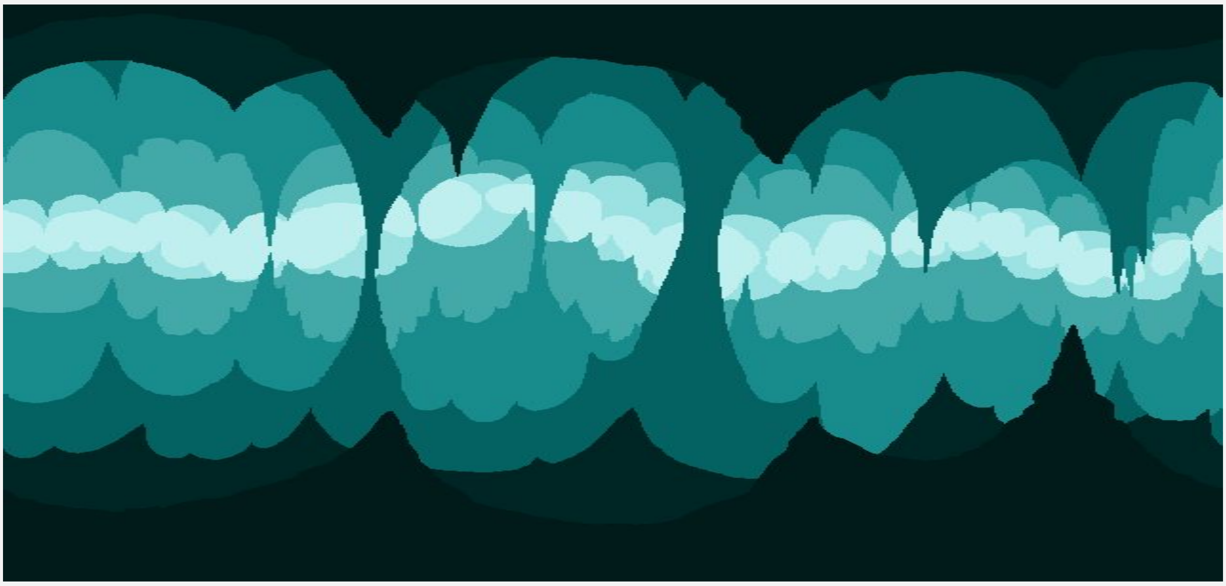
**Acid**
acid spritesheet, where the first row part (left) represents the top layer of the acid, and the second the bottom layer in for example a lake or pond.

**Tilemap used for the world's environment**
consists of two main colour palettes, icy blue and fiery orange/red. These are used to create the levels and environments that build up the game's world map.



## Environment

**Below ground - inside of planet cave background:**

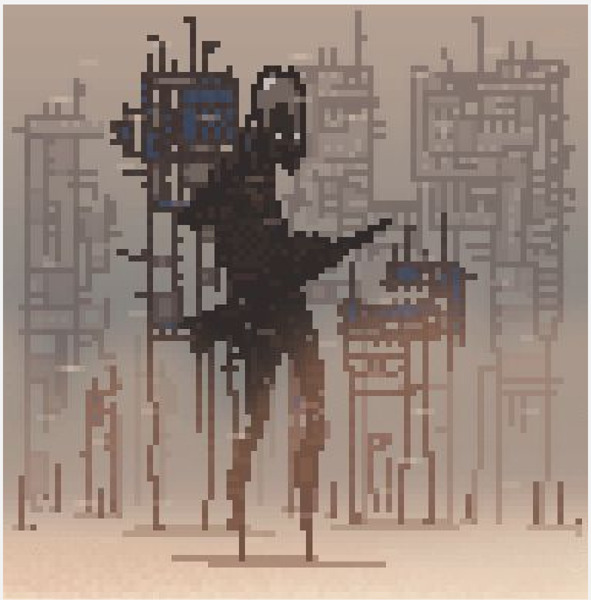


**Planet surface background:**

# Appendix

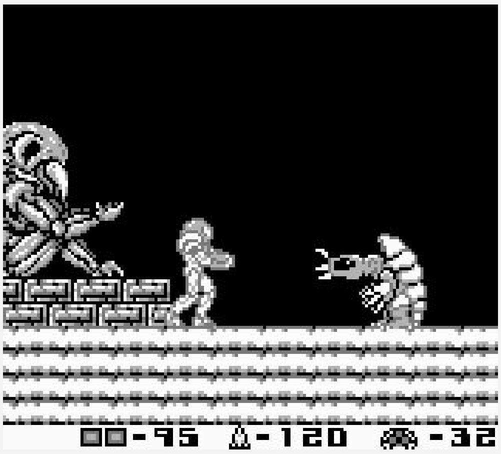
This section includes pictures, art and other relevant things in the game that don’t fit under any particular section in the GDD.

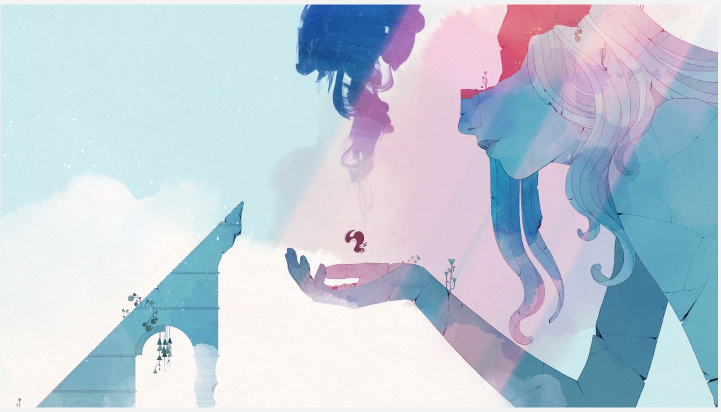
**Pixel art of alien character from unknown source, which was used as inspiration for our character design**

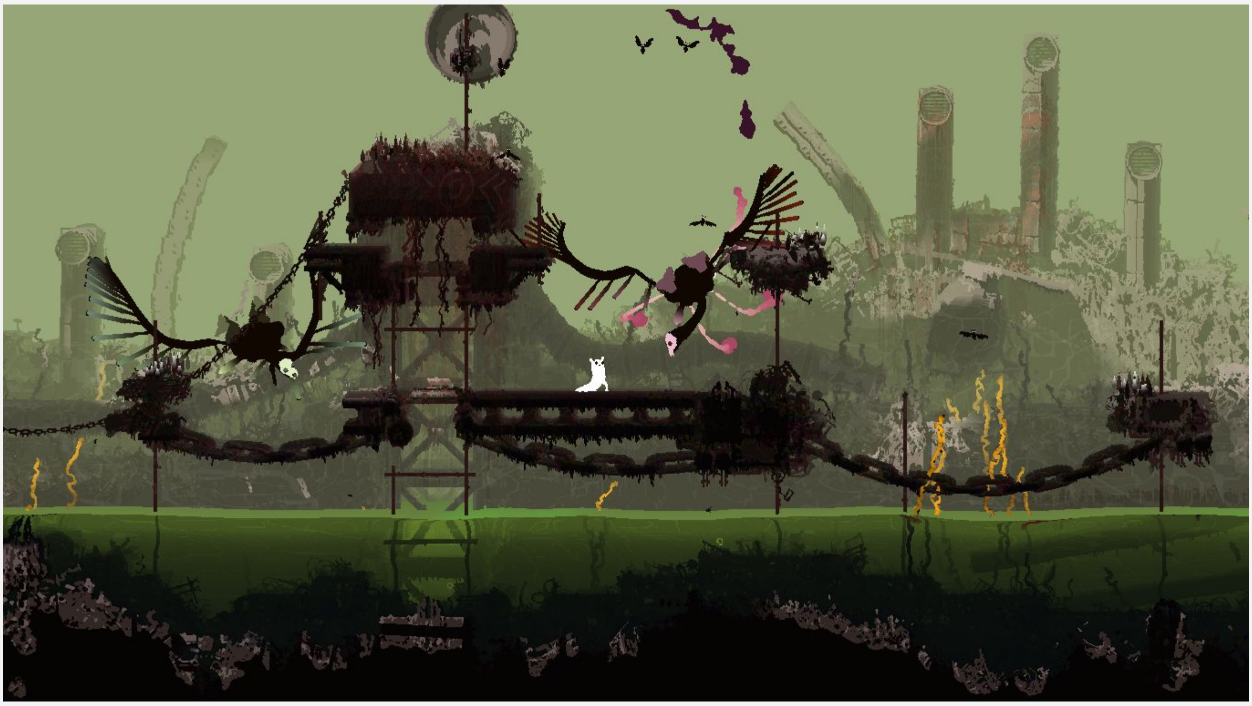


**Concept art for the checkpoint,** which wasn’t used in the end

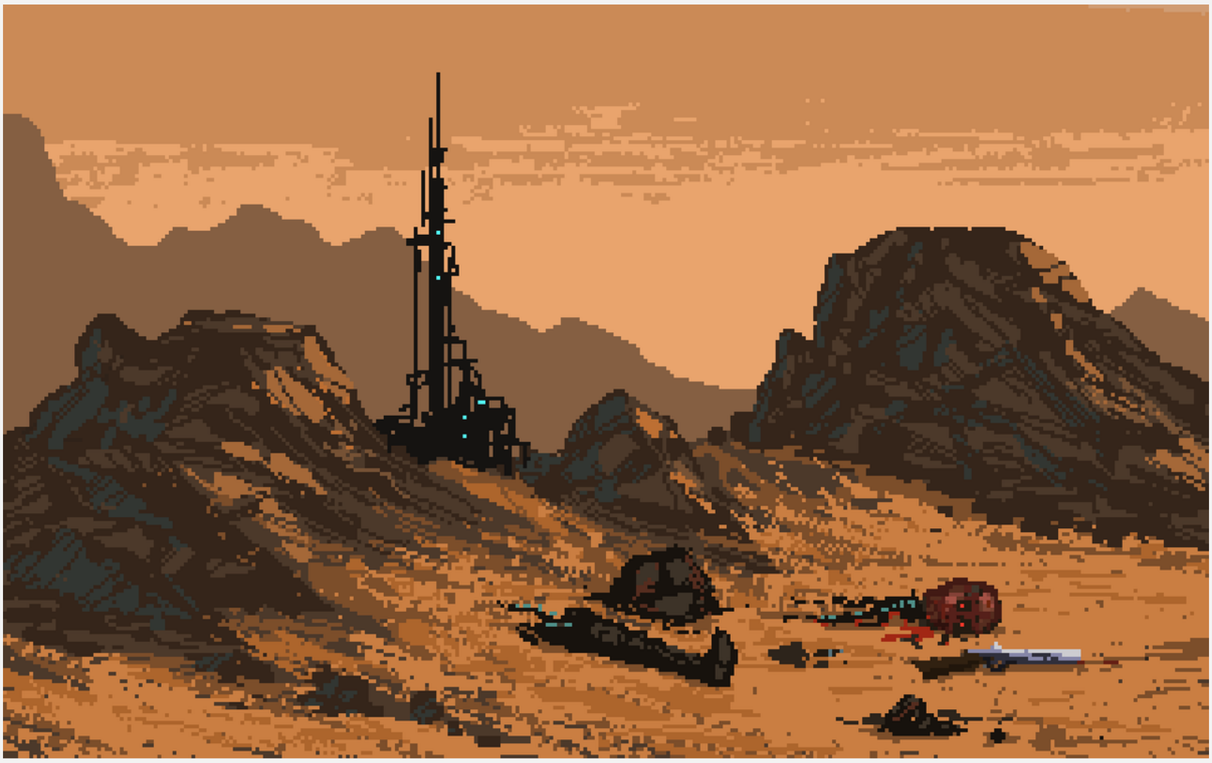
**Pictures from mentioned games that inspired our game:**
**(Metroid, Gris, Rain World)**

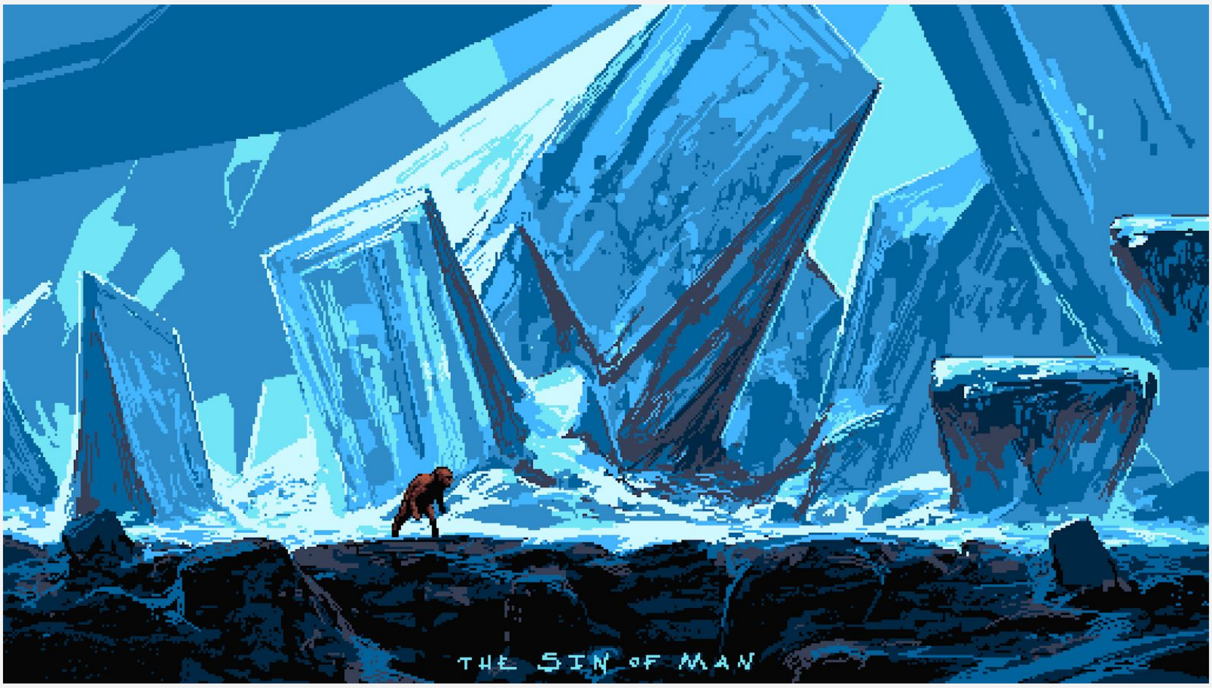






### Inspiration for our backgrounds




THE SIN OF MAN