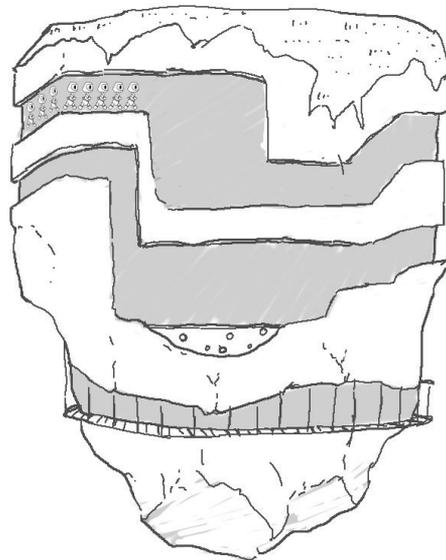


The Lost Creatures

A VR puzzle adventure

aka

Lemmings + Monument Valley in VR



© 2016 Future Games. All rights reserved.

Version 1. Ferruccio Cinquemani. 27-10-16

Table of Contents

[The Lost Creatures](#)

[Design History](#)

[Summary](#)

[Game Overview](#)

[Game Concept](#)

[Feature Set](#)

[Genre](#)

[Target Audience](#)

[Game Flow Summary](#)

[Look and Feel](#)

[Project Scope](#)

[Number of Locations/Levels](#)

[Number of NPC's](#)

[Developments beyond the MVP](#)

[Gameplay and Mechanics](#)

[Gameplay](#)

[Game Progression](#)

[Mission | Challenge Structure](#)

[Puzzle Structure](#)

[Objectives](#)

[Play Flow](#)

[Mechanics](#)

[Physics](#)

[Movement](#)

[Player Physical Movement in VR](#)

[Objectives](#)

[Actions](#)

[Picking Up and Moving Objects](#)

[Switches and Buttons](#)

[Interacting with the creatures](#)

[Rotating the world](#)

[Screen Flow](#)

[Screen Flow Chart](#)

[Main Menu](#)

[Level Select](#)

[Options Screen](#)

[Game Options](#)

[Replaying and Saving](#)

[Story, Setting and Character](#)

[Story and Narrative](#)

[Backstory](#)

[Plot Elements](#)

[Cut Scenes](#)

[Game World](#)

[General Look and feel of world](#)

[Setting for levels](#)

[Levels](#)

[Level #1](#)

[Synopsis](#)

[Interface](#)

[Visual System](#)

[HUD](#)

[Control System](#)

[Audio](#)

[Music](#)

[Sound Effects](#)

[Artificial Intelligence](#)

[Creatures AI](#)

[Player and Collision Detection](#)

[Pathfinding](#)

[Technical](#)

[Target Hardware](#)

[Development hardware and software](#)

[Game Engine](#)

[Scripting Language](#)

1. *Design History*

1.1. **Summary**

Version 1.0 - Oct 27th 2016: First draft

2. *Game Overview*

2.1. Game Concept

A VR puzzle game inspired by the classic Lemmings.

Guide a group of small creatures lost in the cosmos on their journey back home. Solve environmental puzzles on a series of asteroids, small planets and abandoned spaceship floating in space while trying to save as many creatures as possible.

2.2. Feature Set

Interact with the game world through accessible VR controls

Rotate and look around the game world in VR to predict where the creatures will go and see potential dangers

Solve environmental puzzles by interacting with obstacles, contraptions and parts of the environment

Interact with the creatures to give them simple commands

2.3. Genre

Puzzle, with elements inspired by games such as **Lemmings**, **Nebulus**, **Pikmin**, **Captain Toad**, **Mario Galaxy**

2.4. Target Audience

Mobile VR version:

A wide audience made of people that do not necessarily play a lot of “hardcore” games but are interested in new technologies such as VR: **adults** (as children seem to have more motion sickness issues with VR) of **any gender or age owning a portable VR device**, such as Samsung VR Gear or Google Daydream.

The lack of violence, short length of the game, support of low cost mobile VR platforms, simple controls and friendly/quirky graphic style is designed to appeal to the same audience that played **Monument Valley** ([2.5 million copies](#)).

There is an underserved market for affordable VR, with Google Cardboard apps being downloaded [more than 25 million times](#), Samsung’s VR Gear having more than [1 million users per month](#) and new affordable high quality platforms such as [Google Daydream](#) being launched in the near future.

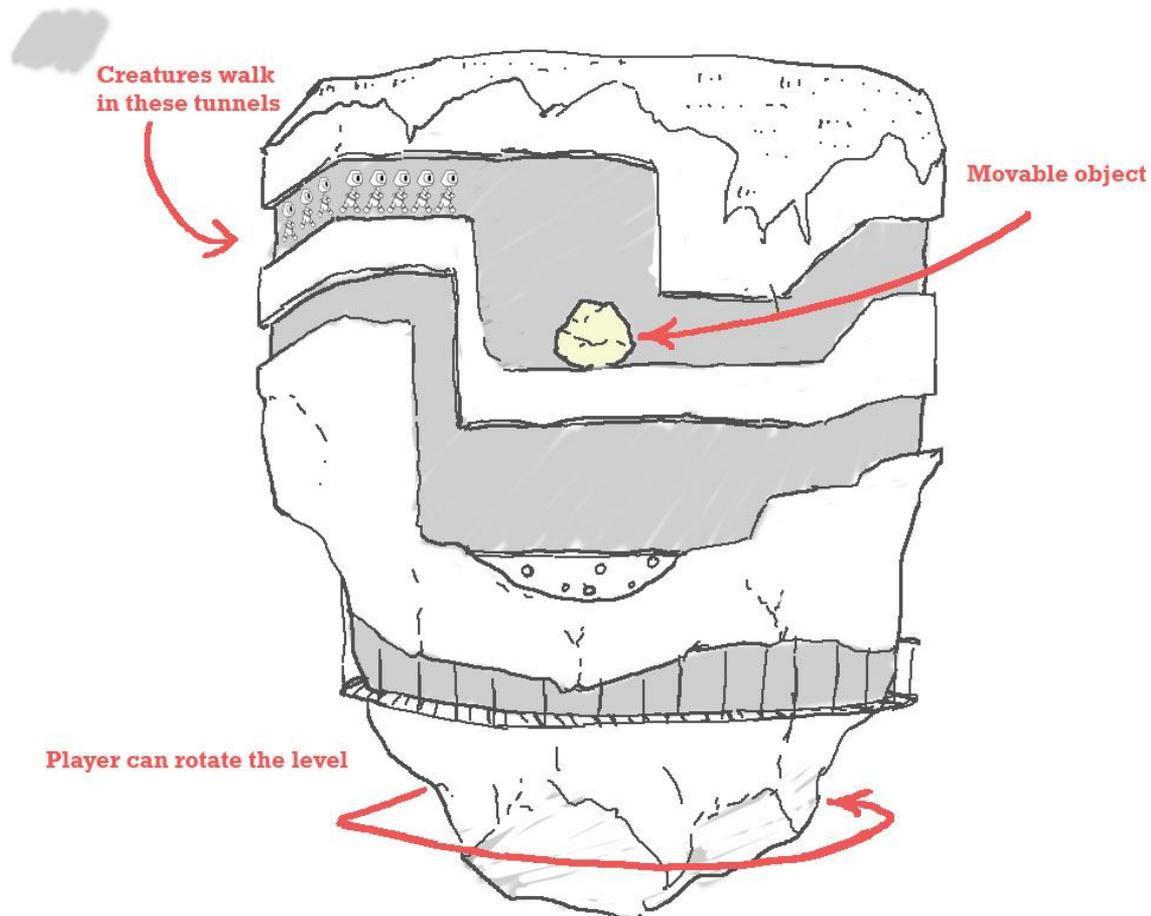
Non mobile VR version:

Playstation VR, Oculus and Vive platforms are still small enough that many players will try any good implementation of VR, but they are rapidly growing, with [PSVR projected to sell](#)

[more than 2.5 million](#) within the end of 2016 and Vive 400000. In that sense, we believe the target audience can be pretty much all owners of VR platforms.

2.5. Game Flow Summary

The player sees an asteroid, in the rough shape of a upstanding cylinder (ca. 1 or 1,5m tall), suspended in space in front of her. On the sides of the asteroid, the player can see tunnels and rooms, like an ant farm. Later levels can be space stations, abandoned satellites etc.



In the beginning of a game, the player can see a row of creatures ready to start walking in a tunnel on the top (or bottom, depending on the level). When the game starts, **the creatures start moving forward**. The player can **rotate the asteroid** left and right to see where the creatures will go. If there are obstacles on the way of the creatures, the player will have to clear the way by **solving environmental puzzles**, let all creatures pass and go check if there are other obstacles later in their path.

In the beginning, obstacles will be fairly simple, such as a boulder to hold suspended while the creatures pass, or a pounding object that needs to be stopped temporarily while the creatures pass. Later, there will be harder obstacles (see Paragraph 3.1.3)

The creatures must reach the bottom of the level (or the top, depending on which level they're playing) where the exit is located.

2.6. Look and Feel

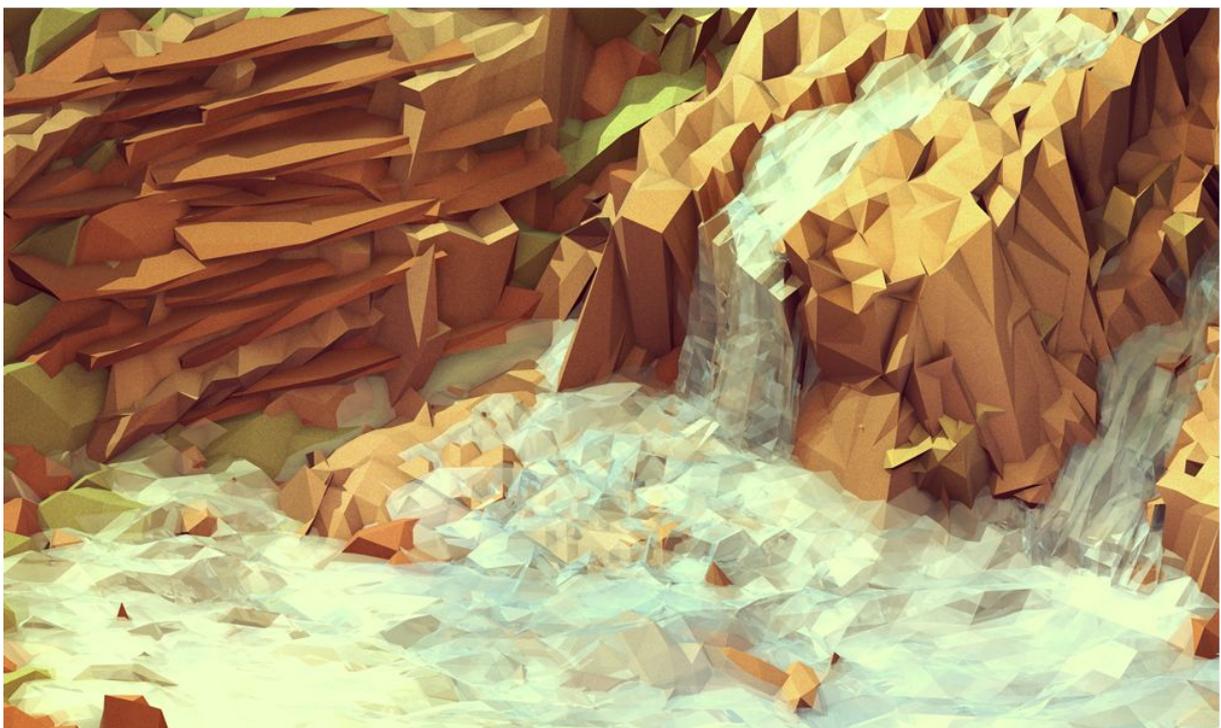
The game has a **low poly style**, with **flat shaded** polygons and **bright colors**. This style is necessary for several technical and artistic reasons:

- VR tends to be relatively low definition, so readable, simple shapes should be prioritized
- Mobile VR has limited performance, which excludes complex shaders or high poly environments
- The games should feel **quirky and pleasant but not necessarily childish** (think Monument Valley or Pikmin rather than Captain Toad)
- The inspiration for the creatures should come from games such as Little Big Planet, Playstation The Playroom, Pikmin, Lemmings. They should be easy to read (so with colors that don't blend with the background) and very low poly, so that there can be many of them onscreen without technical issues.

Inspiration for characters



Inspiration for overall style



2.7. Project Scope

The scope of the game should be limited, with a focus on polish and meaningful experience rather than number of levels. Additional levels can be sold as DLC.

2.7.1. Number of Locations/Levels

Less than 10

2.7.2. Number of NPC's

No NPCs

2.7.3. Developments beyond the MVP

The current design document covers a MVP state of the game. With enough resources and time, there are some things that could be done to extend the scope of the game.

For example: in the most basic version, having puzzles and one way to victory should be enough. However, the game could have more depth with some **secret areas** in each level that will be explorable only after the player unlocked a specific tool to reach them. In that case the creatures would have to **pick up collectible to unlock items** such as boots that allow them to jump, or snorkels to let them survive in water.

Also, the scope of the game could be expanded with more levels.

3. *Gameplay and Mechanics*

3.1. **Gameplay**

The game is about looking around and **rotating the level** to see where the creatures will go and **interact with the environment** to clear their path.

The **creatures always walk forward** until they find an obstacle (see 3.2.2 for more about movement).

The player can **interact with the creatures to stop them** momentarily. There should be some urgency in many puzzles but the game should not feel too stressful. The real hard goal should not be reaching the exit, but reaching it with all creatures alive.

Each level starts with all the creatures alive. The exact number of creatures in each level must be iterated and tested on each level. Too many may become a problem for some puzzles that require timing: we don't want the player to repeat the same puzzle for different groups of creatures. Ideally, they should be less than in a Lemmings level, between 10 and 30.

3.1.1. **Game Progression**

The progression is linear, with each level solved unlocking the next. A level is solved when the player manages to bring at least one creature to safety.

3.1.2. **Mission | Challenge Structure**

Each level is bigger and harder than the previous. Even if the complexity of puzzles increases, the real important thing is that each level should contain at least a new puzzle mechanic. Iteration and one-off mechanics are very important here. We want the player to feel surprised and eager to see what comes next.

3.1.3. **Puzzle Structure**

Each level must be solved through interaction with the environment and timing. The focus is on **clever puzzle solving**, not how quickly the player must act.

We don't want to punish the player too harshly. For example, some gates may be placed right before a puzzle to allow the player to think about how to solve it before letting the creatures walk in.

Example of some puzzles:

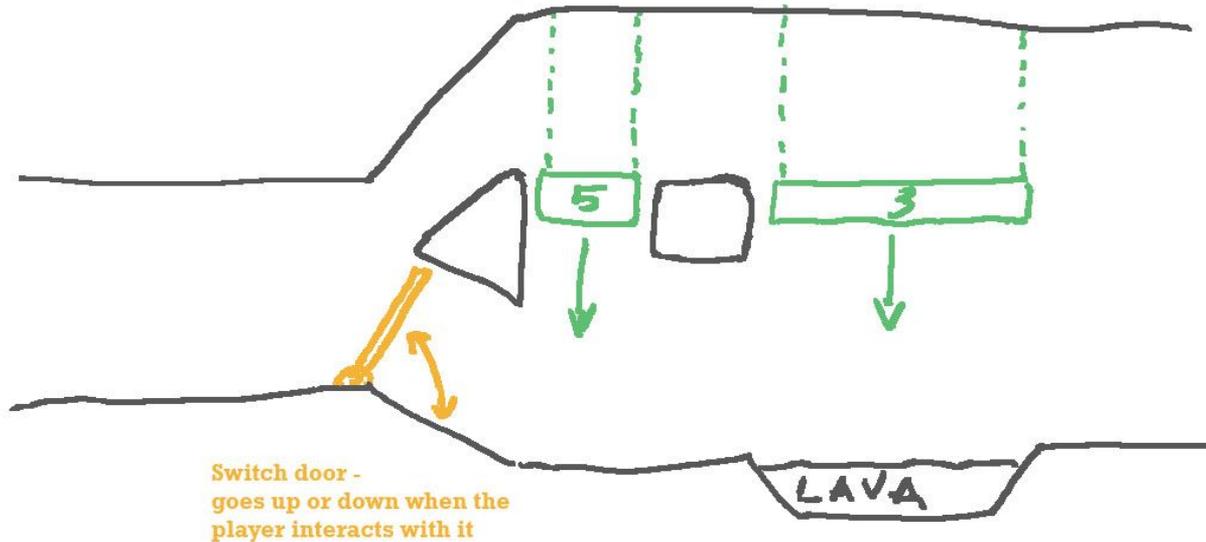
- *Simple*: a rock needs to be lifted so the creatures can go on walking.
- *Medium*: Switches that fork the path of the creatures, so that the player will have to send a certain amount of creatures on road A so that they can lower a bridge for the ones in road B, and so on.

Moving objects in green

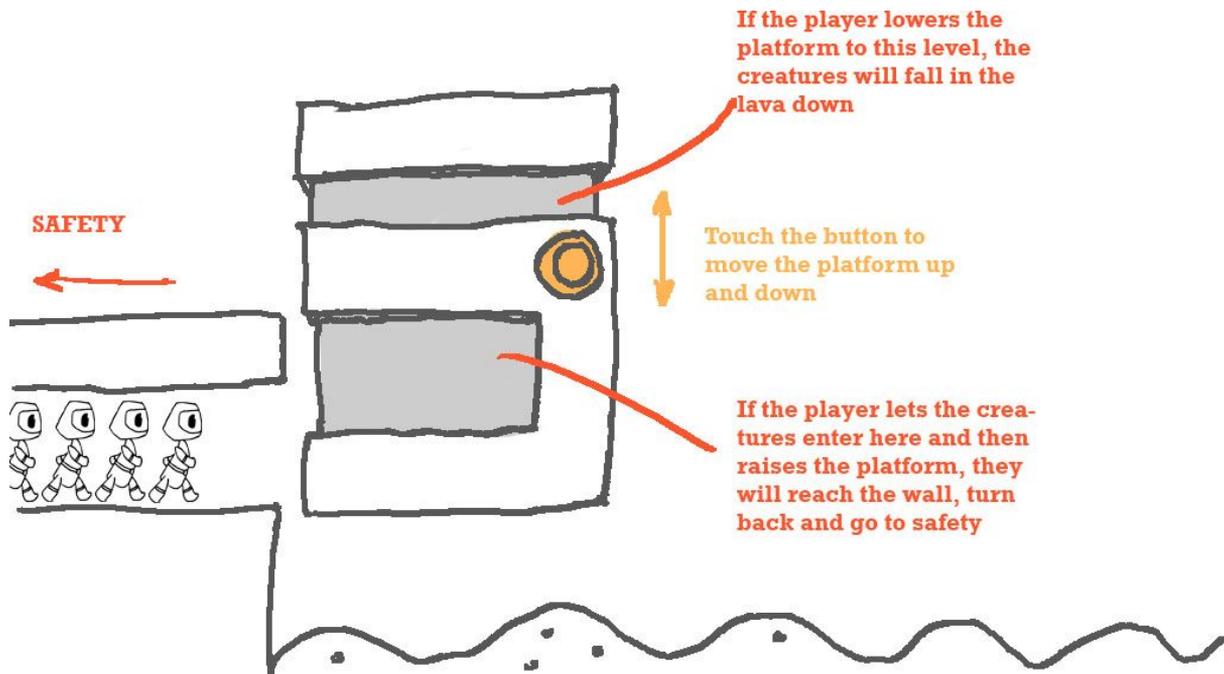
Environment in grey

Interactive objects in orange

Let five or more creatures get onto the first bridge and it will lower down, so all creatures will walk into lava. Time the lowering of the switch door so only 3 pass and they will lower the bridge over the lava letting all creatures to safety

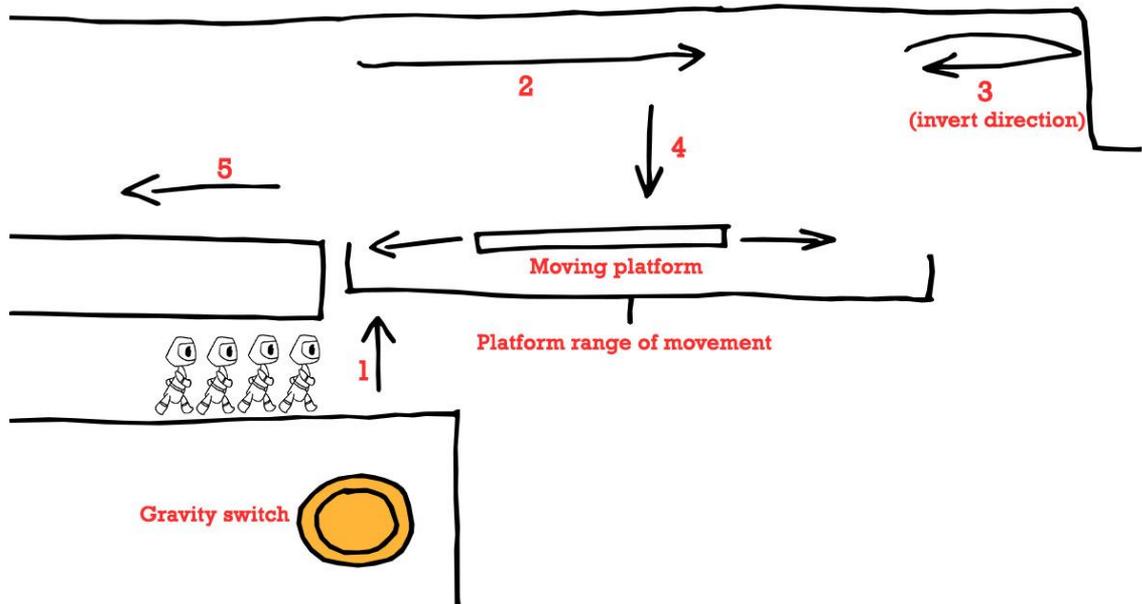


Other switches will move parts of the level up and down.

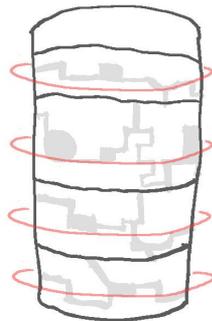


Many medium complexity puzzles can use the fact that the creatures will invert the direction if they will reach a wall. This can be used in creative ways (e.g. the player may see an optional path that is reachable only by inverting the direction of the creatures, and she will have to find a way to make them go to a wall so they change direction)

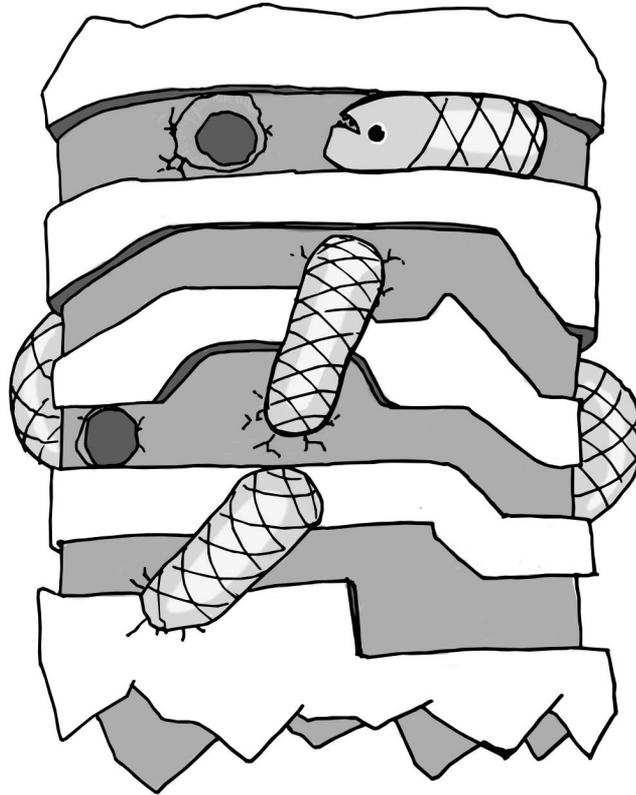
- *Complex*: Switches that invert gravity for the creatures may have to be used together with changes of directions and/or moving platforms.



- *One-off ideas for puzzles and levels*:
 - A level may require the player to rotate layers of the level. In that case, the puzzles will be mostly like a labyrinth, so the player must rotate the layers in a way to prevent the creatures to die, fall into space etc.



- Another level may contain a space worm or snake that pokes in and out of the level and requires the creatures to sneak around (by staying still while the worm pokes its head out near them).



- Another level may have the shape of a cube. In that level, the player would be able to rotate the whole level on the X axis, instead of only left and right. By combining that with a gravity switch, there could be a lot of complex navigational puzzles that could be designed.

One-off ideas can be a big part of the game: it's important to **not recycle too many puzzles**. It's better to have **shorter levels with more variety** than vice versa.

3.1.4. Objectives

Reach the exit of the level with as many alive creatures as possible. Losing creatures shouldn't be punished but saving many of them should be rewarded.

3.1.5. Play Flow

Each level is supposed to take maximum 10 minutes to solve. The player should not get stuck without knowing how to proceed: either he loses all creatures (game over) or ends the level with some or all creatures alive. At the same time, the game should not be too stressful. The focus should be on the cleverness of the puzzles; it shouldn't be a game where the player needs to play the same level dozens of time to beat it. Therefore, the creatures need to walk at a reasonable, not too fast pace.

The levels are presented in a traditional linear fashion, with each level unlocked after the previous one. This allows a difficulty progression to take place and avoids confusion for the player.

3.2. Mechanics

The main mechanic is centered around three actions: **interacting with puzzles** (switches, pickable items), **rotating the world** (to see where the creatures are heading to and what the next obstacle is) and **interacting with the creatures** (they stop for some seconds when clicked upon)

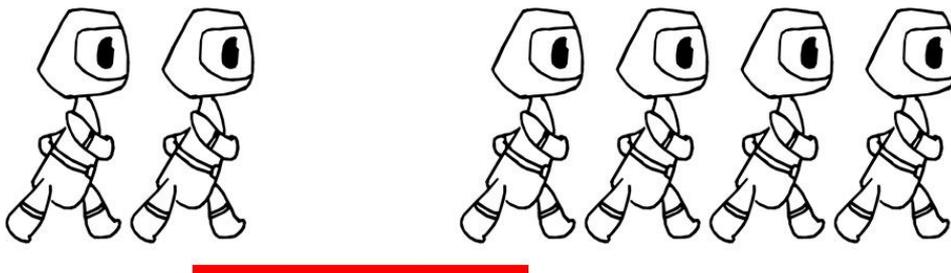
3.2.1. Physics

Some detached items, like a rock, are physics based. Gravity may be flipped in some parts of some specific levels, but probably all physics should be faked, as having the creatures subject to physics could cause unnecessary complications in development.

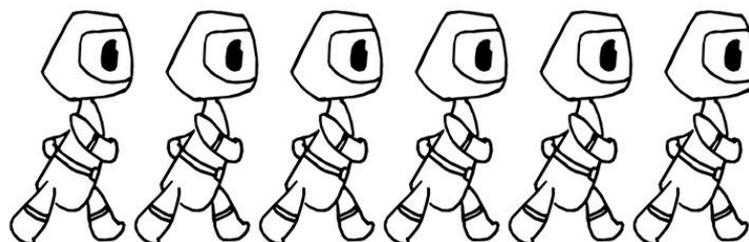
3.2.2. Movement

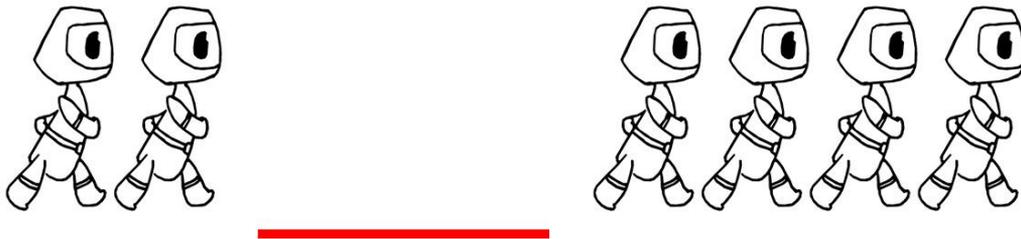
The creatures...

- ...**move always forward** until they reach an obstacle. When they do that, [like Lemmings](#), they change direction. To keep things readable and simple, the creatures **only move along the sides of the level**, never towards the centre: the play area is on an ideal 2D plane wrapped around a cylinder-like object (see 5.1 for more info)
- ...**will fall off a cliff** if there's nothing stopping them.
- ...**tend to stay close to each other**. If one is behind, it will try to get back and join the group. The logic of groups work this way: if the distance between two creatures is bigger than a certain amount (to be tested and iterated), the one behind will run instead of walking until it reaches the one in front of it. If the distance is bigger than the threshold, a new group will form. This allows the player to dynamically break the creatures in different groups and reunite them, which can be used in several puzzles.

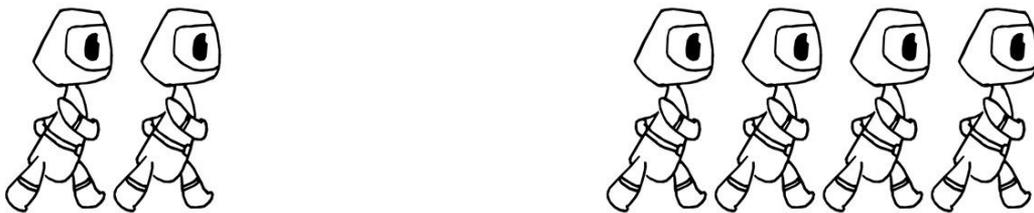


If the distance between creatures is less than a specific value, the creatures behind will catch up with those in front





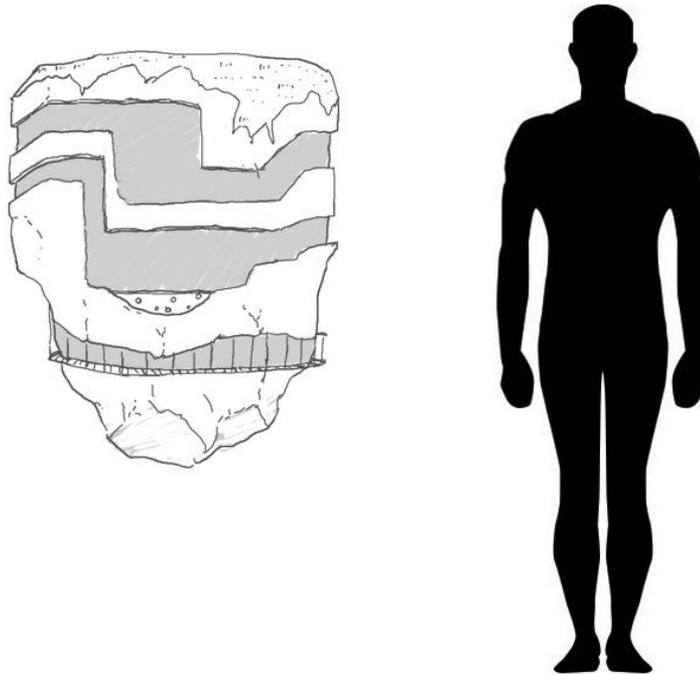
If the distance is bigger, two groups will form until they get closer again



- ... **can be interacted only as a group**. Touching any member of the group stops the whole group.
- ... **cannot jump** (unless some power up get introduced - see 2.7.3)

3.2.3. Player Physical Movement in VR

The player sees the level in front of her in VR, so she's free to move around as much as the VR headset allows (in the case of VR with position tracking) and free to get nearer the level, focus her sight on different areas, etc. The level should be big enough in VR to allow the player to see everything clearly but not so big that she will have to crouch on be on her toes to see the bottom and the top of it. The playable area should be around one meter tall and one meter wide.



3.2.4. Objectives

The **main objective is to reach the end of the level**, either at the top or the bottom, depending on the level. The secondary objective is saving as many creatures per level as possible.

3.2.5. Actions

The player does not control directly the creatures, but can interact with them and the environment in these ways:

3.2.6. Picking Up and Moving Objects

Some object, to be highlighted (with a glow, maybe) can be picked up, mostly just to be taken out of the way or, for example, to be used as a temporary bridge. They can be held by pressing and keeping pressed the action button while pointing at them with the cursor. Releasing will drop the item.

3.2.7. Switches and Buttons

Switches and button activate mechanisms such as drop bridges, moving platforms, rotating doors etc. They are used by pointing at them with the cursor and pressing the action button.

3.2.8. Interacting with the creatures

Using the action button while pointing at a creature with the cursor will make their group stop walking temporarily (maybe 2 seconds; needs to be tested). After that, they will keep on marching again and it won't be possible to make them stop again for a couple of seconds.

3.2.9. Rotating the world

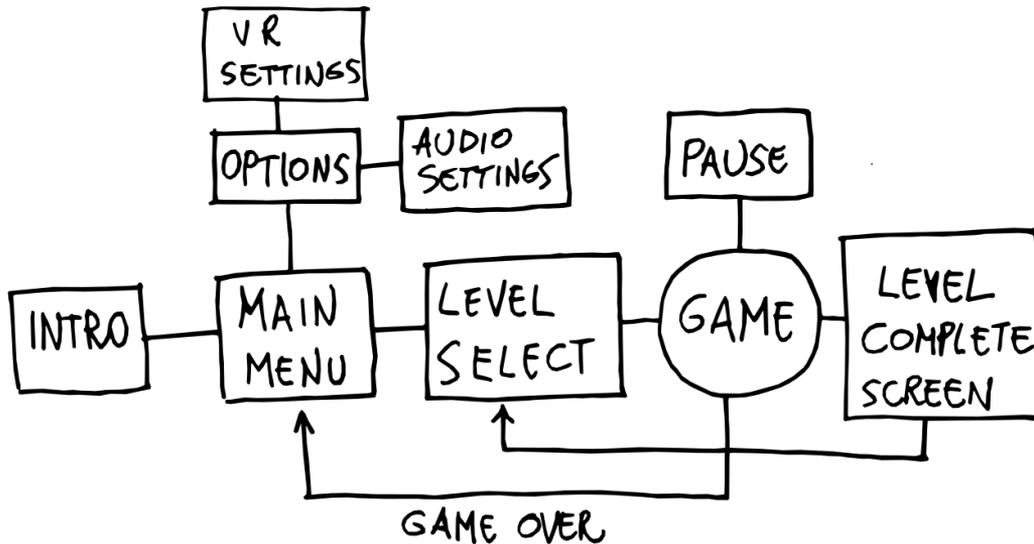
Considering that the level is on the sides of a cylinder, the player needs to be able to rotate it. He can do that by pointing the cursor on a non interactive part of the level, pressing and holding the action button, and move the controller left or right in the direction he wants to

rotate the level. Depending on the VR system (mobile or non mobile) and its controls, there may be a slider/buttons for rotating the level.

3.3. Screen Flow

The whole UI is based on simplicity and quick access to content.

3.3.1. Screen Flow Chart

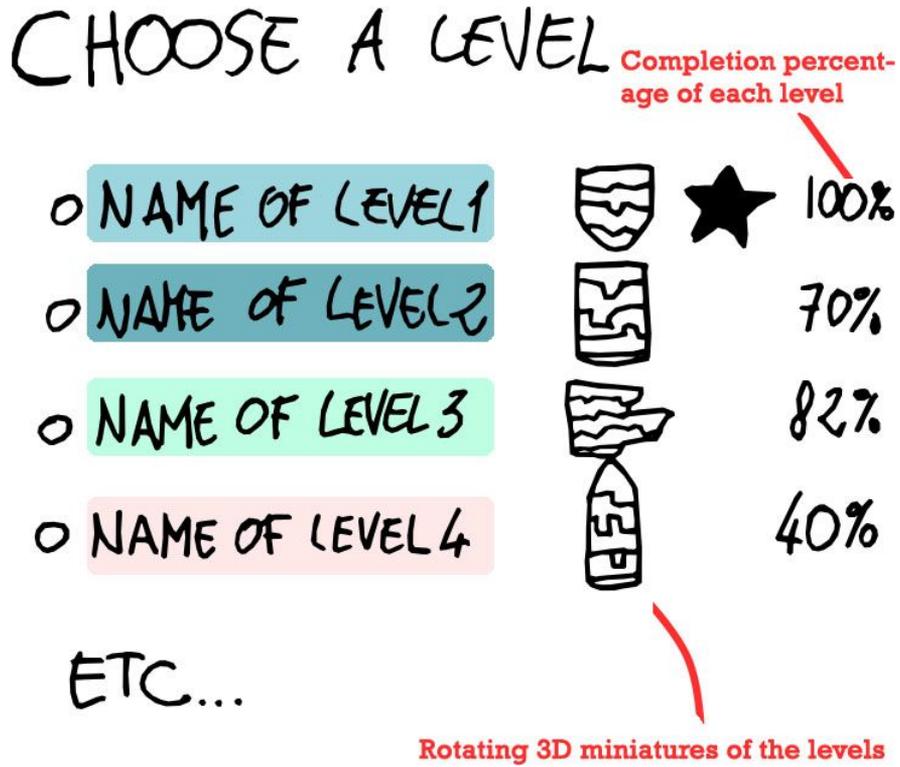


3.3.2. Main Menu

The main menu gives access to Options and Level Select

3.3.3. Level Select

A simple selection with a list of levels, each of them with an indication of completion percentage and a small model of the level itself to better remind the player which level is which.



3.3.4. Options Screen

Options should be limited to VR settings and Audio

3.4. Game Options

Options should be kept to a minimum. A sub menu in the main menu should allow the player to mute music and/or sound effects and another should allow for VR options to be set.

3.5. Replaying and Saving

At the end of each level, the game state is saved and the player gets back to the level selection screen.

4. *Story, Setting and Character*

4.1. Story and Narrative

4.1.1. Backstory

The creatures (temporary name until something more catchy is found) are a **crew of aliens who got lost while exploring the cosmos**. They are stranded on the asteroid where the first level takes place. With the help of the players they reach a portal at the end of the level. The portal doesn't bring them home, though, but instead they find themselves every time in a new place. Level after level, they go from asteroids to planetoids, to abandoned spaceships, always hoping that the next portal will bring them home. At the end of the last level the creatures reach the last portal and disappear, leaving the player wondering about their fate.

4.1.2. Plot Elements

The plot is not the main focus of the game but it has the purpose of setting a mood. Therefore, it should be kept light. We want mostly to give the player a feeling of visiting alien places.

More than a story, the game should be about a feeling. The creatures are **homesick, determined to come back and they have only each other as companions**. Like Monument Valley, the graphics, music and sound can do a lot to convey this kind of mood.

There could be small details to give more personality: for example, at the beginning of each level we could see the creatures getting out of the portal and cheering for a moment, thinking they are home, then immediately after that they look disappointed and then again determined. Much can be done in terms of environmental narration (see Section 4.2)

4.1.3. Cut Scenes

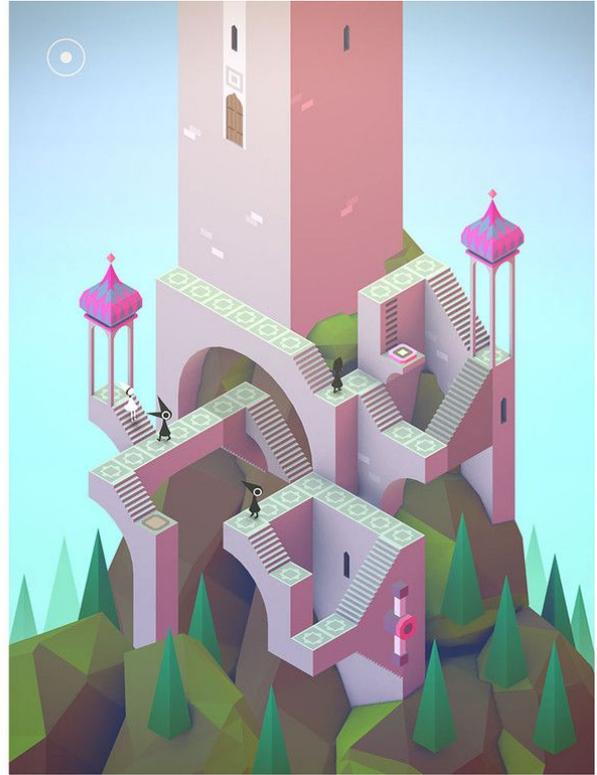
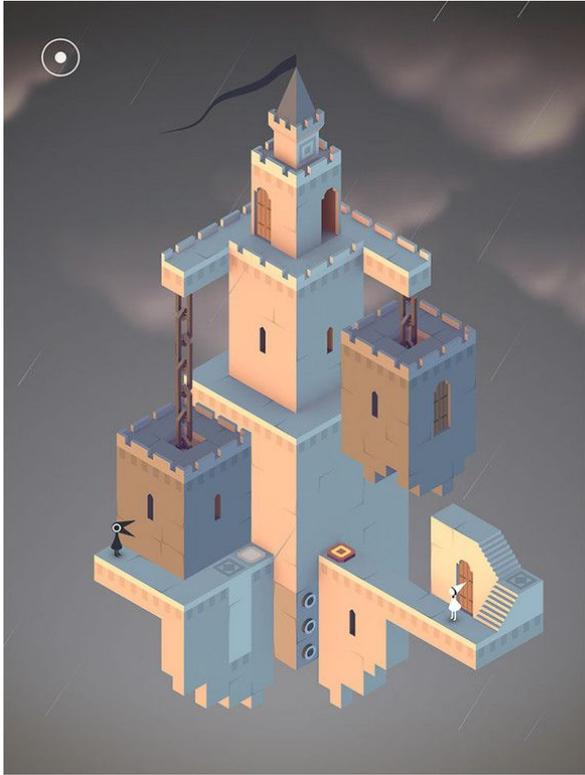
The game has no dialogue or text, but a short cut scene with static 2D illustration could be the intro, showing the back story.

4.2. Game World

4.2.1. General Look and feel of world

The low poly, flat shaded style requires using simple shapes and not focusing too much on details. Each level should have its own mood and backstory expressed through colors shape and sound rather than through detailed props. The general feel should be slightly dreamy, like in Grow Home and Monument Valley:





4.2.2. Setting for levels

Some narrative ideas for levels:

- an abandoned space station with a lush overgrown hydroponic lab
- a temple floating in space inspired by mayan architecture with statues of a scary deity and a worm inhabiting its tunnels etc.
- a clockwork level, with mechanical gears and switches, rotating puzzles and elements that move in time with the music, built for mysterious reasons by an ancient civilization

6. *Interface*

6.1. Visual System

The game is in VR, so any interface should be kept to a minimum to avoid breaking immersion. Any text should be rendered on a 3D plane in the game world.

6.2. HUD

There is no permanent HUD in the game apart from the cursor controlled by the player. Menus, when present, are rendered as a plane in front of the player. The player interacts with the menu by pointing with the cursor and selecting the desired option by pressing the action button.

6.3. Control System

The **controller is used as a pointer**. A ray trace is cast from the tip of the pointer and a cursor will be projected on the point hit by the cast. If the game is ported for a system without a controller (such as Cardboard), the gaze of the player is the vector that projects the cursor onto the world.

The action button of the VR device is used to:

- confirm interactions with whatever interactive object the cursor is pointing at (creatures or puzzle parts)
- move the level if the player is pointing at something non interactive, holding the action button pressed and moving the controller left or right

6.4. Audio

Like the visuals, the general feeling of the audio should feel light but not childish.

6.5. Music

The music should fit the theme of the level (a level set in an abandoned spaceship should give a feeling of mystery, one set on an asteroid a feeling of loneliness etc.), but in general shouldn't be too catchy/too Nintendo-ish. Think, as a reference, to the [Little Big Planet original soundtrack](#) by Daniel Pemberton or [Jim Guthrie's work on Sword & Sworcery](#). The music can be a little melancholic, but it should not be outright sad: the theme of the game is loneliness and feel of being lost, but not in despair.

6.6. Sound Effects

Every interaction should have a specific unique sound. The rotation of the level should have a clicking sound (like rotating a knob), interacting with the creatures should make them emit gibberish sounds (think Toad in Captain Toad or pikmins) etc.

7. *Artificial Intelligence*

7.1. **Creatures AI**

Creatures always walk forward, in line. When they reach an obstacle, they get back (see 3.2.2). They will fall down when they reach a ledge.

7.1.1. **Player and Collision Detection**

Creatures can bump into each other, but they keep a constant distance when going in the same direction.

7.1.2. **Pathfinding**

The level should never have a situation where the creatures can choose where to go. The path must always be obligated; they always go forward - or down if there's no road in front of them.

8. *Technical*

8.1. **Target Hardware**

Mobile VR: Samsung VR Gear, Google Daydream and/or Cardboard, depending on availability and tech analysis of feasibility (can the hardware render the game properly? is lack of position tracking too limiting? etc.)

Home VR: PlayStation VR, HTC Vive and Oculus Rift.

8.2. **Development hardware and software**

8.3. **Game Engine**

Unity seems to be better equipped for mobile VR (support of Daydream is already available), but in general either Unity or Unreal should be feasible depending on the experience of the dev team.

8.4. **Scripting Language**

C# on Unity, C++ or Blueprint on Unreal