Destruction for Dummies

1. Formal Game Proposal

1.1. Game Description

In Destruction for Dummies, players find themselves in a 2D puzzle environment consisting of destructible blocks, where the destruction of elements in a correct sequence will complete the level. The catch is, players destroy all elements they come in contact with uncontrollably. All the time. This includes not only the puzzle blocks, but the very walls and ground that contain them. Sloppy destruction, or failing to keep the pace up, will therefore result in failure and a new attempt.

Players are invited to explore each puzzle level in this fast paced trial and error setting, and express their creative side, ironically, in order to find the correct solutions.

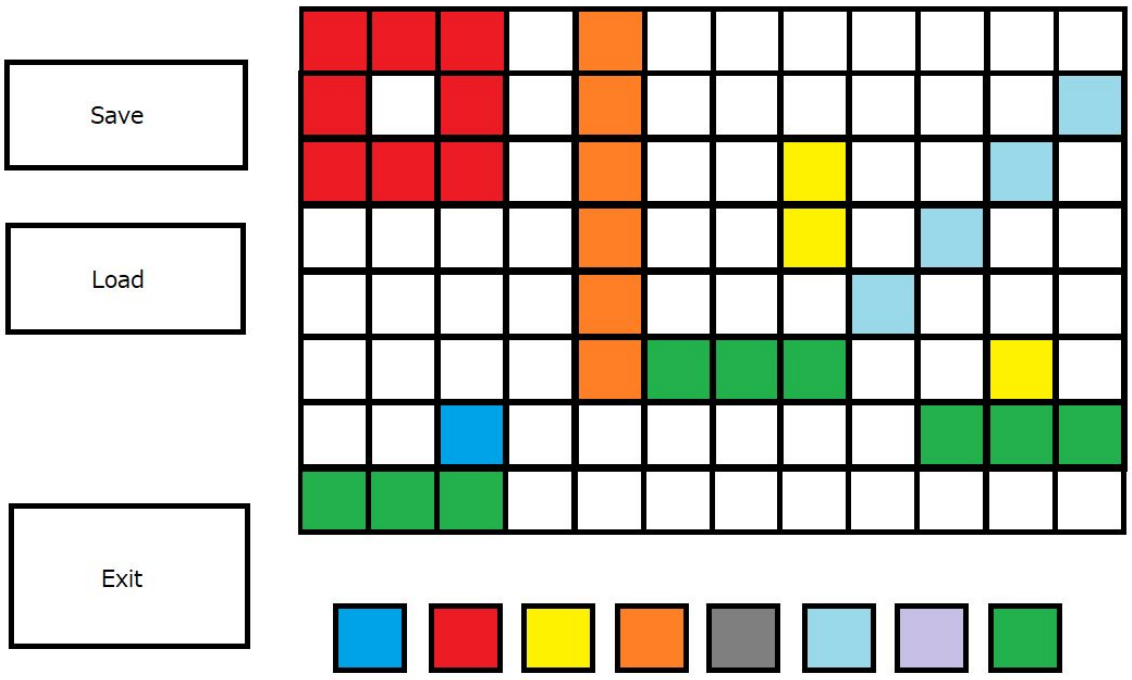
1.2. Block Types

- **Wood Block:** A frail block, breaks down on the first contact
- **Stone Block:** A sturdy block, falls slowly apart on contact and breaks down after a certain amount of contact
- **Air Block:** A block which creates an updraft on destruction
- **Death Block:** The player loses when they touch this type of block
- **Chain Blocks:** Blocks which are connected to each other, when one block of the group is destroyed all the other are too
- **Buff Block:** A block which on destruction gives the player a special power such as:
    - Charging up their destructive energy, allowing them to touch blocks without destroying them and releasing a powerful destruction wave after a certain amount of time, destroying any kind block
    - Providing the player with the ability to perform one dash, which accelerates them in the direction they are moving
- **Resistant Block:** A block which obtains divine power from nearby blocks , can only be destroyed, after destroying the specified amount of other blocks(weakening their resistance)
- **Chaos Block:** A block infused with the power of chaos, it inverses the controls of the player

1.3. Level Editor

Players can create their own levels using the level editor. Within the level editor you can choose between all available block types and place them on a 2D grid. You also

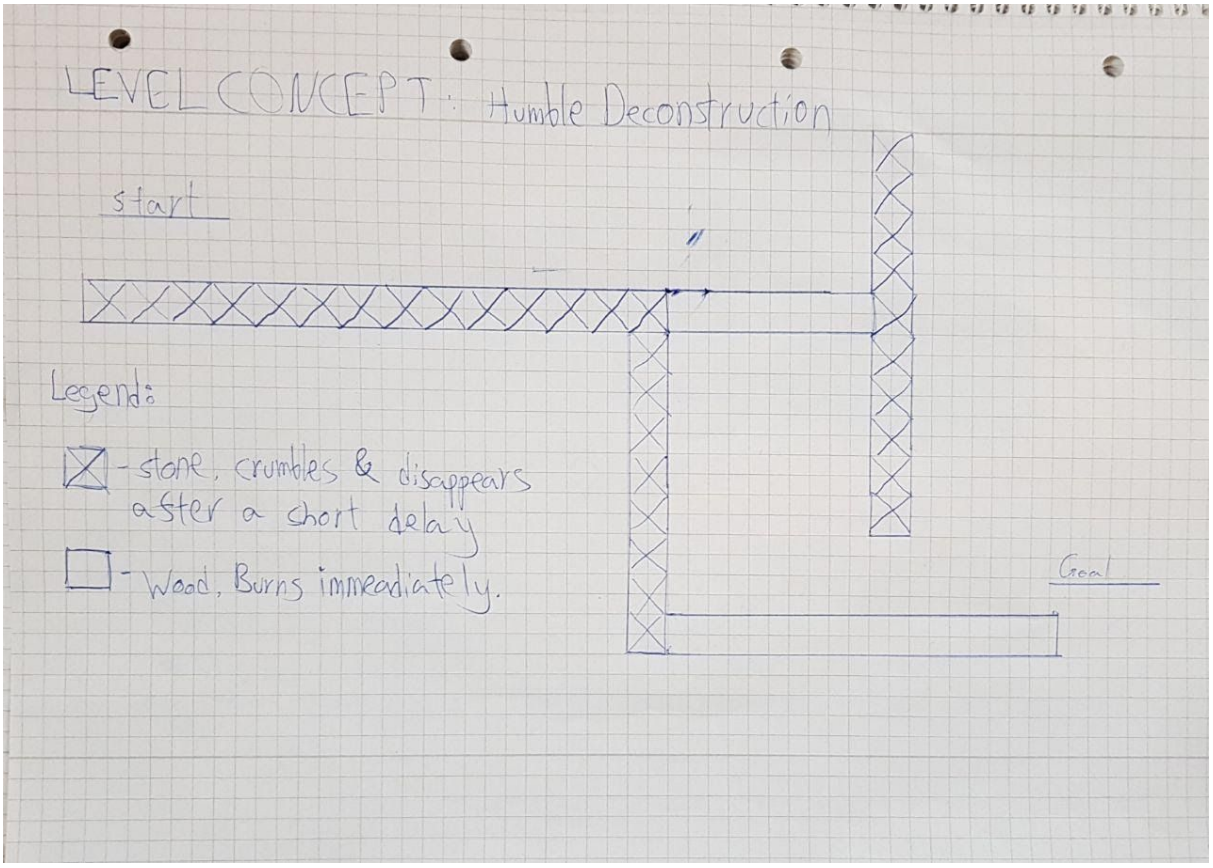
need to choose a starting position for the player. You can save your creation in a file and reload your level into the game later.

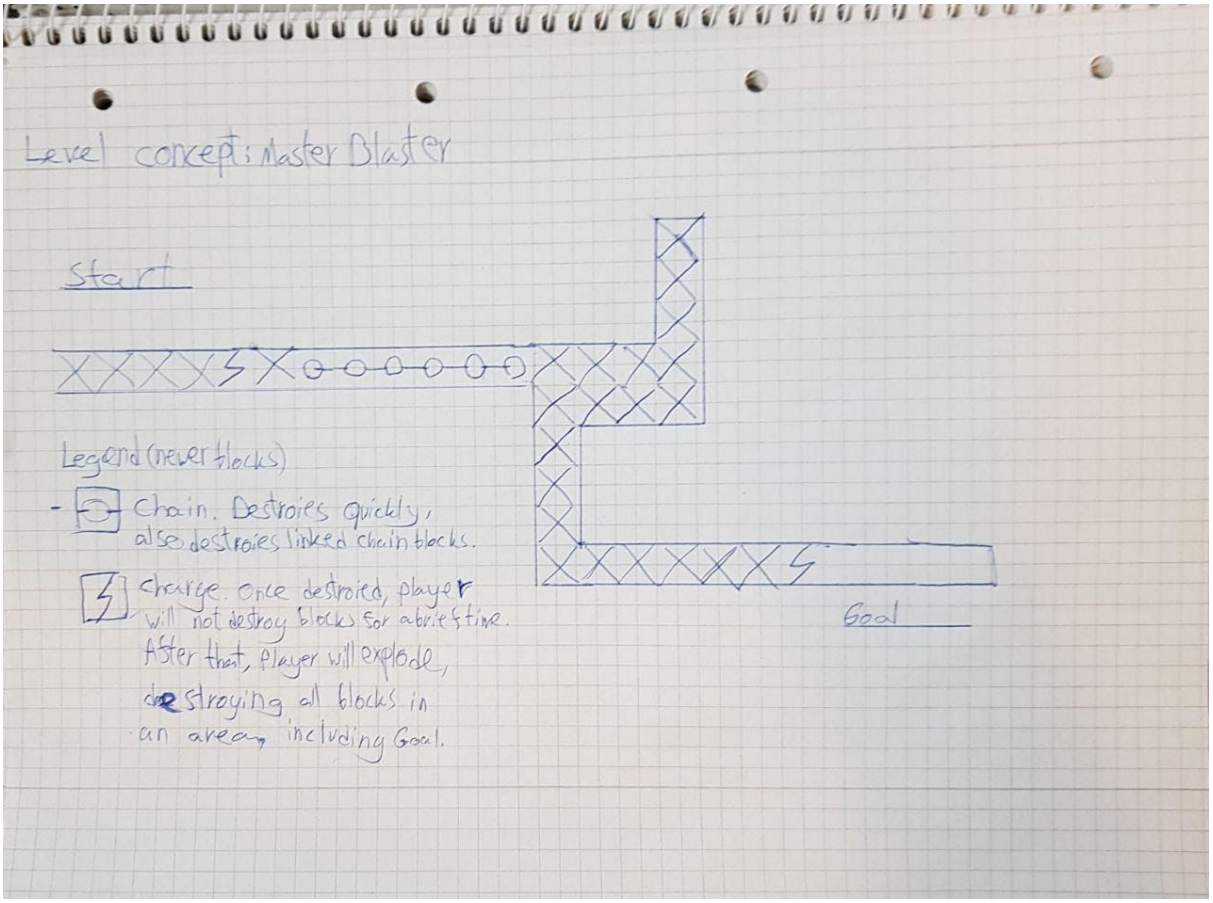


1.4. Story

Like mortals, gods must go through training and certification in order to fulfill their purpose. Balagan, the future god of destruction of the realm of Pangolia, has come of age. He must go through the trials set before him, before he may exercise his immense power. Stripping him of the ability to control his destructive abilities, the wise Da'at seeks to guide the young one through an educational gauntlet. In turn, this experience will teach Balagan that destruction is not just a spectacle to be had for his amusement (and amuse him it does). Though the adolescent god will begin his journey as a hot tempered youth that is eager to begin destroying to his heart's content, the adventure will teach him that destruction plays a key role in all things, including creation itself, and must therefore be conducted not only cautiously, but with great responsibility as well.

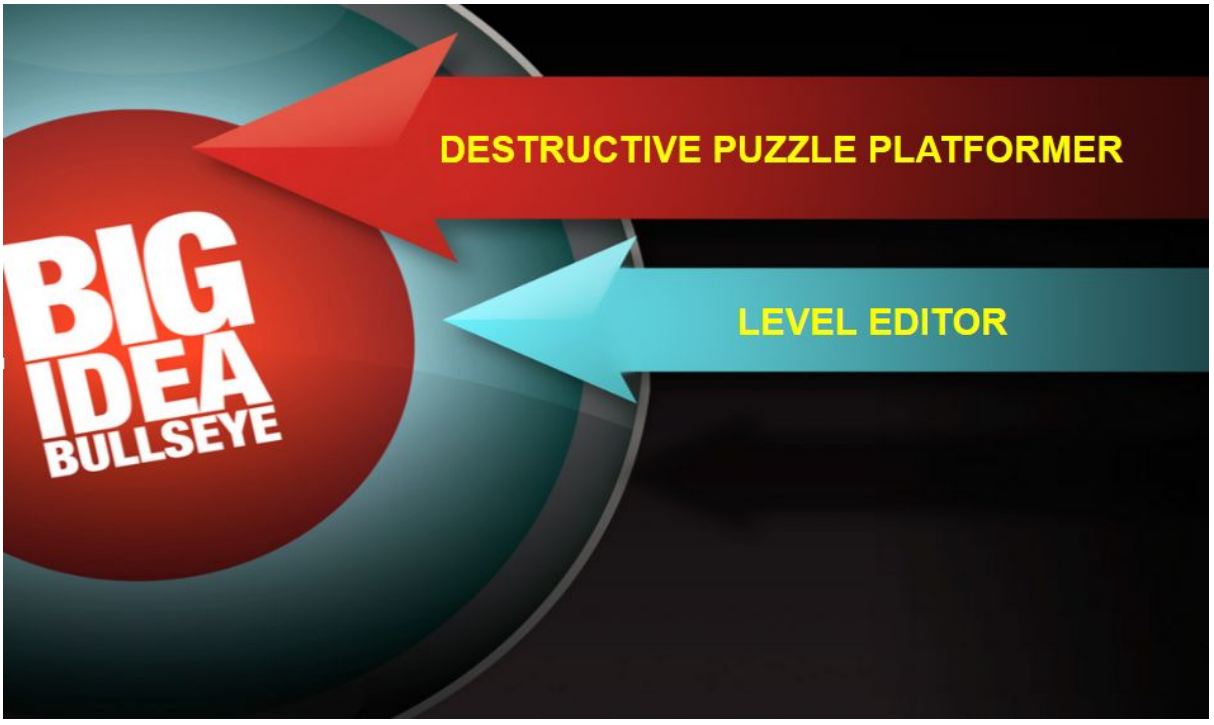
1.5. Example Stages





1.6 Character concept art

1.7. Big Idea and Technical Achievement



The technical achievement of the game is the level editor, allowing the player to freely create their levels and puzzles. In the level editor, the player can choose from a selection of available block types and then place them on a grid. They can export the finished stage after they were able to beat it once or just save it to work on it at a later time.

1.8. Development Schedule

1.8.1 Layered Development Schedule

Functional Minimum:

- Simple Player Character, that can move and jump
- At least 2 different block types(wood, stone)
- Start/Goal-platform
- Level Class(blocks in a grid)
- Input Manager(Xbox/Mouse-Keyboard)
- Simple cube models with color

Low Target:

- At least 5 different block types
- Simple Level Editor
    - Export level
    - Can create and save only 1 level
- Simple Menu UI
- Two playable levels
- Respawn of the player at the beginning of the level
- Completing a level
- Blocks with textures

Desirable target:

- Animated Player Character
- At least 7 different block types
- Background Music & Sound Effects
- Between 7 and 10 playable levels
- Nice looking level selection UI
- Block with simple animated textures
- More complex level editor
    - Playtest level
    - Load level

High target:

- Buff Blocks(Wall Jump, Dash, Energy Charge)
- Better looking blocks(shaders/effects)
- Even more levels
- Moveable blocks
- Level complete comment depending on blocks destroyed
- Extradiegetic blocks
- Story dialogues

Extras:

- Boss fight
- Multiplayer
- Shareable Levels(Online)
- “Mobile Support”

1.8.2 Milestones and tasks

1. Milestone: Game Idea Pitch (06.11.2019)

Task Name:	Who	Hours
Brainstorming	All	5
Documentation and Presentation	All	5

2. Milestone: Prototype (20.11.2019)

Task Name:	Who	Hours
Defining Type of Prototype	All	3
Creating a Prototype	All	7
Documentation and Presentation	All	5

3. Milestone: Interim Results (11.12.2019)

Task Name:	Who	Hours
Player Controller	J	8
Character Model	M	12
Input Manager	L	6
Basic-Block-Development (Start/EndPlatform, Wood,Stone)	L	8
Initial Level Class	J	5
Block Dev + Initial Level Design	L	6
Level editor	J	16
Character Animation	M	12
Level Design(Iteration) + Creation	All	6

Basic Menu	J	4
Documentation + Presentation	All	5

4. Milestone: Alpha Version (15.01.2020)

Task Name:	Who	Hours
Level design	All	8
Background music	J	6
Sound FX	J	6
UI	L	6
Level editor higher features	M	20
More level design	All	8
Energy Charge Block	L	5
Dash Block	J	5
Finish all desirables	All	10
Even more level design + implementation	All	8
Story dialogues + implementation	J	12
Movable blocks	L	6
Wall Jump Block	M	12
End of level comment	J	7
VFX, animations	L	10
Documentation and Presentation	All	5

5. Milestone: Playtesting Results (29.01.2020)

Task Name:	Who	Hours
Testing and Evaluation	All	10

Bug Fixing	All	4
Implement Feedback	All	2
Documentation and Presentation	All	5

6. Milestone: Final Release (12.02.2020)

Task Name:	Who	Hours
Trailer	All	10
Documentation and Presentation	All	5

1.8.3 Timeline

1.9. Assessment

The game offers challenges growing harder by the level, with a twist that will reel in anyone trying this out: everything gets destroyed all the time. This means that a player who first picks this up realises, as the level begins, that they have to run from crumbling floors and chambers as they try to figure out the solution. And also fail a few times on the way.

This process of trial and error, initiated by a frenzied “run for your life” beginning will serve as an engaging gameplay element.

In order to curb the possible frustrations of failing, be it while attempting to solve the level or while exploring, the game will reset the player and level instantly upon failure. The player will be encouraged to try out multiple ideas, and have little to no downtime while playing the game.

As the levels progress, a greater variety of destructible blocks are exposed, leading to more complex interactions and, by extension, harder puzzles to keep the player engaged.

Additionally, creating and sharing levels for the game is made available through the level editor, inviting creativity and social interactions using the games provided mechanics.

2. Prototype

2.1 Prototype goals

- Explore the game concept
- Elaborate on the mechanics
- See if the idea could translate into a fun game

2.2 Prototype description

In order to best meet the goals, while restricting the amount of resources invested in this prototype, we have opted to develop a throw away prototype unity project.
This project will encompass the foundations upon which the envisioned game project rests. Doing so will allow for testing and elaboration of the game idea. Is this fun? Does it feel right? How can we improve this?

To that purpose, a carefully selected collection of elements has been to make an early playable version in this prototype:

- Character, controllable - absolutely needed in order to experience an earlier version of the game.
- Some types of blocks the player can interact with - coming up with interesting puzzles might be challenging with a fewer amount of elements to choose from. However, it will serve the purpose of this prototype.
- Start / Finish platforms - necessary for the player, so that they may comprehend where they need to go to in order to complete the level.
- Basic level completion / failure mechanics - clear cut outcome to the puzzles.
- Quick respawn mechanic - to convey the feeling that the actual game will boast, of rapid trial and error with little down time.

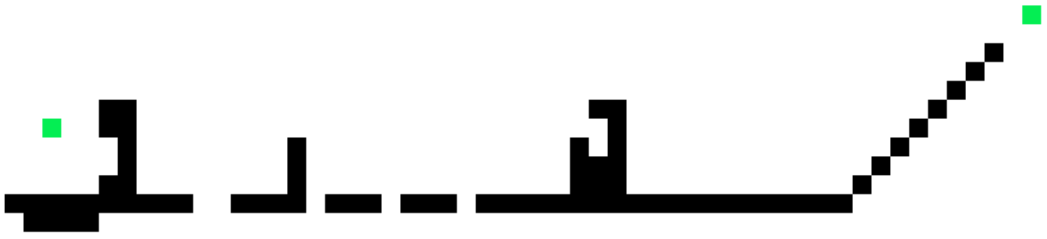
Early Levels:

The prototype will have 2 levels, both contributing to the exploratory nature of the task at hand.

The first is a much simpler level, designed to appear in the earlier stages of the game. It's linear. It showcases some basic obstacles, to be used as components in larger puzzle systems in later levels of the actual game. It has a very straightforward trial and error process of learning, so that it can be completed in several tries.
The reason behind this is simplistic design is: It should be kept down to earth without any as of yet unnecessary complexity to obscure the tester's opinion, and distract the development process. If the game concept turns out to be fun, the prototype will convey this feeling through this level alone, with all of it's boasted simple level elements.

Goals of this level:

- Basic movement, block interactions
- Level start / finish
- Player elimination and quick respawn (reload in this case)



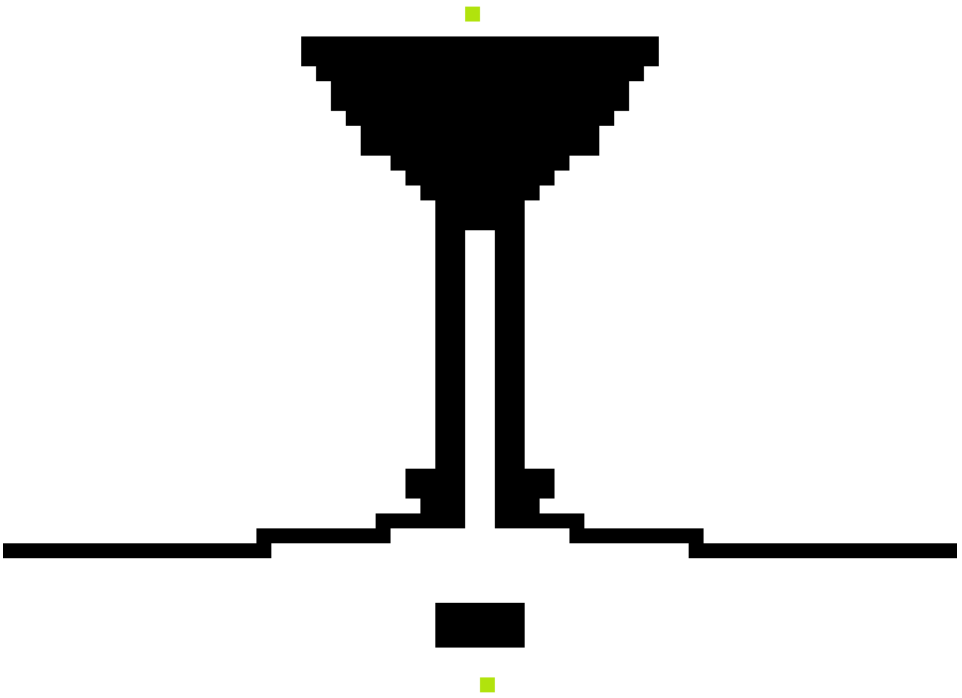
The second level:

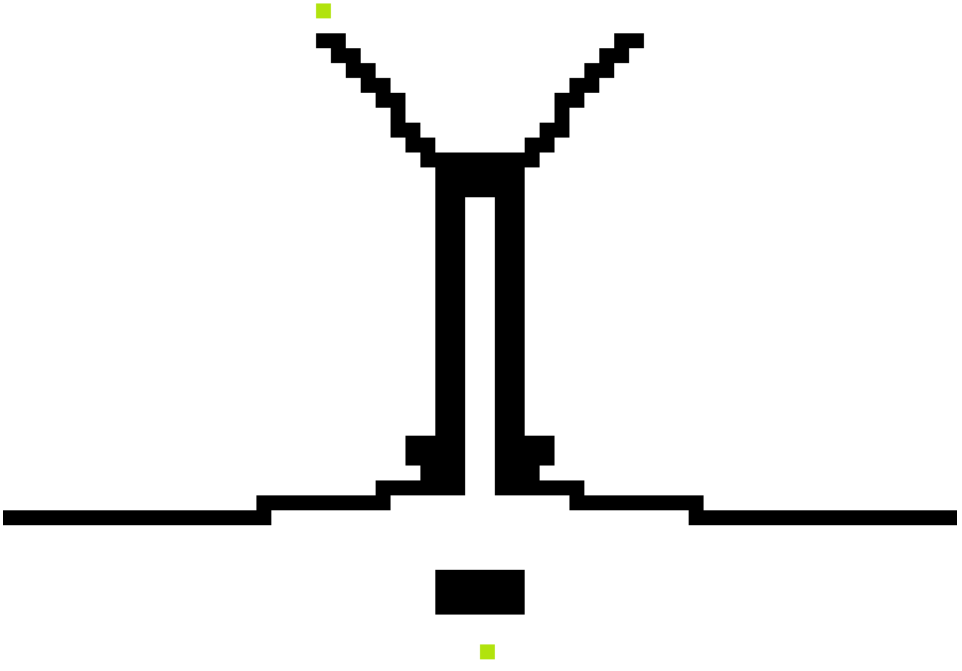
Once the basics are rounded up and evaluated into the most basic of packages for prototyping, the second level offers an opportunity to bring to life some of the more advanced concepts dictated by the game plan, saved for the later levels found in the game. Whilst the first uses the most fundamental blocks in the most basic realization possible, this next level attempts to replace the later but still use the former. The solution: A level built around the idea of exploration. A player who attempts to solve this puzzle will either start to drill their way straight down and see that they were lucky to do so, or wander off to the sides only to later figure out that taking one of the side routes leads to an inevitable restart of the level. Once the first phase has passed, the player must descend to the goal that is located in the central bottom part the stage.

Having exploration available as part of the level design enables the evaluation of the idea of having these elements in actual puzzle levels of the game. If it doesn't work within the rules of the game, or doesn't feel right for any particular reason, it should be omitted. If it does prove, however, to be a compelling aspect of the puzzle solving process, more of these will be incorporated into the final product.

Goals of this level:

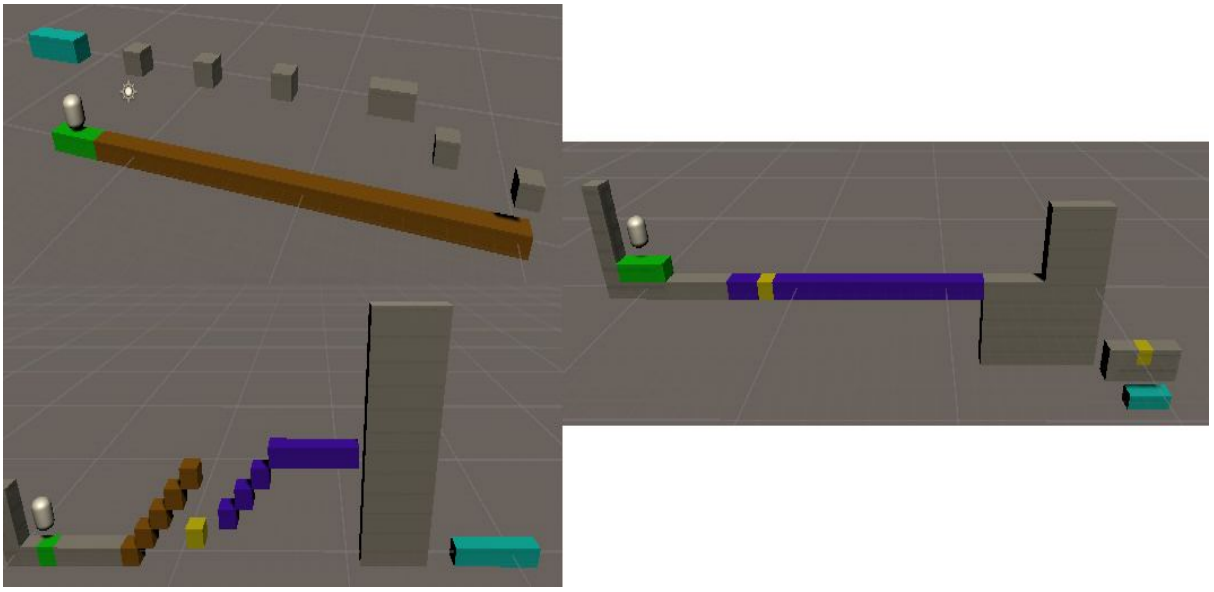
- Evaluate exploration
- Give a wider variance of options for puzzle solving actions
- See the basic interactions in a come together in greater scale





2.3 Prototype levels

The 3 prototype levels we implemented look like this:



The first level introduces the player to the wood and stone blocks, as well as the core game mechanic of uncontrollably destroying blocks touched by the player. Should the player jump on one of the above stone blocks in this level, they will disappear, rendering the level unsolvable, resulting in the player's fall.

The second level introduces the charge and chain blocks. The player is offered to use the charge block under the wood block stairs. If the player takes the offer, and does touch the charge block, they will not be able to destroy any other block for 5 seconds. After this time period, the player will have a force outbreak, destroying every block around them. If the player masters this mechanic they will be able to destroy the stone block wall without falling down. In case the player ignores the charge block and tries to reach the stone wall via the chain blocks, they will all disappear within a second.

In the third level the player once again is challenged with chain and charge blocks. If the player touches a chain block before touching the charge block, they won't be able to pass the bridge of the former. The second charge block at the end is a trap. If the player touches it they won't be able to reach the end, as they can't destroy the other stone blocks for 5 seconds, and after those 5 seconds the force outbreak will destroy the stone blocks, but also the end block. Failing and restarting the level ensues.

2.4 What we have learnt

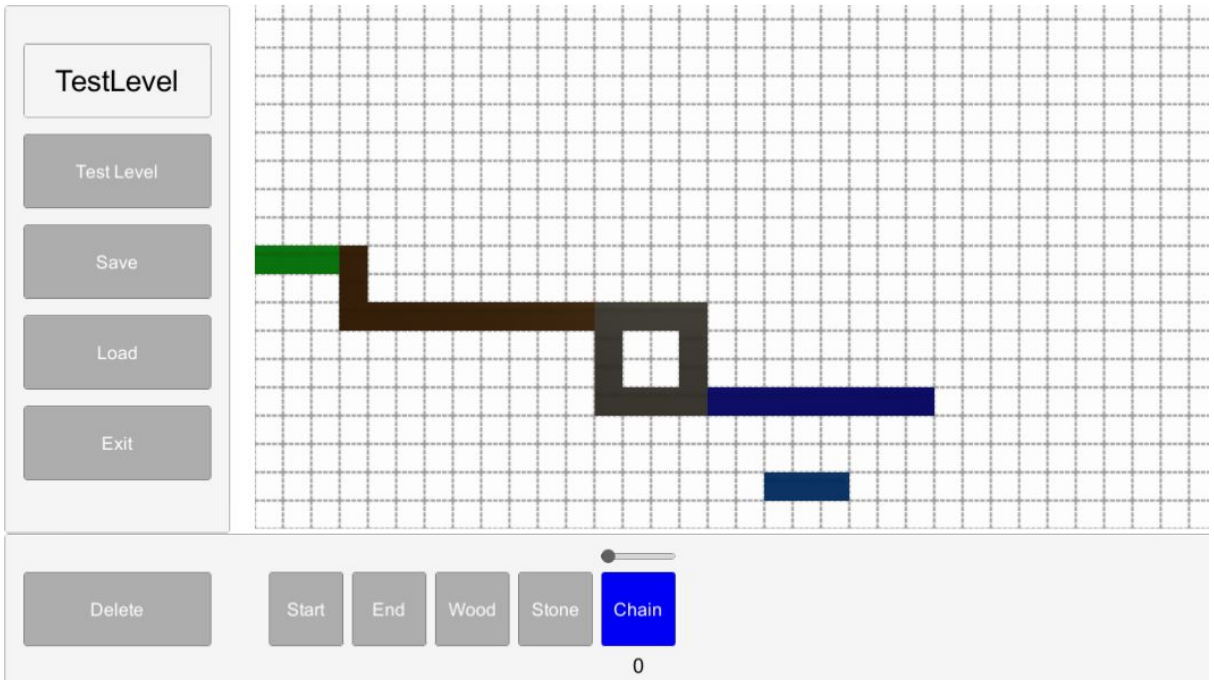
By creating a throwaway version of the game we got preliminary insights on how the game feels and on what problems could occur while playing the game.
During the creation and playtesting of our prototype, we have learned the following things:

1. Manually creating a levels is tedious: While we started to build the prototype levels per hand it became quickly clear that copying and putting every block in the correct position is slow and unexciting. This shows us that we should put even more emphasis on implementing the level editor so we can use it to build and change our levels faster.
2. We could also confirm that we can have the block vanish with a certain delay so that the player feels forced to move on quickly while giving them enough time to move over or jump from the block before it completely vanishes. Also the main gimmick of the game should become apparent quite fast even without textures indicating a block is crumbling. The player might die once or twice while not knowing that blocks can vanish but after a couple of tries it should be clear that they need to be fast.
3. The short levels were really fun to play. As mentioned before dying is part of the game and the learning experience and because the levels in our prototype are quite short containing only one or two challenges before they reach the goal it does not feel frustrating at all to die. We implemented a quick respawn(yet not optimized) in order to bring the player swiftly back into the game.
4. Last but not least we found out that we need to reconsider the playersize. At first it was planned that the player would be high as 2 blocks but by playing with a player of this size in our levels we found out they were to large. The large player caused issues where they needed to jump up to a platform but because of their size they would bump into blocks which were placed higher up meaning building things like stairways was pretty hard. That’s why we adjusted the jump physics a bit and shrunk the player down to 1.6 times the size of a block. But we think this needs still further testing.

3. Interim report

3.1 Level Editor

For our technical achievement we have chosen to build a level editor for the game. It consists of three areas. The first one includes functionality for saving levels, loading levels, testing levels as well as exiting the level editor.



The second area shows a grid where the player can place the different block types. The grid at the beginning consists of empty blocks, which have no functionality and only exist in the level editor. Loading a level with empty blocks into the play scene will automatically delete all empty blocks, and only leave the normal block types.

The third area includes one button for each available block type in the level editor. Wood and Stone blocks can be placed by first clicking on the corresponding button and then on the position in the grid. While one block type is selected the player can place as many blocks of that type as they want to. Holding down the mouse button and moving the cursor over the grid will place multiple blocks at once.

When adding chain blocks to a level in the editor, the user also will specify a blockID in order to determine which chain blocks belong to the same group of chain blocks. This is done by having the player choose a number between 1 and 5 on a slider.

The start and end platform can only exist exactly once per level, so if the player clicks on the start button he can move the start platform in the grid, but is not able to add additional start or end platforms.

The player can delete blocks from the grid using the delete button in the same way they can add blocks to the grid. The delete button internally adds empty blocks to the grid.

Saving a level works by giving the custom level a name in the text field and pressing the save button. As of right now the level gets saved into a fixed folder on the personal computer.

Loading a level is as simple as typing the level name into the text field and pressing the load button. The test level button functionality has not been implemented yet.

During the implementation of the level editor we made sure that the user interface is separated from the grid, otherwise clicking on a button could have caused an unwanted addition of a block behind the UI element.

3.2 Block Types

Apart from the start and goal platforms, the game includes 3 types of blocks at the time of writing, namely:

- Wood - Destroyed shortly after the player's touch
- Stone - Like wood, but takes a longer time to disappear
- Chain - groups of blocks that will vanish simultaneously after touched. The group of a block can be decided upon in the editor.

Blocks have been abstracted to Block data, which contains the information describing the blocks, and to the actual block prefabs to be used in the scene. A saved level stores the information of block data associated with a location on the grid, in order to instantiate a block prefab in this location when the actual level is loaded.

Blocks will instantiate a simple particle effect upon destruction. It is common to all destruction types as of now, but will be replaced by ones tailored to the block's

desired effect in later stages of development, such as wood burning and stone crumbling, to name a few.

3.3 Player Controls

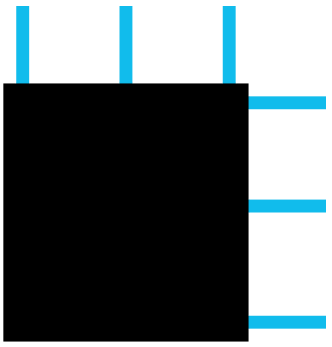
Summary: The player is capable of the following interactions as of now:

- Running (constant velocity, no acceleration)
- Jumping and falling (constant fall acceleration, limited maximum fall velocity, constant initial jumping velocity)
- Colliding with blocks to interact with them

The player controls should feel right, responsive, and not find themselves at the mercy of unity's physics engine when it is not required. Therefore, they have been decoupled from the physics engine completely, opting for some manual physics calculations, such as movement and fall acceleration, up to a limited fall speed value.
Doing so allows for true virtue over the players controls, tweaking them to the likeness of the standard set by critically acclaimed 2D platformers.

When opting to not use the physics engine, the problem of collision detection presents itself. In order to resolve this issue, a ray tracing based solution has been introduced to the control script. It is loosely based on this blog post by Nicholas DiMucci:
http://overdevelop.blogspot.com/2013/12/2d-platformer-collision-detection-in.html

In the following example, the player object (shaped like a box, scaled to have the proportions of a humanoid character) is moving to the right and up. Using the ray tracing solution, rays are being shot from the center of the object, as well as some translated rays on each axis. These rays will detect any colliders that the player will come in touch with, respecting the direction of movement. The same is used in order to determine if a player should start falling (casting downwards). In all directions, some padding and margins have been set in place in order to make the collisions feel more realistic given the character's model in its environment.



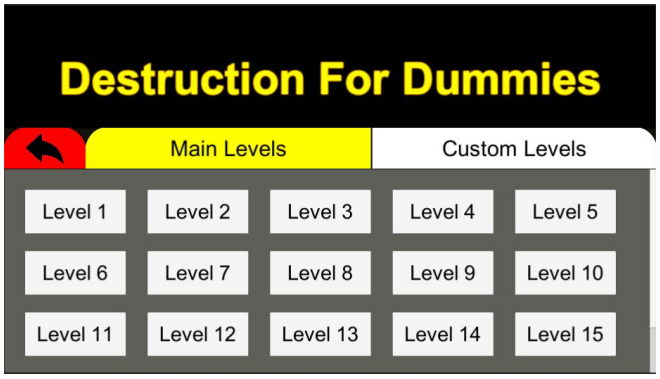
Additionally, it is the player's collision with the block objects that triggers the objects destruction mechanic.

In order to accomplish this, the player has a cube collider attached to it, which is padded slightly to allow trigger collisions with the surrounding objects.

Standard animation controller and state machine for some simple yet very serviceable character animations. The player model is made a child object of the actual player object, decoupling model and object from each other.

3.4 Menus and GUI

A simple main menu has been implemented, where the players have the choice to play the course of levels provided by us, create levels of their own, or select a specific level they want to play. The level selection is divided into the main levels and the custom levels sorted by the time of their creation.



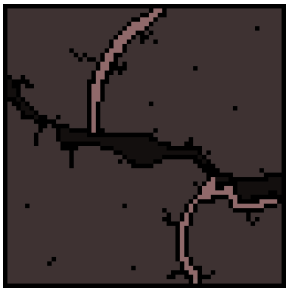
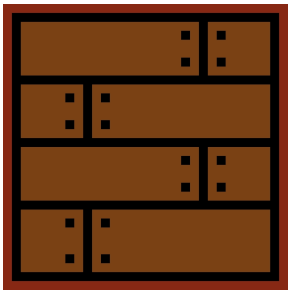
3.5 The levels

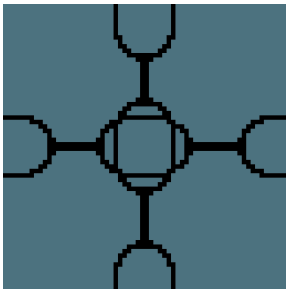
Over 8 Core levels implemented, sorted by their estimated difficulty.
The levels make use of all implemented block types and offer a mediation between platformer difficulty and puzzle solving.

3.6 Art

The player's model is a placeholder model. It has been provided by mixamo, along with the animations.

Block textures have been created with PiskelApp:



3.7 Goal achievement summary

In summation, we have managed to implement all the points that we set for the functional minimum and the low target, with the exception of the Xbox controls, which we postponed, as the game can be played perfectly with keyboard controls. Resulting in a fully functional game, where the player can choose one of the provided levels or jump into the level editor and create their own levels and then play them.

Concerning the desirable target we managed to implement some points:

- Animated Player Character: We decided on not modeling the player character ourselves and instead use a model from the website: https://www.mixamo.com. The reasoning for this is because we do not have team members who are versed in modeling and creating an animated model would require too much time and resources. Therefore, we focused on implementing an existing model and aligning the animations with the movement.
- Nice looking level selection UI: Although not initially planned for the interims, the level selection is also fully implemented, showing all existing levels and dynamically adding levels the user creates.
- More Complex Level Editor: We could partly complete this point as loading is already possible for us but the testing is still a work in progress.

4. Alpha release

4.1 Block Types

4.1.1 Death

The death block kills the player upon touching it. The death block can be used to create areas where the player is unable to go to or to make certain parts of a level more challenging by having an instant death condition.

4.1.2 Lock & Key

Lock blocks are indestructible until you hit the corresponding key block. Once you hit the key block with the correct id / color, all the lock block with the same id / color will become destructible and change their texture and the key block will vanish.

4.1.3 Respawn

Respawn blocks, as the name indicates, will respawn after a certain ingame time has passed. The block will not respawn if the player stands on the spot where the block would respawn. This block type is great for making levels where the player has to revisit an area again.

4.1.4 Charge

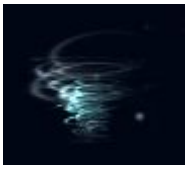
Once the player touches the charge block they get charged up with energy and can't destroy any other blocks anymore for a duration of ingame time. Once the time has passed the player outbreaks a strong force destroying all blocks around the player.

4.1.5 Restore & Restorable

Restorable blocks are not interactable until the corresponding restore block is touched by the player. They are only present by a black grid. Once they got restored they change their appearance and behave like normal wood / stone blocks. Blocks that intersect with the player will not be restored.

4.1.6 Updraft

The updraft blocks acts an upward force on the player allowing the player to reach higher blocks that they would normally not be able to land on.



4.2 Character changes

The height of the player has been readjusted in order to better fit with the environment.

Some physics improvements have been implemented.
Animations have been added, their transitions have been fine tuned.

Additionally, a guide character has been implemented for the core levels of the game. The player may interact with this character, who is always placed right next to the starting platform. Usually the characters will exchange story driven messages, but a few tips are also included. This character has also been modeled and animated through mixamo.

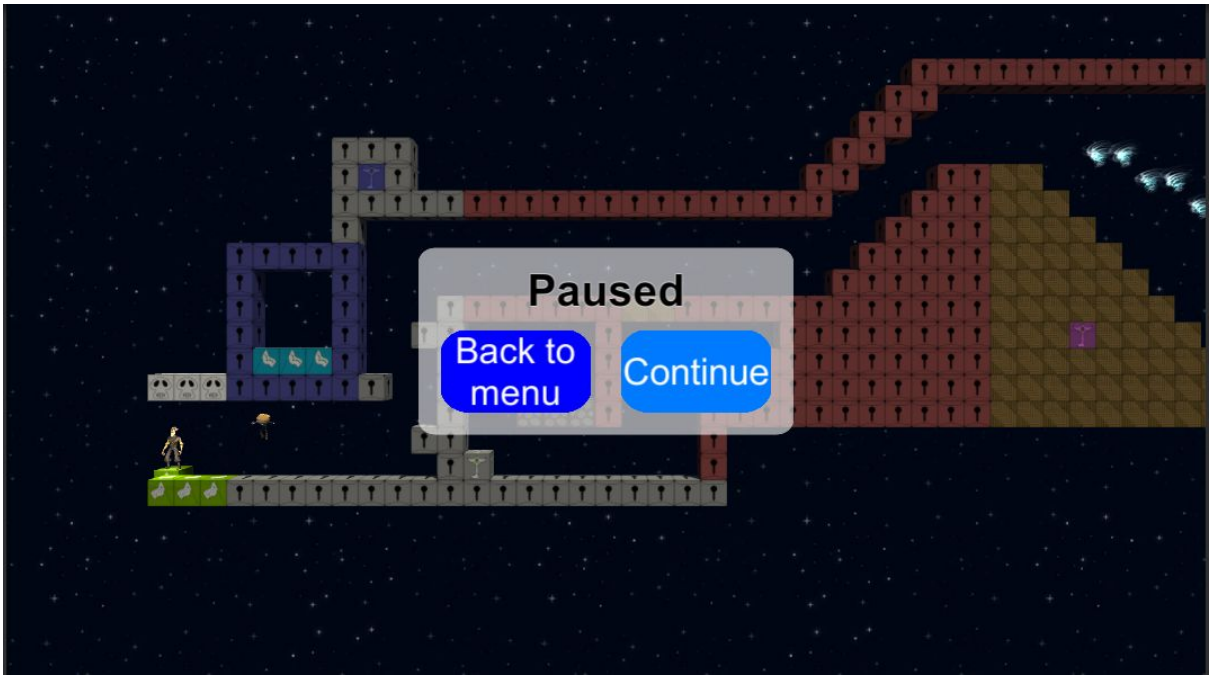
4.3 Play scene

The final play scene has a simple tileable starry night sky background representing the vast universe which the god of destruction has to watch over. To reduce the repetitiveness we randomly generate star signs created by the stars onto the background.

We added the advisor character into the main level scenes, to which the player can talk to revealing the story and tips for the level. We decided to make the story interaction optional as players might get annoyed by forced story cutscenes. We also reward the player for reading the dialogues because they do not only contain the story but the advisor also sometimes gives hints to the player on how to complete the level.

They player can exit the play scene whenever they want from the pause menu which opens when they press the esc-key/select-button.

Last but not least, we updated the ending animation. Now the god of destruction does not jump after reaching the goal, but instead they dance celebrating their successful trial completion.



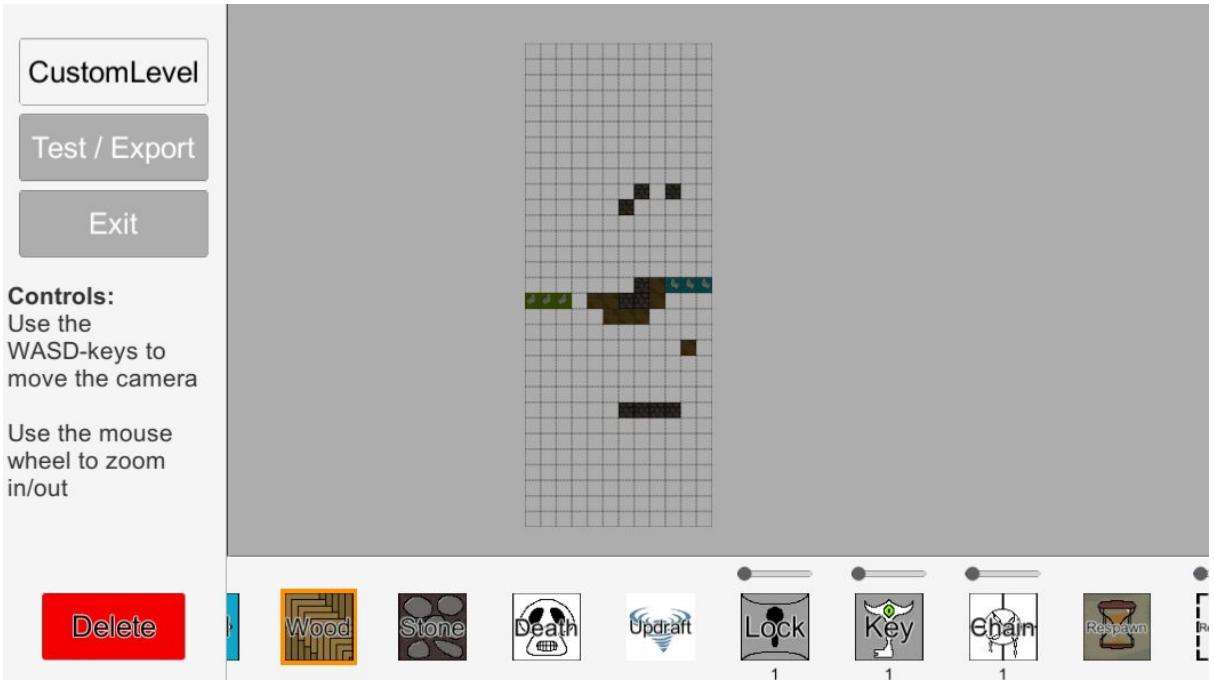
4.4 Level editor

The Editor now starts with a menu where all the created levels are displayed. The player can create a new level by inputting the name and the dimensions. If they want to scrap a level and start anew they can press the delete button and choose the level they want to remove. By selecting a level from the menu player changes into the edit mode.

In the edit mode, the player has a selection of up to 12 different blocks through which they can scroll through per mouse wheel or dragging. Per default, only the wood, stone and death blocks are unlocked. By progressing in the main campaign the player is introduced to new block types and after completing the introduction level they will be able to use them in the editor. The currently selected block, which is by default the woodblock, is highlighted orange. Some blocks also have a slider above them indicating they can have different versions. By moving the slider also the color of the block changes to identify corresponding blocks more easily. Moreover, hovering over a block reveals a short description text reminding what effects the block has.

The player can now navigate through the grid by using the "WASD"-keys or by dragging with the right mouse button. Also, they can zoom with the mouse wheel. The movement area and the zoom distance is limited by the size of the level.

In regard to the left menu, we removed the load button as this is now done in the start menu. Also, we changed the test level button into a test and export level button. This allows the player to test their level and after they completed the level once they are allowed to export the level until they add a new block.



4.5 Level structure

We structured the levels of our main campaign in a way so that the player only starts to know their ability to destroy blocks they touch, shown by wood blocks. Every couple levels a new block is introduced and between the introductions are level consisting of the known blocks to challenge the abilities of the player. In general the difficulty of the levels increases over time, except some introduction might be easier than the previous levels in order to demonstrate the new mechanic of the block more clearly.

The order of blocks we introduce is as follows:

Wood block -> Stone block -> Death block -> Updraft block -> Lock/Key block -> Chain block -> Respawn block -> Restorable/Restore block -> Charge block

4.6 Art & Sound

The game boasts a plethora of in game sound effects that trigger when blocks get destroyed, conveying the feeling of material burning, cracking and crumbling.

Additionally, background music is now available when playing the game. Royalty free music from the website: Bensound.com. Track name: Dance.

Regarding visual art:

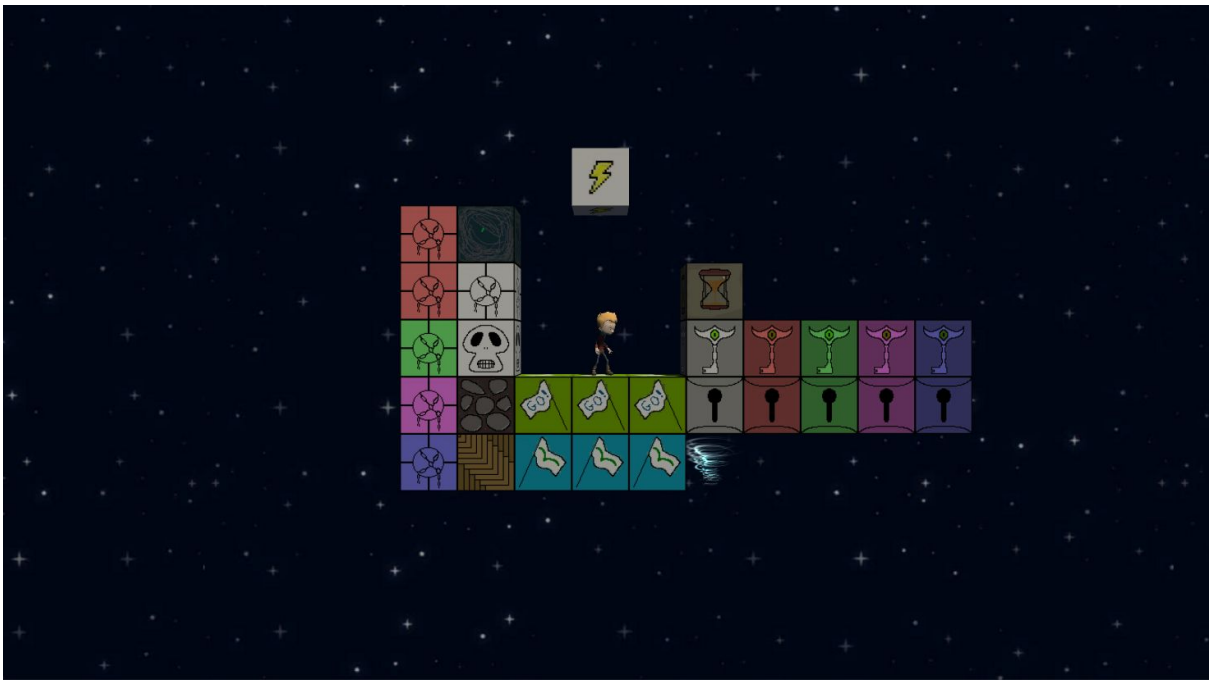
2D art: have been completely overhauled. Most blocks conform to the pixel art style used in the game's interim iteration, as it was easy to produce with a non artist team. All blocks also feature a modified destruction texture that is used to notify the player that the block has been touched and is about to call the interaction.

Particle Effects: Almost all blocks have unique particle effects upon destruction. These are distinct from each other and help make the blocks more intuitive in their own way. Examples: Respawn blocks will have particles fly out, halt, and then fly back in just in time for the blok to reappear. A charge block will cause the player to visually charge around them, in a radius that matches the following explosion, and so on and so forth. Other destruction particle effects serve a visual purpose to make the materials feel more dynamic, such as wood crackling and burning, stones crumbling, chains flung from chain blocks, keys from key blocks etc. In combination with the new sound FX, the blocks now feel a lot more engaging then they have in the past.

4.7 Challenges during development

A major challenge has been all art assets. None of the programmers on board for this project bring any significant art skills to the table, so there was a lot of adapting to undertake.
Some of the tasks have been fairly well mitigated using royalty free music and 3D models that are rigged, and animations that are available for free. This did however require us to abandon our old character designs.

As we wanted the blocks' art in game to be consistent, finding suitable art for it was very challenging. In the end we opted to create our own art assets for these, as a pixel art style was easy to create and keep consistent.



4.8 Progress and changes of the development schedule

Using the source control software GIT we were able to successfully follow the major part of our time schedule by dividing the tasks to us three developers. Each one developed a new feature in a new feature branch, once one of us was done the other two team members controlled the merge request before merging to master.

During development Liou focused on the level editor, Jonathan on the player movement scripting and 2D art, and Marco on the different block types.

Looking at the original planned timeline for the alpha release we were able to fully implement the following features:

- Background music & Sound FX
- Working and decent looking UI
- Higher features of level editor
    - User can select ID with certain block types
    - User can only export their level if they can beat their own level.
    - User can select the name, width and height of the custom level.
    - User can zoom in and out for easy level editing
- New block types
    - Death
    - Charge
    - Lock & Key
    - Respawn

    - Restore & Restorable
    - Updraft
- Story dialogues
- Tons of levels for the player to explore

As we already mentioned in the interim report none of us has any acceptable skills in modeling, so after some tries of making our own character we decided to not put that much effort into it and got an animated model for from the internet.

We could not find enough time for our high targets of having moving blocks and wall jumps functionality.

5. Playtesting

5.1 Conducting the playtesting

5.1.1 Procedure

We conducted playtests with friends, family, and colleagues as participants in private. All in all, we had 8 testers play our game and fill out the feedback form. We planned on conducting more playtests with more people but the session we had turned out longer than we expected. The general planned structure of the playtest looks like this:

Welcome and short introduction	~5min
Playing the main game	~15min
Testing the level editor	~5min
Filling out the feedback form	~10min
Casual talk about the game experience	~5min

We began with a short introduction about the game and their goal but stayed silent about all the specific block mechanics and controls as they are supposed to learn it from playing.

Afterward, they should play the game for around 15 minutes which was the estimated time to play through most of the provided levels. Unfortunately, we misjudged the difficulty of the levels and many participants took way over the estimated time to finish all the levels up to over an hour. Afterward, they could test the level editor and create their own level.

Last but not least they filled out the feedback form provided in 5.1.2 and we talked about their playing experience. With the talk, they could give us more direct feedback allowing us to gain further insight.

5.1.2 Feedback from

The feedback form was created with Google form and the questions below are marked with their answer type in the brackets. The types consist of linear scales, multiple-choice and text answers.
We also included a segment with questions from “The Game Experience Questionnaire” by IJsselsteijn, W. A., de Kort, Y. A. W., & Poels, K. (2013) in order to calculate a score table for our game.
The playtesters also needed to place our game in the play matrix. For that, we showed them an image of the matrix with examples on it and asked them to define where the game lies on each axis.

1. User Data
    a. Gender [Multiple-Choice]
    b. Age [Multiple-Choice]
    c. Rate how well versed you are with video games.[Scale]
2. General Experience[Scale]
    a. I was interested in the game's story
    b. I felt successful
    c. I felt bored
    d. I found it impressive
    e. I forgot everything around me
    f. I felt frustrated
    g. I found it tiresome
    h. I felt irritable
    i. I felt skilful
    j. I felt completely absorbed
    k. I felt content
    l. I felt challenged
    m. I had to put a lot of effort into it
    n. I felt good
3. Editor
    a. How easy to use was the editor?[Scale]
    b. How much did you like the process of exporting/testing a level?[Scale]
    c. How would you improve the editor?[Text]
4. Game Design
    a. How far did you get? (without cheat)[Number]
    b. How much did you like the ...?[Scale]
        i. Updraft Block
        ii. Lock/Key Blocks
        iii. Chain Block
        iv. Respawn Block

v. Restore/Restorable Blocks
vi. Charge Block

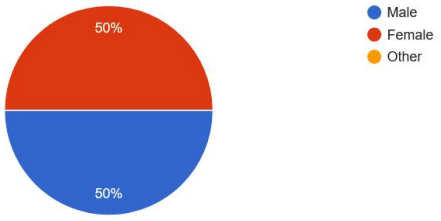
c. Explain why you like or dislike certain blocks?[Text]
d. How did you like the block effects / animations? [Scale]
e. Do you have any suggestions for new block types?[Text]
f. Rate how good the player controls felt.[Scale]
g. Rate the games physics.[Scale]
h. Did the game's physics hold up to your expectations?[Long Text]
i. PlayMatrix:
    i. How skill or chance based do you think the game is?
    ii. In which way does the game challenge the player?

5. Final Remarks
    a. Rate your overall game experience.[Scale]
    b. Which aspect of the game did you like the most?[Text]
    c. Which aspect of the game did you like the least?[Text]
    d. Describe how you would improve the game.[Text]
    e. Additional remarks which did not fit to any questions above.[Text]

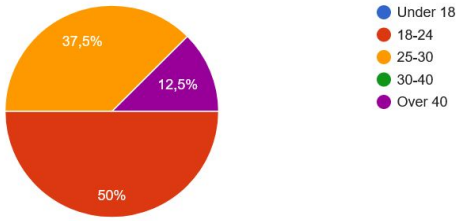
5.2 Playtesting results

5.2.1 User data

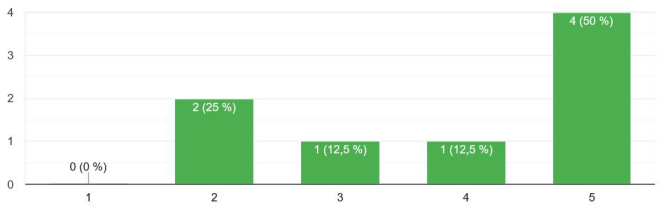
We had an even contribution of 4 male and 4 female testers.



Half of our testers were in the age group of 18 to 24, three in the age group of 25 to 30 and one was above 40.



The majority of our testers were well versed with video games.

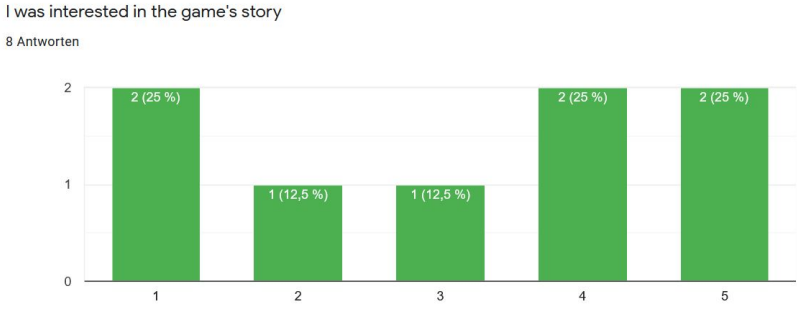


5.2.2 General experience

For this part we used the “The Game Experience Questionnaire” ingame module from
IJsselsteijn, W. A., de Kort, Y. A. W., & Poels, K. (2013)
Technische Universiteit Eindhoven

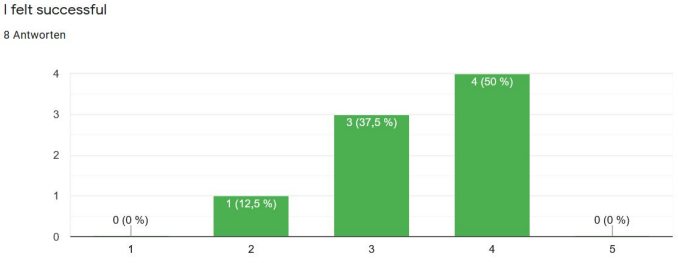
This allows us to use the scoring guideline defined in the paper in order to calculate how flow, tension, immersion, and challenging our game is and what the negative and positive aspects of our game are.

Since a good part of our game is based on the story of the teenage god of destruction Balagan we asked how interested our testes were in the story

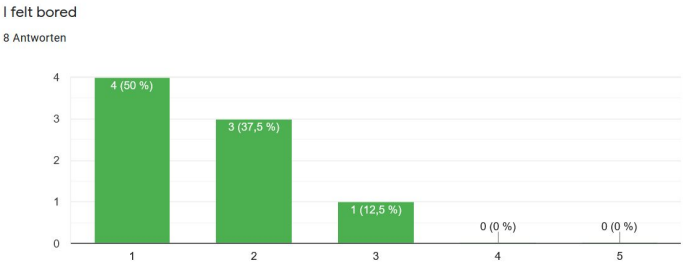


While some listed to all the dialogues in the game others skipped most of the story parts and were only interested in the gameplay.

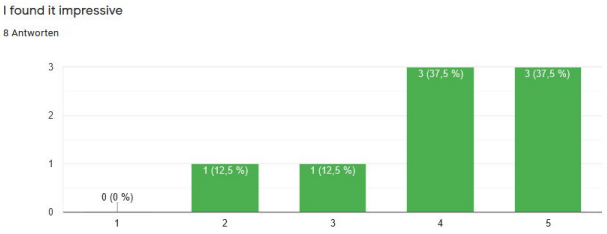
Players felt more or less successful.



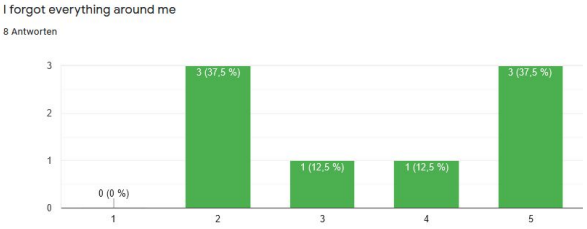
Nobody was bored playing our game.



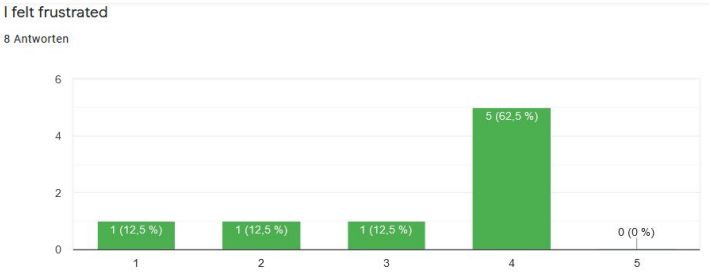
Most people were impressed by our game.



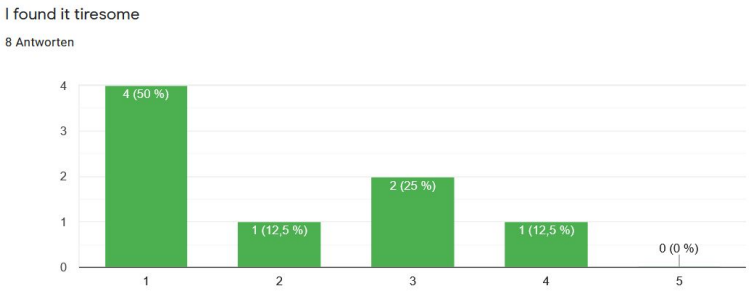
When we asked if the player forgot everything around them the results were differentiated



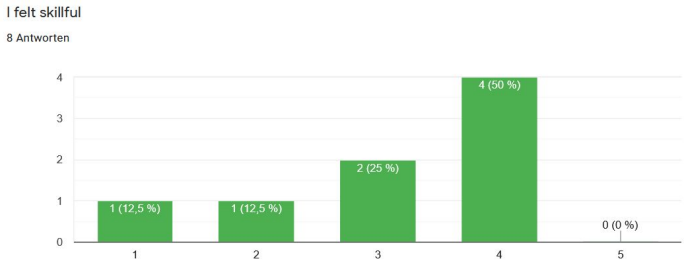
Quite a lot of testers felt frustrated, so this is definitely a point where we need to improve our game’s experience.



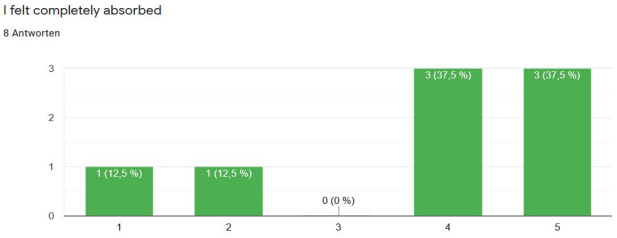
The majority of our testers didn’t find our game tiresome.



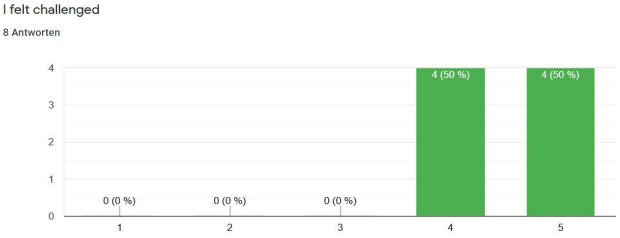
Many testers felt skillful.



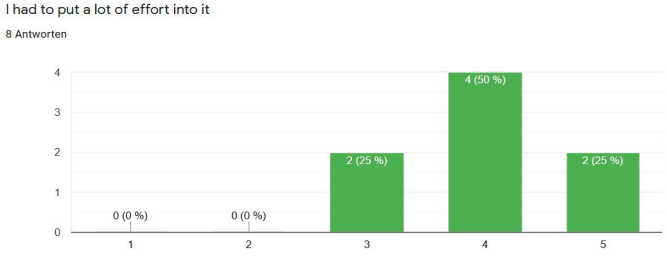
The game completely absorbed most test players.



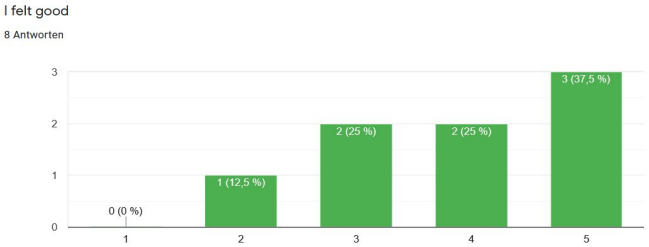
Due to the high difficulty of our game almost all testers felt challenged, so this was another point where we needed to improve the game.



Most testers also felt like they had to put a lot of effort into the game.



When we asked our testers if they overall felt good while playing the game the results were positive, but not as positive as we hoped for:



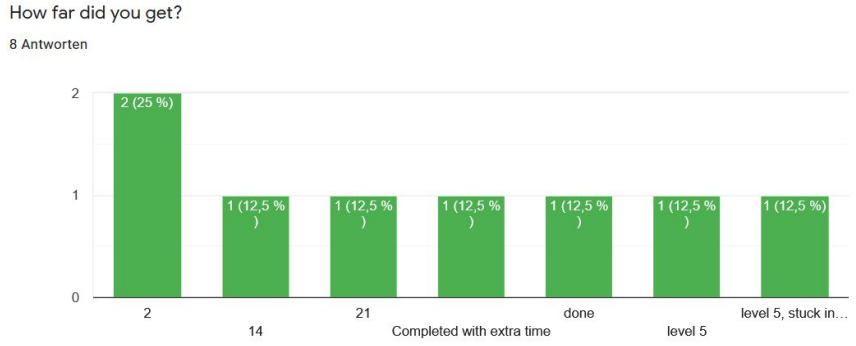
Using the scoring guidelines from the user experience paper, we calculated the following score table:

Competence	Immersion	Flow	Positive affect	Challenge	Negative affect	Tension
3.3	3.5	3.6	3.4	4.5	1.8	2.9

The score goes from not at all (1.0 Points) to extremely (5.0 Points).
Green color indicates that we want a high score and red that we want a low score for that column. Orange means we want a medium score of 2.5.
Since all of the results except for challenge are acceptable, we made several improvements towards making the game easier at start and more difficult over time in order to make the flow of the game feel more intuitive.

5.2.3 Game design

From a total of 8 testers, only 3 were able to complete the entire 21 levels we had so far. We did not expect player to already get stuck on level 2 or level 5 as often as they did. Therefore we decided to make easier versions of those two levels and overall change the level structure towards less difficult levels.



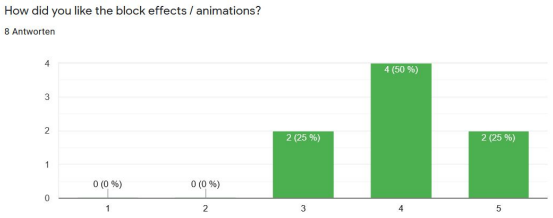
In the following table you can see how much our playtesters liked the different block types:

Updraft	Key / Lock	Chain	Respawn	Restore	Charge
3.9	3.3	3.0	4.0	3.2	2.7

To our surprise the charge block, into which we put the most effort got the lowest rating.

The main reason for this seems to be that the testers didn't understand what was going on after they touched it. Due to this we decided to visualize which blocks are affected once the explosion takes place.

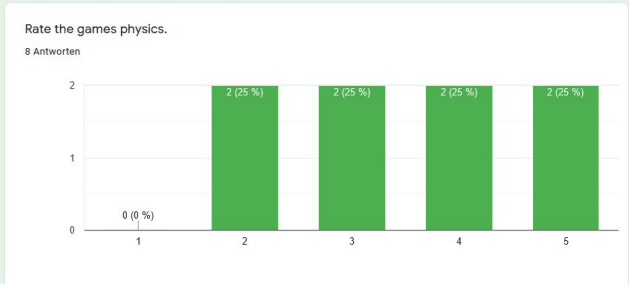
The block sound effects and animations on the other hand were rated overall good:



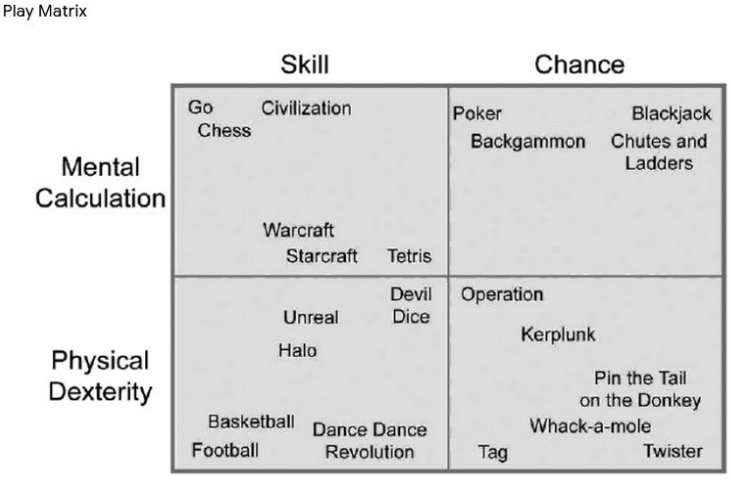
Testers have suggested us new block types, here is a list of the block types that were suggested to us and what we think about those ideas:

- Invincibility blocks
  - Doesn't make much sense with our story of a teenage god of destruction that can't handle his power.
- Speed boost blocks
  - We originally planned this block, but we tested the speed boost feature and didn't find it well fitting with the game.
- Story blocks
  - Might be nice, but we think the story dialogues with NPC's are more immersive.
- Teleportation blocks
  - We don't intent levels to be so big that you need to teleport at any time.
- Moving blocks
  - Implementing these into the editor is extremely difficult, therefore we decided to skip moving blocks for alpha release.
- Savepoint blocks
  - We don't intend levels to be so big that you need to savepoint at any time.
- Invisible blocks
  - We don't want to make levels artificially more difficult by adding these kind of blocks, the player levels should be able to see the path they need to take right from the start of the level.

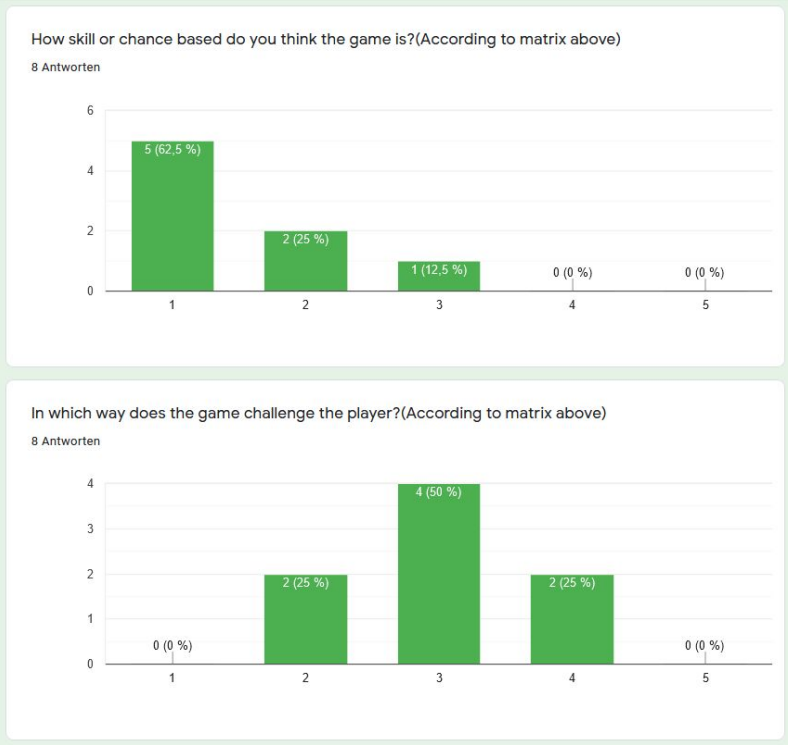
As for the player controls and physics the results were very diversive, but all testers agreed that the player collider is to big and doesn't fit with the model, so we downsized the player collider a bit.



We also asked the player to rate the game based on the play matrix:



with the following results:

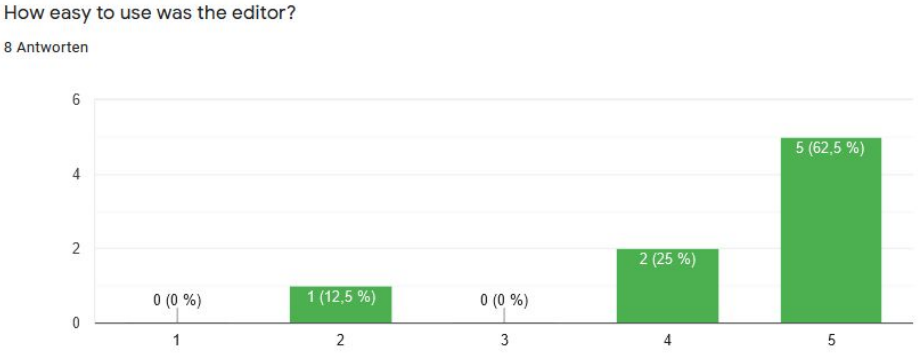


So the majority of player felt like the game was skill-based and challenged them equally in both a mental and physical way which matches our intention of the game as some levels are very mechanically based with precise jumps and some other ones are puzzles where the player needs to think through their moves to reach the goal.

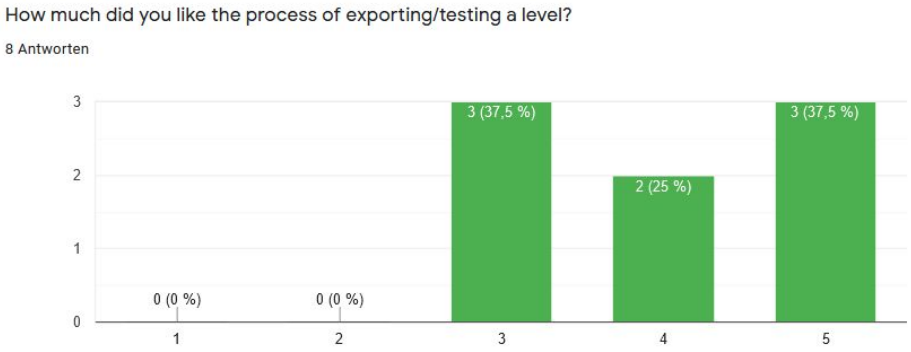
5.2.4 Level editor

The big majority of players loved the level editor feature.

Almost all agreed that the level edit was easy to use:

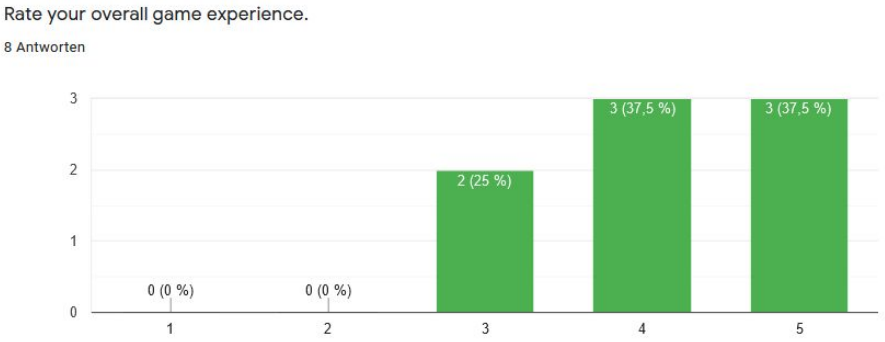


The restriction of having to successfully test custom made levels before exporting them was well received by our testers;



5.2.5 Final remarks

The overall game experience was rated good:



However we received three main critic points, that we needed to deal with:

- Collider of the player was to big.
- The difficulty of some levels is too high.
- The need for a better zoom feature, as the current implementation doesn't show the entire level if the level is very big.

5.3 Improvements based on playtesting results

5.3.1 Collider and physics

The player's collider has been tightened to better fit the assigned player model. In order to keep the imported model from slipping out of the collider's bounds, a new empty parent object has been implemented to serve as the axis for the player's model.

5.3.2 Zoom function

Revamped. Using zoom will now do 1 of 2 things:
If the player is on the starting platform (in the beginning or upon returning), the camera will pan to the center of the level, and zoom out. This reveals the entire level.
If, however, the player uses zoom anywhere else, then the camera reveals the area around the player, and the zoom is reduced by comparison.

5.3.3 Level difficulty

The game's difficulty curve has been streamlined.
Level 2, which has been notoriously difficult for the 2nd level, has been shortened and streamlined.
The stone block's life time (before the destruction) has been increased to 2 seconds, allowing the player more spare time.
This has alleviated the frustrations from level 5 and some others, making the hard segments not as punishing.

Some other levels have been modified since the intended mechanics were blatantly skippable, such as the introduction of the chain blocks.

Other levels, which have offered an intricate way of circumventing the presented mechanics, have been kept in place as rewards for clever players who notice these.

6. Conclusion

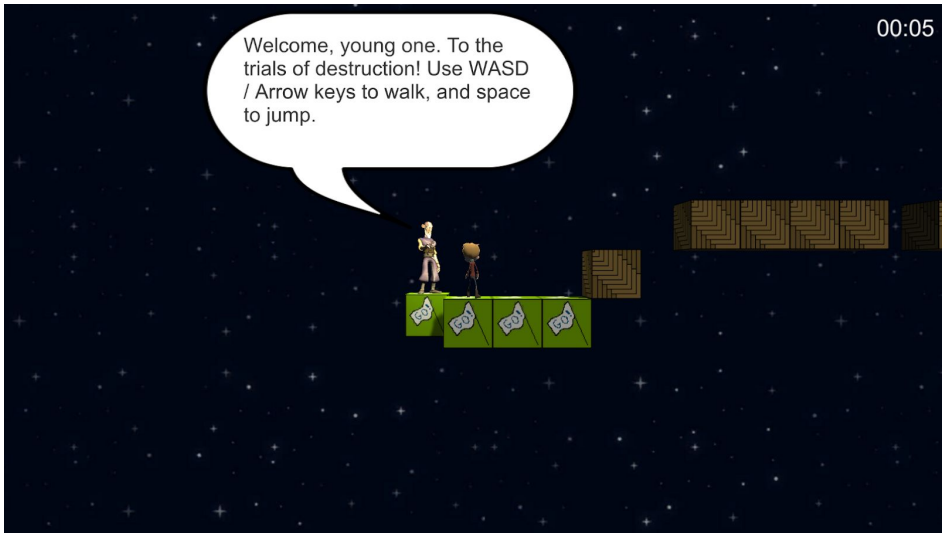
6.1 The Final Product

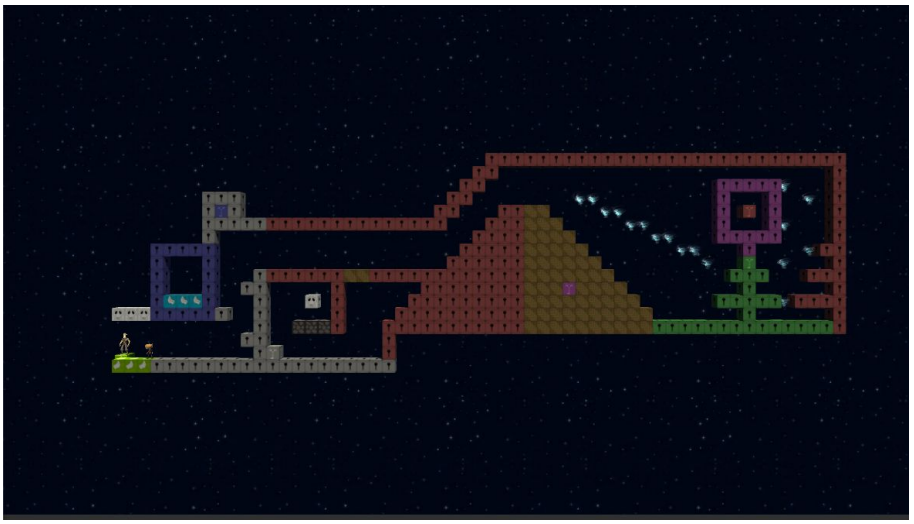
Our game is a destructive platformer where the player plays a future god of destruction who needs to learn how to use their destructive powers. Charging straight ahead without a plan will lead to the doom of oneself, therefore they must use smart planning and precise movements to reach the goal.

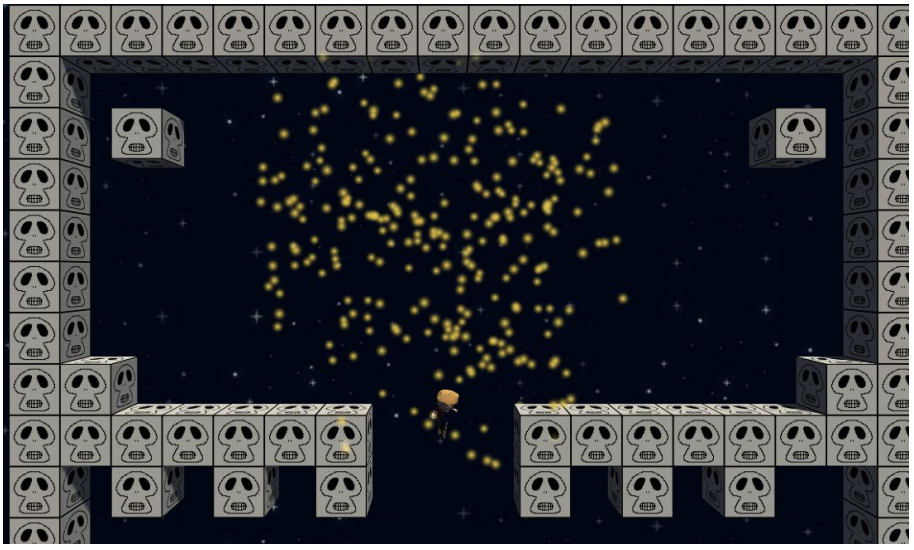
The main game consists of 21 levels built from 11 different block types and most of them will be demolished by the gods' destructive powers upon contact. Each type of block reacts differently to one's destruction and produces unique effects, which the player has to plan around to make it to the goal platform. Some levels require the destruction of blocks in a specific order, other demand precise platforming.

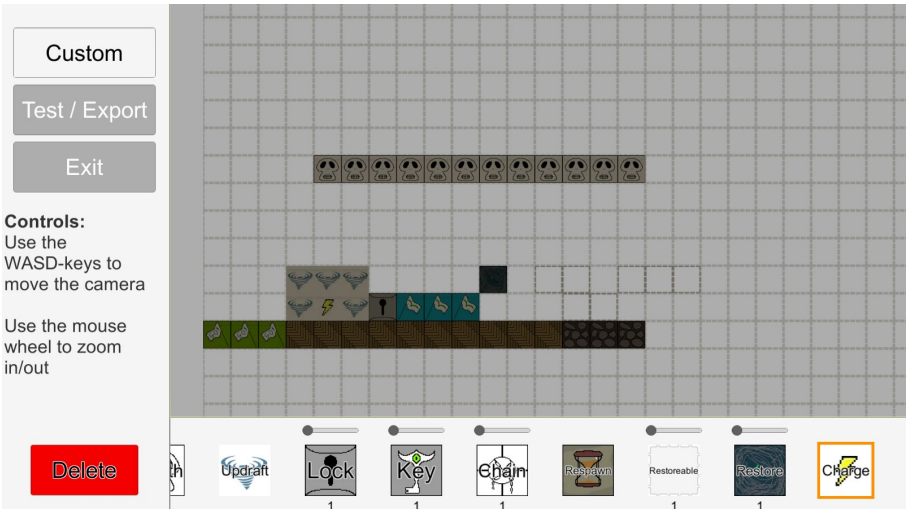
The player is accompanied by an advisor who gives them information about their controls, hints on how to solve some levels and talks about the story.

In the case the provided levels are not enough, the player can create their levels in the level editor. There they build their levels by placing the blocks, which they unlock by playing the main game, unto the grid and export them after they were able to complete the created level.








Custom
Test / Export
Exit
Controls:
Use the WASD-keys to move the camera
Use the mouse wheel to zoom in/out
Delete
Updraft
Lock
Key
Chain
Respawn
Restoreable
Restore
Charge

Changes since Alpha release:

- Changes we described in the playtesting chapter (5.3)
- Charge Block: We improved the tutorial text for the charge block to make it more transparent what the charge block does. We also added a particle circle with the effect radius and an effect where all the blocks within the explosion radius are darkened in order to show which blocks will get affected.
- More Effects: More destruction effects were also added, such as a respawn effect for the player and unique effects for the restorable blocks.
- Timer: We also added a timer that counts up when a player starts a new complete run from the main menu. With this timer, players can try to compete on who played through the game faster.

6.2 Experience

We are happy with the final state of the game. We managed to create many unique levels and puzzles which require a lot of attempts but feel satisfying when you solve them. This was exactly how we envisioned this game when the idea came during the brainstorming phase. When we created the milestones we planned to realize around half of the high-level targets, which we succeeded in although some planned features needed to be left out.

In general, the development schedule was more like a rough guideline instead of an exact plan. As students the time we have to work on the project varies week by week, therefore we usually set our own goals for each week depending on how much time we had. In terms of communication, we used Discord for online meetings where we discussed tasks, problems, and ideas. For each call we wrote documents were the most important points are noted. All in all, we had around 20 calls for this project and a handful of in-person meetings in the university.

Although we created a digital instead of a physical prototype besides showing us how important an editor implementation was, we scrapped most of the throwaway code and started from scratch.

The main development phase was at the beginning quite rough because many parts of the project depended on each other. After we finished the editor though, it went quite smoothly. We could in parallel develop new blocks and build levels with the existing ones. Because we could build the levels within our editor we enjoyed the level design/creation phase a lot. We could immediately build our ideas in the editor and test them.

6.3 Personal Impressions

Q: What was the biggest technical difficulty during the project?

Liou: The greatest difficulty was developing the editor in the beginning. Blocks needed to be structured so they can be placed, replaced in the editor and efficiently saved so we can load the levels later.

Jonathan: Definitely the editor. It provided many challenges, and has filled the role of bottle neck for development on multiple occasions. It is also an area where some compromises to our vision had to be made, such as having the editor usable with a controller or providing undo / redo functionality.

Marco: I agree that the editor was by far the most difficult part of the development. From having to make sure that raycast don't go through UI elements, make new block types work with for example the delete button, up to the advanced save mechanics the level editor always provided us with new challenges.

Q: What was your impression of working with the theme?

Liou: Destruction is a very funny theme and especially not a restrictive theme. It allowed many funny ideas during the brainstorming phase.

Jonathan: Destruction is open ended, but also a theme that is prevalent in video games, so it did feel familiar as a theme. It has led to interesting project suggestions such as a demolition derby.

Marco: Destruction works for a wide range of games, from shooter, jump & run, real time strategy, role playing games to puzzle games. The theme can also be interpreted in both a positive (e.g. destroying evil monsters) or a negative way (Godzilla destroying a human city).

Q: Do you think the theme enhanced your game, or would you have been happier with total freedom?

Liou: The theme was so non-restrictive that in my opinion most ideas one could come up with could be mapped to the theme and therefore it would make no difference.

Jonathan: Finding a game that would satisfy this constraint was a creative process, and I'm glad that we found something that does satisfy it but could also be considered a fun game independently.

Marco: I prefer working with a theme over working with total freedom. A theme allows you to build your game around, if you don't have a theme then often times the game becomes to crammed with different features and ideas that may not work together as well.

Q: What would you do differently in your next game project?

Liou: Nothing really, the whole development phase went very smooth from start to finish.

Jonathan: Would try go get some proper artists, I believe the game would benefit a lot from some visual upgrades.

Marco: This way the my first university project that we were able to finish in time with almost all features that were realistically possible in the given time, so I'm very proud of our game and would work the same way with the next project.

Q: What was your greatest success during the project?

Liou: Finishing the editor and being able to create levels in there, was in my opinion the greatest success. From this moment on we could start building real levels and play them.

Jonathan: Personally, I feel like creating the addicting game loop of trial and error, with near instant retries, is a large contribution to the game's fun factor.

Marco: I agree with Liou that finishing the level editor and making new levels and therefore more user content a lot faster was the biggest success but I also want to mention that finishing new difficult block types like for example the restorable blocks and charge blocks also felt very good.

Q: Are you happy with the final result of your project?

Liou: I am really happy with how the game turned out. It is really challenging and the destruction mechanic allows a lot of different level designs.

Jonathan: Very. It is fun and challenging, and allows for user created content. I would like to keep working on it for a future release.

Marco: Yes, what made me especially happy was the fact that on the demo day from the start of the event up to the end our seats with the alpha version of the game were always full with people playing the game.

Q: Do you consider the project a success?

Liou: Yes, with the given time frame I think that the game came out great. We have many unique and challenging levels and the effects and art look great although we did not really have an artist.

Jonathan: Yes. Given the time window for development, and limited resources, we were still able to pull off a game that is fun and easy to get into, yet crushingly difficult to master at higher levels, and allows for level creation.
It would have been great to have some more original art though.

Marco: Yes. While the game idea may have been seem a bit simple at the start seeing the final result shows that adding a just the small concept of destroying every block in the level made the from a casual jump & run to a very challenging and fast paced jump & run that gave our game testers a unique experience.

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

Liou: We mostly met the milestones during the project phase. We replaced some block ideas because they were either more interesting or the old ones were too time consuming.

Jonathan: More so than not, we met our milestones in a timely manner.
As expected, during development our visions were clearer than before, and we were able to prioritize the important features over the others. Some fun ideas were dropped, but ultimately, the end product is very satisfying.

Marco: We mostly met our milestones, sometimes we replaced a block idea with a less difficult one. Only the art and animation we were not able to do ourselves due to none of us being an artist.

Q: What improvements would you suggest for the course organization?

Liou: I did the course before and I think the winter course is time-wise more relaxed because of the christmas holidays. I would like that for the summer course the time-frame would reflect more the winter one.

Jonathan: I believe that a group counseling during the project elaboration phase would be greatly beneficial (For each group separately). Some of the project ideas might improve a lot with some technical / creative guidance early on.

Marco: I believe that the course is fine as it is. Having to present the next milestones always gives you a goal to work towards.