Computer Games Laboratory, Summer Term 2018

ARTISTA

Project Notebook

Erik Franz

Katharina Franz

Frederic Fulghum

Tommy Ngo

Daniel Probst

Luis Wiedenmann

Lucky
CHARM

Contents

1 Formal Game Proposal

1.1 Game Description

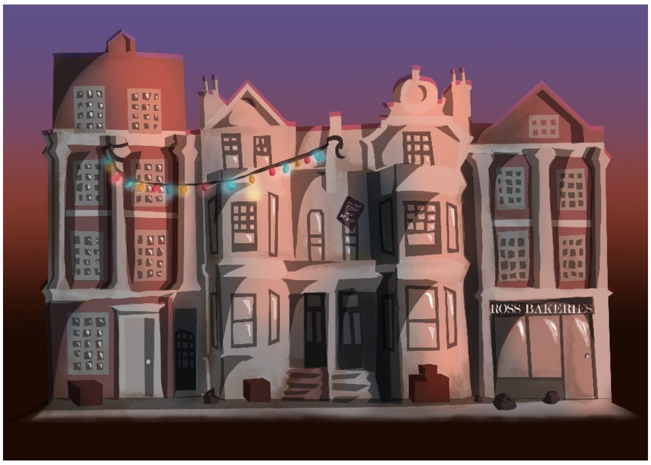
Artista is an up to four player arena fighter in which your only weapon is telekinesis.

The Players take the role of one of four different Circus Members and fight short telekinetic battles against each other to the top of an arena with destructible floors full of objects ready to be thrown around. The Arena also features randomly spawning power ups that create a more dynamic and interesting game flow.

1.1.1 Setting

Worlds End has arrived out of nowhere, and everything is falling apart and being swallowed by dust and smoke. The Circus crew of Artista was currently in London when this apocalypse hit them. But instead of running away they decided to go down with style. And so, they decided to have one great final show in their arena. The Arena is being magically pulled together by the circus manager, who uses her telekinetic powers in order to hold on to the arena long enough for the contest to play out.

1.1.2 Arena

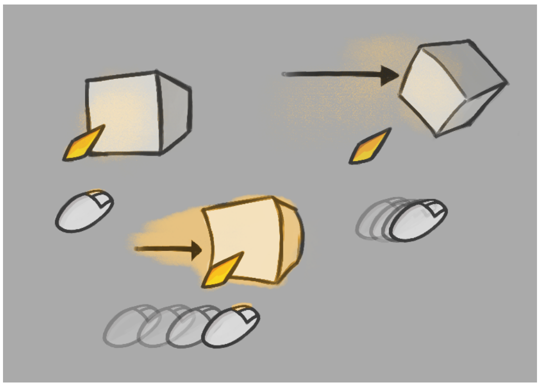


The whole game takes place in an arena with multiple floors. Most parts of the floors are destructible, however there are some platforms that can not be destroyed to keep a way to the top open. The Arena will be filled with an array of objects that can be thrown by players. The remains from destroyed parts will also be available for throwing.
The Arena size is adjusted to the number of players starting the game, so that for example two players will find themselves in a smaller area that forces them to engage and keeps the downtime to a minimum. Seeing how potential player numbers include two, three and four means three different sizes will be needed. The Arenas will be put together from the same assets, and as such having one means having the other two fairly quickly as well.

1.1.3 Characters

There are 4 different playable characters, all of them being performers that are part of the circus crew. They come from all around the world, and as such are quite a diverse and contrast rich collection. The differences between them are purely cosmetic, in order to allow everyone to play the character they feel like playing unhindered by their attributes. Furthermore it allows for a quicker visual distinction between players, and as such should make the gameplay more easily readable. Something that is certainly only in the High Target or Extra Category, is to ad voice lines to all the characters in order to give them more personality.

1.1.4 Telekinesis



Our core gameplay pillar is the use of telekinesis as a way to fight. The player has to hover over the object he wants to control with his mouse. As soon as his cursor is over the object he has to hold the right mouse button keep control over the object. Moving the cursor while controlling the object will move it by adding forces on it, so the objects follow the cursor depending on their weight and momentum. To pull the object closer of push it further away the player has to use his mouse wheel. As soon as the player lets go of the right mouse button he will stop controlling the object. If the object has momentum to it this is retained, and players are able to throw boxes in this way. There is a maximum distance between the player and the objects he can control and trying to move the objects further away while having it picked up is not possible.

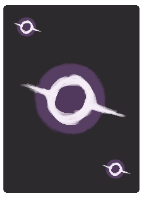
1.1.5 Stamina

Additionally to the Range limitation there is also a stamina system which stops the player from running around like a madman while at the same time flinging huge boulders left and right. Keeping control over an object by telekinesis will slowly decrease the stamina of a character, while actually moving an object will do so more quickly. Stamina also depletes when the character is running or does other energy consuming tasks like jumping. Stamina will start to regenerate after a short period of time in which the player is not performing any stamina consuming actions. If the High Target feature of taunts is achieved it would include taunts that the players can use to speed up their own stamina regeneration at the cost of a short time where they cannot move. This may also apply a stamina regeneration debuff on enemies or drain their current stamina for a small amount.

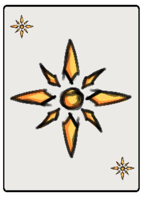
1.1.6 Power Ups

Artista includes three different Power Ups order to create a more dynamic and interesting flow of combat as they are something that the player will be forced to plan with. Power Ups spawn at random places in the arena, but never too close to one of the players. They are represented by glowing playing cards that rotate and hover above the ground. Walking over one of these power ups will give a certain amount of uses, depending on what turns out to to be fair and balanced during playtesting. As soon as one player has collected it, the current pick up disappears and it takes some time until the next one will appear in the arena.
There are Three Power ups which the player can obtain and use:

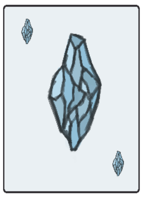
Singularity The player can use this ability whenever he has an object under his control. Doing so will turn the object he is holding into a singularity which pulls objects and players towards it. The sphere exists for several seconds and repels nearby objects with a small outward burst of energy at the end. The singularity is represented in the game with dark purple and black colors, both in its card design as well as the hand effect.



Overcharge Overcharge works similar as the singularity, as it is only usable when controlling an object. But instead of creating a pull, overcharge goes into the opposite direction and will create an explosion around the object, pushing players and objects away and potentially hurting them. The overcharge is represented in the game with orange and red colors.



Freeze Being the most defensive ability, freeze allows the player to freeze one or multiple object and stop it in its momentum. In order to do so the object cannot be controlled by another player. The frozen object will thaw out after several seconds, or if too much force is applied to it. Freeze is represented in the game with light blue colors.



1.1.7 Gameplay

A full game is called a contest and consists of a number of smaller rounds in succession. The player winning most of the rounds is declared winner of the contest, and if there is a draw a final round is added. All players start each round at the bottom of the arena and their goal is to reach the center of the top floor. In order to do so the players have to use the loose objects with their telekinetic powers to climb and build their way upwards. They can also throw objects at each other to push their adversaries down or at structural parts of the arena to make floors collapse to keep players from advancing. The remains of destroyed structure will also be available for throwing. Indestructible platforms, that are part of each floors, will allow players to quickly catch up if pushed down. A round ends if a player reaches the center of the top floor. If no player is successful after a certain time floors will start breaking away from the bottom.

1.1.8 HUD

Besides from the main menu Artista uses as little HUD as possible, in order to keep the screen free of visual clutter and distractions. Playtest are highly needed here in order to see if these systems are understandable for new players and might lead to more traditional UI choices if needed. The information that would normally need a representation in the HUD, Stamina, available power ups and the round timer, are all worked into the game environment.

Stamina is represented by a glowing area around the character, which is bigger and more strongly glowing when he has full stamina and is smaller and less visible when he has less stamina left.
The currently selected Ability is shown as a colour wise matching glow around the hands of the character completed by fitting SFX and a distinct sound.
The Timer is delivered by short voice lines from the circus leader and by increasing dust and fog gathering in the arena, as well as small Earthquakes that signal the start of the last 10 seconds.

1.1.9 Menus

All Menu icons will take the form of old style Circus Tickets (rectangles with little spikes at both ends where they were torn from a longer piece of paper). To keep the screen from being too boring, some artworks of the characters and the environment will be displayed, fitting the menu the player is currently in. The Circus leader will give some comments in the form of voice lines (additionally displayed through subtitles) on actions done in the menu, like turning sounds on again or choosing to play.
Needed Menu options:

- Play Game
  - Host Game
  - Join Game
- Options
  - Sound
  - Controls
- Credits
- Quit Game

1.2 Technical Achievement

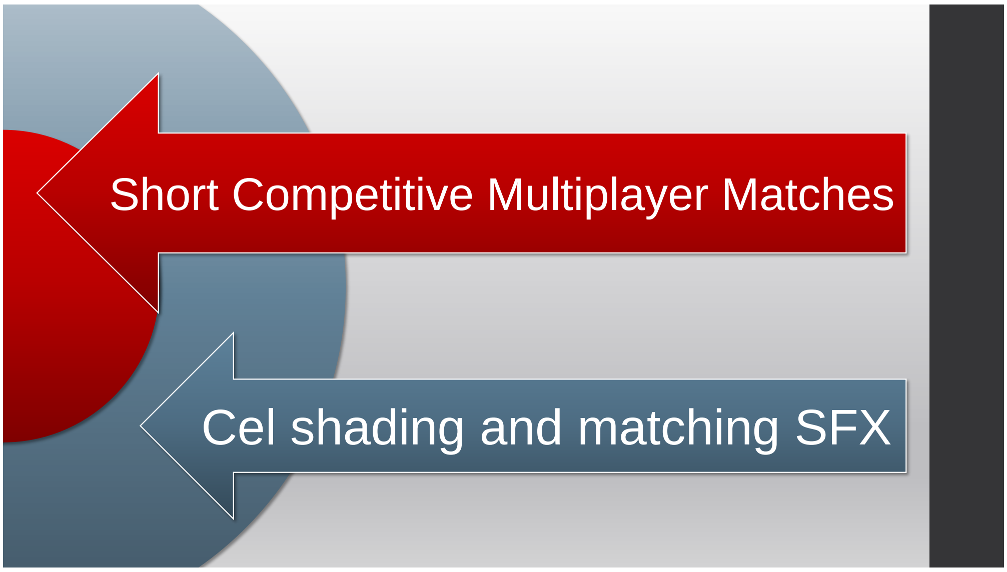
Our main technical achievement goes along with the course theme ”High Contrast” as well as the design and art focus of the MD.H. It will be the implementation of a custom, unified Cel Shaded look. In addition to the basic Cel Shaders for the environment and characters, we will create our own SFX, using particles and shaders. This will include effects for the basic telekinesis, different special abilities, ability pickups and environmental effects. The in-scene UI mentioned above will not only use its own SFX (e.g. for carried abilities) but also ties into the main environment and character shaders as well (e.g. the glowing stamina circle projected in the environment).

1.2.1 Technical Features

The game will be implemented with the Unity game engine for PC. For the successful implementation our game will also need several other technical features.
As the core gameplay is physics based we need a sound physics simulation. Although the basic simulation is handled by Unity, we still need to build the telekinesis controls, interaction of special abilities with the simulation and a damage mechanic on top of it.
The pickups need to be randomly placed on the map and player need to be able to pick them up.
The game needs to manage health, stamina and special abilities for all players.

We chose a 3rd Person movement and camera control scheme. The camera controls have to support both player movement and telekinesis controls.
As this is a directly competitive game we need to integrate a networked multiplayer with the standard Unity functions that synchronizes all of the above.

1.3 "Big Idea" Bullseye



Short Competitive Multiplayer Matches

Cel shading and matching SFX

At its core our game is a short competitive arena fighter with physics based combat.

1.4 Assessment

Artista will be a lightweight game in a joyful atmosphere that is easy and intuitive to understand for new players. To help with that goal this game will feature compelling, unified Graphics with a Cel-shaded look. Players will fight each other in a carefully designed arena with various telekinetic abilities.

1.5 Development Schedule

We will at first put our focus on getting all the technical aspects of the game up and running, with just as much art as we need to test all of them. Following the advice to plan in layers this is how we categorized the tasks needed to create the game.

1. Functional Minimum
   - (a) Working telekinesis
   - (b) Core Game Logic: for example, Life Points as well as loosing and winning conditions
   - (c) Cell Shader for environment and characters
   - (d) Working Multiplayer
   - (e) Movement and Camera
2. Low Target

(a) Power Ups
(b) Different Object Types: for example, Crates, Barrels and Lanterns
(c) SFX
(d) Arena layout
(e) Coarsely destructible arena
(f) Round Timer
(g) Stamina System
(h) One Player Character Model with animations

3. Desirable Target

   (a) Three more Character Model's with animations.
   (b) Interactive Tutorial
   (c) Circus Leader
   (d) UI
   (e) Character Ragdoll
   (f) Death cam
   (g) Sounds

4. High Target

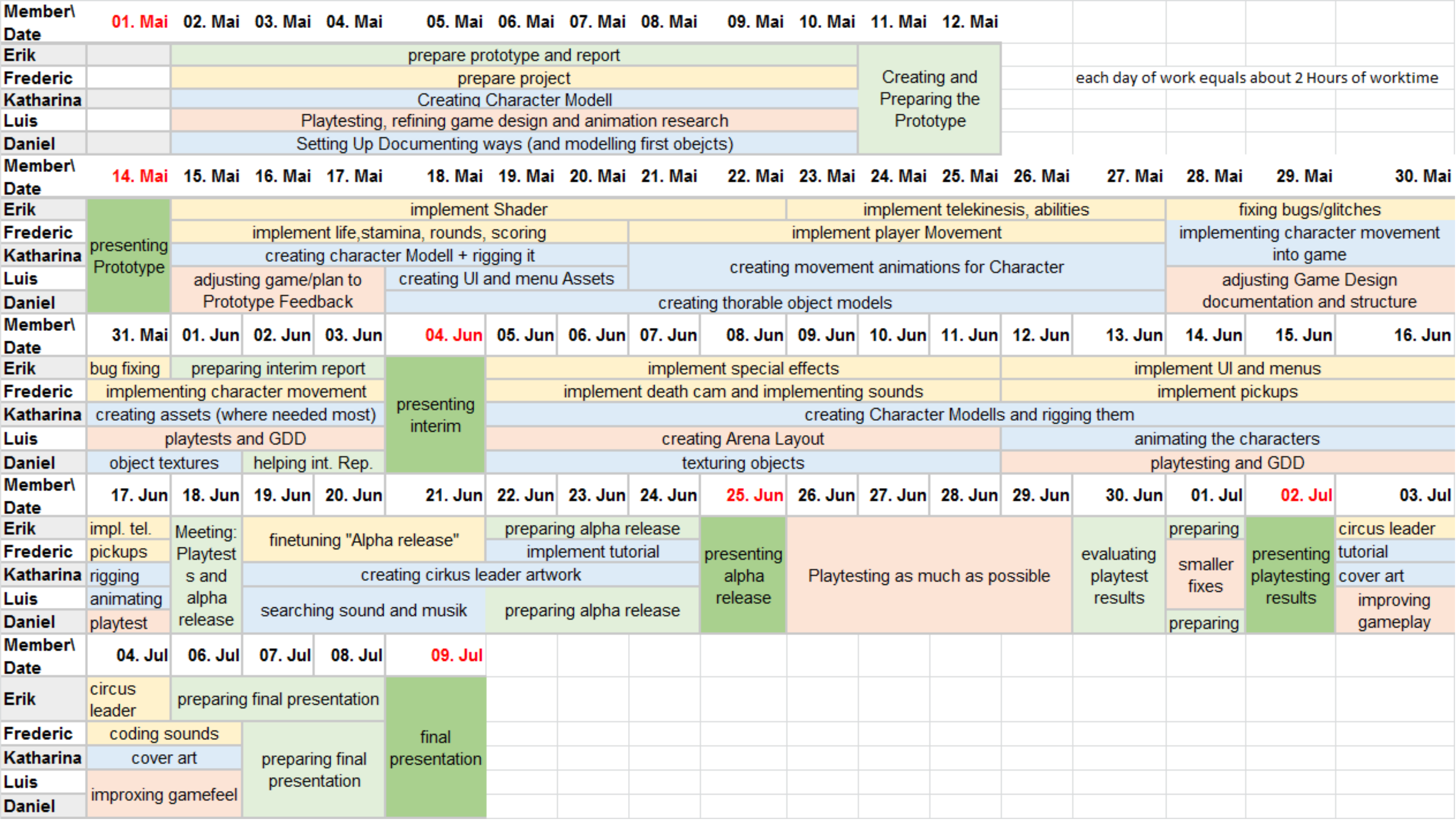
   (a) Destructible Environment
   (b) Loadingscreens
   (c) Singleplayer Sandbox and/or High score Mode
   (d) Music
   (e) Taunts
   (f) Voice Acting

5. Extras

   (a) Additional Game modes: Capture the flag, King of the Hill, Sudden Death,
   (b) Additional Power Ups

1.5.1 Task List and Timeline

Task List Erik Franz
core game mechanic:
character movement (run, jump, ground detection) 4 iterations, still needs work
character climbing prototype
character stamina (telekinesis, abilities, sprinting)
camera controls
basic telekinesis (grab, hold, throw)
advanced telekinesis (grab points for objects)
special abilities (freeze, overcharge, singularity) → interaction with objects
destructible structures (coarse system) → interaction with telekinesis, collisions and forces
networking:
lobby, player setup
player sync
dynamic object sync
player setup and sync
overall game loop
graphics:
basic Cel shader (main light, point lights, diffuse, normal, emissive)
→ additional for SFX: noise based effects, fresnel, transparency
→ global effects (circle effects on objects)
post processing: bloom, tone-mapping for HDR (used in emissives and SFX)
SFX for telekinesis
SFX for stamina
SFX for special abilities
(SFX include particles, shaders and connection to core mechanics)
simple menu/UI
Unity editor:
UI for shader and controllable objects
indicator tool for structural dependency



Member\ Date	01. Mai	02. Mai	03. Mai	04. Mai	05. Mai	06. Mai	07. Mai	08. Mai	09. Mai	10. Mai	11. Mai	12. Mai					
Erik		prepare prototype and report									Creating and Preparing the Prototype						
Frederic		prepare project												each day of work equals about 2 Hours of worktime			
Katharina		Creating Character Modell															
Luis		Playtesting, refining game design and animation research															
Daniel		Setting Up Documenting ways (and modelling first obejcts)															
Member\ Date	14. Mai	15. Mai	16. Mai	17. Mai	18. Mai	19. Mai	20. Mai	21. Mai	22. Mai	23. Mai	24. Mai	25. Mai	26. Mai	27. Mai	28. Mai	29. Mai	30. Mai
Erik	presenting Prototype	implement Shader								implement telekinesis, abilities					fixing bugs/glitches		
Frederic		implement life,stamina, rounds, scoring						implement player Movement							implementing character movement into game		
Katharina		creating character Modell + rigging it						creating movement animations for Character									
Luis		adjusting game/plan to Prototype Feedback			creating UI and menu Assets										adjusting Game Design documentation and structure		
Daniel					creating thorable object models												
Member\ Date	31. Mai	01. Jun	02. Jun	03. Jun	04. Jun	05. Jun	06. Jun	07. Jun	08. Jun	09. Jun	10. Jun	11. Jun	12. Jun	13. Jun	14. Jun	15. Jun	16. Jun
Erik	bug fixing	preparing interim report			presenting interim	implement special effects							implement UI and menus				
Frederic	implementing character movement					implement death cam and implementing sounds							implement pickups				
Katharina	creating assets (where needed most)					creating Character Modells and rigging them											
Luis	playtests and GDD					creating Arena Layout							animating the characters				
Daniel	object textures		helping int. Rep.			texturing objects							playtesting and GDD				
Member\ Date	17. Jun	18. Jun	19. Jun	20. Jun	21. Jun	22. Jun	23. Jun	24. Jun	25. Jun	26. Jun	27. Jun	28. Jun	29. Jun	30. Jun	01. Jul	02. Jul	03. Jul
Erik	impl. tel.	Meeting: Playtests and alpha release	finetuning "Alpha release"			preparing alpha release			presenting alpha release	Playtesting as much as possible				evaluating playtest results	preparing	presenting playtesting results	circus leader
Frederic	pickups					implement tutorial									smaller fixes		tutorial
Katharina	rigging		creating cirkus leader artwork														cover art
Luis	animating		searching sound and musik			preparing alpha release											improving gameplay
Daniel	playtest														preparing		
Member\ Date	04. Jul	06. Jul	07. Jul	08. Jul	09. Jul												
Erik	circus leader	preparing final presentation			final presentation												
Frederic	coding sounds		preparing final presentation														
Katharina	cover art																
Luis	improxing gamefeel																
Daniel																	

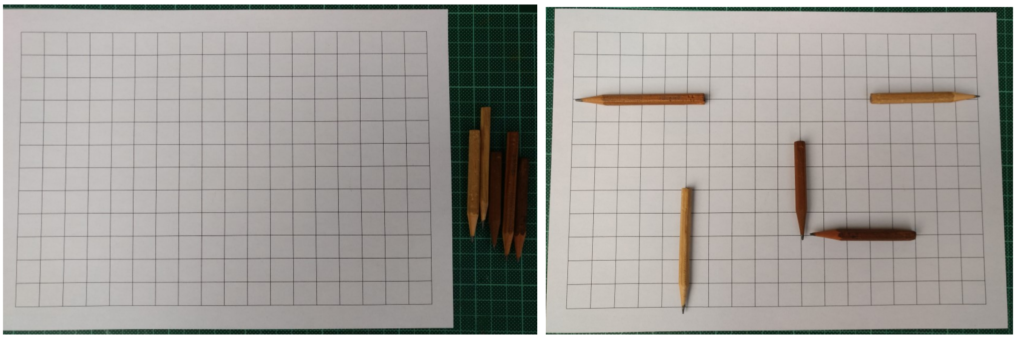


Figure 1: The base map for 2 or 3 players and short pencils used for walls.

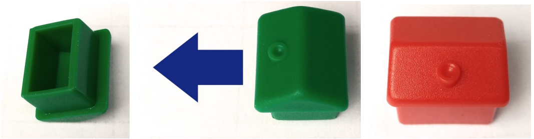


Figure 2: Dynamic objects. still, moving (chimney as direction indicator), grabbed.

2 Prototype

We created a physical prototype to evaluate some of the core mechanics in an easier controllable environment. In addition to the physical prototype we have tested the dynamics of the telekinesis throwing mechanic in a software prototype.

2.1 Prototyping Goals

The core combat mechanic of physically throwing objects at each other will not be modeled in the physical prototype, as we think it is save to say that it is generally considered fun. Furthermore we explored the telekinesis mechanic in a software prototype (sec. 2.5.1). With this prototype we focus on exploring: arena size (to get a rough idea of how large it needs to be), availability of objects, stamina balancing, limitation of the telekinesis grab range and the player number influence on gameplay. This will be modeled in a 2D time and space discrete environment: as a turn based game on a grid.

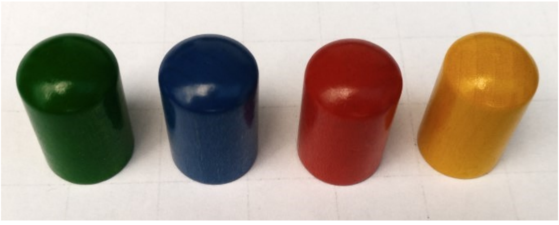


Figure 3: The 4 player pieces.

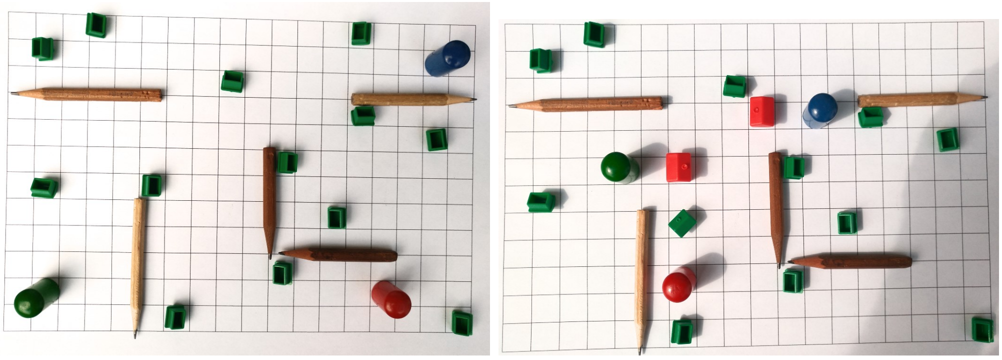


Figure 4: Start setup and example of a mid-game state.

2.2 Setup

The 2D map consists of a simple grid and static walls are represented by short pencils, see Fig. 1. There are dynamic objects,which can have 3 states: still, moving and grabbed (Fig. 2) and up to 4 players, which are represented by the colored player pieces (Fig. 3). Each player has to remember their stamina per round and grabbed object.
Special abilities, including their pickups, are not modeled.

2.3 Game Loop

The final game loop we came up with is as follows:

Players start each round with 3 stamina
Players can move according to the Movement rules
Players can use Telekinesis according to the rules
Physics rules are applied

2.4 Game Rules

The prototype has, same as the game, a free-for-all gameplay. There are no teams and no direct coordination between players is allowed.

Movement The player can move 2 fields in any direction unless it is blocked by an object in any state, a wall or a player. If the player is carrying an (grabbed) object movement cost 1 stamina per field. The player can also choose to drop a carried object (for free) to move without stamina cost.

Telekinesis The player can grab any still object in ca. 3 fields range (see Fig. 5) for 1 stamina. The player can also attempt to catch a moving object for 1 stamina with a success chance of $\frac{1}{3}$ once per object and round. If an object is grabbed it can not be grabbed by another player (it is locked) and a player can only grab 1 object at a time. Once the player has grabbed an object they can move it in any direction within the grab range (Fig. 5) for 1 stamina per field. If the target field is occupied by another object the "Physics"-rules apply. Finally the player can choose to throw the object in any direction for 1 stamina or drop it for free.

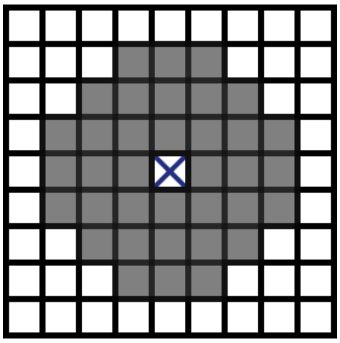


Figure 5: Player grab range.

Physics move moving object 2 fields in their direction and resolve collisions according to these rules:

updated object	hit object	consequence/action
moving	wall	stop
moving	still	move still one field if not blocked by something else, both stop
moving	moving	both stop
moving	grabbed	stop
moving	player	move player one field if not blocked, stop, player looses 1 health
grabbed	wall	blocked
grabbed	still	move still in ”correct” direction if not blocked by something else
grabbed	moving	can stop moving for 1 stamina → grabbed - still
grabbed	grabbed	blocked
grabbed	player	move player in ”correct” direction if not blocked, 1 additional stamina

2.5 Results

In the end the prototype worked surprisingly well, given the dynamic core mechanics of our game. However, it was quite hard to find a model that was playable on paper and still close enough to the original game idea to be able to translate findings from the prototype to the game. It was also hard to find a game loop that was fair. In the first iterations the last player could always kill other players without them having a chance to evade.

Even without the special abilities the ruleset became rather complex. It showed how many possibilities for interactions between players and objects there are and gives an idea of how hard the game will be to balance. Because of the many rules to consider the prototype was rather cumbersome to play at first, but once we got into it it was quite fun. These concerns don’t translate to the game however, as most rules will be handled by the engine.

The grab range limitation and object travel time, combined with the ability to evade thrown objects gives an incentive to come close to other players. We also found that it is possible to defend against thrown objects with grabbed ones, which can results in some form of melee. The availability and placement of objects is tied to that and turned out to be quite important, as the whole gameplay depends on it. Another concern that translates to the game is the ability to grab and hold objects behind walls, where the player can not see them. This can be resolved with LoS-checks. The arena size and layout is an difficult and important aspect since it depends on many other factors (e.g. player and object speed, grab range and number of players). Playtests with dummy arenas will be necessary to determine good arena sizes.

Despite the lack of dynamics we gained surprisingly many insights about the underlying physics. While many of them will be handled by the engine some might require special attention.
Overall the physical prototype was an interesting approach and insights from a different point of view. Sadly the prototype had not much to do with the game in terms of gameplay. It is rather slow and strategic in contrast to the intended fast and reaction based gameplay of the game.

2.5.1 Software Prototype

Our software prototype originates from an early tech test for testing the implementation difficulty and complexity of various required features in Unity. This prototype also includes a rough version of the telekinesis throwing mechanic. We used it to test the fun and controls of this core mechanic and if was considered fun by all team members. This Tech Test also showed the many parameters and possibilities for balancing, similar to the physical prototype.

3 At MD.H: Game Design Session

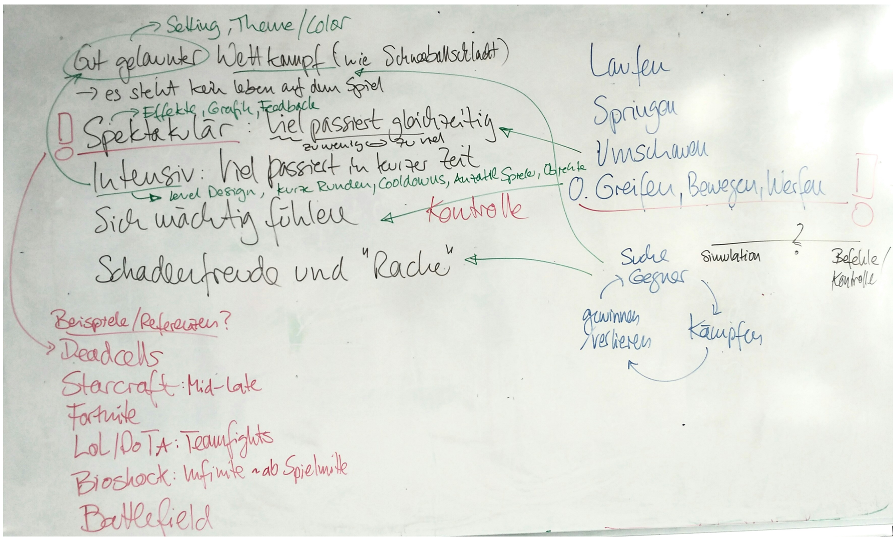

Setting, Theme/Color
Gut gelaunter Wettkampf (wie Schneeballschlacht)
→ es steht kein leben auf dem Spiel
Effekte, Grafik, Feedback
Spektakulär: Viel passiert gleichzeitig
zu wenig ↔ zu viel
Intensiv: Viel passiert in kurzer Zeit
level Design, kurze Runden, Cooldowns, Anzahl Spieler, Objekte
Sich mächtig fühlen
Kontrolle
Schadenfreude und "Rache"
Beispiele/Referenzen?
Deadcells
Starcraft: Mid-Late
Fortnite
LoL/DoTA: Teamfights
Bioshock: Infinite ~ab Spielmitte
Battlefield
Laufen
Springen
Umschauen
O. Greifen, Bewegen, Werfen
Simulation
?
Befehle/Kontrolle
Suche Gegner
Kämpfen
gewinnen/verlieren

Figure 6: Experience goals (left), core mechanics (top right) and game loop (bottom right).

During a game desing session at the MD.H on 16.05 we defined several experience goals our game should fulfill as well as the core features of our game. We figured out how these features relate to the experience goals and which are most important to our game, see Fig. 6.

While play-testing of our software prototype we found out that our core mechanic of controlling objects with telekinesis does not match the experience goal of feeling mighty and control as it was simply to hard to hit the relatively small player characters. To overcome this issue we could either make it easier to hit other players by simplifying the control scheme or change the goal of the game where player must hit a larger target. We decided to change the goal and adjust the gameplay as follows:

The primary goal of the players is now to reach the top of a arena with multiple, mostly destructible floors. Telekinesis is used to aid in climbing and building a way up and to hinder other players by destroying structural parts of the arena by throwing objects. This way the targets are mainly the larger arena structures, while it is still possible to hit players directly. Section 1.1 has been updated to reflect these changes.

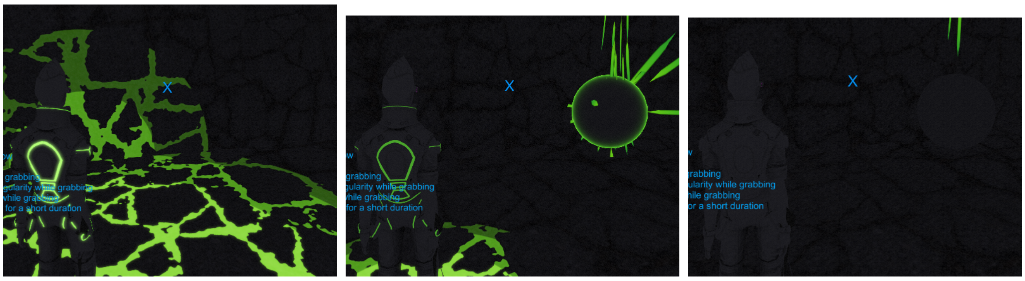


Figure 7: Stamina indicators for full, half and empty stamina levels.

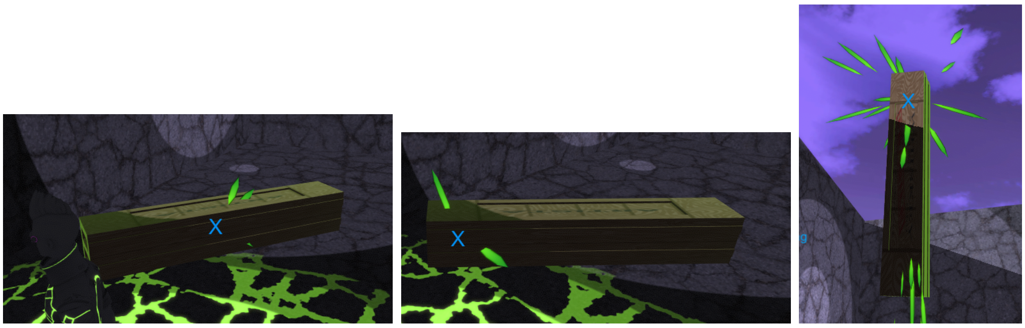


Figure 8: Indicators for controlling an object. Particles for the point and a Fresnel effect on the whole object in the player color.

4 Interim Report

So far we achieved the most progress in the implementation of the necessary game features while assets and level design are still missing. The game includes all of the necessary building blocks to make a game and the design team is currently working on an arena layout to build a first arena to test the gameplay. Art is working on character models and other assets.

At this point the game includes a simple menu and player lobby UI. Up to 4 player can join together. When all players are ready the game loads into a game scene which is currently a test scene where all the various features can be tested.

4.1 Player Controls

Good player controls are vital for our game and have proven a bit harder than expected. They are constantly being improved and currently work as follows: Directional Player movement is controlled with WASD. The player can also jump and has slight directional control while airborne. The character can walk on slopes up to a certain angle and can reliably stand on moving objects. The camera is controlled with the mouse.

A player's stamina level is indicated by a circular effect in the player's color on objects near the character. We also chose to add emissive elements to the character model itself to improve stamina visibility e.g. while airborne, see Fig.7.

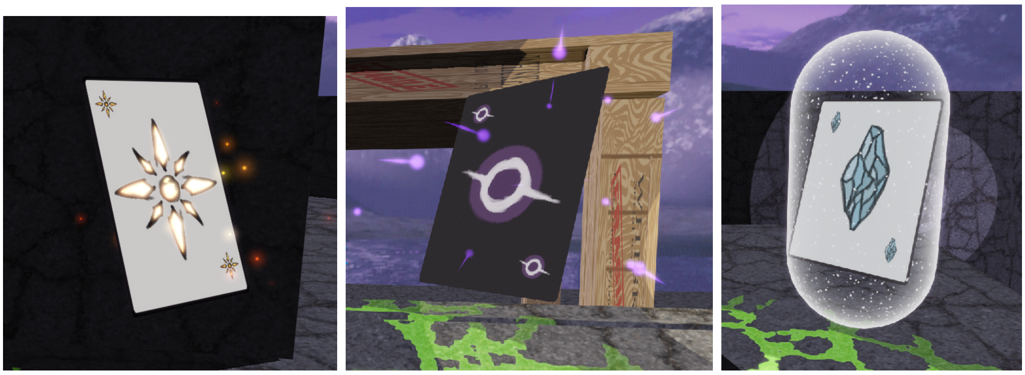


Figure 9: Pickups for the three abilities: Overcharge, Singularity and Freeze.

4.2 Telekinesis

The arena will be filled with different objects that can be controlled telekinetically. These objects have different properties like mass and friction. When the player looks at a free object that can be controlled the object will be highlighted. We also added the option to have multiple point on an object where it can be grabbed. The currently selected point will also be indicated with an effect, see Fig.8.

When the player clicks and holds [RMB] while a controllable object is indicated the player will grab the object. It is then continuously moved to the center of the screen by applying a force to the object, as described in Fig. 18. The distance to the screen can be controlled with the mouse wheel. While a player holds an object stamina is drained. Upon stamina depletion the object is dropped and stamina will start to regenerate after a short time. A player can throw a controlled object forward (from the camera) by clicking [LMB]. This will apply a certain speed to the object rather than a force to make the throwing action more consistent between object with different mass, the velocity diminishes with mass however, see Fig. 18 right side.

4.3 Abilities

To be able to use an ability the player has to pick up the corresponding pickup first (Fig.9). A pickup will grand a number of chargers to use an ability, on every use a charge is removed.

Singularity: The singularity ability (Fig.10 left) can be used when an object is actively controlled. Pressing [R] will then convert the object into a stationary gravitational singularity that will pull all objects towards it for a few seconds. The pull force depends on the objects mass. Optionally the singularity can be set to explode when it expires, pushing all objects away (see Overcharge, it is the same mechanic).

Overcharge: This is a straightforward ability, the player can make a controlled object explode by pressing [Q], pushing all other objects away (Fig.10 middle). The applied impulse currently only depends on the distance to the between the objects, not on the overcharged object's mass.

Freeze: Freeze comes in 2 variations. The ability (Fig.10 right) freezes all objects near the player when pressing [E] for a short time. The player can also freeze a single controlled object by clicking [MMB]. This single freeze action is always available and will cost stamina instead

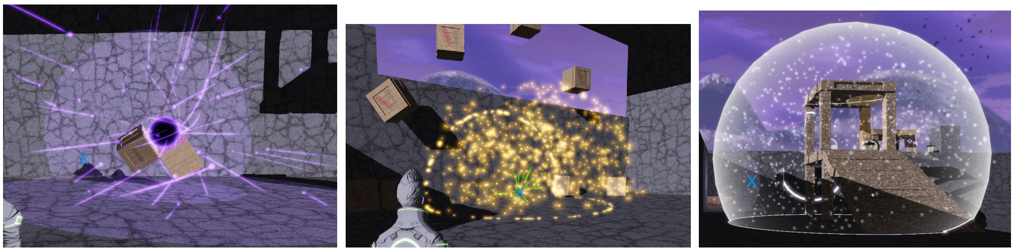


Figure 10: The three abilities in action: Overcharge, Singularity and Freeze.

of a pickup-charge. A frozen object will instantly stop moving and can not be controlled or otherwise moved until the freeze effect expires. Upon expiration the object will resume it previous movement. That an object is in the frozen state is also visually indicated.

4.4 Destructible Structures

We have implemented a simple system to realize destructible structures (Fig.11). Such structures consist of the same objects as the free ones that can be controlled by a player, but are flagged as ”static” and can’t be controlled in this state. They are organized in a hierarchy of objects supporting other objects. Each object has a list of other objects it supports and a number of how many supports it has itself. Furthermore there a 3 threshold that a determine when an object is set free, and therefore controllable by a player: direct and indirect destruction and impact impulse.

If the number of supports is below the direct threshold the object can be freed by hitting it with another free object with an collision impulse grater than the impact impulse threshold or an explosion. If the number of supports drops below the indirect threshold the object will automatically break free. When an object breaks free either way, it will notify all object it supports that they have lost a support, which can lead to cascading breakdown of a structure. ”static” object can be affected by the area freeze ability which will prevent them from breaking down for the duration of the effect, see Fig.10 right.

4.5 Miscellaneous

The biggest problems during implementation were network related, joining players together and keeping state. So far we have not been able to successfully return all players to the lobby to start another round. This will be the next main focus for the programming team. The synchronization of physics and object worked without problems. The system for destructible structures was surprisingly easy to implement.

We already implemented a number of shaders and SFX (see also: sec. 1.5.1 → graphics) for testing and showing what is possible. They are however merely placeholders and will change once the art team has made concepts and assets for them.

In the development schedule the technical aspects are, apart form the networking problems, well in layer 3. Design and Art are starting to come along, but are fallowing a different schedule.

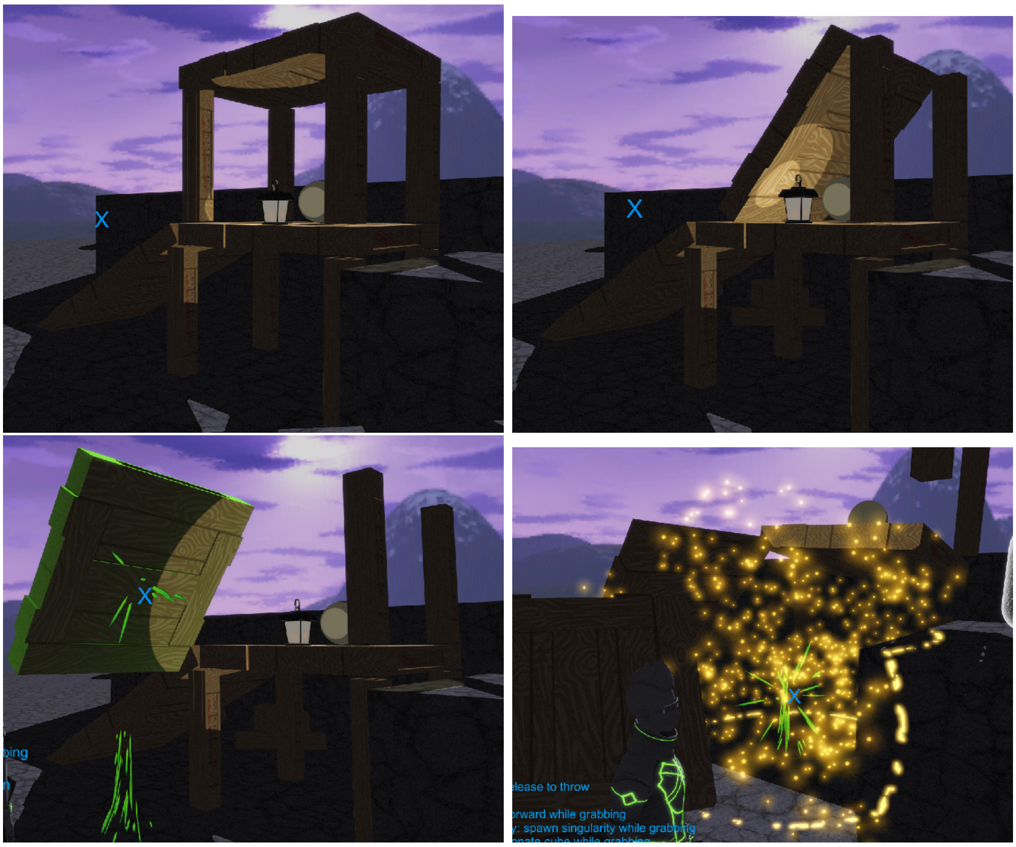


Figure 11: Destructible structure. Intact; Slightly damaged after objects were thrown against 2 of the "pillars"; The elements are controllable when set free; Interacting with an instance of overcharge that completely destroys the structure.

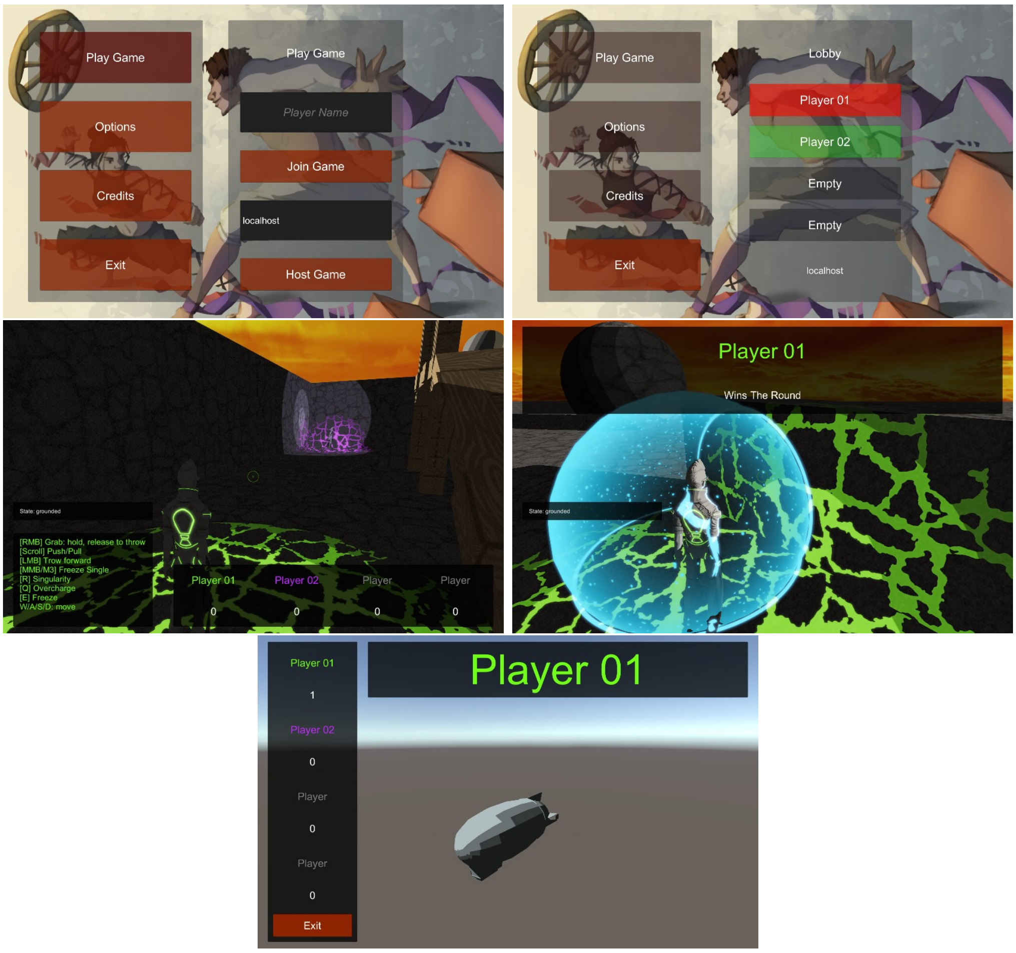


Figure 12: Game Loop screens. Play Game Menu, Lobby, UI during play, a player wins a round, Game End UI

5 Alpha Report

The interim version of the game already included all required (core) mechanics for the game, but was not as playable as such. It was missing the overall game loop: play rounds, win rounds, win game, restart. Since the interim milestone we mainly focused on adding these missing elements to make the game playable.

5.1 Overall Game Loop

Most of the work since the interim milestone went into making the game loop work, especially the networking aspects of it.

After starting the game the player is in the main menu, where they can choose to play, set a name, and then host or join a game. Up to 4 players can play together (1 host, 3 clients). In the multiplayer lobby players can set their ready state (and later choose their character and

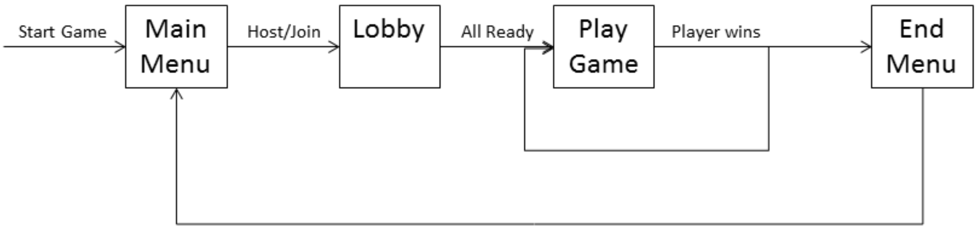
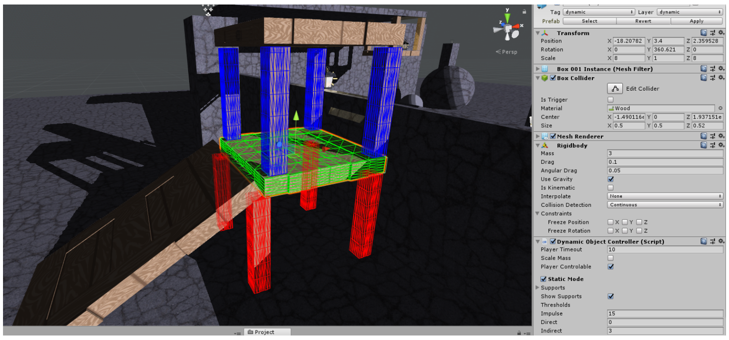


Figure 13: Game Loop



Tag: dynamic
Layer: dynamic
Prefab: Select | Revert | Apply

Transform
Position: X -18.20782 Y 3.4 Z 2.359528
Rotation: X 0 Y 360.621 Z 0
Scale: X 8 Y 1 Z 8

Box 001 Instance (Mesh Filter)

☑ Box Collider
Edit Collider
Is Trigger: □
Material: Wood
Center: X -1.490116e Y 0 Z 1.937151e
Size: X 0.5 Y 0.5 Z 0.52

☑ Mesh Renderer

Rigidbody
Mass: 3
Drag: 0.1
Angular Drag: 0.05
Use Gravity: ☑
Is Kinematic: □
Interpolate: None
Collision Detection: Continuous
Constraints
Freeze Position: □ X □ Y □ Z
Freeze Rotation: □ X □ Y □ Z

☑ Dynamic Object Controller (Script)
Player Timeout: 10
Scale Mass: □
Player Controlable: ☑
☑ Static Mode
Supports
Show Supports: ☑
Thresholds
Impulse: 15
Direct: 0
Indirect: 3

Figure 14: Custom Unity Editor

color). When all players are ready, the game starts. During play there is a new UI for help and player scores, which is shown while holding [TAB]. The players then play the game (as described in section 4) until one player reaches the goal (or the time is over). This player wins the round and is indicated in the UI. After a short time all players and the arena are reset and a new round starts. Once a player has won enough rounds, they win the game. The game then switches to the end scene, instead of starting a new round, which shows player scores and the winner (will later show a "reward" scene/cinematic). at this point the networking is shut down. The players can then individually return to the main menu and start a new game. Figures 12 and 13 show the UI and flow for the game loop.

Implementing this caused a lot of problems with some of unity's networking components. Often we had to create our own solutions or workarounds, which took some time. Main issues: player name and score sync, loading the correct scene and not destroying state and connections while doing so, gracefully destroying connection and players at the end while keeping the state (score, names, winner) on all parties for the end scene. The final return to the main menu is no problem since no networking is involved.

5.2 Other Additions

We also added a few smaller features.

A custom unity editor UI for the objects that can be controlled telekinetically by players and are also used to build destructible structures. The editor includes a tool for showing structural dependencies between objects (Fig. 14).

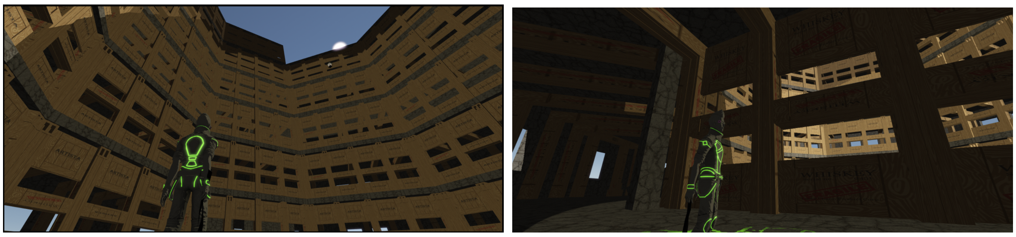


Figure 15: Arena

We added clipping for the player camera, such that it no longer clips through the environment as much.
We refined the ability management on players. Abilities can now be set exclusive (whether or not players can carry multiple abilities) and overriding (if exclusive, does a new pickup overrides already carried abilities?) and have a max capacity per ability (these are all global properties).
We added an experimental climbing mechanic that allows the player to climb up on edges of objects they could stand on. When a player stands close in front of higher ground the player is moved up, up to a certain height. This includes step up mechanic that allows the player to walk over small steps in the environment. The verticality of our level and gameplay might require this. For now it is a simple (experimental) implementation just for testing.

5.3 Design & Art

We have build a prototype arena to use for playtesting (Fig. 15). This arena is build from a few assets we already have and not representative for the final style. Regarding art, we have concepts for lighting and graphical style and the first character (Fig. 16) and assets for the environment are (being) modeled. The main shader is implemented. After we have a sufficient number of assets to build a authentic arena we can test lighting (according to lighting concept) and then tweak the shader accordingly.

5.4 Progress

This evaluation only considers tech progress for TUM which was up to now implemented by Erik Franz alone. A progress chart can be found in Fig. 17.

Layer 1 Done.

Layer 2 Done, only the round timer is missing.

Layer 3 Here, the major missing feature is Sounds. At this point the game doesn't contain any audio at all. Ragdoll and a Tutorial are also missing.

Layer 4 The largest tech feature, Destructible Environment, is implemented. Loading screens don't seem necessary at the moment.

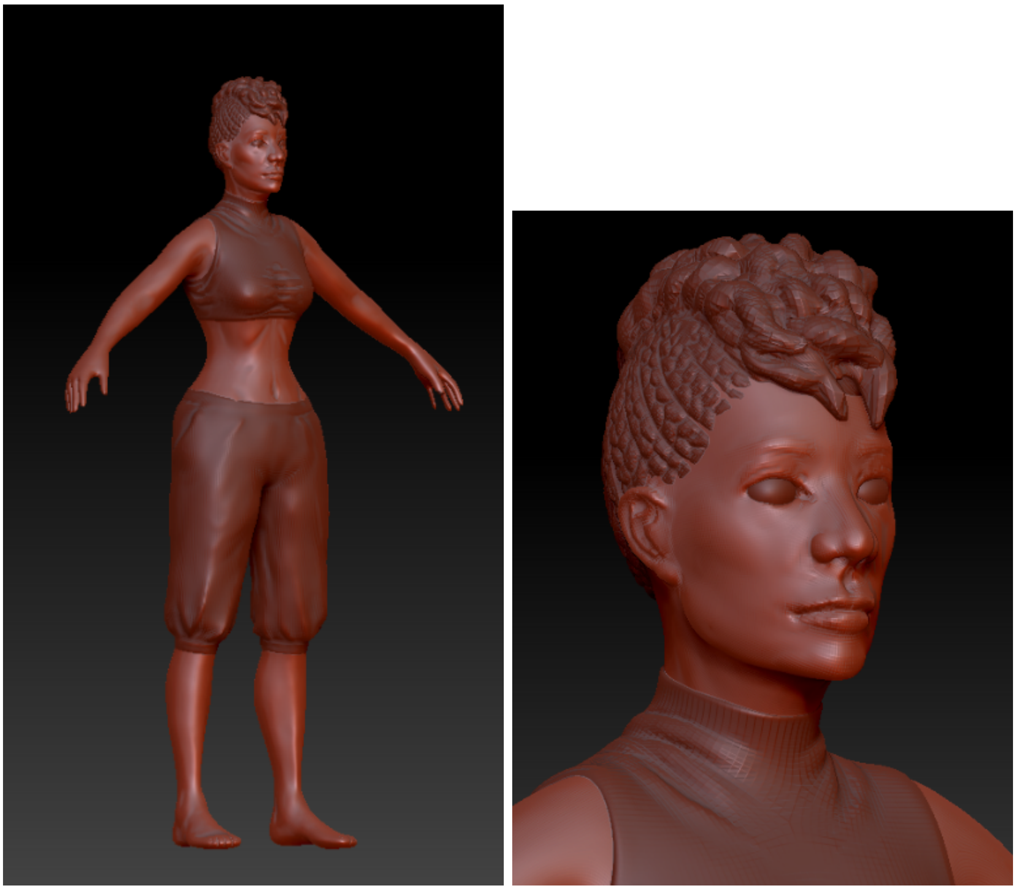


Figure 16: Character model for our first character ”Spade”.

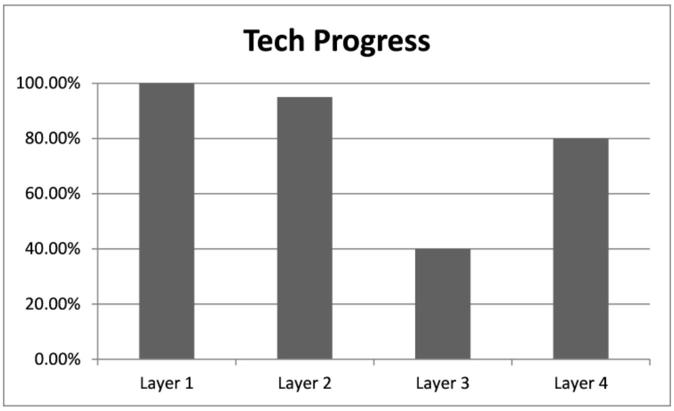


Figure 17: Tech progress chart

6 Playtesting

6.1 Organization

We held several separate playtests, one at TUM, 2 at MDH and one with friends. Players were given a short oral tutorial prior to the game, as we don't have a meaningful tutorial yet. At TUM we had some hardware issues, resulting in effectively one player playing alone. The individual playtests were organized as suggested. We observed their actions during the game and wrote down feedback and any issues we noticed. The play sessions were about 15-20 minutes long. Afterwards they answered the questions and we discussed their experience.
Our testers at both universities are all more seasoned gamers, 2 in each test. The friends are 3 casual gamers.

There is also an official playtest scheduled at the MDH for next Wednesday, where all project groups take part in each others tests.

6.2 Results

6.2.1 Questions

what do you think about the controls? could they be improved?
what do you think about the overall gameplay?
did anything feel odd/ out of place/ not fitting the game?
what do you think about the arena setup?
what do you think about the pace of the game?
did you feel that something was missing?
have you been bored at anytime during gameplay?
have you been confused/you didnt know what to do at any time during gameplay?
was the game difficult or easy?
did you have problems to achieve what you wanted? If so, what was the reason?
how easy or hard was it to hit enemies?
something you would add or remove?
how could the game be improved?
anything else you would like to mention?

6.2.2 Answers

The playtesting results were pretty consistent between the different sessions (TUM and MDH). Here is a sorted summary:

Movement

- Jumping feels off. It's too far/fast and poorly controllable.
- Walk - Run movement feels off, walk is not used/usefull.
- Stamina system was not evident end felt too limiting.
- Experimental climbing mechanic caused players to get stuck.

Telekinesis

- Generally, TK forces should be (or feel) stronger (for power fantasy).
- Heavy object feel too heavy.
- Directional throw when releasing objects is too strong and uncontrollable.
- Object selection mechanic is unclear.
- Missing indicators for carried/available abilities.

Arena

- It's too large and too empty, but also feels too tight at the same time.
- Obstacles are not recognized as such and some do not work as intended.
- Inconsistent visual distinction between fixed and destructible objects.
- Players got stuck in some spots.

Graphics

- The arena is too dark.
- Freeze effect to weak and badly visible.
- Overcharge and Singularity effects are liked.

Overall our testers enjoyed the game's core features, telekinesis and the destructible environment.

6.3 Changes

Based on this feedback we decided on the following changes. The order corresponds to the order above.

Movement

- Increased gravity for player to make jumping less floaty. Maybe increase in air control.
- Change from walk - run to jog - sprint movement with jog being the default.
- Re-balance stamina, set circle indicator to appear on all objects.
- Improved climbing mechanic by adding more checks for valid climbing situations.

Telekinesis

- Increase TK forces and control.
- Make TK more consistent for object with different weight (use same non-force velocity scaling as for throwing, Fig. 18)
- Remove the extra force when releasing objects.
- Make object selection cross-hair more visible.
- Indicators for abilities are planned, in the meantime add a simple UI.

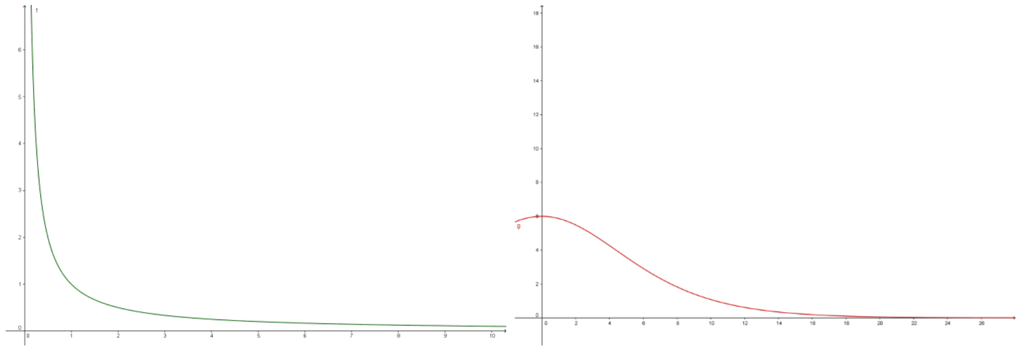


Figure 18: Qualitative comparison of velocity. x is object mass, y is velocity. Left uses a constant impulse I to determine the objects velocity: $v = I/m$. Right is our velocity scaling for throwing objects. It's the derivative of a logistic function: $4 * a * \frac{e^{-b*m}}{e^{-b*m*2}+2*e^{-b*m}+1}$ with a and b being configurable constants. This makes the throwing velocity more stable while still diminishing with increasing object mass. For moving the object while holding it we apply the force $\vec{F} = (\vec{d} - \alpha\vec{v})a\sqrt{m}$, with $\vec{d}$: distance from object to desired location, $\vec{v}$: objects current velocity, α: dampening factor, a: move speed factor, m: mass of object (sqrt to make heavier object feel heavy while making the falloff not too drastic) . (not in the graphs).

Arena

- Arena is reworked by the designers.

Graphics

- Made stone color brighter. Arena is set to be made from white stone later.
- Made the freeze effect emissive (from lit) and brighter.

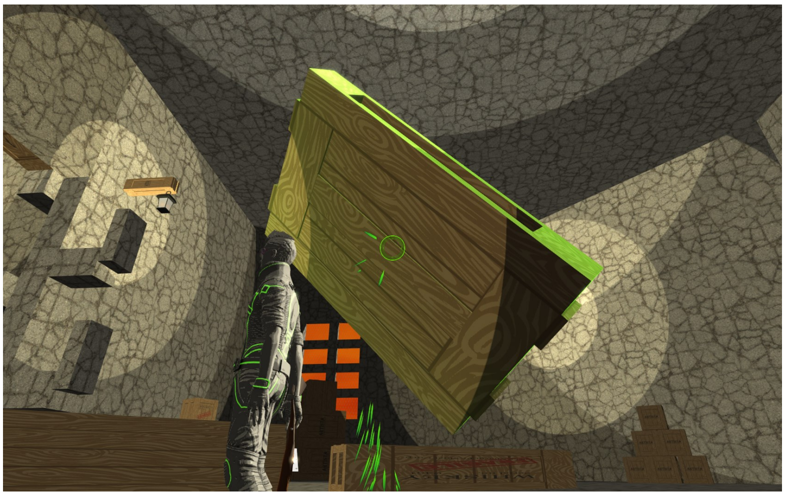


Figure 19: Telekinesis.

7 TUM Release

This is the final release for the TUM project, but we will continue to work on the game for the MD.H project.

7.1 Changes since Playtesting

All of the changes indicated in 6.3 plus some more options for how objects interact with abilities (can't be destroyed, can't be frozen, don't resume momentum after being frozen) have been implemented. We also added ability indicators right away instead of adding a placeholder UI. During the MD.H playtesting session we discovered some networking bugs, some of which have been fixed by now. In this state it is possible to play one round together, subsequent round may exhibit bugs.

7.2 Achievements

This section summarizes what has been integrated into the game so far. It focuses on the implementation and SFX by Erik for the sake of the TUM project requirements. Most of the required features were already implemented at the interim milestone (Sec. 4), so it will be referenced here. The second half of the project focused mostly on multiplayer, see Sec. 5. Arena design and building and some assets made by MD.H team members are also included in the game.

7.2.1 Character Controls

Character controls are described in Sec. 4.1. They have undergone some changes during development, especially the jump behavior. We also added a climbing mechanic that is still in development.

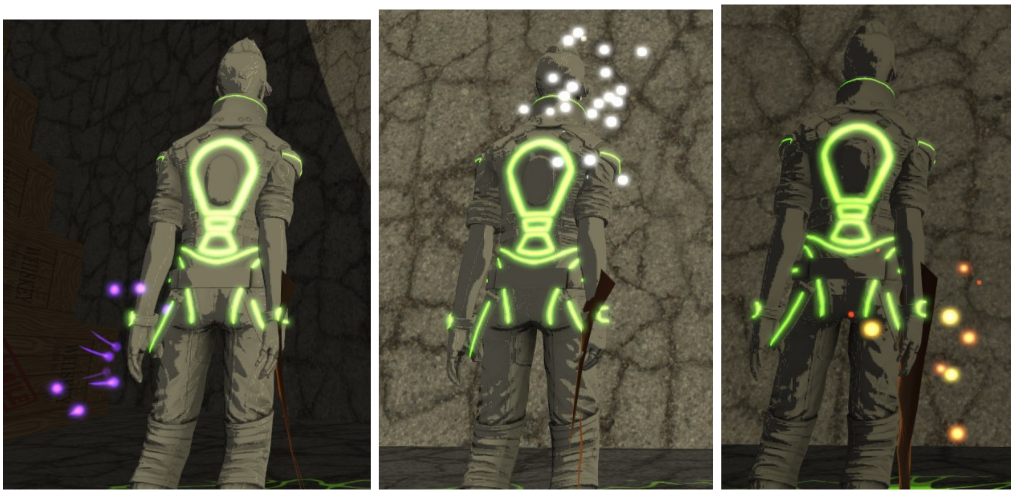


Figure 20: Ability indicators on the player model: singularity on the left hand, freeze on the shoulders and overcharge on the right hand.

7.2.2 Telekinesis

A player can grab, hold, move and throw all loose object in the arena, Fig. 19. The objects behavior is influenced by it's mass. Telekinesis is also described in Sec. 4.2. A Fresnel effect in the players color highlights objects that are selected for grabbing or currently grabbed. Particle effects indicates where an object would be or is grabbed as well as where the object is above the ground. These effects can also bee seen in Fig. 8. Object locked for other players while hold and holding an objects drains the players stamina, see Fig. 7.

7.2.3 Abilities

The game includes 3 abilities, as described in Sec. 4.3, with pickups (Fig. 9) and indicators for showing which abilities are available to a player (Fig. 20). The indicators might be replaced by the cards from the pickups attached to the players back. We also added the possibility to make abilities exclusive, allowing only one ability to be carried at a time. Exclusive abilities can be overriding, replacing already carried abilities, or not.

7.2.4 Destruction

Destructible environment was initially in the high target. After making changes to the design (Sec. 3) it became more essential to our game and we implemented it in the form of destructible structures as described in Sec. 4.4. Later we added a tool to visualize the structural dependencies between objects (see Fig. 14) to make building the arena easier.

7.2.5 Networked Multiplayer

The networking components were our main focus in later development and tie in all other systems/features of the game. Most of our implementation problems were directly related to Unity networking. Fig. 12 and Fig. 21.

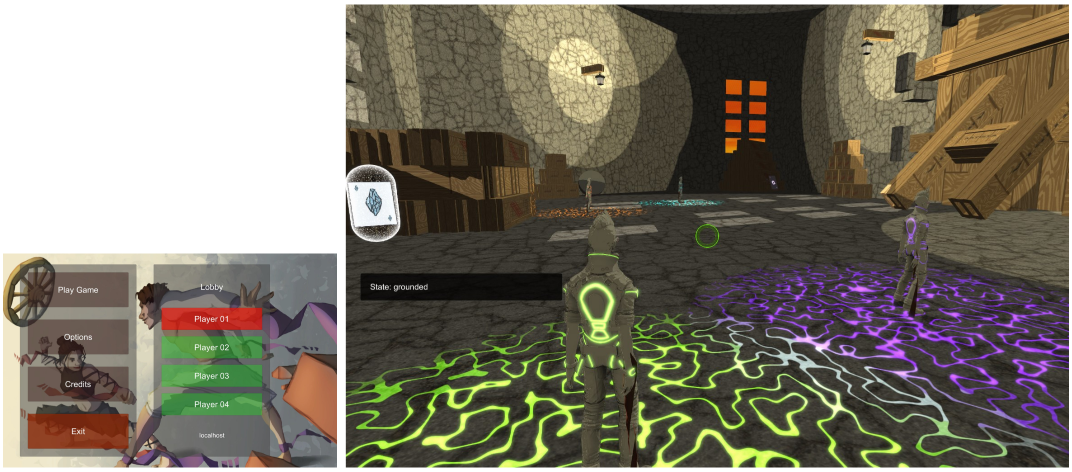


Figure 21: Multiplayer Lobby with 4 players and 4 player at the start of the game.

7.2.6 Shader and SFX

We implemented a custom shader for our desired Cel-Shading. This includes brightness steps for global and point lights as well as support for emissive, global and noise based effects (Fig. 22). We also added a number of special effects for telekinesis, the abilities and their pickups and indicators and when hitting an other player.

7.3 Evaluation

Overall i would say that the project was successful in terms of the TUM schedule, we achieved what we set out to: implementing all necessary tech features to have a playable game at the end of the TUM project phase. The only missing feature is sound. The hardest and most time consuming part was the implementation of the networked multiplayer, which was not our initial tech focus. But I learned a lot about how to implement multiplayer (in Unity). The remaining features, apart form good player controls, were surprisingly easy to implement, even the destructible structures.

7.4 Impressions

More time and larger scope would be nice to build something larger and polish and finish it as a game. Also I think that the end of the project falls into the busy end of the semester it not ideal. My idea would be to make the organization similar to the MD.H project. Have a longer concept and prototype phase during the semester, then a break during exam time and finally a dedicated project phase after the exams to focus on implementing the project.

7.4.1 Collaboration with the MD.H

What I liked about the collaboration with the MD.H was the larger and structured Team with defined roles (Design, Programming, Art, Leads for each role and a Producer). It was also interesting to see how the designers and artists work and their contribution was quite valuable as these roles are typically not present in TUM projects. We also got good feedback from the MD.H supervisors, especially the Game Design professor.

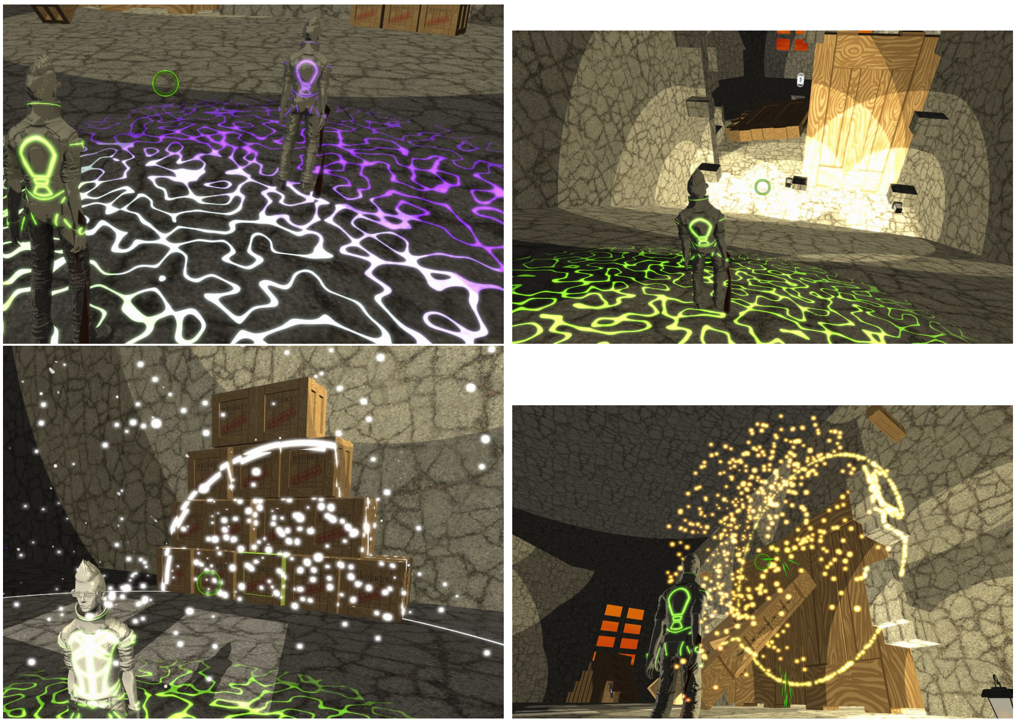


Figure 22: Some examples of the shader effects on objects. For more please also refer to the many other figures that show screenshots.

A downside is the increased organizational overhead. We had to manage 2 different schedules and make the others aware of deadlines of either project. It also meant more work for me as I had to make most the TUM deliverables and presentations myself.