



Game
Programming
Laboratory

Dust Off

"Ultimate Genie Cleaning Battle Palace"



Lucas Habersaat - Producer, Game Designer, Programmer
Jonas Passweg - Game Designer, Programmer
Vithurjan Visuvalingam - Game Designer, Programmer
Jonathan Ehrat - Game Designer, Programmer
Sascha Forster - Game Designer, Programmer

Formal Project Proposal

1.1 Game Description

1.1.1 Overview

Dust Off is a fast-paced spell-throwing arena fighting game for up to four players on the same couch. Each player takes over a genie and dives into a palace that was devastated by a sandstorm trying to clean up the most. There, they need to collect sand and use it to throw spells, which in turn disrupt the other players from collecting sand. Additionally, strategic positioning allows the genies to utilize better fighting spots and elemental objects to make sure they clean up more sand than the others. This free-for-all 2.5D top-down party game uses a very limited amount of spells and brings variety through combos with natural elements like fire or water. The simple yet intricate design allows players to easily enter the game while leaving some strategies to be discovered only after playing for a longer time. Finally, players are going to feel the sand that is lying all around, as this valuable resource also slows when walked over. The sand is cleaned up and dropped all the time making the fighting arena a changing and dynamic place.

Key Features:

Sand everywhere: While sand is not the only thing on the map, it is definitely the main and only resource and can be found nearly everywhere, scattered around the palace. It is the genie's source of magic, so, it is needed to use sand spells and the player having the most sand wins. As such, the players need to decide how to manage this precious resource and when to collect it. Additionally, being hit by a sand spell makes the player drop a portion of their sand and walking over heaps of it slows the player drastically. With this, the player will learn to love having a lot of sand while hating all sand laying around or thrown by other players, making them the perfect genie cleaners.

Element Interactions: To make the gameplay more interesting, the sand will interact with everything it touches. Most notably, elemental objects like water or fire are placed on the map. Spells that go through these elements are transformed into new spells, like sand becoming a mud ball when going through water or glass when going through fire.

Non-Stop Action: Between planning, collecting, throwing, and dodging, there is no time to take a break. The players need to plan their positioning and attacks carefully, but should not hesitate as being the first player to get an elemental buff for example, might give a big advantage. However, hitting spells is the main goal next to collecting the most sand possible. As such, each player should always have something to do, something to run from or after.

Arena: In addition to the sand, the arena engulfs the players in a palace set into the desert. It has a low-poly, cartoonish, soft, and simple feeling and the sand itself gives a touch of reality to the game. Additionally, the palace and its interior help to define a border around the fight and provide classic strategic aspects like cover.

1.1.2 Background Story

After a brutal storm in a desert south of the violet sea, the sharp-witted sheik had his grand, majestic palace devastated. His forgetfulness let the door open during the rude sandstorm. Luckily, this sheik is the owner of many genies. As many times before, his highness goes to his genie-chamber to fetch a few magical lamps, brings them to courtyard of his palace and summons them. And, as many times before, he promises to free the genie that cleans up the most sand. Happy to turn his misfortune into entertainment, he watches the genies fight in what can only be called an epic dust off!

1.1.3 Design Decisions

As our game "Dust Off" is a fast-paced battle arena game, the core mechanics focus on fighting, movement and strategic positioning. To keep the game simple during the mayhem, there is only one resource: sand.

Players are tasked with collecting as much sand as possible from the map. Since there are no weapons in the game, basic attacks consist of throwing sand at enemies.

These attacks cost a portion of the collected sand, but the prospect of hitting enemies, which drop sand when hit, makes the expense worth it. The sand attacks are close range to encourage risk-taking and have the additional effect of putting sand on the ground, slowing every player that walks through it.

An aspect that forces players to consider their positioning is elements that are distributed around the map. These elements transform and power up the basic attack to long-ranged and heavy-hitting projectiles with element-dependent extra effect.

The theme of sand plays a central role in the game. Sand covers the map and impacts the movement of the players. It is thrown at everyone during battle as attacks and the sand gains diversity through elemental transformations, for which it is the main ingredient.

Movement is needed to evade attacks and find a favorable position to strike opponents. As a central part of the gameplay, we want players to have as much control over their movement as possible and therefore provide them with omnidirectional movement. This is perfectly enabled by the joystick of controllers, which is why we first focus on controller support and only later focus on bringing the same feeling to keyboard and mouse.

Figure 1.1: *Combat in a devastated palace.*

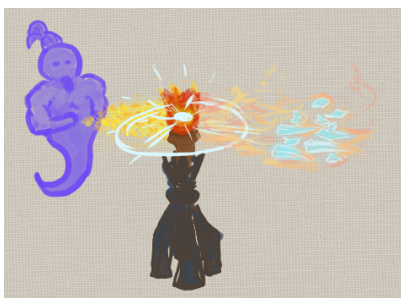


Figure 1.2: *Sand turning into glass splinters.*



Figure 1.3: *Sand turning into a mudball.*

1.2 "Big Idea" Bullseye

It's coarse and rough and irritating and it gets everywhere and you have to dust it off the palace! But it allows you to disrupt your friends, which is fun!

The core aspect of our local multiplayer game is resource management in terms of sand. The goal of a genie is to collect more sand than the other genies. This can be achieved by collecting sand that is lying around on the map and by sabotaging other genies by attacking them using sand spells. When such sand spells hit the opponent, genies lose some of their sand. The twist is that every time you use a sand spell, you also use some of your own sand. So the winner of the game will be the one who manages the balancing act between collecting and spending sand.

1.3 Technical Achievement

Make sand feel nice, authentic, and satisfying to look at and interact with.

As sand is our main resource and a lot revolves around it, we want to focus on that. Sand lying around the palace, spells, and elements shall look realistic, appealing and give the player an intuitive and satisfying feeling when interacting with it.

To achieve this we have to touch upon many aspects, like visual effects, sound effects, realistic feedback like slowing the character, and satisfying controls.

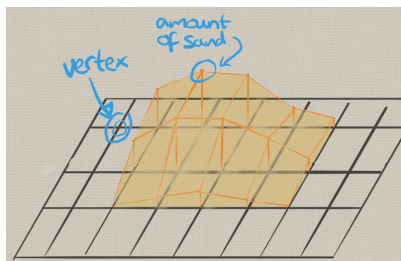
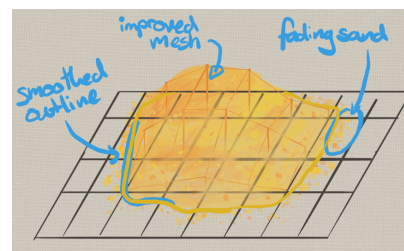
As a first step, we want to put our attention to the sand lying around the arena. We believe, that the **visual aspect** of sand will contribute the most to the nice feeling of sand. So, our actual technical achievement will be making the **sand heaps** in the arena **behave** realistically by using dynamic meshing or a vertex displacement shader on a planar mesh. Further, we want the sand to **look** appealing using some fragment shaders that capture the grainy and typical texture of sand.

In conclusion, our technical achievement will be making sand look, behave and ultimately feel nice, authentic and satisfying using shaders.

1.4 Development Schedule

1.4.1 Layered Task Breakdown

The five stages of the development of our game are described here. For each layer, the tasks are listed, categorized, numbered, and further described. For each item in the lists, a more detailed description of each task will be found as an Issue on GitLab. Of course, this is all subject to change as we iterate, gain more insight and adjust our plans, the game itself, and its features. New tasks might arise and existing ones changed or removed. Many mechanics and features are deliberately defined a bit loosely, as there are often multiple options. With time, prototyping and iteration we have to find out what works best.

Figure 1.4: Sand Rendering in the game Journey**Figure 1.5:** Sand mesh logic.**Figure 1.6:** Sand mesh after shaders.

Layer	Update Name	Content
1. Functional Minimum	Make It Work	Setup, Movement, One Spell & Element
2. Low Target	Make It Sandy	Add Sand as Resource and Fire
3. Desired Target	Make It Beautiful	Add Visuals, Sounds, and Balance It
4. High Target	Make It Shiny	Menu, Story, Advanced Visuals/Audio, More Spells

Functional Minimum or *Make it Work*

The absolute minimum consists of the **basic setup** of our game engine, the introduction of the fundamental **game objects** and **basic interactions** between those objects. The setup for the game objects consists of rendering the 3D models, defining a 2D hitbox, and handling the input for player characters.

1 Formal Project Proposal

Setup	Game Objects	Interactions
1.1 Mono Game Setup	1.4 Arena	1.9 Char/Arena Collision
1.2 Fixed Camera	1.5 Character	1.10 Char/Char Collision
1.3 Controller Input	1.6 Character Movement	1.11 Spell/Char Hit
	1.7 Spells	1.12 Spell/Water Modifier
	1.8 Water Element	1.13 Char/Water Modifier

Setup

With the given MonoGame template project, some things need to be set up. Our most simple game engine needs a fixed camera, to render models, and some base classes for our game objects. Further, we want to set up the controller interface and make it recognize connected controllers and detect input.

Game Objects / Input

The game consists of the above-listed game objects. The **arena** consists of a ground, a boundary and later obstacles.

There is a **character** for each connected controller that can be moved around with the analog stick. **Spells** are spawned by players and travel a certain distance through the arena. The first basic attack will be a wave of sand projectiles shot in a cone. Further, there is a modified attack, a mudball, which can spawn under the right conditions. Lastly, we have **water elements** in the arena in some form (e.g. fountains, puddles) that will interact with the player and spells (and create said mudball).

For all these game objects, we need very basic primitive 3D models and 2D hitboxes. The character reacts to player input for movement and spell casting.

Interactions

Nearly all game objects interact with each other on contact in some way. **Characters** collide with the arena boundary, obstacles, and other characters and cannot overlap or move into each other. **Spells** are spawned by characters and can hit other characters on collision, which has a behavior that will be implemented later (damage, sand heap drop, destruction of spells, slow enemy, ...). Lastly, the **water element** modifies the player or spells on collision. Characters can cast mudballs instead of sand projectiles for the next few attacks or for a certain time. Sand projectiles colliding with the water, go through and are transformed into mudballs.

Low Target or Make it Sandy

For this stage, we introduce the core mechanic next to elemental interactions: the **Sand** as a resource for each player. It represents health and spell cost (like mana in other games) of the character and collecting it is the main objective. Further, we add a new element, fire, interaction with which will turn sand into glass. Finally, more obstacles are added to the arena for cover.

Sand

- 2.1 Sand Resource for Character and UI
- 2.2 Add Sand Heaps
- 2.3 Collect Sand (new Action)
- 2.4 Spells Cost and Drop Sand
- 2.5 Vertex Displacement Shader
- 2.6 Sand Heap Simulation

Fire

- 2.7 Add Fire Element
- 2.8 Add Glass Spell
- 2.9 Char/Fire Interaction
- 2.10 Spell/Fire Interaction

Arena

- 2.11 Add Obstacles
- 2.12 Spell/Obstacle Interaction

Sand

Introducing the resource mechanic of sand, we touch up most aspects of the existing game. Each player now has their sand that represents their health and mana. Sand heaps lie around on the map randomly in the form of stacks that can be collected by the player with a separate action. Spells cost sand, however, when another character is hit, sand will be dropped from their pool that can be collected again. The sand heaps are implemented with dynamic meshing using some technique like a vertex displacement shader and to make the it move and spread like real sand, we add some basic simulation step that distributes sand from higher to lower altitudes based on the angle. A lot of balancing will be required to make the sand management work, motivate the players to fight, and keep the game fun.

Fire

Another element is added to the arena: fire, which turns sand into glass. Sand projectiles are turned into glass splinters, which have a different effect, like increased damage. The fire might be a fireplace or a torch in the arena. Some interactions are still missing, like mudballs with fire, which we will focus on in a later stage.

Arena

To make the arena more interesting, we add obstacles for cover which also further support the theme of a palace. The obstacles collide with characters and spells stop at obstacles.

Desired Target or Make it Beautiful

A lot of this stage is not about the actual gameplay, but elements around it. **Art** and visual effects will be introduced properly here: models, animations, and other visual effects as well as sound. **Balancing** is quite crucial to make the game stable, fair and keep players engaged. We want to try out a few things and iterate on some mechanics. Further, we add some **more spell interactions** and elemental combinations. Lastly, we implement the **game mode**. A game might consist of multiple rounds with some winning condition. What the game mode looks like

1 Formal Project Proposal

exactly, is something we also need to iterate on.

In general, a lot of our technical achievement will reside in this category, where the goal is to make the sand and spells beautiful and satisfying to interact with.

Visuals/Audio

- 3.1 Character Model & Animation
- 3.2 Spell Model & Animation
- 3.3 Arena/Obstacles/Element Models
- 3.4 Sand Effects/Shader
- 3.5 Basic Sound Effects

Spells

- 3.6 More Elemental Interactions
- 3.7 Spell Bounce
- 3.8 Dash Ability

Balancing

- 3.9 Balance Stats
- 3.10 Try Out Various Additional Mechanics

Mode

- 3.11 Multiple Rounds
- 3.12 Winning Condition

Visuals/Audio

So far, the game only had very basic placeholder models, such that we can focus on gameplay. This shall be improved. We craft game models as good as we can in a coherent low-poly and cartoonish style. Focus is on beauty in simplicity. For the character and spells, animations add a lot of authenticity. This can be further achieved with sounds and shaders.

Spells

The number of spells and their interactions is pretty low at this point, so, we want to add more interactions and combination options. For each pair of spells, there should be an effect, if they collide. We iterate on what happens when spells hit some border; one option is to introduce a bounce. And lastly, the characters lack mobility and with some dash ability the game will become more fast-paced.

Balancing

How much sand do spells cost? How much damage do they do? How much sand can be collected per interaction? Are there multipliers? Where exactly does sand drop? All those questions must be answered and the sweet spot found to maximize fun. Further, we reserve some time to try out some additional mechanics, like knockback, attack charging, or invincibility frames when hit.

Mode Until now, the game starts right away and goes on forever without any winning condition. We will change this here and add some form of conclusion to the game. How many rounds, the time limit, and the exact requirement to win, are things we must find out.

High Target or Make it Shiny

As our high target, we focus on polishing and making the game shiny. Of course, this late in development, it is hard to guess what we will be doing exactly. One thing we will be adding for certain is the **menu** system including a story trailer, and some configuration options for the mode. Further, we can improve **audio and visuals** and add **more spells or elements** or adjust existing ones.

Menu

- 4.1 UI System
- 4.2 Main/Play Menu
- 4.3 Configuration
- 4.4 In-Game Menu
- 4.5 Story Trailer

Better Audio/Visuals

- 4.6 Advanced Sound Effects
- 4.7 Background Music
- 4.8 Spell Shaders

More Spells

- 4.9 Defensive Sand Wall
- 4.10 Another Attack Spell
- 4.11 Ultimate Spell

Menu

Keeping it as simple as possible, there will be one main menu, where the players can join the game. Some options for configuration of the game, like the number of rounds, or the time limit gives players more freedom. Besides that, we add some form of story trailer before the game starts, to make clear what the goals are and to add a humorous touch.

Better Audio/Visual

Fitting background music and better sound and visual effects contribute to the ambience of the game and immerse the players more.

More Spells

With only three actions (attack, dash, collect), the players are somewhat limited in their actions and by adding more spells like a defensive sand wall or another attack spell, we open up more possibilities. One possibility could even let the players choose spells between rounds or upgrade them in some specific way. To add a feeling of progress, we thought of adding some sort of Ultimate Spell, that is unlocked when collecting a certain amount of sand, which will have a big impact on the gameplay.

Extras

Art

- Better Models, VFX, Animations
- Improve/Balance All Spells
- More Arenas

New Spells/Elements

- Wind Zone (Boost/Slow Down)
- Gunpowder & Explosions
- Sandstorm Ultimate

New Mechanics

- Stunned When Blinded
- Spell Upgrades Between Rounds
- Non-Combat Player Interaction (Contracts, Healing, ...)
- Stun

Others

- Button Remapping
- Keyboard Support
- Different Game Modes
- Easter Eggs

1.4.2 Task List

We present a list of the tasks directly related to the creation of our game, who they are assigned to and the estimated time required to complete them. It is shown in Table 1.1.

1.4.3 Timeline

Figure 1.7 shows the milestones for writing further chapters in this report, as well as our current estimated timeline for the different categories of tasks within each layer.

1.5 Assessment

The game is aimed at groups of friends who want to get together and measure themselves against each other in an action-packed brawl. They will immerse themselves into an antique, magic world inspired by the 'Arabian Nights', where they take the role of genies to help the sheik clean the sand flooded palace while sabotaging each other.

The fighting happens around the singular resource sand. While this makes clear what players need to pay attention to, it also demands stellar resource management from them. Sand can be combined with elements scattered around the map to unleash more potent attacks. If done right, a variety of spells will fly all over the p(a)lace, hitting enemies with a powerful oomph, knocking them down and allowing you to collect their sand.

Overall, we aim to create a beautiful game with some simple mechanics, whose interactions provide enough complexity to be challenging. Attacking and dodging incoming projectiles should be satisfying and be accompanied by according visual and sound effects.

Table 1.1: Tasklist showing tasks, assignees and estimated time required in hours

Task	Assigned To	Time	Task	Assigned To	Time
Functional Minimum			Low Target		
Setup			Sand		
Mono Game Setup	All	2	Sand Resource (Char, UI)	Jonathan	8
Fixed Camera	Sascha	6	Sand Heaps	Lucas	10
Controller Input	Lucas	6	Sand Collection	Vithurjan	6
Game Objects			Spell Cost	Sascha	4
Arena	Sascha	4	Sand Drop	Jonathan	8
Character	Lucas	4	Fire		
Character Movement	Jonas, Lucas	4	Fire Element	Vithurjan	4
Spells	Jonas, Vithurjan	10	Glass Spell	Lucas	6
Water Element	Jonathan	4	Char/Fire Modifier	Jonathan	4
Interactions			Spell/Fire Modifier	Jonas	6
Char/Arena Collision	Vithurjan	6	Arena		
Char/Char Collision	Lucas, Vithurjan	6	Obstacles	Sascha	6
Spell/Char Hit	Jonathan, Vithurjan	8	Spell/Obstacle Interaction	Sascha	6
Spell/Water Modifier	Jonas	6	Vertex Displacement Shader	Lucas, Sascha, Vithurjan	20
Char/Water Modifier	Jonathan	4	Sand Heap Simulation	Lucas, Jonathan	16
Desired Target			High Target		
Visuals/Audio			Menu		
Char Model/Animation	Lucas, Vithurjan	16	UI System	Sascha	18
Spell Models/Animation	Jonathan	14	Main/Play Menu	Jonas	16
Arena/Obstacles/Elements Models	Jonathan	12	Configuration	Jonathan, Sascha	14
Sand Effects/Shader	Lucas, Jonathan, Sascha	20	In-Game Menu	Vithurjan	10
Basic Sound Effects	Jonathan	8	Story Trailer	Lucas	30
Spells			Visuals/Audio		
More Element Interactions	Jonas	8	Advanced Sound Effects	Vithurjan	20
Spell Bounce	Sascha	6	Background Music	Jonathan	8
Dash Ability	Jonathan, Lucas	8	Spell Shaders	Lucas	20
Balancing			More Spells		
Balance Stats	Jonathan, Vithurjan	14	Defensive Sand Wall	Sascha	14
Try Different Mechanics	Sascha	6	Another Attack Spell	Lucas	10
Mode			Ultimate Spell	Lucas	12
Multiple Rounds	Jonas	8			
Winning Condition	Lucas	6			

Figure 1.7: Timeline: Showing report milestones and categories of tasks for each layer

