

DDJD Team B - Game Design

Index	1
Story (Story Bible)	2
Premise (high concept)	2
Game Interface	3
Character Abilities and Items	7
Active element	7
Melee attacks	7
Spells	7
Default	8
Melee Attacks	8
Default dash	8
Jump	9
Fire	9
Melee Attacks	9
Dash	10
Air Movement Skill - Hover Effect	10
Fire shot	11
Fireball	11
FlameThrower	12
Earth	12
Melee Attacks	12
Dash (Jump with rock):	13
Air Movement Skill	13
Defense Wall.	13
Earth shot	14
Big Rock Shot	14
Wind	14
Melee Attacks	15
Dash	15
Air Movement Skill - Air Walk	16
Tornado	16
Push Attack	17
Electricity	17
Melee Attacks	17
Lightning Dash	18
Air Movement Skill - Hover effect	18
Electric Walls	19
Electric Ray	19
Electric Bolt	20
Items	20
Game World Description & Design	21

Earth Village	21
Wind Village	22
Electricity Village	23
Fire Village	24
GLD - Game Level Design	25
Missions	25
Technical Design Document	27
Concept Art (Art Style Guide)	32
Project Plan	34
Resource Plan, Budget, Schedule, and Milestones	36
Resource Plan and Budget	36
Schedule and Milestones	36
Test Plan	37

Story (Story Bible)

Premise (high concept)

The game takes place in a small open world that features several coexisting villages, each powered by a unique gem representing an element. These gems not only shape the villages' environments but also grant them special skills and access to valuable resources. The very existence of the villages relies on these gems, as they provide the necessary energy for the local wildlife and sustenance.

Our main character (MC) begins as a regular resident of the earth village. On a tragic day, their village is mercilessly attacked by an evil organization seeking to control all the elements. Despite the valiant efforts of the MC's mother, a powerful wizard, to repel the raid, she tragically loses her life in the process. With a heavy heart and a thirst for vengeance, the MC is determined to avenge their mother's death, protect their village.

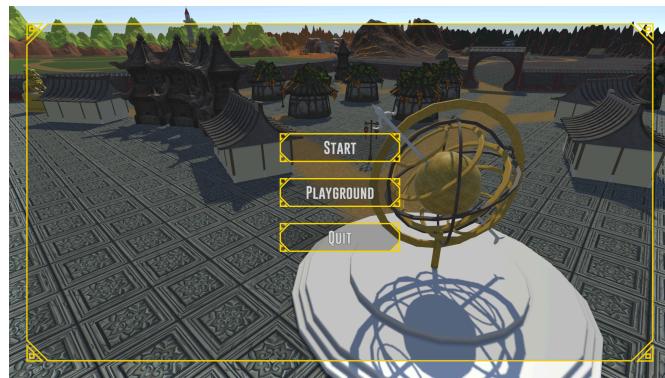
The player's primary objective is to avenge the death of their mother by dismantling the responsible organization. This involves infiltrating other villages that have been conquered by the organization and eliminating the captain stationed there. As the player liberates these villages, they gain the trust and support of the villagers, unlocking new skills. This is achieved by touching the gem of each liberated village, learning from the locals, and acquiring items and abilities from vendors.

Alternatively, the player can choose a darker path by stealing the gems for themselves, empowering their existing skills with the stolen element. This path, however, comes with its own set of challenges and consequences as the player taps into forbidden powers for personal gain.

Throughout the MC's journey, they will face various trials, encounter intriguing characters, and explore diverse environments. The choices made along the way will shape their destiny, determine the fate of the villages, and unveil the true nature of their character. With courage, determination, and strategic decisions, the MC must bring justice to their mother's memory and safeguard the earth village, ensuring its survival in the face of imminent danger.

Game Interface

Main Menu



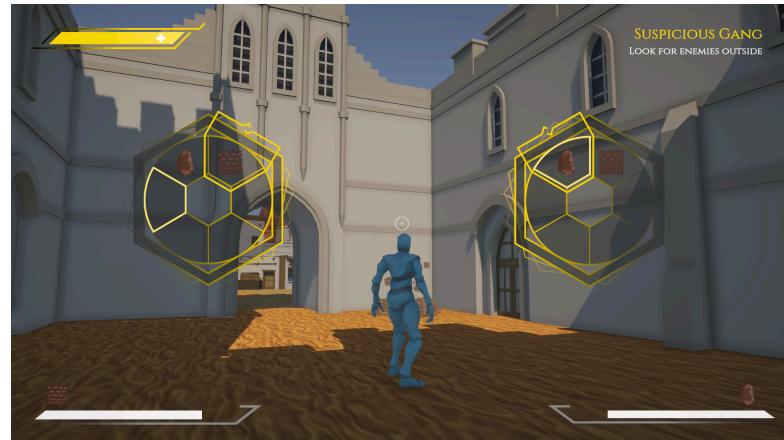
When the player opens the game, they are presented with a screenshot of the game along with three options. If they click *Start* the game will simply start, if they click *Playground* they will be presented with an alternative version of the game in which all spells are unlocked and it's mainly for testing and fun purposes, and if they click *Quit* the game will be closed.

Playing Interface

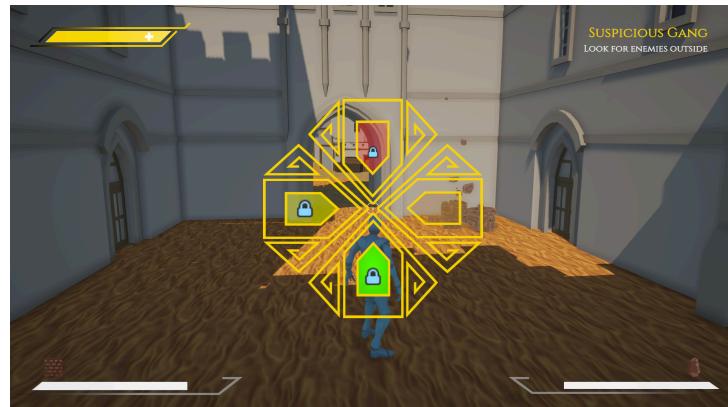


Once you enter the game you will be presented with the playing interface, which by default contains the health bar on the top left corner, that shows how many health points you have left, and two mana bars at the bottom corners, showing how much mana you have from the elements of your selected spells.

By clicking *Q* or *E*, you will be presented with the spell wheel from the left or right hand respectively. Here you can select the spell you want to use and see what spell is currently active.



By clicking *Left Alt* you will be able to change your active element, which will be the element of your dash and melee attacks. The currently active element is the one that is not faded. Some elements will be locked, indicated by the blue locks.



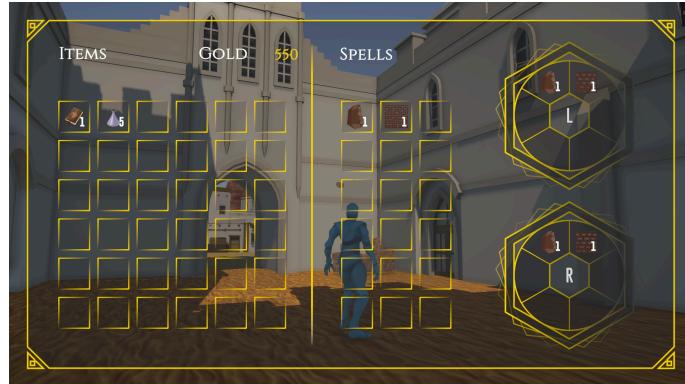
If you use your dash, it will have an associated cooldown, and that is evident by the spinner that appears on the bottom of the screen.



Inventory

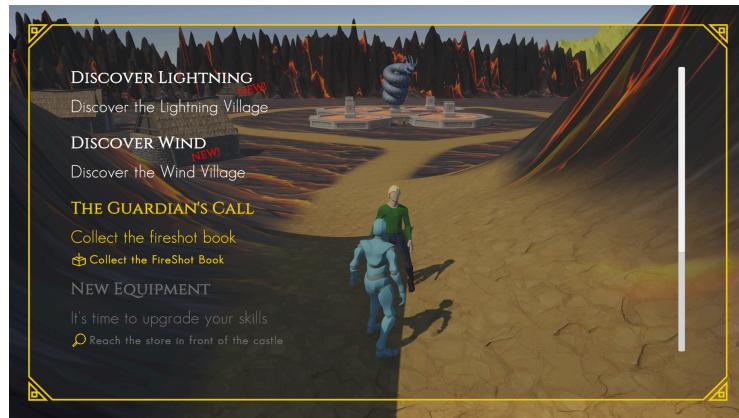
By clicking *I*, the inventory will open. Here you can see how much gold you have and what items you have. Items are divided into general items and spells. Items are quest items, potions or books, the two latest being consumables, which you can

consume by clicking on them with the right mouse button. Spells can be placed on the spell wheels on the right side, making them available on the corresponding spell wheels in the playing interface.



Missions Menu

By clicking M, you can open your missions menu, where you can see the missions you have available (marked with *New*), ongoing (in white) and completed (in gray) and see what goals compose each mission. By clicking any of the missions, its title and first incomplete goal will be shown in the playing interface.



Interactions with NPCs

When you approach a Non Playable Character, if they have a yellow diamond on top of their heads, the hint "Press F to interact" will appear.



If you do so, a dialog will appear, which you can go through by clicking *Enter* or leave by clicking *Escape*.



Tutorials

Some missions will have the aim to introduce the player to some game concepts or mechanics. In such a case after an interaction, a tutorial interface will appear describing to the player how that new concept or mechanic works, with an illustrative image. The player can cycle to the pages of the tutorial with *Enter* or leave with *Escape*.



Map

The map can be accessed by pressing the TAB key. It can provide the player with helpful information about next missions or goals.

When a npc has a mission, a yellow square appears on the map and a red circle appears on the map indicating that a goal is in the area.



Character Abilities and Items

All character abilities are inspired by the respective elements, including spells, dashes or special jumps.

Active element

The player can have one active element at a time, the elements can be unlocked by consuming the respective element stone, Fire Stone, Wind Stone or Electricity Stone.

Each element has a different dash and an air movement skill. The active element also changes the VFX of the player's melee. In a future version, each melee has different effects. If the element dash is on cooldown the player can also dash, but it will do a default dash with no special effects.

Melee attacks

The player can melee attack enemies while not aiming. While it's not possible to move while attacking, at the start of each attack, the player will push himself towards the movement input, in order to adjust the position while attacking. It's also possible to dash while attacking, which increases the damage and knockback.

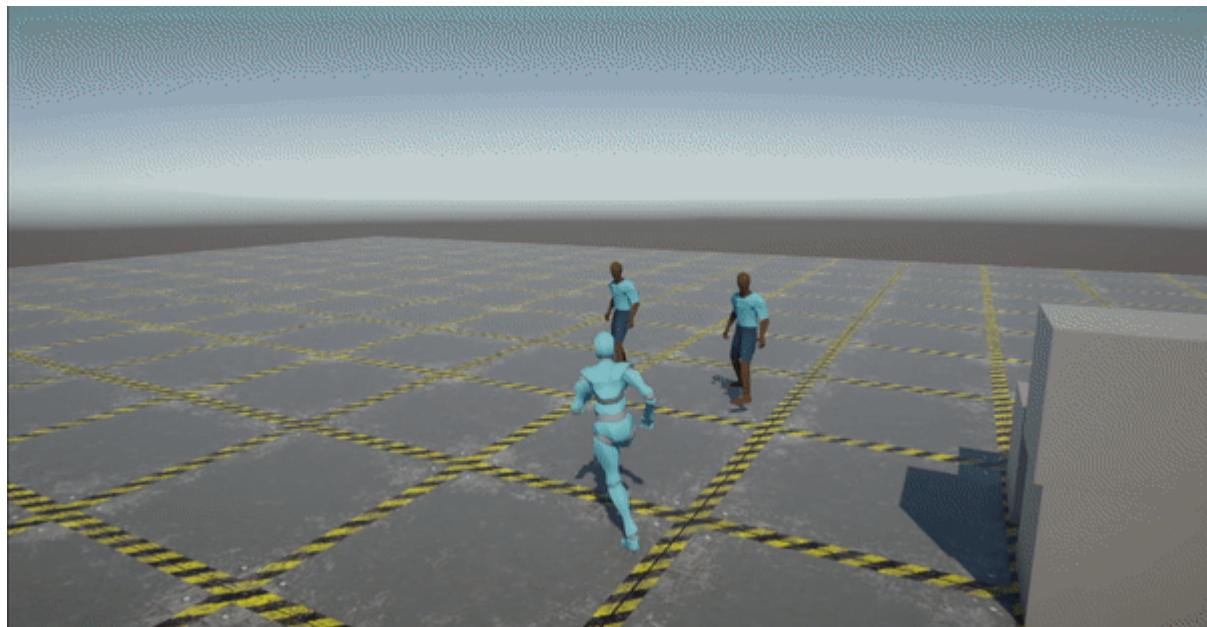
Spells

The player can cast spells (previously acquired) by switching to **aim mode**. It's possible to have two active spells and use them simultaneously, and the spell doesn't have to be the same element as the active element. The player needs to assign a spell for the left or right hand, in order to cast it

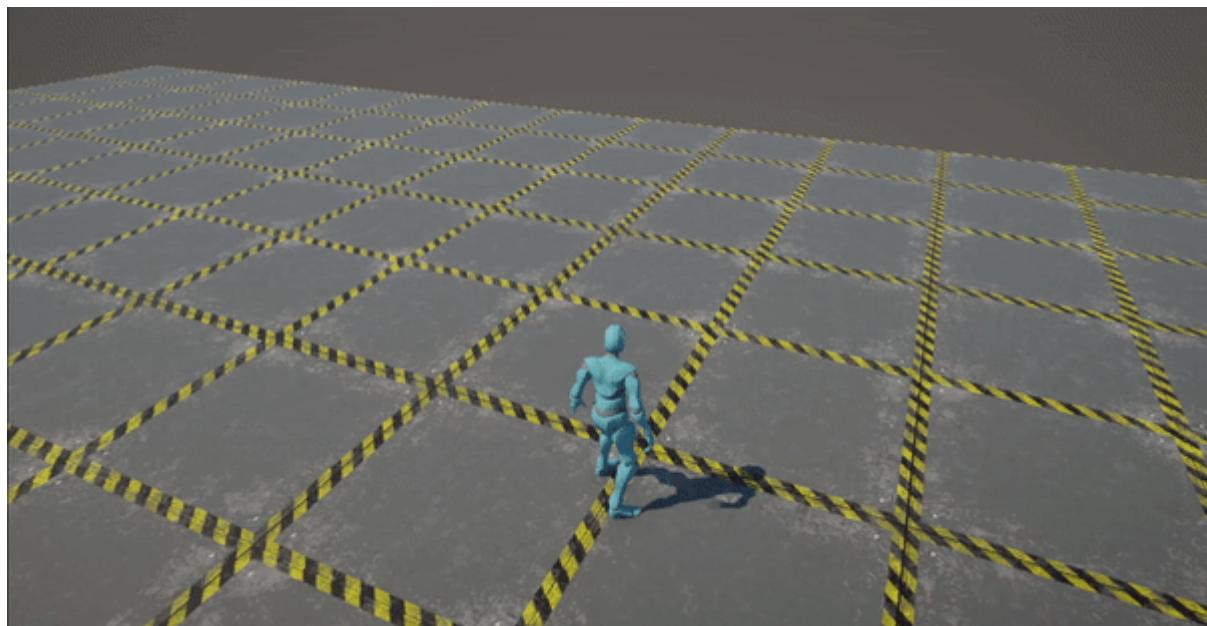
Default

These are the abilities the player can use without using any kind of element.

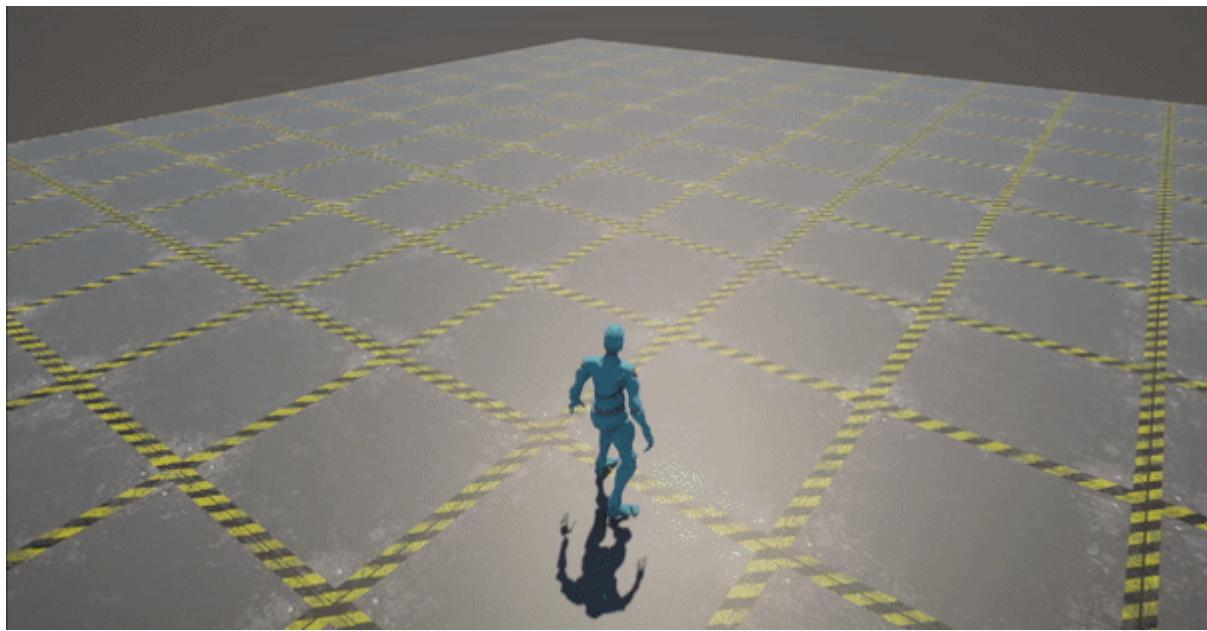
Melee Attacks: for close range melee fights.



Default dash: used to quickly dodge enemy attacks and to improve the overall movement of the player.



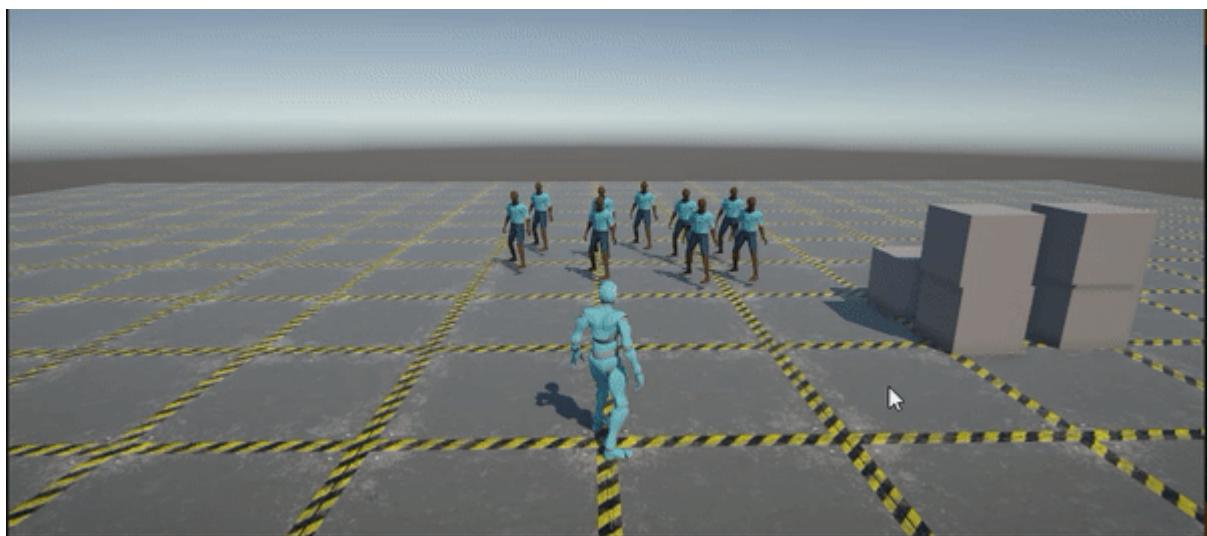
Jump



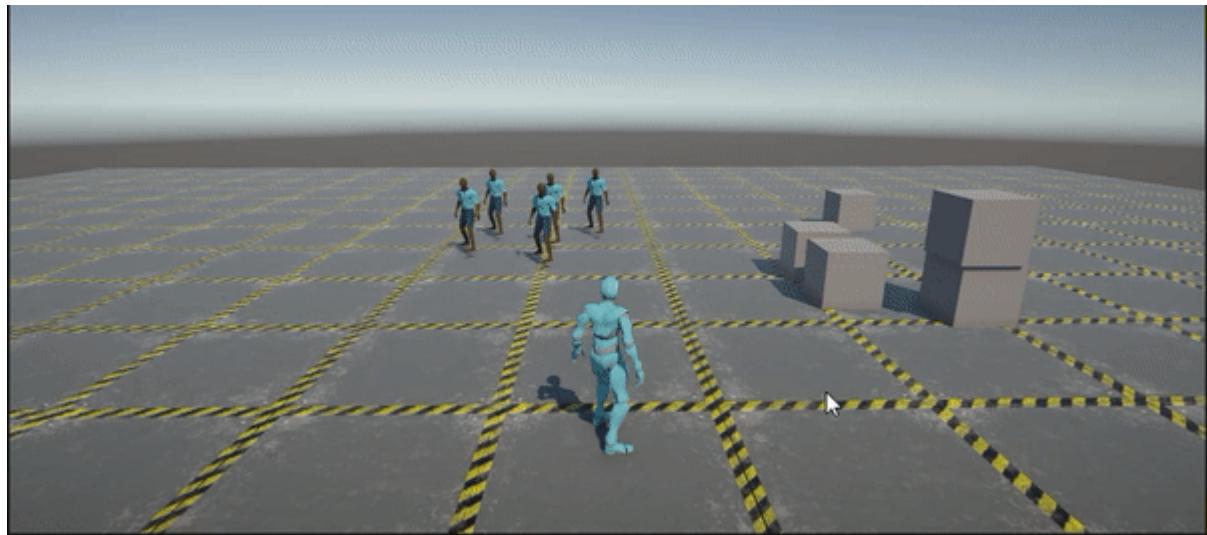
Fire

The spells associated with the fire element focus on dealing direct damage to enemies in the form of direct attacks or area of effect.

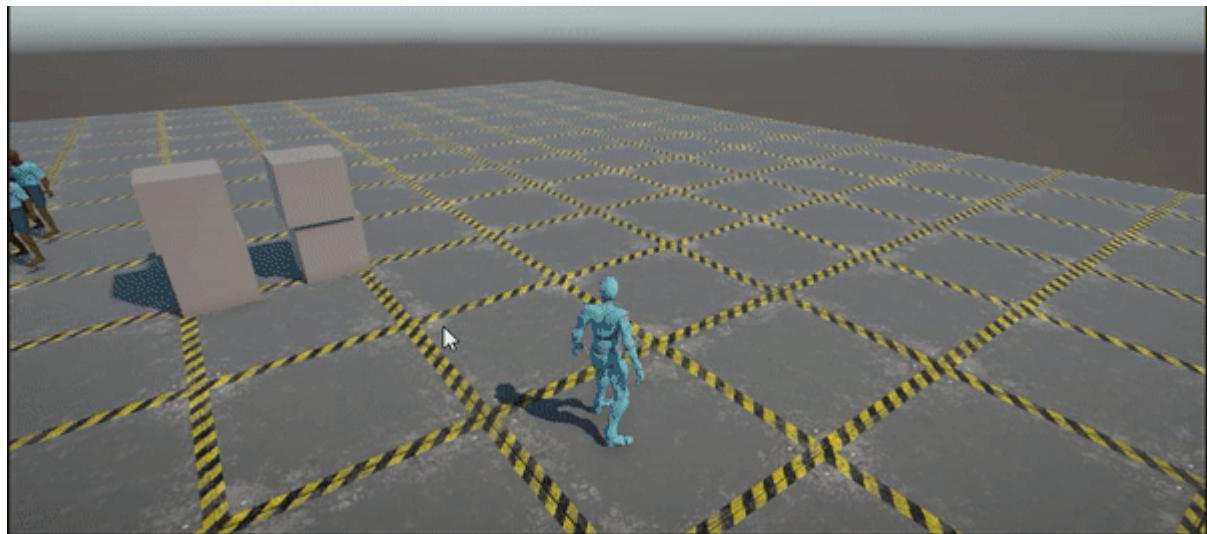
Melee Attacks



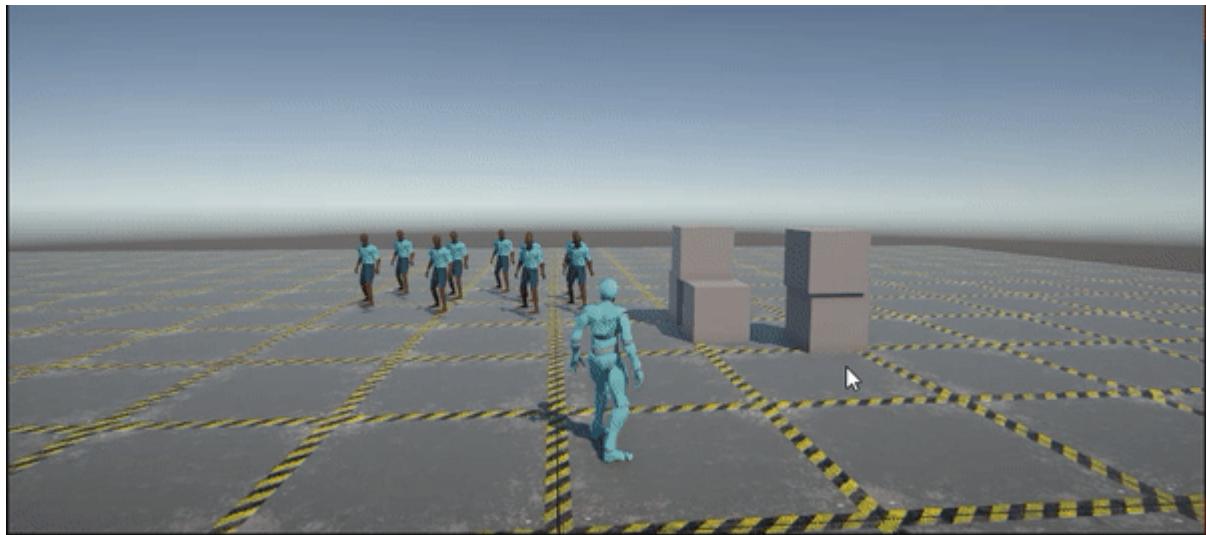
Dash: creates an explosion that damages nearby enemies. Has a considerable cooldown.



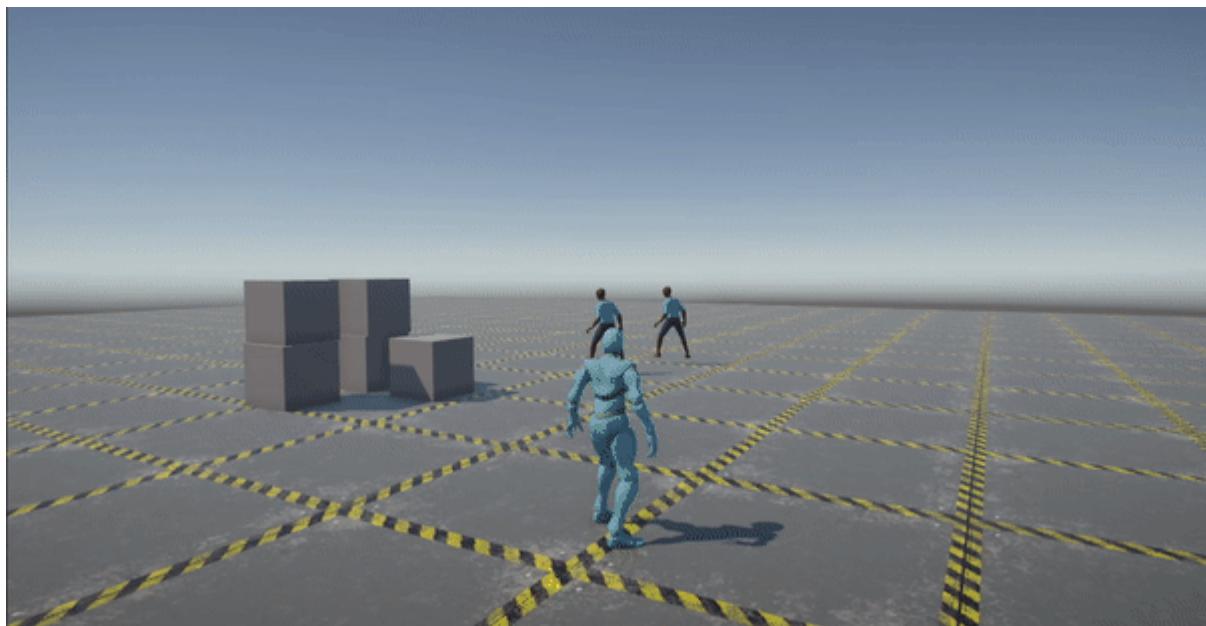
Air Movement Skill - Hover Effect: allows the player to glide midair. This is similar to the electricity jump variation, with the exception that this one allows the player to move forwards more easily than upwards.



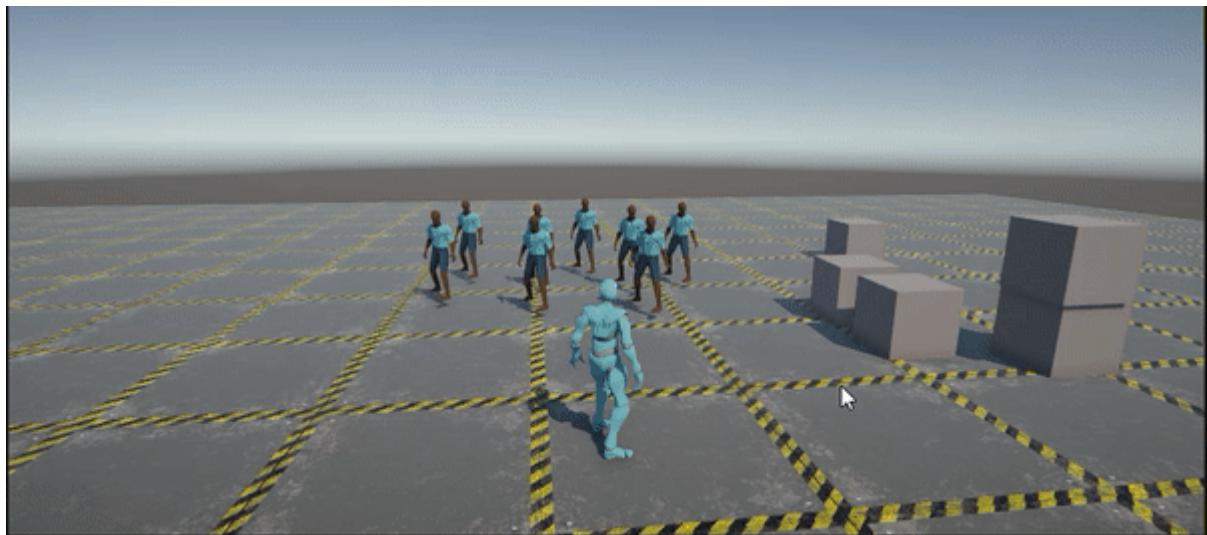
Fire shot: small fireball with low cooldown that deals low damage and requires little mana.



Fireball: charged fireball with high cooldown and capable of dealing high damage in a radius. Requires considerable mana.



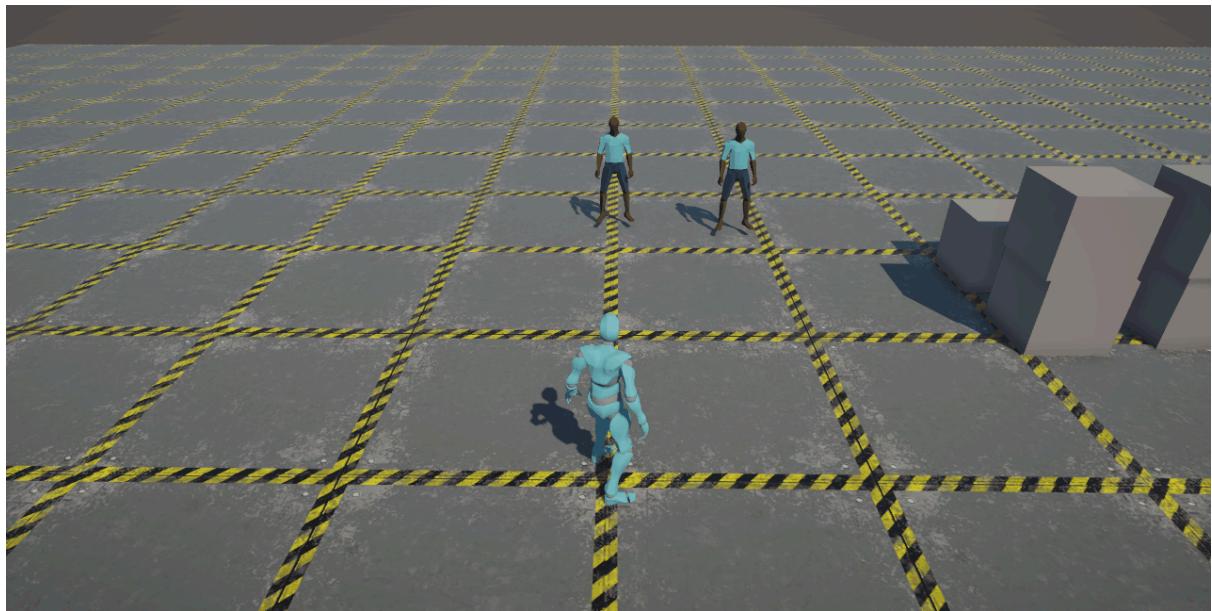
FlameThrower: continuous fire that deals damage and spends mana over time.



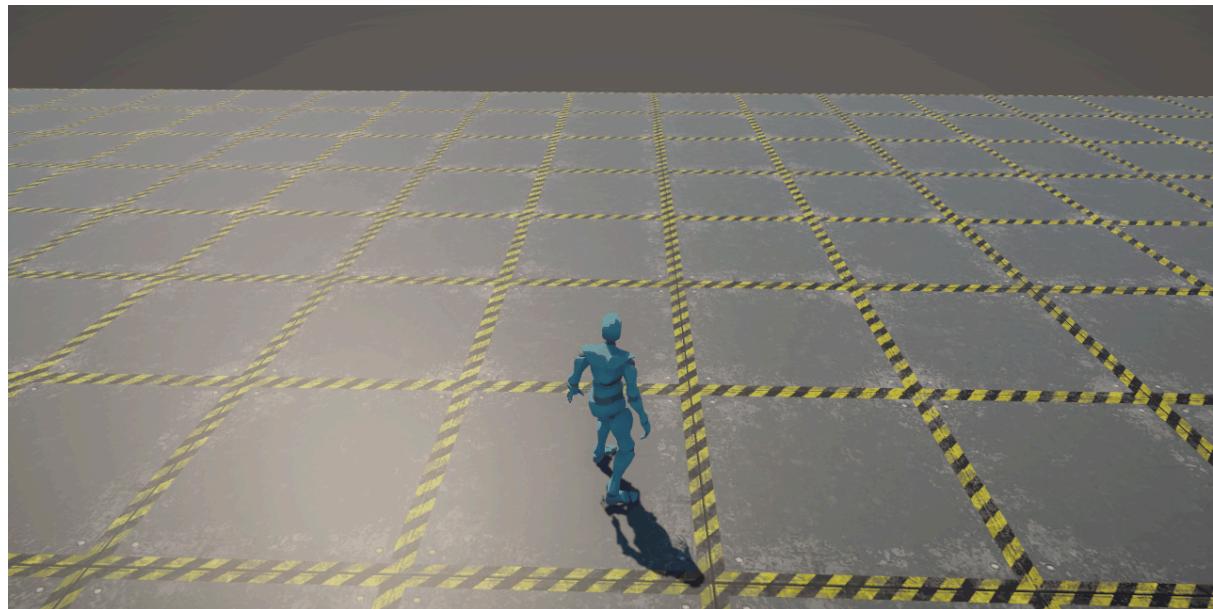
Earth

The spells associated with the earth element have a balance between damage and support abilities, including unique defensive options.

Melee Attacks

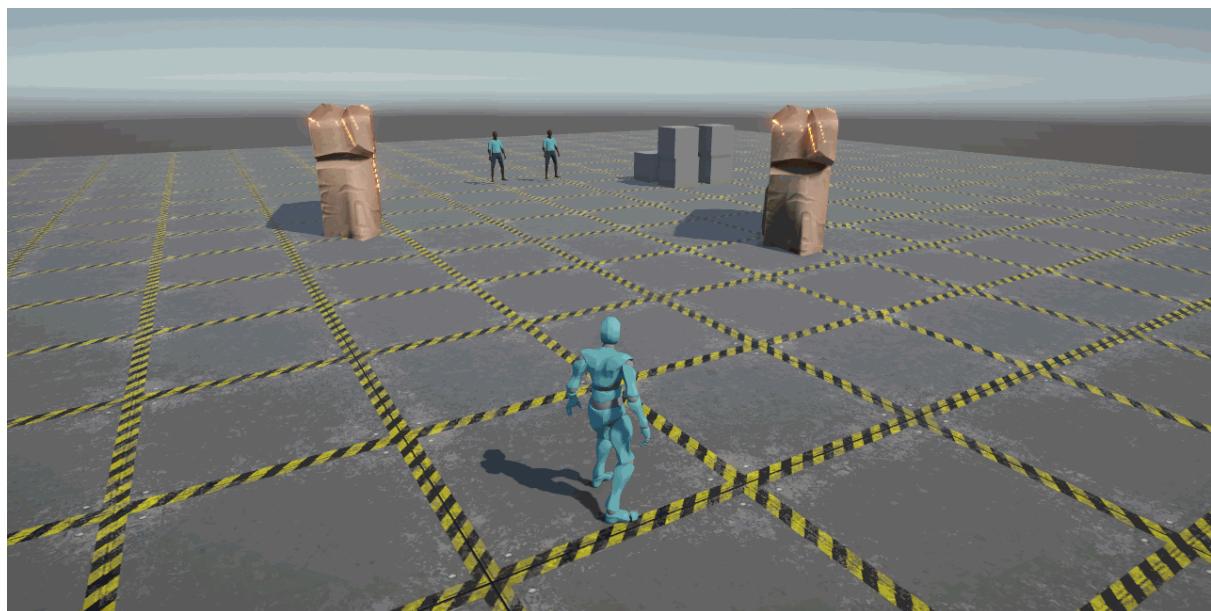


Dash (Jump with rock): lets the player jump higher than usual by spawning a large boulder underneath him. Has a considerable cooldown.

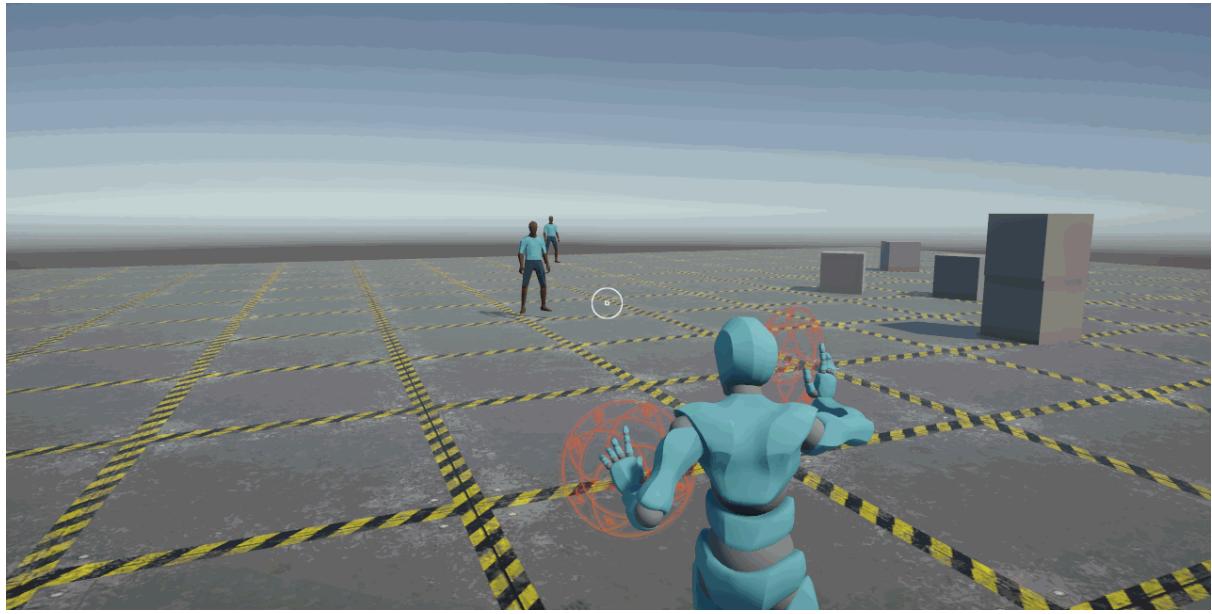


Air Movement Skill: has no air movement skill

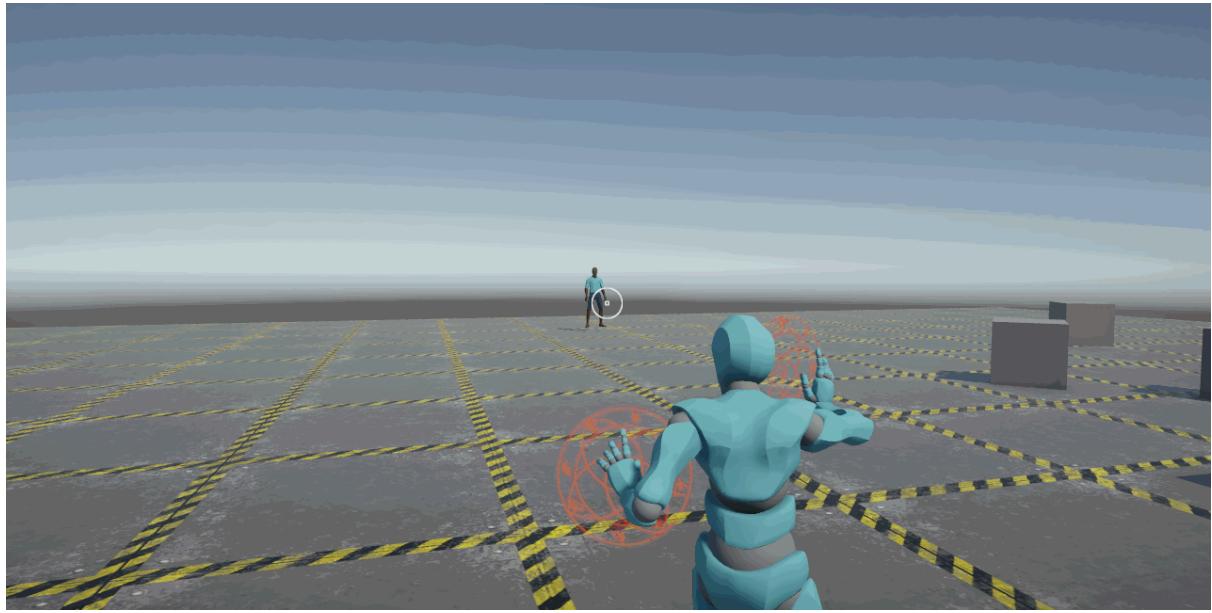
Defense Wall: spawns a wall to block attacks from enemies and offer cover to the player. Also damages and knock enemies that collide with it.



Earth shot: throws a small rock that deals low damage. Has low cooldown and spends little mana.



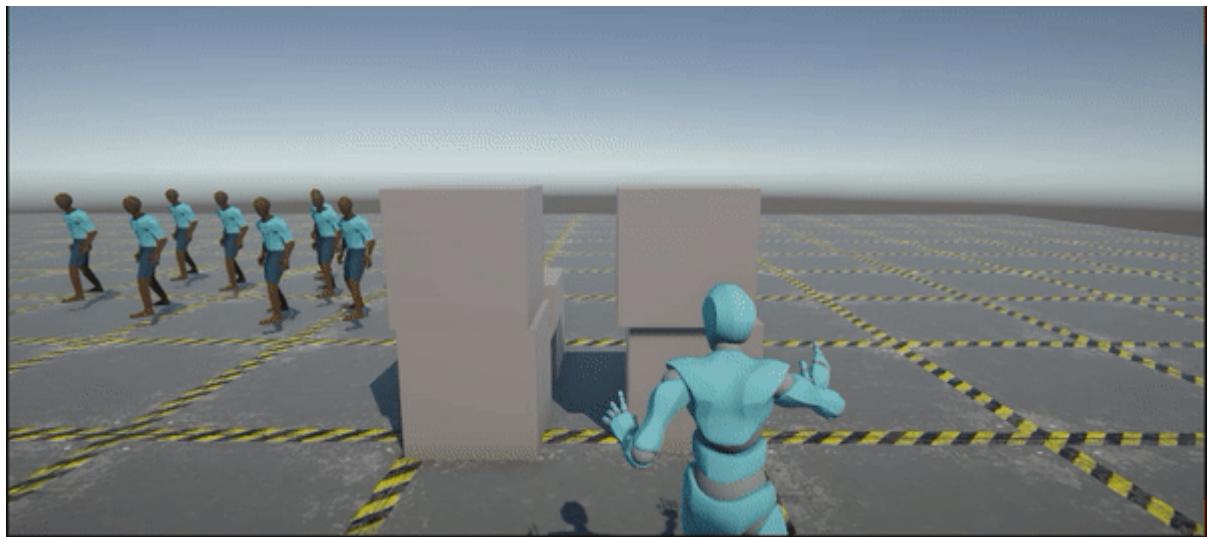
Big Rock Shot: throws a chargeable rock capable of dealing high damage in radius and knocking down enemies. Has medium cooldown and spends considerable mana.



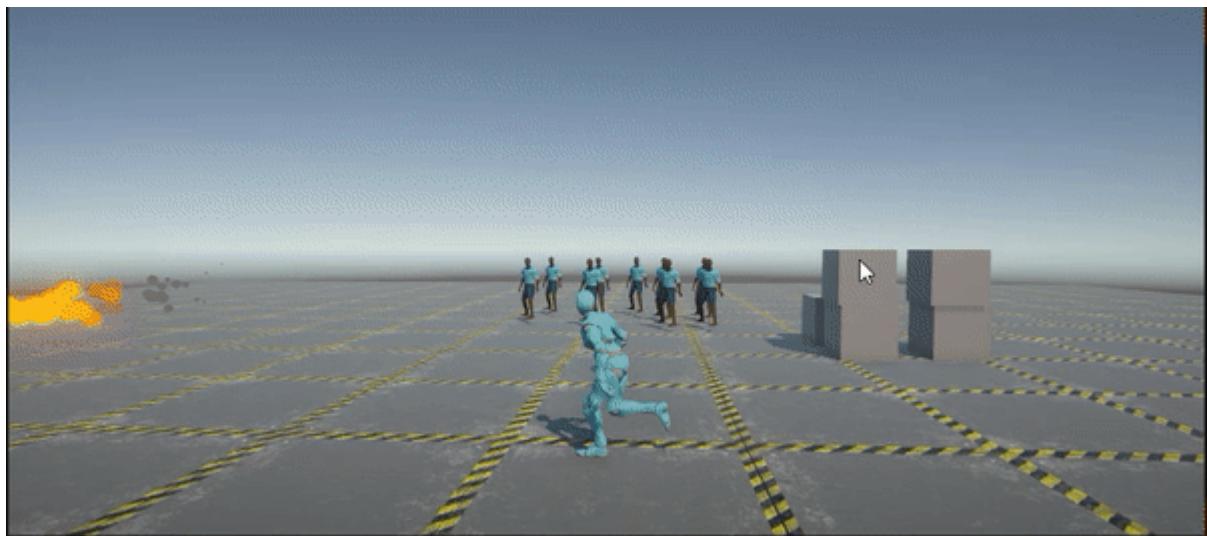
Wind

The wind element focuses on giving defensive abilities and improving the player's movement that greatly increase their capabilities when fighting enemies or exploring the map.

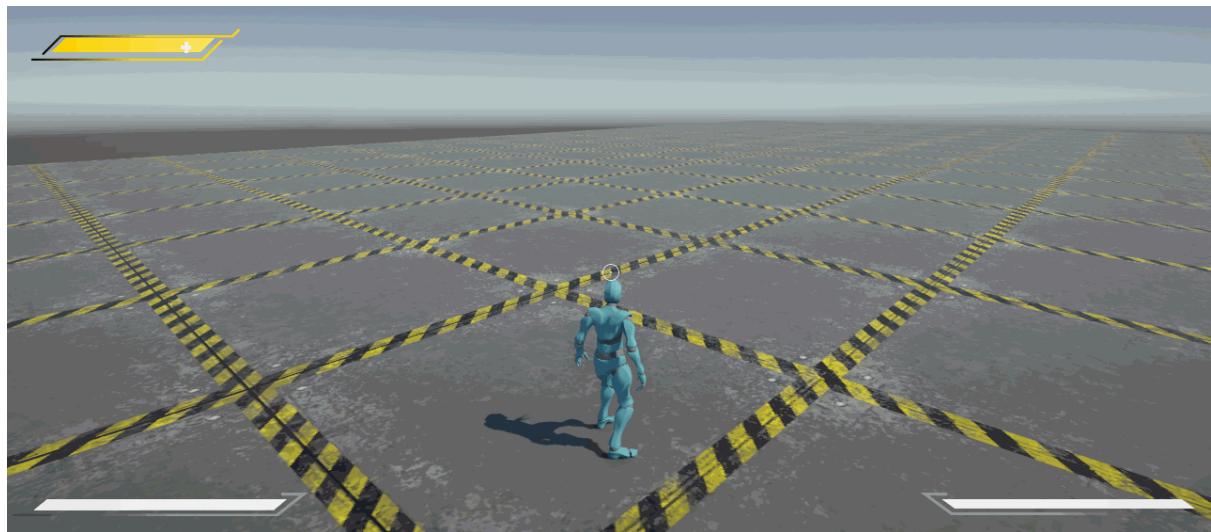
Melee Attacks:



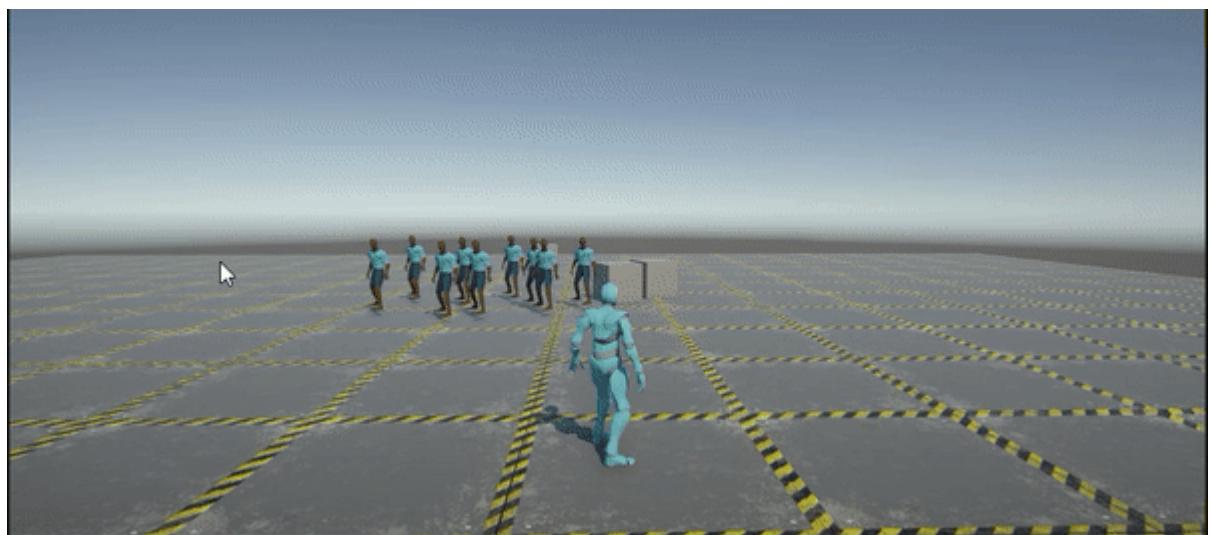
Dash: has low cooldown and high speed, allowing for more freedom of movement.



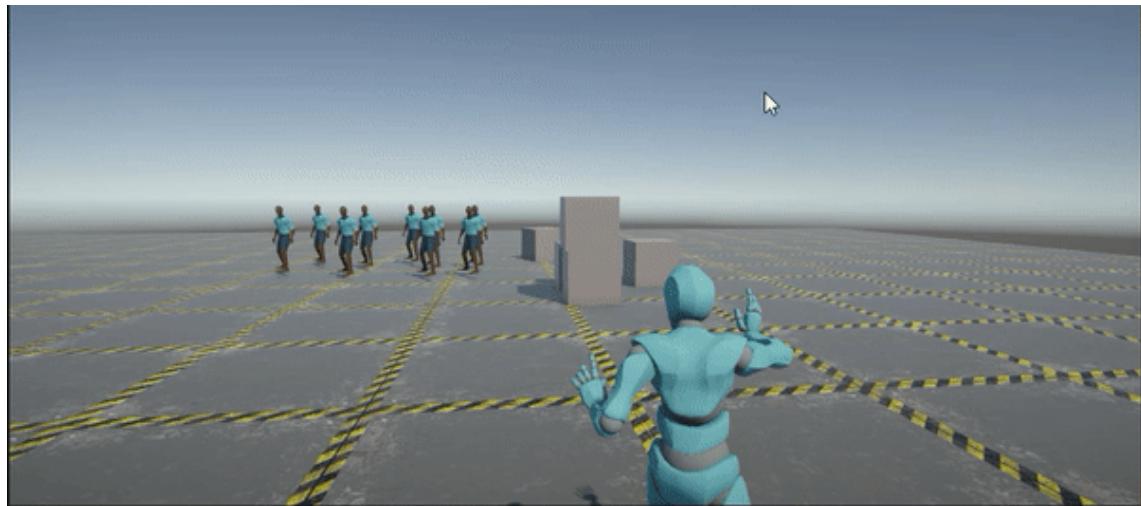
Air Movement Skill - Air Walk: Allows the player to jump midair



Tornado: damages and knocks enemies over time, while also deflecting projectiles. Has high cooldown and spends considerable mana.



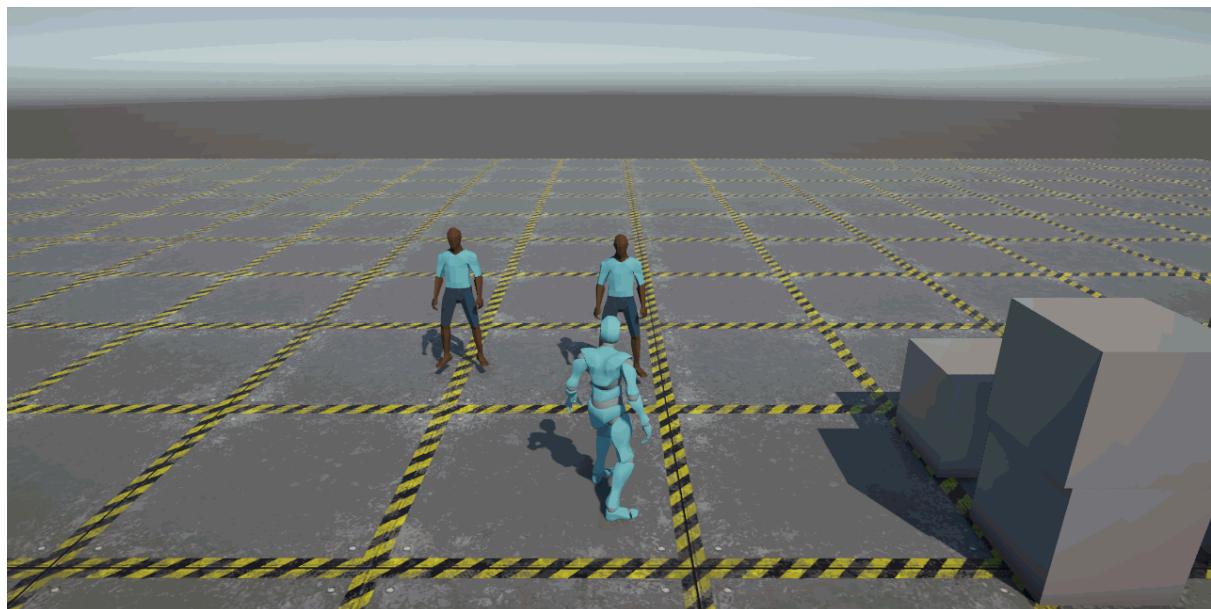
Push Attack: Deflect projectiles and objects. Does very low damage, has low cooldown and spends little mana. Useful for deflecting fireball attacks



Electricity

The spells associated with the electricity element focus on dealing precise damage to single targets. It also has some unique abilities that change the way the player can play the game.

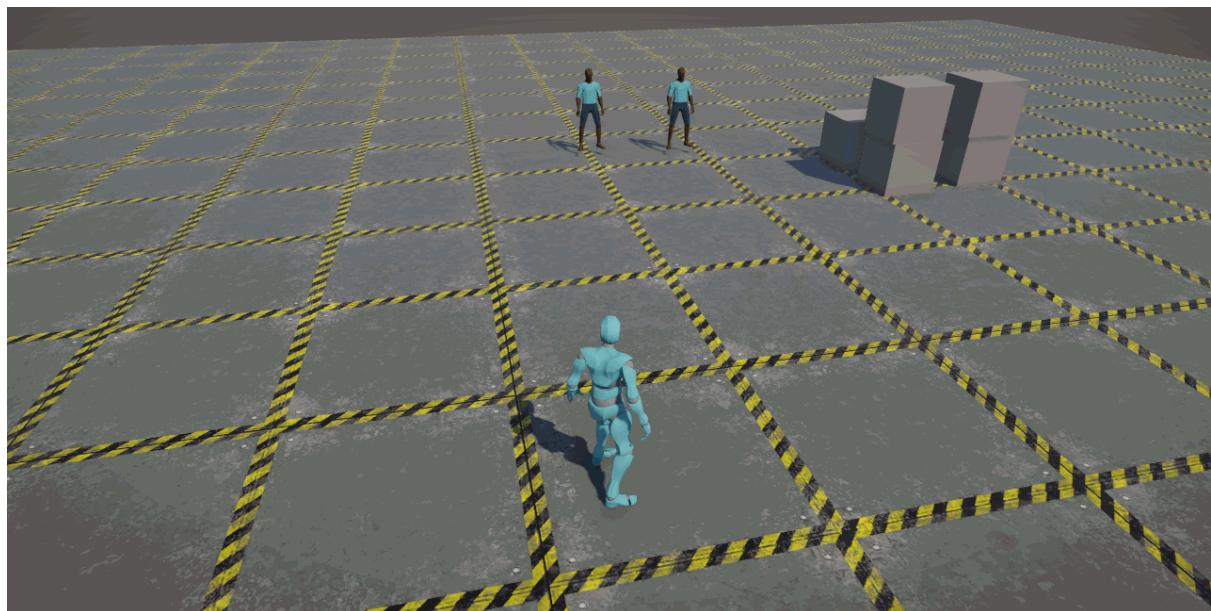
Melee Attacks:



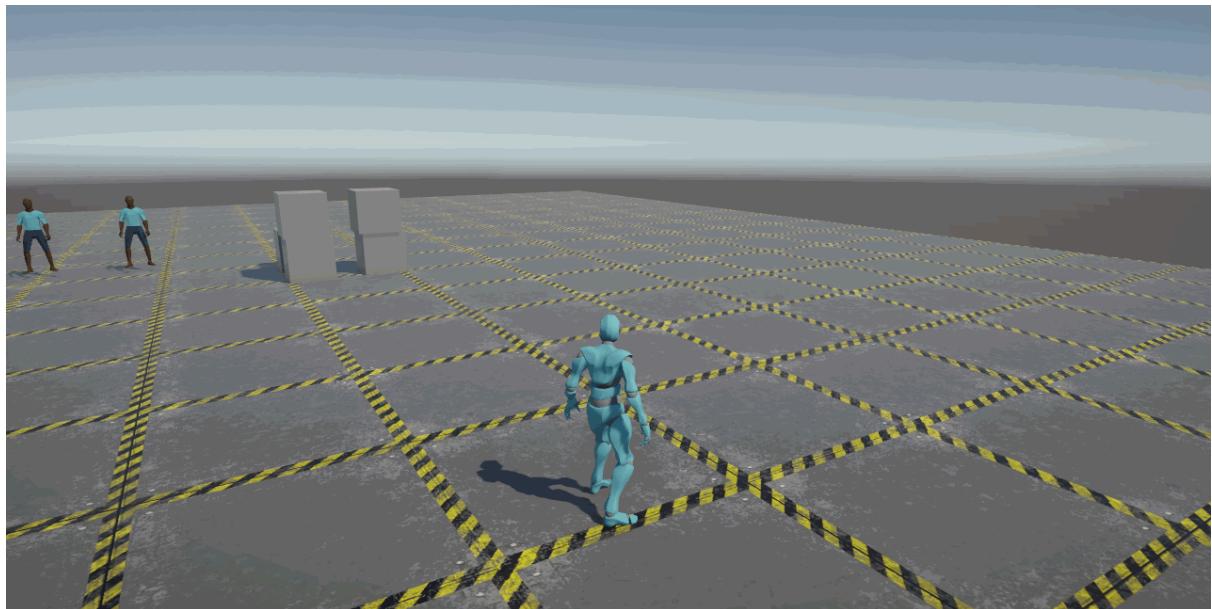
Lightning Dash: temporarily transforms the player into a lightning ray that dashes through enemies while dealing damage over time. Has more cooldown compared to other dashes.



Air Movement Skill - Hover effect: allows the player to glide midair. This is similar to the fire jump variation, with the exception that this one allows the player to move upwards more easily than forwards. Useful for shooting midair.



Electric Walls: unique objects that can be temporarily placed in the ground (dealing little damage) to form customizable traps that deal considerable damage if an enemy steps on it. Has low cooldown and spends little mana, to allow for multiple placements.



Electric Ray: continuous lightning ray that deals damage over time in a specific spot the player is aiming at.



Electric Bolt: quick lightning bolt that deals damage in a specific spot the player is aiming at. Has a medium cooldown and mana cost.



All of the spells and abilities can be combined by equipping them in the left or right hands (in the inventory or *quick switch* key), or by selecting the active element (*ALT* or respective key) in case of the melee, jump and dash.

Items

The game has a few important items that can be acquired with mission rewards, collectibles or by buying them in the shop. They can be checked or used by accessing the inventory.

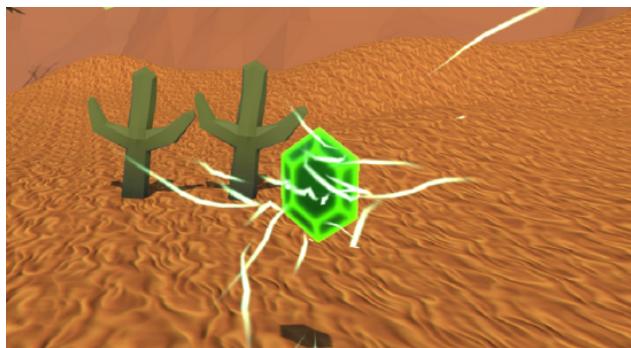
- Health Potions: Consumables that restore a portion of the player's health. It can be used in the inventory or in the *quick use* key (C).



- Spell Books: Items that can be used to unlock new spells.



- Element Gems: Items that can be used to unlock new elements (and respective jump and dash).



- Gold: Currency that can be acquired by completing missions and used to buy items in the shop.

Game World Description & Design

The world of Mystical Realms is set in the medieval age, in a universe where magic is part of everyone's life via the power of the elements. As the player, you start in the Earth Village which is the only town that is yet to be conquered by the evil organization. The remaining villages: Wind, Electricity, and Fire villages are in the hands of the enemy and must be freed.

Each village has a very unique architecture, making it easily recognizable both inside the city and outside in the wilderness. The following section depicts the unique elements of each village.

Earth Village

- There is a castle as the main building of the city. It is possible to get inside and even reach the ceiling.
- The city is surrounded by mountains primarily composed of sand, giving it a very isolated feel, with a single entrance containing 2 watchtowers.
- The rest of the village is composed of houses that fit the desert theme and additional objects, such as a water well.

- The wilderness can be distinguished by its flora, composed of dry trees and cacti.
- The ground is primarily made of sand and is irregular. Objects like an abandoned raft can be found on it.



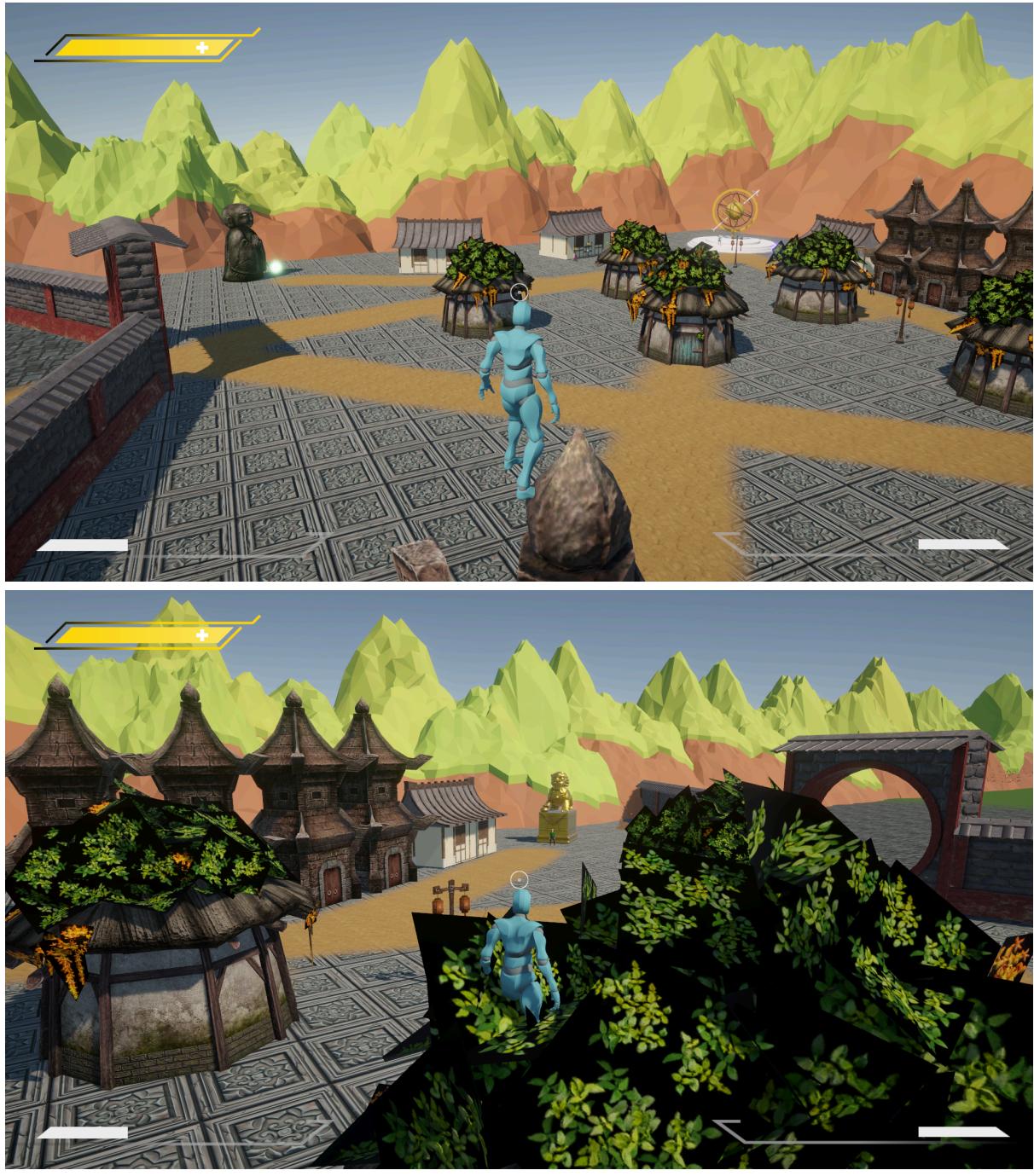
Wind Village

- The village can be seen from afar thanks to the tall tower in the back of the city.
- It is well protected by wood and stone walls, allowing the entrance through 2 gates.
- The inside is also made of wood and stone, resembling a medieval town.
- The power of the wind can be felt through the trees' movement.
- The trees in the wilderness are birch and oak trees and the grass is rather tall.



Electricity Village

- The city has an oriental theme.
- Unlike the other villages, it has 3 main buildings: the Golden Lion, the Temple of Olympus, and The Great Buddha.
- Diverse types of houses and oriental street lamps.
- In the wilderness, the ground is made of stone. However, the city consists of a more polished tiled floor.



Fire Village

- It is surrounded by volcanoes and steep mountains with lava flowing.
- The unique building is right in the center. It's a complex structure made of lava fountains with a dragon climbing a pillar on top.
- It is noticeably more deserted than the other cities. The buildings are scattered around the map, emphasizing the lack of organization here.
- The environment is very dry. The ground is composed of magmatic material, with darker tones and lava.



GLD - Game Level Design

Missions

The game is mission driven. Players will embark on a series of thrilling quests. Each mission offers a generous reward in the form of gold, with the amount varying based on the mission's difficulty level. Additionally, some missions present valuable items as additional rewards, enhancing the player's progress.

The adventure starts in the Earth Village, serving as a captivating starting point that acquaints the player with the game mechanics. The player will uncover the nuances of gameplay, master their skills, and prepare for the challenges that lie ahead.

Earth Village

1. New Adventure - Talk with Bella to learn basic mechanisms (Movement and Jump), who redirects him to the seagull in order to learn fighting mechanisms (Attack and Change Spells).
2. Surprise Attack - Talk again with Bella which will trigger an attack on the village. After killing all the enemies, the player needs to interact once more with Bella
3. New Equipment - Talk to Bella in order to find the Earth's shop and talk to the owner, Maria.
4. Fear - Talk to Edward, the castle's guardian, who will let the player know the fear felt in the village, triggering another fight. After being done with the enemies, the players need to talk with Edward to finish the mission.
5. Promising Future - Reach Bella to unlock the other village's missions.

Fire Village

1. Collect Fireshot Book - Talk to the Elder Raelin, the village's guardian, and find the fireshot book which unlocks the fireshoot spell. Once the player collects the book, it needs to interact with Elder Raelin.
2. Fire Warriors - Reach Richard and fight with invaders of the village. After that, talk with Richard again, who will redirect him to this village store.
3. Brothers Betrayal - Speak with Sarah, that will tell him to go and talk with her brother, Bruno. Talk with Juan, who will help the player find Bruno. Once Bruno is found and the player interacts with him, the player starts being attacked. After dealing with the enemies, the player needs to reach Bruno and Sarah again, receiving a fire stone when the mission is completed.
4. Collect Coins (secondary mission) - Talk with Elder Raelin, who will ask him to recover stolen coins, which have been dropped by a thief. After collecting them, reach Elder.

Electricity village

1. Discover Lightning - Talk with the guardian, Juan, who will welcome him to the village.
2. Collecting Lightning Book - Talk with Kanya, who asks him to find the hidden Lightning Trap Book. The player has to talk to her again in order to finish the mission.
3. Thunderstorm - Speak with Rui, who will trick him into a fight. Once the fight is over, the player talks again with Rui who will inform the player that some important item is near the Lion's statue, the lightning stone.

Wind village

1. Discover wind - Reach Aang, to be welcomed to this village.
2. A Mysterious Power- Talk with Master Shifu who gives him the wind stone.
3. Invasion - Talk with Beatrice to fight with the FireMighters and reach her again to end the mission.
4. Look Up There - Speak with Robert, who will inform the player to talk with the village's greatest wizard, Harry, located on the top of the tower, retrieving a book as a reward, the Tornado Book.
5. Lost Potions - Talk with Elysia, that tells the player that a wagon, which was full of potions, suffered an accident, resulting in potions lost all around the village. The player needs to collect them all and speak to Elysia again, once he has gathered them all.

Final Boss

Reach Bella to trigger the final boss fight in order to save all the villages.

Technical Design Document

- List of all features

1. Mission System with 3 different goals, Interact, Collect and Fight.

Each Mission has an Interaction to begin it. They can have any number of goals, and in order to complete each goal, all the previous goals must be completed. At the end of the mission, it should unlock all the next missions specified in a List.

2. Collectable objects

Collectable objects could be either potions, books or quest items.

3. Inventory to use potions, unlock spells/elements, and assign spells.

Inventory should have 3 areas, the item space, the spells space and the spell wheel space(for Left and Right hands).

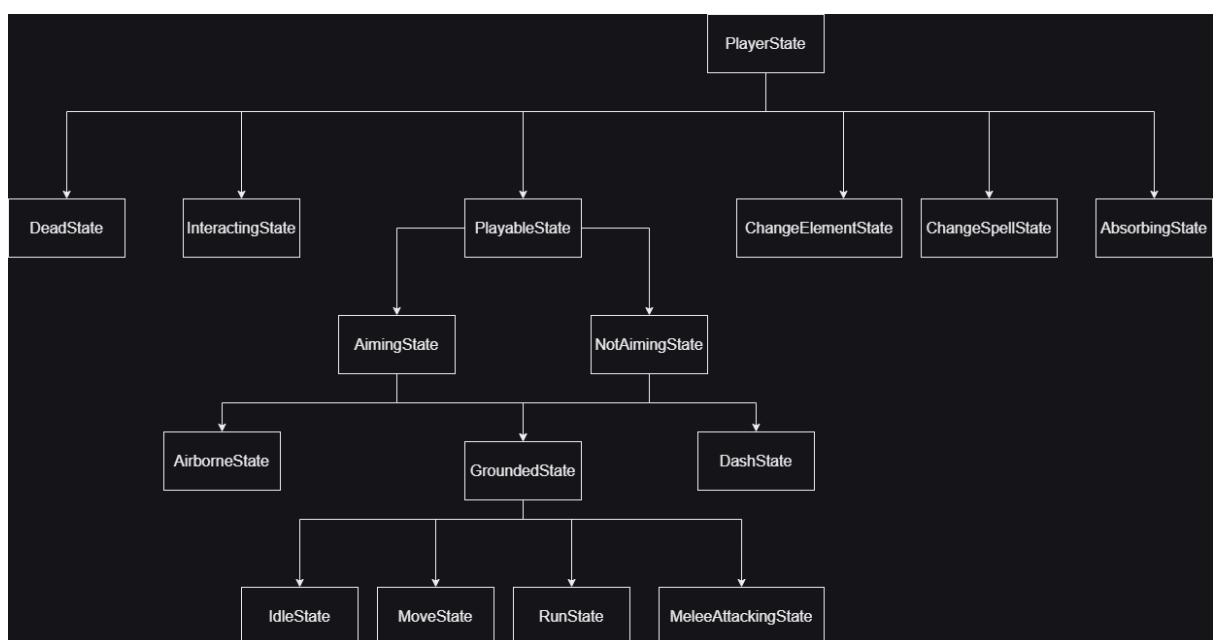
Books, element stones and potions should go to the item space.

Spells should go to the spells and spell wheel space.

The spell wheel space contains the spells that can be changed anytime.

4. Player State Machine.

The Player State Machine should be a Hierarchical State Machine, in order to support the complex mechanics in a scalable way.



*MeleeAttackingState can't have AimingState as a superstate in the tree

5. Player Movement (jump, walk, run, falling)

Players should be able to jump, walk and run when in the PlayableState. Each state should update the Animator, so the animation matches the action. The movement should be done with RigidBody Physics, adding a force to the player's rigidbody when the max velocity isn't reached.

To make jumping feel more dynamic, the physics material is changed to a frictionless material when the player is landing, in order to maintain momentum.

6. Dash

The Dash Mechanics should be versatile to be used by the player and enemies. When the player is dashing, the FOV of the camera can be changed, depending on the dash stats.

7. Air Movement skills, Hover and Airwalk

There are two Air Movement Skills, Hover and Airwalk.

Hover adds a continuous upwards and forwards force, and airwalk is a simple jump midair. Each Element can have an *AirMovementSkill* associated with it. When an element is changed, it adds a component to the Player in order to use the *AirMovementSkill* while in the *AirborneState*.

8. Spell System (SpawnSpell, InstantSpell, ChargedSpell, Continuous Spell)

Each Spell is contained in a ScriptableObject *Skill* that, depending on the type, may have different stats. Each Skill has a Prefab associated with it which represents the Spell in the Scene. Any unique feature of the Spell, should be placed as a Component in the Prefab.

There are 4 types of spells:

- SpawnSpell, Spawns an object in the scene (EarthWall should be in this category, to be refactored in the future).
- InstantSpell, Shoots the spell on keydown.
- ChargedSpell, Starts charging spell on keydown and shoots on keyup. Requires the spell Prefab to have a *ChargedSkillComponent*.
- ContinuousSpell, Shoots on keydown, and stops shooting on keyup.

9. Melee Attacks

Melee Attacks should be versatile so that it can be used by Enemies(Could be implemented in the future) and the Player.

By continuously attacking (Holding the attack button), a small combo is performed. It should be possible to dash while attacking.

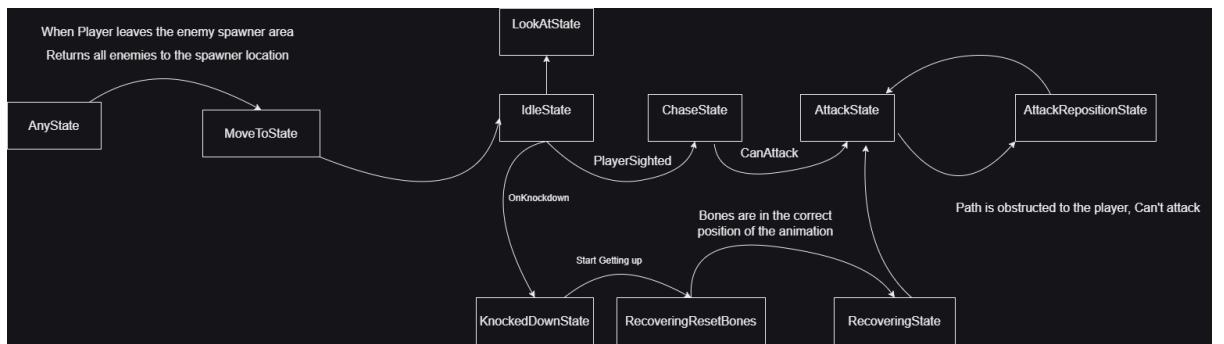
At the start of every melee attack, it is possible to move the Player/Enemy in a desired direction, so that the melee system becomes more dynamic.

10. Change spells in combat with time slowed down

When not aiming, it's possible to change spells mid-combat. Time will be slowed down while the spell is being chosen. The same applies to changing the element.

11. Enemies state machine

Enemies will detect the player when they see them(Line of Sight), hear them(When the player is close to them) , when the player damages them, or when a nearby Enemy has detected the player.



Each enemy has a ScriptableObject **EnemyInfo** that has a Prefab and which behaviors it should have(Use dash and Attack, chase and idle strategy).

*KnockedDownState, RecoveringState and RecoveringResetBones are related to the ragdoll system.

When the enemy can't hit a player with its spells, because the path is obstructed, it changes to the RepositionState, repositioning itself to a new position that can hit the player.

12. Enemy spawners

Each enemy camp has a number of enemies that it contains, the type(Unlimited, Normal), the trigger (OnProximity, OnCombat, OnGoalStart), the location of the spawners, the initial enemies, and what enemies it can spawn. The prototype didn't use this feature to its full extent, only used OnGoalStart as a trigger, and didn't use Unlimited as a type.

13. Enemy messages - PlayerSightedMessage, MoveToMessage

Each enemy can communicate with other enemies and enemy camps, to alert of the player's position. It's implemented in a way to support additional messages, however due to time constraints, they weren't added.

The enemy camp may send a `MoveToMessage` to instruct it's spawned enemies to fall back to the spawner's position when the players gets too far away

14. Ragdoll implementation

The enemies can get knocked down, and put in a ragdoll state.

The ragdoll is able to get back up, by using the `KnockedDownState`, `RecoveringState` and `RecoveringResetBonesState`. They handle what happens during the recovering phase.

15. Enemy Boss

The Boss implements the enemy states, while only changing the `AttackState` with minor changes to its behavior.

It additionally spawns 4 different types of runes to attack the player, in different phases.

`StaticRune` - Shoots at player, doesn't move

`HoveringRune` - Shoots at player, follows the enemy that spawned it

`ChaseRune` - Shoots at player, and chases the player

`DefensiveRune` - Deflects player's projectiles, and follows the enemy.

The boss has 5 different phases, each increases the number of runes, and starts spawning a new rune. When damaged, an explosion is triggered when a new phase begins.

16. NPCs

In the game, you'll find NPCs scattered throughout the villages, each with their own unique roles and stories. To engage in conversation with them, players must approach them closely.

Certain NPCs serve as mission triggers, identifiable by a distinctive yellow diamond hovering above their heads. This diamond acts as a visual cue for players, indicating that an opportunity for a mission is available. Once the mission is completed, the diamond vanishes, signifying that the NPC has returned to their regular state.

To bring the world to life, NPCs are designed with different animations depending on the player's interaction. When engaged in conversation, they showcase a speaking animation, allowing players to visually connect with them. In contrast, when left idle, NPCs display an idle animation, adding depth and realism to the game environment.

17. Mana System and each Element Source

Each element has its own mana, and can be recharged in its respective element source scattered around the map.

The *CharacterStatus* which has the health of the character(player or enemy) contains the mana for each Element.

The player can recharge the mana when he is in NotAimingState and facing an element source.

The enemies don't use mana in the prototype.

- Choice of Game engine

Unity is the ideal choice for developing a 3D open-world game due to its exceptional features, flexibility, and industry-wide popularity. With its robust toolset, Unity provides an efficient and streamlined development environment, empowering game creators to bring their visions to life. Additionally, all the team members had previous experience with the engine.

Furthermore, Unity's extensive asset store and community support make it a treasure trove of resources for game developers. The asset store offers a vast collection of ready-made 3D models, animations, scripts, and plugins, which significantly accelerates the development process and reduces the need for creating everything from scratch. Additionally, the thriving Unity community provides ample opportunities for knowledge sharing, troubleshooting, and collaboration, ensuring that developers have the support they need to overcome challenges.

Unity's cross-platform capabilities are another compelling reason to choose it for a 3D open-world game. With Unity, you can develop your game once and deploy it on multiple platforms, including Windows and Linux.

Moreover, Unity's built-in physics engine, animation system, and visual effects simplify the implementation of realistic interactions and behaviors within the open-world game. These features, combined with Unity's extensive documentation and tutorials, empower developers to create dynamic and interactive environments that enhance player immersion.

- Details about the 3D objects, terrain, and scenes
 - The assets used in Mystical Realms were taken from the Unity Asset store. Some were customized to our needs using Blender or Unity built-in tools.
 - The unique village statues were built from the following packages: [Dragon Statue](#), [Olympus Temple](#), [Golden Lion](#), and [The Great Buda](#).
 - The unique village buildings, such as the Earth village castle and the Wind Tower were derived from the following packages: [Electricity Village](#), [Fire Village](#), [Earth Village Canyons](#), [Electricity Village Gates](#), [Wind Village Gates and Houses](#), [Desert Objects](#), and the [Earth Castle & Houses](#).

- The ground textures and materials came from the following packages: [Electricity Wilderness](#), [Fire and Earth ground](#), [Wind soil](#), and [Electricity Village floor](#).
 - Some shaders were developed to enhance the theme of the respective villages, such as a lava shader, a wind shader, and a water shader.
 - The terrain was sculpted using the Unity Terrain Tools. This enabled the creation of the environment according to our needs, including mountains, slopes, trees, grass, and floor textures.
 - In order to improve the performance of the game and lower the requirements, we implemented a chunk-loader system that renders chunks according to their distance from the player. This is accomplished by the use of Addressables. Addressables use asynchronous loading to support loading from any location with any collection of dependencies. Thus, objects are loaded/unloaded from the scene as the player traverses through the map.
- Networking (none)
- Delivery platform & hardware/software requirements for running the game on a system
 - **Minimum Specification**
 - OS: Windows 10/11 or Linux (Debian or Arch based).
 - Processor: AMD Ryzen 5 4500U (64 bit)
 - RAM: 8GB
 - Graphics: AMD Radeon Graphics (integrated)
 - Storage: 2GB available space

Concept Art (Art Style Guide)

The art style should follow a cartoonish look giving a fantasy immersion. Some screenshots of other games can be seen for initial inspiration.



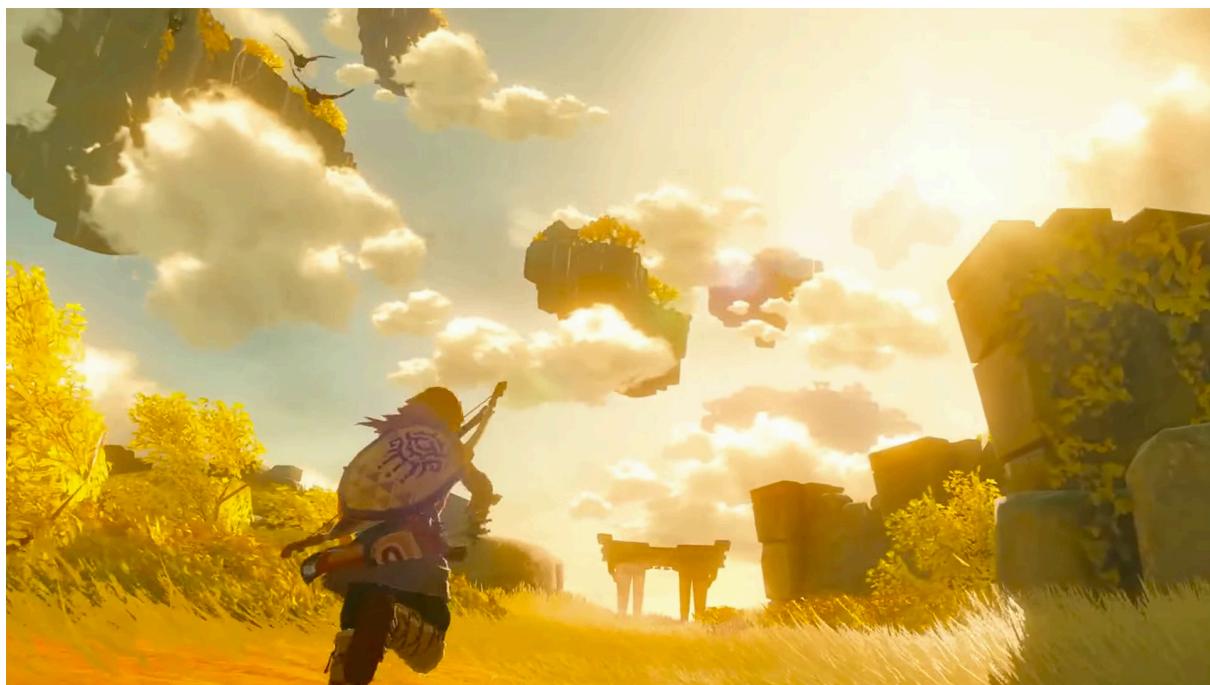
Immortals Fenyx Rising



Spellbreak



Genshin Impact



The Legend of Zelda: Breath of the Wild

The interfaces follow a cartoonish style, consisting on dark transparent backgrounds with yellow gold-like borders.

Project Plan

The creation of the game was divided with a focus on six key areas:

- World creation
- Character's movement and gameplay
- Mission system development
- UI implementation
- Sound design
- 3D modeling

By dividing the tasks into these categories, the game development was much faster, having different people focused on different parts.

World Creation

- Plan the map geography, including the village borders, connections, and wilderness paths.
- Sketch the architecture of each village, following a unique theme.
- Search for models, materials, and textures for each village environment.
- Development of the Chunk System, improving the performance of the game through distance-based loading of the map.
- Build each village, using terrain tools and assets.
- Development of appropriate shaders, such as the lava shader.

Character's movement and gameplay

- Implement the main character's physics and movement.
- Implement a system to use and cast spells.
- Implement a melee combat system.
- Implement all spells for each of the elements.
- Implement special variations of jumping and dashing.
- Create AI enemies with specific behaviours.
- Create enemy spawner camps to be used in the map.
- Test and change the parameters to balance the game (spells, enemies, health, etc.).

Mission System Development

- Implement different user mechanisms, such as interact with objects and NPC's on approach and fight with a spawn of enemies
- Design a mission system to support different types of missions.
- Create generic missions with objectives and rewards.
- Implement the mechanisms for the different types of missions (collect, interact with NPCs, fight)
- Test and iterate on missions to check the difficulty, and overall player experience.

UI Implementation

- Main menu

- Playing interface
- Inventory
- Missions menu
- Interaction with Npcs
- Tutorials
- Map

Sound design

- Abilities: sounds for different effects
- Combat sounds: damage taken, melee
- Movement: walk, run, airwalk, etc
- Dynamic Ambience: unique ambient sounds tailored to the specific village the player is in.
- UI: interaction with the menus
- Music: generic battle music, wind battle music(not implemented) and menu music

3D modeling

- Movement and take damage animations taken from Mixamo
- Aiming, airwalk

Resource Plan, Budget, Schedule, and Milestones

Resource Plan and Budget

The game development process was subjected to significant budget and resource constraints as it was a university project undertaken by students without any monetary funding. Consequently, the team relied on self-created assets or freely obtained royalty-free resources from the internet. Additionally, time posed a considerable limitation, as the team had approximately three months to complete the entire project.

Notably, the team consisted of the following human resources:

- 7 developers who were assigned to various aspects of the game.
- 2 sound designer tasked with creating crucial sounds and music for the game.
- 1 game modeler responsible for creating the most important models and animations in the game.

Schedule and Milestones

There was a general schedule plan proposed by the teachers that can be found below:

Week (Thursday)	ALL	DEV	3D	SND
16/03/2023	Work on concept and GCD.			
23/03/2023	Work on concept and GCD. Start work on Game Design			
30/03/2023	Work on game and level design. Define list of assets to be developed (consider UI as well)	Submit GCD Review concept art Test simple model export and import with [3D] Start experimenting with main mechanics	Provide concept art Test simple model export and import with [DEV] Start modelling	
06/04/2023		Easter		
13/04/2023	Work on game mechanics and level design (2nd sprint)	Develop first obstacles, enemy and power-ups Define Interactions with [3D]	Define Interactions with [DEV] Continue modelling	First contact with project teams Game Sound Doc first draft. Define Interactions with [DEV]
20/04/2023		Refine Mechanics and Interactions Implement more obstacles, enemies and power-ups	Continue Modelling Provide updated assets to [DEV]	Test FMOD integration with [DEV]. Updated Game Sound Doc.
27/04/2023	Work on game mechanics and level design (3rd sprint)	Integrate assets in the game Test FMOD integration with [SND]	Provide updated assets to [DEV] Start creating blending animation (states)	Game Sound Doc Detailed. First Sound Mockups and proposals
04/05/2023		Integrate draft animations from [3D]	Provide draft animated versions to [DEV] Provide videos of draft animated versions to [SND] Continue animating	Provide updated assets to [DEV]. Sound place holders (events) implementation - v1
11/05/2023		UI Development with [3D]	Support [DEV] on UI Development	Work with videos provided by [3D] Provide updated assets to [DEV]. Refine events and parameters
18/05/2023	Refinements			Main Sounds Implementation and first running tests
25/05/2023		Start final integration and testing with assets from [3D] and [SND] Intermediate presentation in practical class	Provide final assets to [DEV] Assist in testing	Global Sound implementation. Minor changes and updates. Implementation assistance, fine tuning and sound mixing with FMOD connected to realtime gameplay
01/06/2023				Provide final assets to [DEV] Assist in testing
08/06/2023		Final adjustments and submission (except docs)		Final adjustments and submission (except docs)
Week of 12/06/2023	Final presentation (final date to be confirmed)			
Week of 19/06/2023		Documentation delivery		Documentation delivery

Other than that, the main milestones the team defined specifically for this project were the following:

- Development of the main mechanics (fight, movement, spells, etc.) - **developers**
- Design and conception of the world - **developers**
- Design and implementation of the user interface - **developers**
- Implementation of the mission system - **developers**
- Script for the narrative and missions - **developers**
- Design and conception of the sounds and music - **sound designers**
- Design and conception of the models and animations - **modelers**
- Play testing, refinements and bug fixing - **all**

Test Plan

The plan for testing the game was to make a component by component testing, so each developer was responsible by thoroughly testing each feature they

implemented. When there was an integration of two or more components, the developer making the integration was responsible for testing it, as well as any developers responsible for the individual components. Finally, testing was performed more extensively when the game was near its completion to find any additional bugs.