Devon Richey

CSCI 43700 Final Project

Magic Tower – Game Design Document

Design Summary:

Magic Tower is a 2D Rogue-lite Action Platformer developed in Godot in which the player takes on the role of a knight attempting to reach the top of a deadly tower, making use of the magical artifacts and secrets they find along the way to overcome the tower’s challenges. Gameplay largely follows an alternating pattern of arena fights and platforming challenges with stat upgrades being offered at the end of each room. At regular intervals throughout the tower, the player will face off against more challenging boss encounters, the completion of which will grant the player a new weapon or spell to add to their arsenal. The player is challenged to reach the top of the tower without dying, and as quickly as possible. Death resets the player to the beginning of the tower with none of their collected upgrades. Each attempt to climb the tower will offer the player different item choices, different bosses, and different room combinations.

Demo Goals:

With a limited timeframe and scope for the game, a demo should show off the core gameplay loop, offering an example of each type of room and showing some of the types of choices the player will have to make. This should include a skippable tutorial room displaying the basic controls, a sample combat arena, a sample platforming challenge, and a sample boss fight. The player should be given an item choice prior to the first combat arena, and stat upgrade choices at the completion of each the combat arena and platforming challenge.

Scene Structures:

Rooms:

Each type of room scene extends a Room node. The Room node contains a tilemap which represents the static terrain of the room and the collision for the terrain. The Room node also contains an Area2D which acts as the exit from the room. This exit is initially blocked when the room first loads, and becomes accessible when the player fulfills the exit condition corresponding to that room. Each room also contains an item choice. This item choice is spawned upon load in the tutorial safe room, and upon the fulfillment of the exit condition in all other rooms.

Room Type 1 – Combat Arena:

A combat arena is a smaller room, matching roughly with the size of the camera. A combat arena room also contains a list of enemies to spawn. Most rooms will spawn enemies in waves, waiting for all enemies in a wave to die before spawning the next wave. Some arenas will spawn additional enemies on timers, forcing the player to deal with enemies quickly or become overwhelmed. Once all enemies of all waves have been defeated, the exit condition is fulfilled, the item choice is offered, and the exit door becomes accessible.

Room Type 2 – Platforming Challenges

Platforming challenge rooms are generally larger to accommodate additional movement obstacles. These rooms often contain moving parts, such as additional platforms or hazards to increase the difficulty of the platforming. The exit condition is fulfilled either by flipping levers scattered throughout the room, or by simply reaching the exit door at the end of the room. The exit door is generally located on a safe platform, where the item choice is also offered.

Room Type 3 – Boss Arena

Boss arenas are similar to combat arenas, however, only contain one more powerful boss enemy as opposed to multiple waves of basic enemies. The exit condition is fulfilled when the boss is defeated.

Player:

The Player Node is a KinematicBody2D with an AnimatedSprite node operating the visual aspect of the player’s behavior. The Player Node also contains two Area2D nodes which operate as area checks for the melee attack behavior. Another KinematicBody2D is included along with an additional AnimatedSprite to act as a Reticle, which is used by the player’s ranged attack behavior as both a visual representation to the player of the direction they are aiming, and for the game to know where to spawn the Spell and Projectile nodes.

Spell:

The Spell node is instanced when the player uses their ranged attack. This node is given the id of the Player’s currently held spell item, as well as the stat multipliers related to the ranged attack. The Spell is then instructed to spawn Projectile nodes according to behaviors related to the spell id. The Spell class has a lifetime timer which allows for the Projectiles spawned to complete their functionality, after which the Spell removes itself from the scene.

Projectile:

The Projectile node is a KinematicBody2D with an AnimatedSprite node that is instanced by the Spell node when the Player uses their ranged attack. The Projectile contains values related to its size, speed, and damage output. These values are modified by the Spell class upon instantiation, after which the velocity of the Projectile is set. Upon collision with anything in the scene, the Projectile will apply damage to that object if it is an enemy, and then remove itself from the scene. Some Projectiles have additional behavior, such as the FireProjectile, which possesses a large Area2D which it uses to deal damage to enemies in a large area after a delay, mimicking the effect of a timed explosive.

Item Choices:

An item choice offers the player two random options from the pool of either items or stat upgrades. The item choice object will disappear when the player interacts with either of the two objects, adding the item to the player’s inventory. The added item will update values stored in the player object and change some aspects of the player’s behavior.

Item – Weapon:

Weapon type items modify the melee damage dealt by the player, as well as the rate at which the player can use their melee attacks. Planned variants include Quick Slash, which has a higher attack speed and lower base damage; Heavy Blow, which has a lower attack speed and higher damage, with a added knockback effect; Vampire’s Fang, which has a slightly higher base damage and has an added health regen effect on hit; and Spellsword, which has a slightly higher attack speed and has an added mana regen effect on hit.

Item – Spell:

Spell type items modify the damage and projectile behavior of the player’s ranged attack. Variants include Ice, which fires three projectiles in a cone pattern for slightly higher damage; Fire, which fires a slower projectile that applies much higher damage in a large area after a brief delay; and Lightning, which fires at a much higher rate at a less accurate spread.

Item – Super:

Super type items grant the player access to powerful area of effect spells. Variants include Ice Storm, which fires off a large number of ice projectiles at various angles for a short duration; Mega Flare, which deals high damage to all enemies in a large area centered on the player; and Call Lightning, which causes lightning bolts to strike random areas around the screen.

Item – Stat Upgrades:

Stat upgrades provide a small multiplier to the base stats of the player, and are offered as choices after each combat arena and platforming challenge.

Enemy:

An Enemy is a KinematicBody2D with an AnimatedSprite node representing its visuals. Each Enemy has an amount of health and the ability to take damage. The behavior of each Enemy is dictated by its own node.

Enemy – Guard:

This enemy patrols between two points until the player enters its detection range, after which the enemy will pursue the player. The enemy gets the position of the player, then tries to reach a position within melee range of the player in order to attack. If the player is unreachable for a period longer than the enemy’s attention timer, then the enemy will resume its patrol. This enemy only attacks if the player is in melee range.

Enemy – Crawler:

This enemy exclusively patrols between two points along a given platform and constantly spews projectiles at a random upward angle. These projectiles are affected by gravity.

Enemy – Spitter:

This flying enemy hovers in place until it detects the player, then seeks to reach a position above the player. Once in this position, the enemy will fire projectiles at the player.

Enemy – Diver:

This flying enemy patrols an area until it detects the player, at which point it will begin to chase the player, dealing damage whenever it collides with the player.

Current Development State:

The development goal for this submission was the demo described above. Due to time constraints related to learning a new engine, the current state of the game includes basic player functionality and item pickups for spells, but does not yet include the player’s super abilities or other types of item pickups. In addition, only a single combat arena has been implemented, with only one enemy type representative of the enemy class. This still allows for the showcasing of most player mechanics, as well as combat and item pickup functionality, but does not fully showcase the full breadth of the gameplay loop.

Issues were also encountered in the functionality of the player’s melee attack not properly reading other bodies in the scene which it overlapped, preventing the melee attack against enemies and the collection of items from functioning properly. A similar issue is present in the scene transition object, preventing the scene from changing.