12/17/2018

* Progress
  + Project backlog started
* Current State of Project
  + Spell generator is functional
  + Spells can be instantly shot, held, or charged
  + Basic damage system is in place
* Questions
  + How to handle physics calculations for Character Controllers?
    - Should consult SCAM project for details

1/1/19

* Progress
  + Added mesh changer spell modifier. Proven successful. Mesh will need to be centered properly.
  + Spread shot has been added
  + Animation States are functioning for NPCs
  + Added status effects
    - Damage over time
    - Paralysis
    - Transmutation(a little jank)
* Current State of Project
  + Spell modifier test is successful
* Questions
  + How to prevent NPCs from walking off cliffs when they want to chase the player?
    - NPCs don’t utilize NavMesh in Chase State
    - Use raycast just ahead of them to detect cliffs?
  + Current status effect implementations may be mild performance/security risk?
    - Is this worth it?
    - Are alternatives possible?
  + All damageables will need to account for a parent damageable
    - Example: transmuted objects

1/2/19

* Progress
  + Started map building system
  + Added Tile MonoBehaviour
  + Added TileData class to store information on specific tile
  + Added LevelBuilder
  + Created separate scene for level building testing
* Current State of Project
  + Added above files. Relatively unchanged since 1/1/19
* Questions
  + How should Tile be initiailized?
  + Need to spend time going over design of levelbuilding
  + NavMeshes will not work with procedurally generated levels
    - Consider pre-built levels instead?

1/9/19

* Progress
  + Level Builder Build Room function operational
  + Room sizes can be defined
  + Room blueprints exist and can be modified
  + Established concept of TileData
* Current State of Project
  + Rooms can be built, but tile pieces are not yet oriented based on location
* Questions
  + Procedurally generated navmeshes
  + Specialized rooms for specific situations
  + Defining a room ID
  + Tiles will print CEILING if asked to display their tiletypes even if they’re not

1/10/19

* Progress
  + Implemented several helper functions for IntVector3
  + Tested room building functions for LevelBuilder
* Current State
  + LevelBuilder is capable of establishing a room and instantiating appropriate materials for said room
    - Difficulty orienting objects according to room layout
    - Need to determine scalability design in a future date
  + Considering putting procedurally generated levels on hold for the time being
* Questions
  + Would it be better to shift gears away from procedurally generated level?
    - Very complicated and large task
    - Good for singleplayer experience
    - Shift gears towards arena shooter design?
    - Long start to finish level design with minor dynamic elements
      * Spawn points
      * Critical Path objects/locations
    - Possibly skip forward to potential multiplayer design philosophy?
  + NEXT FOCUS: DESIGN AND GRAYBOX CASTLE\_01 ARENA STYLE LEVEL
    - Theme: Main castle areas: hallways, dining hall, entrance hall, staircase, etc.
    - This should be the main/basic map, very even fair spread with moderate highs/lows
    - Indoor setting with some large rooms

1/14/19

* Progress
  + Redesigned wall instantiation
  + Single room is generated with properly angled/positioned walls
* Current State
  + Level Builder generates a single room with properly oriented walls
  + Room contains 4 entrances hardcoded into the center of the walls
  + Will generate doors at higher levels
* Questions
  + Override room designs and generate doors where ever desired?

1/15/2019

* Progress
  + Allowed for configuration defined number of doors instead of random
  + Added a wall texture for testing purposes
* Current State of Project
  + Spawns primary structure of room(floor, walls, door)
  + Will automatically orient walls/doors to sensible positions/rotations
* Questions
  + Spawning and connecting multiple rooms: How do?
  + Idea 1: Generate a list of rooms from some kind of configuration
    - Can be based on size of total map
    - Create a list of rooms that have a “preferred set of dimensions”
    - Build each room out, shrinking dimensions as needed to allow for sizing.
    - NOTE: this will require some method that determines largest available space in grid
    - Leverage RoomBlueprint scriptableobject to contain preferred set of dimensions
      * Maybe also contain an override preference for what room it wants to connect to
      * Ex. Bedrooms that have 2nd entrances may connect to a bathroom
      * Ex. Dining Hall must connect to at least one kitchen
    - Double doors: BuildDoorTile algorithm needs to check for neighboring doors in order to build double doors
      * Will also need to do double pathfinding for hallways(limited)

1/28/2019

Progress

* Stopped procedural level generation production. Need to rethink structure
* Begun building player base as 1st level of game.

Current State of Project

* Wizard tower level has been ¾ designed. Building has begun

Questions:

* Consider full length story to tie hand built levels into.
* If returning to procedural generation, reconsider:
  + Prefab based system combined with grid registry system
  + Rooms to be built with preset, empty room prefabs that will register themselves onto the map grid(a return to initial design of Capstone)

Tasks:

* Build battleground scene to test combat within
* Build basic melee enemies to test combat in

5/14/2019

I really need to get better at this. Maybe I’ll make it weekly?

* Progress
  + Object pooling
  + Melee and ranged enemies
  + Started save game data
  + Raycast casting method
  + Hold person spell
* Current State of Project
  + Game is playable
  + No outstanding bugs
* Questions and Concerns
  + Hold person spell needs tuning and considerations
    - Moving around while holding the person should move the person relative to caster(like a transform parent child relationship)
    - What should occur when a simple single shot hold person hits a target?
  + Loot system should start implementation
    - Tiered loot
    - Tiered chests
    - Tiered enemy drops
    - Should be design tailorable
* Tasks
  + Design loot system
    - What can be considered loot?
      * Health bubbles
      * Mana bubbles
      * Spell components
      * Cloaks/outfits?
    - Tiered Chests
    - Tiered Loot(?)
    - Will this require a redesign of spell component systems?
      * Rather than making serialized spell components with configurable data, spell components should have blueprint ScriptableObjects that are used to generate runtime spell components with configurable stats?
      * Would require a few days worth of refactoring?
      * Advantages:
        + Allows for more sensible loot tier system where tiers are assigned based on runtime stats of object
      * Disadvantages:
        + Large refactor
        + Will require save data to store even more information about object

5/29/19

* Progress
  + Basic loot implementation successful
* Current State
  + Game playable
  + Chest can be opened and health/mana orbs can be retrieved
* Questions
  + Scene transitions
    - Transition from menu view to gameplay
    - Data required
      * Spells
      * Other equipped items
  + Spellbook to be deprecated?
    - Spells don’t normally appear in the world
    - Purely player created -> no need to use monobehaviour representation
* Next Steps
  + Engineering
    - Build spell inventory view
    - Build current arsenal spell inventory view
    - Create select and back events to allow swapping spells in and out of arsenal
    - But maintain spell inventory
  + Design
    - On player death, what are the consequences
      * All rewards gathered are lost
      * Player is returned to menu screen
    - Iconography
      * How should spell components be represented?
      * ***How should player created content be visually represented in UI?***

7/27/2019

* Progress
  + Spell and inventory prototype view
  + Begun graybox level prototype
* Current State
  + Basic game functionality established
  + Tutorial level has been started
* Questions
  + Consider entire game loop for player
    - How will levels be started?
      * One of a few locations with some differences in each location?
    - How will levels end?
  + Consider building custom nav mesh/ navigation system
    - Unity’s navigation system is finnicky at best
    - Or perhaps, reduce the complexity of unit intelligence?
    - Note: raycasting algorithm
      * A ray fired from a point ***within*** the polygon will intersect the sides of the polygon an odd number of times
      * A ray fired from a point outside the polygon will intersect the sides of the polygon
    - Colliders in unity have “collider.bounds.Intersects(Bounds bounds)” which is useful for procedurally generated navigation
      * Floor tiles can use this function against all walls in its tile
      * Floor tiles can also use this function against all other obstacles within its room
* Next Steps

8/30/19

* Progress
  + Game State Machine first pass implemented
  + Started work on loadout editing
  + Arena ultra basic prototype implemented
* Current State
  + First basic game loop much closer to completion
  + Still only placeholder graphics
* Questions
  + Need to perform various art passes on UI and models/textures
  + Should consider if certain UI elements can be abstracted for more generic implementations
* Next Steps
  + Complete loadout edit art pass
  + Complete loadout edit implementation
  + Create win/lose conditions for arena mode
    - Stop player control on death
    - Display lose screen UI panel
    - Give player arena stats
    - Give player option to try again/quit