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)