OVERALL GAME IDEAS:

-Unreal Tournament Java Bot API

-pros: pre-built game, only implement AI

-cons: haven’t used before, limits our options for PCG (and AI techniques?)

-Unity roguelike/dungeon crawler

-pros: endless options for PCG and possible AI techniques

-cons: have to build from scratch

AI TECHNIQUE IDEAS:

-decision tree (“learning” optional)

-as AI enemy agents attack/get killed, they follow a decision tree that dictates which action they should perform next (optional: learn which techniques work best and update the tree to be more biased towards “good” decisions)

-Difficulty: how do we determine when a decision produces a good result? (i.e. when the agent attacks… does the target lose health? Or more health than the agent? Or what happens after the attack?)

OPTIONAL IDEAS (IMPLEMENT IF POSSIBLE):

-genetic algorithm for enemy/player AI

- Player model learning/adjustment of AI based on how player performs

**PITCH TOPICS:**

-Overall Game

-roguelike dungeon crawler

-player shoots, enemies attack via melee

-win game by progressing through set number of rooms and killing final boss

-pathfinding uses procedurally generated gridworld

-separated into rooms for size limit

-PCG rooms

- all rooms generated at the beginning of the game

-rooms contain walkable space, non-walkable obstacles, enemies, and the enter/exit doors

- dictated by size, obstacle placement, door placement

- enemies randomly spawn minimum distance from door player entered

- max percentage of room blocked by obstacle

-Decision Making

-at beginning of game, manually created decision trees are used by enemies which become better as the player progresses

- possible actions will include: attack player, run away, dodge bullets, block door, wander, etc.

- if possible, we would like to implement a learning component for the decision trees