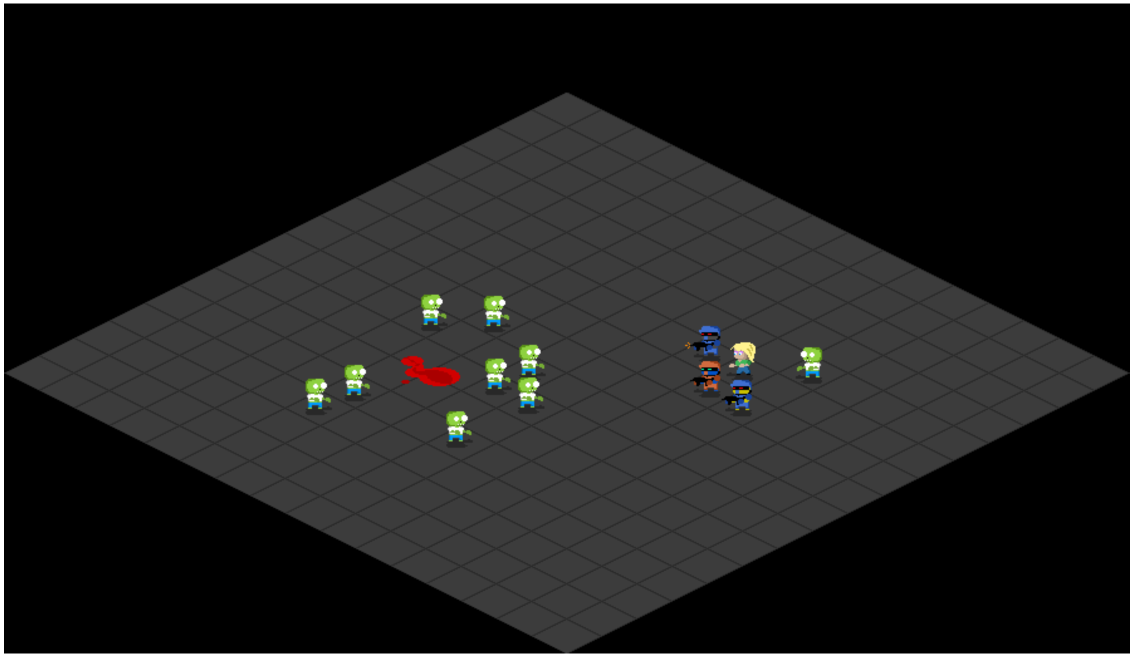
Rogue Zombie Prototype v1



Testing Essentials

- Testing Combat
  - human fights zombie
  - zombie fights humans
- Testing Movement
- Testing NPC interaction
  - Talking to NPC to get mission
- Testing Item interaction
  - Picking up key
- Inventory
- Mission Log
- Scripted Events

Level

- 3 story office building
- Floor 1: Lobby
- Floor 2: Offices / Cubicles
- Floor 3: Office with Desk Key

Mission 1: Escape Building

- Start on first floor as human with a gun, and pack of smokes
- Mission begins “Find the key to get out of locked building”
- Go to NPC to get info on key location on floor 3 and get access card to floor 2 by trading smokes
  - (Can kill NPC for keycard too)
- Go to second floor
- Fight zombies
- Go to level three
- At level three collect key
- Event: Attacked by zombie and bitten

Turn into zombie
Mission is discarded and new mission is given

Mission 2: Eat NPC

Return to second floor
Fight humans
Go to first floor
Fight NPC
Bite them and eat them
Leave Front door
Roll credits

- **Houring**
  - 🟢 1 - 4
  - 🔵 4 - 8
  - 🔴 8 - 16

- **Programming**
  - Architecture
    - 🟩 ~~Set up repo with SFML 2.1 to target cross platform~~
    - 🟥 ~~Setup compile platforms for Mac, Linux and Windows~~
      - ~~http://www.ludumdare.com/compo/2013/04/24/holy-trinity-of-sfml-2-0-how-to-set-up/~~
    - 🟦 ~~Port Bit Engine and fix H/CPP stuff~~
    - 🟥 Server client architecture
  - Tile Rendering
    - 🟥 Fix world tiles in normal 2d array translated to a drawn iso view
    - 🟩 TileMap and Tile class for rendering
  - Mobile and Static Objects
    - 🟩 Mobile characters
      - Tile relationship
    - 🟩 Static objects such as walls and containers

      - Tile relationship
    - 🟩 Ability to render them on tiles
  - Sound
    - 🟩 Ability to play sound effects
    - 🟩 Ability to play music
  - Blocking Heuristics
    - 🟩 Each mobile object needs its own block heuristic to determine if it:
      - Move through target tile
      - See through target tile
  - Non-Combat Movement
    - 🟩 A* Navigation to visible or previously visible tiles
    - 🟩 Cardinal tile by tile movement
    - 🟩 Adding direction to mobile objects for rendering direction
  - LoS (Line of Sight)/ FoV (Field of View) Algorithms
    - 🟩 Ability to check line of sight between tiles (raycasting)
      - change bresenham to use tilemap indices versus pixel coordinates
    - 🟦 FoV shadowcasting
    - 🟦 To render and save visibility around character based on FoV and explored tiles
  - Doors
    - 🟩 Static objects that can be opened to reveal more area
    - 🟩 Positioned between two walls
    - 🟩 Locked and Unlocked states and Closed and Opened states
  - Mission
    - 🟩 Trigger mission state changes
      - Active
      - Completed
      - Failed
    - 🟦 UI for Mission tracking
      - Ability to track a mission objective
      - Ability to track mission steps
  - Inventory

  - 🟩 Ability to assign items to player
    - Character must be able to carry a keycard
    - Character must be able to carry a pistol
    - Character must be able to carry ammunition that is decremented with weapon use
  - 🟩 Ability to equip/unequip items on the player not in inventory
  - 🟩 Sprite rendering for items within characters (guns etc.)
  - 🟦 UI for inventory management
- NPC
  - 🟦 Ability to interact with NPC
    - Click and hold or right click to get mini-menu of interaction choices
  - 🟥 UI / Ability to read and explore dialog
    - Dialog can trigger missions
    - Dialog can trigger add/remove inventory items
- Combat
  - 🟩 Ability to die and game over
  - 🟩 Ability to engage in hostile combat with friendly NPC’s
    - Friendly combat is triggered by shooting the NPC
  - 🟦 New movement system based on combat action points
    - Visualize movement radius
    - Action Point abilities
      - Melee
      - Shoot
      - Move
      - Reload
      - Skip
  - 🟩 Statistics for combat
    - Calculate Chance to Hit based on LoS, obstacles, distance and type of gun
    - Calculate health, damage and damage mitigation (armors)
    - Calculate initiative
  - 🟩 Combat visual effects
    - Bullets

    - Blood and Damage
    - Downed enemies
  - 🟩 Turn Queue
  - 🟩 NPC’s must be able to die and turn into corpses
  - 🟩 Corpses must be lootable
    - Interaction menu for loot all items or item by item
- Unload / Load levels
  - 🟩 UI for interacting with objects (move upstairs / move downstairs)
  - 🟦 Moving back and forth from levels should persist item states
    - Keeping X adjacent levels in memory for simulation
- Advancing Combat
  - 🟩 Enemy Zombie AI
  - 🟩 Combat between more than one enemy
  - 🟩 Differentiation between friendly and enemy among NPCs
    - In Combat
    - Out of Combat
  - 🟩 Wandering AI for zombies to open or bash doors
- How NPC’s act during level transition
  - 🟩 In mid-combat if we move back to level one:
    - Zombies chase to next level via A* to level entrance
- Ability to interact with objects
  - 🟩 Must interact with desk to get key
    - Objects must be able to contain items
    - Triggers mission step
- Scripted Events
  - 🟦 Ability to code scripted actions into game objects
    - Must make zombie appear
    - Must have guarantee to bite player
    - Player will shoot zombie
    - After being bit player will turn
  - 🟩 UI for story narration

        - 🟩 Old mission is failed, new mission is given to Eat first NPC
    - Ability to transition Player Character
        - 🟩 From human to zombie
    - Ability to change friendly factions
        - ■ Becoming a zombie makes any zombies on level 2 friendly
        - ■ Becoming a zombie makes humans enemies
    - Advancing Combat
        - 🟩 Enemy Human AI
        - ■ Combat between player, enemies, and other friendly
    - Ability to Eat Downed Enemies
        - 🟩 Improve health
        - ■ Trigger mission
    - End Game
        - 🟩 Leave first floor with key
        - 🟩 Roll Credits
    - Main Menu
        - 🟩 Sound Options
        - 🟩 Graphics Options
        - 🟩 Bit Maverick Links
        - ■ Credits
        - ■ Start Game
        - 🟩 Quit

Estimates 20140208:

- 44 green = 44 - 176 hours
- 9 blue = 36 - 72 hours
- 4 red = 32 - 64
- (Best)   44 +9 +4   = 112 hours
- (Worst)   176 +72 +64   = 312 hours
- (Average) 88 +54 +48   = 190 hours (avg green = 2, avg blue = 6, avg red = 12)

Thoughts on next versions

1. Scaling to X number of floors
2. Minimap
3. More abilities and statistics
4. Random room generator
5. Random item generator
6. Bashing barriers and doors
7. Intelligence to use human items
8. How to manage levels in memory and hard disk
9. Multiplayer

Random Notes

- Blocked Hallway Problem
    - In scenarios where an NPC blocks a hallway you can avoid it being permanently blocked by swapping tile positions between the NPC and the Player
    - This should always work because the player should not be on a blocked tile and neither should the NPC
    - Having a simple boolean of is_swappable will allow certain NPC’s to purposely block hallways