CPSC 427: GAME PROPOSAL - Dungeons Inc.

Team Members

Harvir Dhaliwal	28474294
Sarah Gu	29720588
Michael Lam	89315733
Samuel Yang	70253513
Jae Hyune Yea	49834898
Al (Meiqi) Yu	85158723

Story

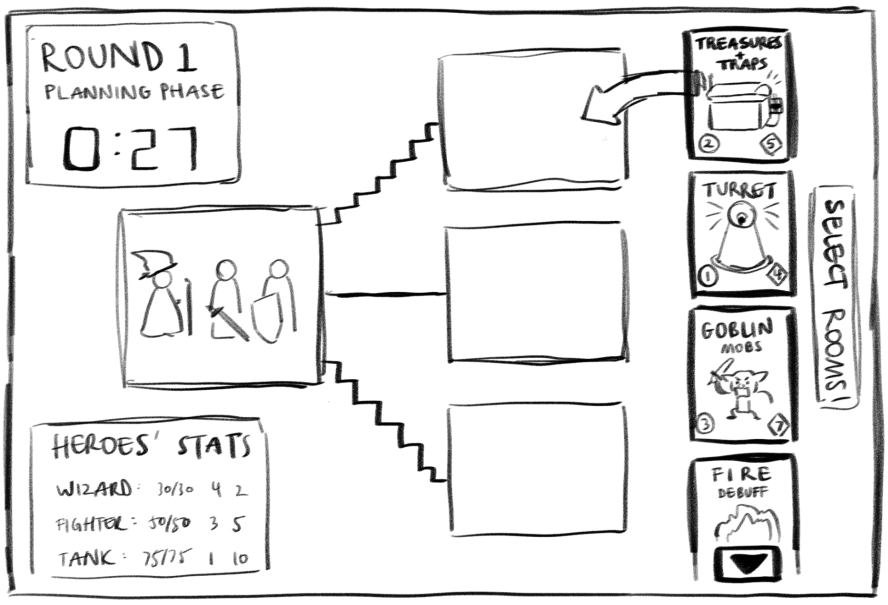
[Little Guy]'s sick and tired of serving as just another generic mob fodder in dungeons. He yearns to be one of the top dungeon masters in the world, but lacks any talent or strength to get there. What should have been another day at work turns into a chance at ascension: a deck of enchanted cards appears by an old, musty cave near his home, promising an escape from his corporate lifestyle. Play as [Little Guy] as he manipulates his newfound magic to turn a crusty cave into a renowned dungeon, using cards to design different trials and defend his new lair from the unsuspecting hero. Will he be able to evolve and climb the ladder to become the greatest dungeon master ever, or will he falter under the might of the unending kingdom of heroes?

Gameplay

The player is a wannabe dungeon master that defends their dungeon-that's-more-of-a-dank-cave from invading adventurers (NPCs controlled by AI). Using a deck of magical cards, they can draw cards to alter empty dungeon rooms to include mobs or traps. After this planning phase, the adventurers' AI will determine the most optimal room to enter next and engage in the room's combat. During this combat phase, the player can draw cards to cast spells against the enemies. A level ends when either the adventurer(s) kill the player (who is waiting in the last room), or if the player kills the adventurers without dying.

Concepts

Planning Phase



During the planning phase, players have a set amount of time to analyze the adventurers stats and place rooms into three spaces.

Combat Phase

During the combat phase, adventurers will enter one of your rooms, and the player can place spell cards to affect the flow of battle.

Technical Elements

Rendering

The basic outline of levels (room slots, background), as well as partitions for different UI components (adventurer stat boxes, cards, menu), will be rendered.

Assets

Sprites will be used for the majority of the graphics, including characters, cards, rooms, spells, buttons, text boxes, health indicators, etc. Simple animations will be used for combat effects, such as flying projectiles or particle effects (eg. blood, dust).

2D Geometry Manipulation

Transformations will be used for character movements and projectiles. Cards can be dragged and snapped into room slots to set room types and cast spells.

Gameplay Logic & AI

During the planning phase, adventurers will use a search algorithm to determine the most optimal path to reach and kill [Little Guy]. During the combat phase, the computer will use the stats of the adventurers, the room, and any casted spell cards to determine adventurer HP loss and / or death.

Physics

Cards cannot move outside of the game window and should automatically snap into a nearby room slot, which requires collision detection.

Advanced Technical Elements

Overworld Hub

Similar to the hub areas of Pikmin 4, Pokemon Mystery Dungeon, and Super Mario Galaxy, the player can interact with NPCs to learn more about the world and side narratives. This requires additional animations (character movements) and collision handling (characters encountering walls).

- Impact of Skipping: Reduced story-based content available. This would impact user experience, though mostly for those interested in the game's storyline. Core gameplay elements will not be affected by skipping this feature.
- Alternative: Remove character controllability to avoid implementing additional animations and collision handling. The semi-open overworld is replaced with a

static image of the hub area, with non-animated sprites of the NPCs. The user can click on an NPC's sprite to talk to them and read the story that way.

Animation & Cutscene Engine

To reduce asset creation and game file size, certain animations and cutscenes will be generated by a basic engine. The engine will need to handle collisions (characters encountering walls, projectiles hitting walls) and determine the desired animation based on the setting (eg. fast idle animation when an adventurer is at full health, slow idle animation when an adventurer is at low health).

- Impact of Skipping: Reduced visuals and story-based content available. This would impact user experience, as gameplay will be rather static without animations and may be less fun without an actively-developing story (in the form of cutscenes). Core gameplay elements will not be affected by skipping this feature.
- Alternative: Reduce the number of animated entities and/or the number of key-frames for each animation. State-specific animations (eg. slower animations at lower health) are removed entirely. Cutscenes will be created as an .mp4 or .mov file and played during the necessary points so that no proper engine needs to be created.

Narrative Tracker

The game will have a central narrative with 3 branching storylines. Additionally, different choices and certain quests can impact the narrative ending. A system will be needed to keep track of the player's choices and completed quests in order to provide the correct storyline cutscenes and dialogue.

- Impact of Skipping: Reduced story-based content available. This would impact user experience, though mostly for those interested in the game's storyline. Core gameplay elements will not be affected by skipping this feature.
- Alternative: Remove branching storylines and keep a singular narrative. Remove player choices and the majority of quests. For the quests that aren't removed, make them required instead of optional so that we don't have to account for quest completion affecting the storyline.

Devices

Mouse Controls (Primary)

- Click, hold, and drag cards to move them

- Click and / or hover the mouse on a card to view its effect
- Hover the mouse over an adventurer to view their stats

Keyboard Controls (Secondary)

- Arrow keys to move the camera around the map
- Esc key for pausing and accessing the game's menu
- Space bar for advancing dialogue

Tools

Itch.io

Sprite assets and music assets will be taken or modified from free-for-use game assets.

Digital Art Software (Adobe Photoshop, Procreate, Clip Studio Paint)

Sprite and other creative assets will be created using different digital art software, depending on the desired medium (eg. sprites, 2D cutscenes, character portraits).

Audio Editing Software (Audacity, Adobe Audition, sfxr.me, ChipTone)

Sound effects and background music will be created and mixed using audio editing software.

Team Management

Task Assignment & Tracking

Weekly Meetings

- At least one meeting will happen per week where we discuss the week's milestone objectives, track progress on assigned tasks, and test new features.
- Meetings are scheduled for Saturdays at 8PM via a Discord group call.

Github Issues

- Issues will be created for each milestone objective and assigned to members.

Internal Deadlines

All weekly milestone objectives should be completed 1-2 days before the weekly meeting. This is so that feature testing can be done beforehand, allowing for bug fixes and/or improvements to be suggested during the meeting. Deadlines may be adjusted according to milestone deadlines and the individual schedules of the team members.

Policies

Code Review

- Code must be reviewed by at least one person before being merged to the repo.

Coding Style Guide

- snake_case is used for variable and method names.
- CAPITALIZED_SNAKE_CASE is used for constant names.
- Open curly braces start on the same line as classes/methods/if-statements.
- One sentence is required for each function to explain its purpose.

Development Plan

Skeletal Game (Milestone 1)

Week 1 - Sept 25th to Oct 1st

- Basic card system (card and deck UIs)
- Dungeon map display and generation (room creation, basic pathway generation)
- User controls (mouse movement and clicks)
- Simple adventurer AI and stat window
    - *Basic Creative Component*: Room / Path selection (BFS algorithm)
- Damage calculations between room and adventurer stats (HP, ATK, DEF)
- One fully-designed level (level 1)

Week 2 - Oct 2nd to Oct 8th

- Textured geometry for models and UI layout
    - Models: [Little Guy], Adventurers, Rooms, Cards
    - UI: Card and Deck layouts, Room selection
- Key-frame / State interpolation (adventurer animations)
- Basic 2D transformations (card movement and placement)
    - *Basic Creative Component*: Changeable camera position
- Simple collision detection & resolution
    - Card over room bounding boxes, with snap-to-place as resolution
    - Game-space boundaries (cards do not go off-screen)

Week 3 - Oct 9th to Oct 12th

- Stress testing and code optimization (reduce lag and computation requirements)
- Basic gameplay balancing (card adjustments, possible mana gauge addition)

Minimal Playability (Milestone 2)

Week 1 - Oct 13th to Oct 22nd

- Expansion on adventurer AI and traits / stats
    - Addition of 3 unique classes with special stat spreads
    - Intermediate room / path selection (greedy algorithm)
- Expansion on card system and UI

    - New cards, outside of basic rooms and spells
    - Ability to merge cards into stronger and higher-tiered versions
- *Basic Creative Component*: Saving and loading states

Week 2 - Oct 23rd to Oct 30th

- Introductory tutorial level / guide (level 0)
- *Basic Creative Component*: Glowing card effect
- Splash screen creation (for team name)
- UI improvements (set game resolution and aspect ratio, improved visuals)
- Stress testing and code optimization (reduce lag and computation requirements)

<u>Playability (Milestone 3)</u>

Week 1 - Oct 31st to Nov 5th

- Implement level selection menu and UI
    - Allow for level replayability
- Quality management and control (handle exceptions and crashes)
- Memory optimization (reduce memory requirements, improve memory usage)
- Extreme stress testing and code optimization (reduce lag, remove bottlenecks)

Week 2 - Nov 6th to Nov 12th

- Begin creating basic animation engine (animated battles during combat phase)
    - Handles simple projectile physics and responsive entity deaths
- Memory optimization (reduce memory requirements, improve memory usage)
- Extreme stress testing and code optimization (reduce lag, remove bottlenecks)

Week 3 - Nov 12th to Nov 20th

- Begin creating character dialogue engine (during planning and combat phases)
    - Generate text and dialogue bubbles (narration, character speech)
    - Display character portraits (could add minimal speaking animations)
- Three more simple levels (level 2, 3, 4)
- Memory optimization (reduce memory requirements, improve memory usage)
- Extreme stress testing and code optimization (reduce lag, remove bottlenecks)

<u>Final Game (Milestone 4)</u>

Week 1 - Nov 21st to Nov 26th

- Begin cutscene engine (displayed before, after level start / completion)
    - Static images and / or character animations with narrative dialogue
- Add parallax backgrounds to level select menu
- Create player configurations and settings menu
    - Handles audio levels, adventurer difficulties, and resetting level data

Week 2 - Nov 27th to Dec 4th

- Finish cutscene engine
    - Dialogue choice selection and character emotive effects
- Four more fully-designed levels (level 5, 6, 7, ??? [SECRET])
- Update all sprites to have a consistent style
    - Mix of custom and free-to-use assets
    - Adds different mob sprites
- Implement main narrative ending and secret ending
- Memory optimization (reduce memory requirements, improve memory usage)
- Extreme stress testing and code optimization (reduce lag, remove bottlenecks)