Game Proposal: ALL-IN

CPSC 427 – Video Game Programming

Team: High Stakes Studio

John Man 30038483

Andy Lee 13349634

Sunny Nie 59484840

Noel Illing 65046468

Jasraj Johal 49833585

Story:

Briefly describe the overall game structure with a possible background story or motivation. Focus on the gameplay elements of the game over the background story.

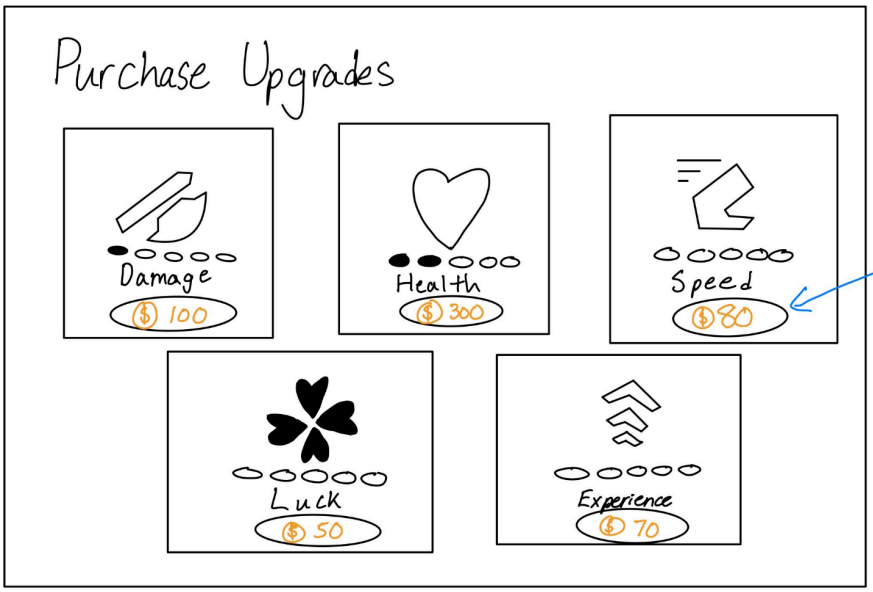
Our game takes place in a fantasy world where a once-prosperous casino city has fallen into chaos. A curse turned the city into a battleground where gambling meets the supernatural. The main character is a gambling addict who lost it all. With nothing left, he decides to seek revenge against the casino using weapons that can be found scattered throughout the casino dystopia. The casino tries to force our protagonist, Chase Damooney, into submission through supernatural effects, such as forcing the main character to gamble for buffs/debuffs, and by swarming the protagonist with casino-themed enemies. Chase Damooney has to clear rooms of enemies, sometimes encountering bosses, with the goal to survive as long as possible. Upon entering a new room, the casino will curse Chase Dammoney with random effects.

Scenes:

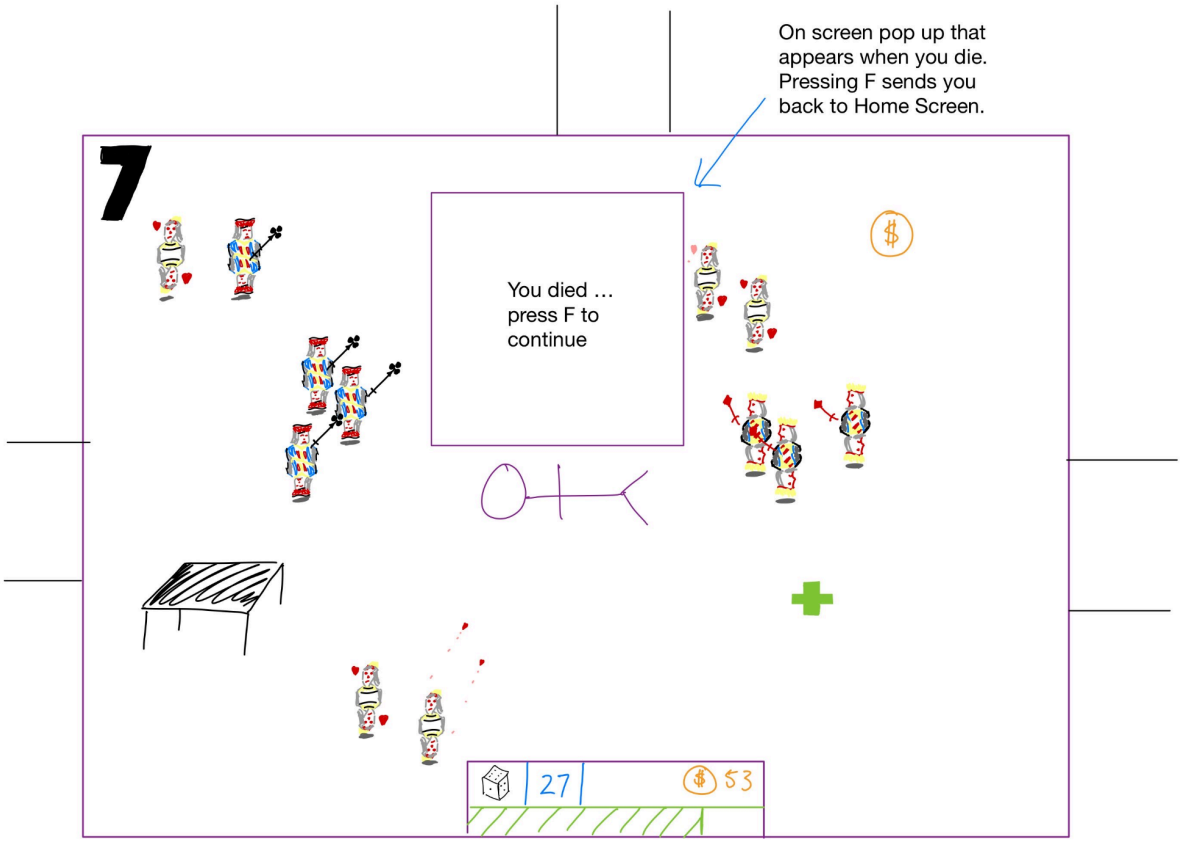
Produce basic, yet descriptive, sketches of the major game states (screens or scenes). These should be consistent with the game design elements, and help you assess the amount of work to be done. These should clearly show how players will interact with the game and what the outcomes of their interactions will be. For example, jumping onto platforms, shooting projectiles, enemy pathfinding or 'seeing' the player. This section is meant to demonstrate how the game will play and feeds into the technical and advanced technical element sections below. If taking inspiration from other games, you can include annotated screenshots that capture the gameplay elements you are planning to copy.

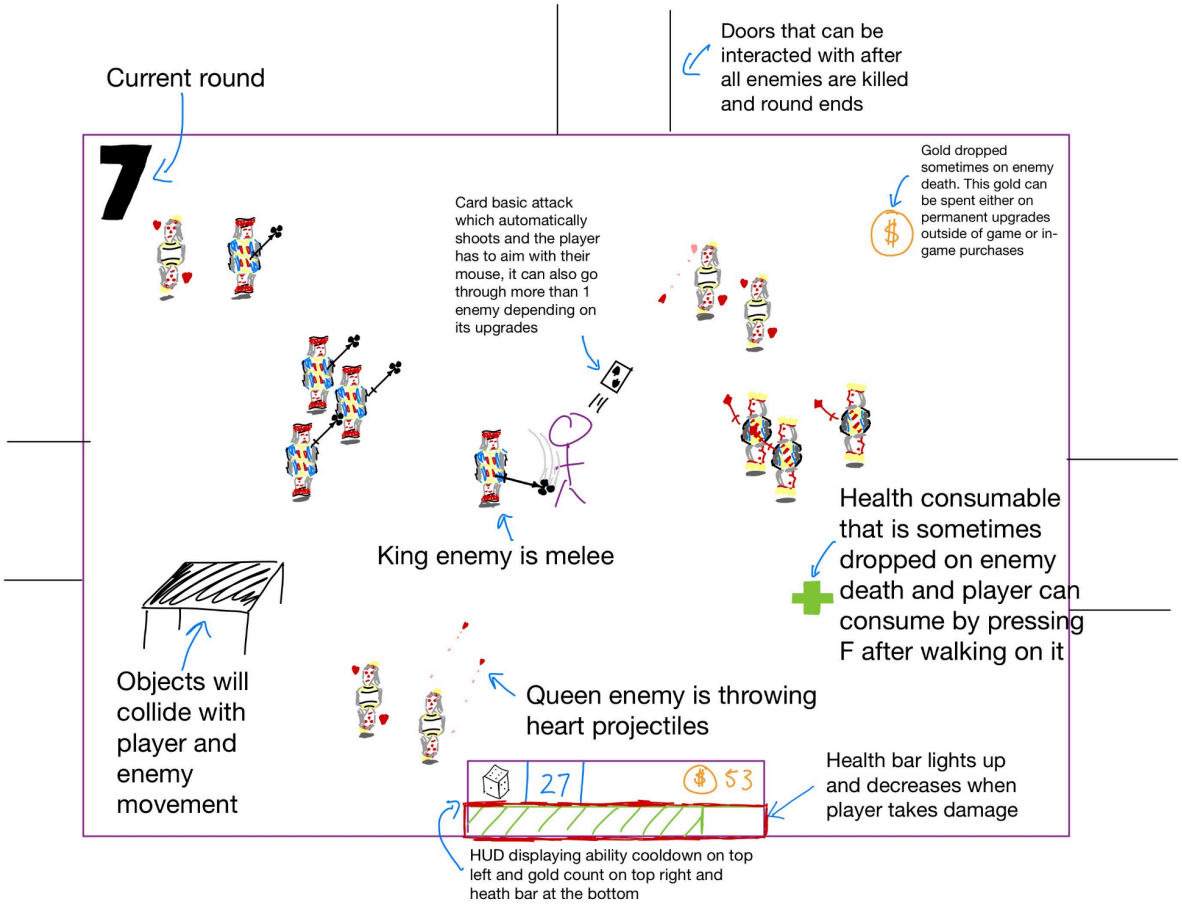
Procedurally generated rooms, at least size of screen. After defeating all of the waves in the room you will be prompted to enter a new room. Each room will have specific buffs and nerfs applied to the player, which the player can consider before choosing which door to go through. Going through the corridor/door generates the next room which the player will stay in until they either beat all the waves of enemies or if they die.

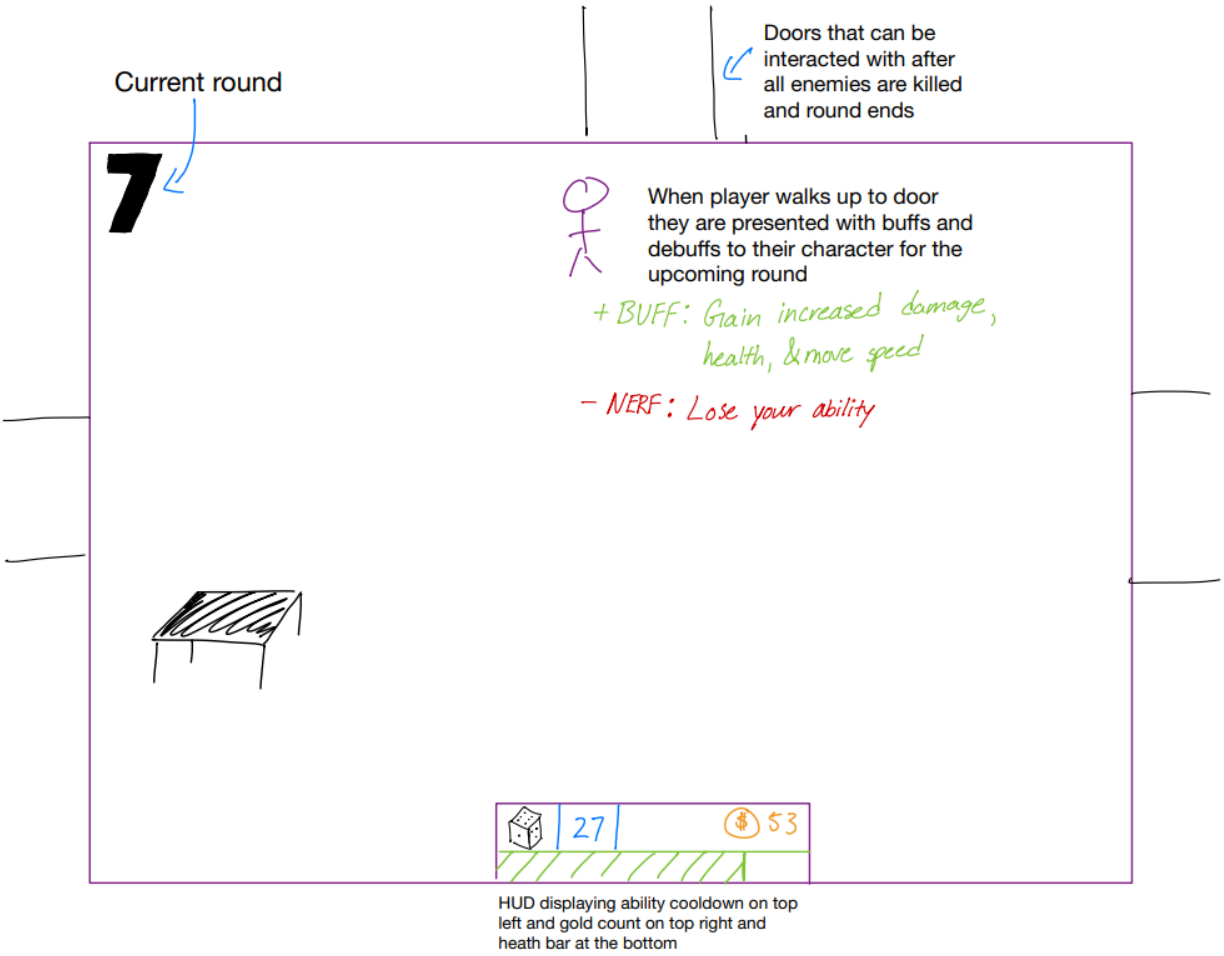
scenes needed to draw with descriptions:



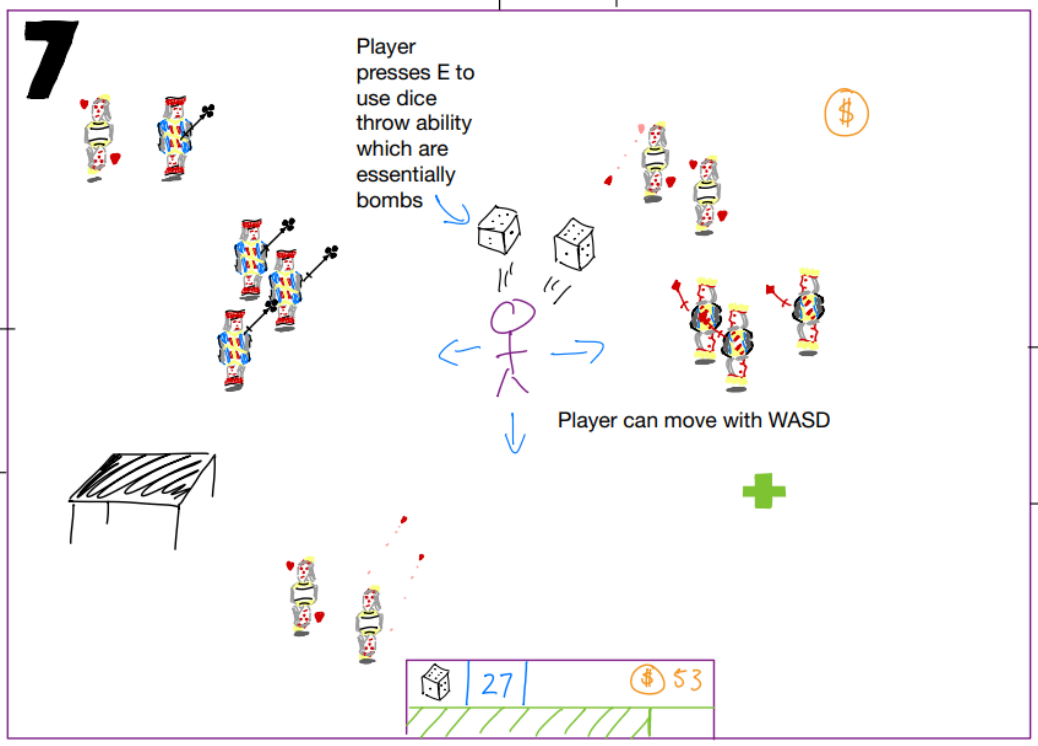
Purchase upgrades with in-game currency that persists between different games

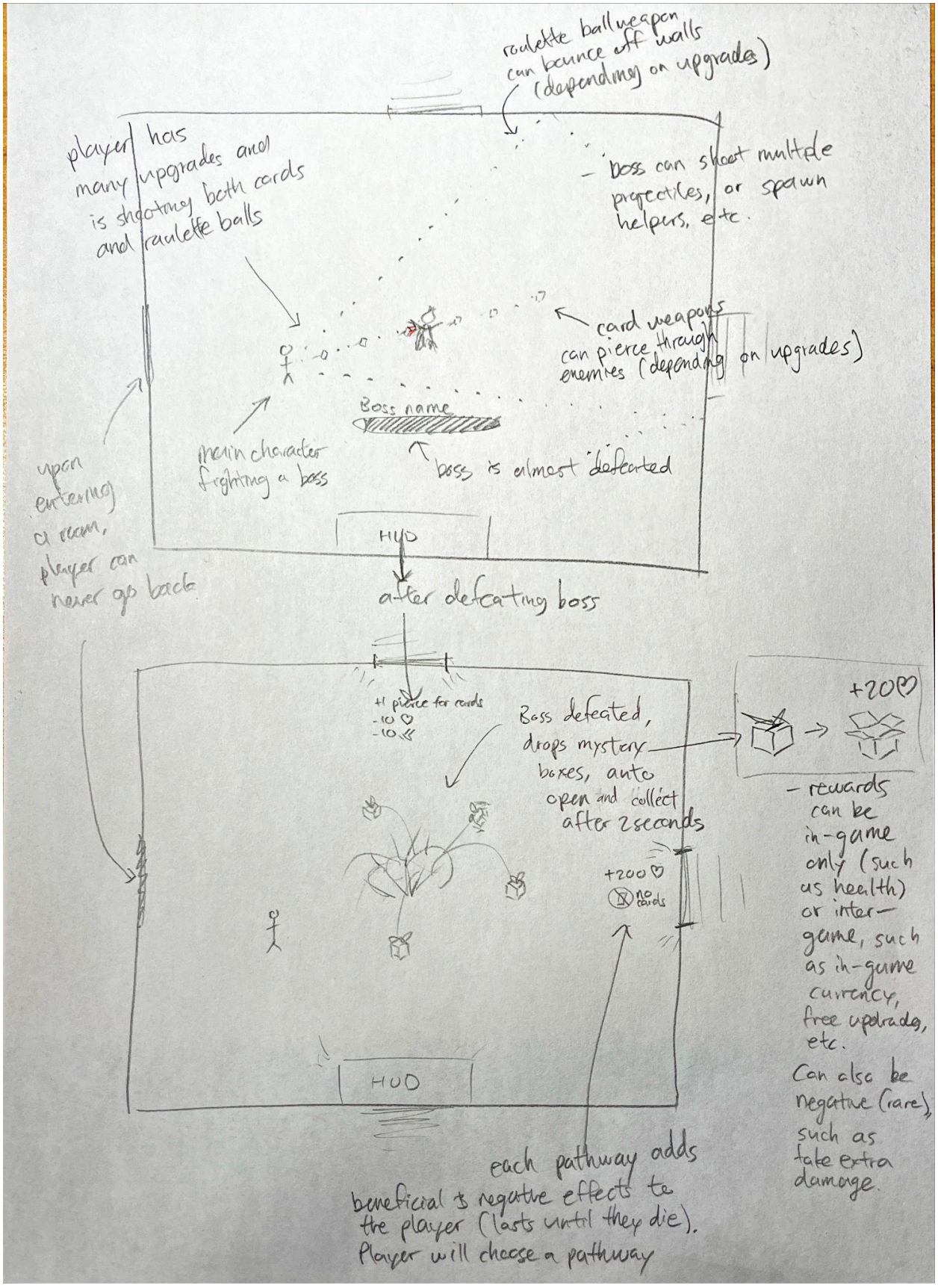

On screen pop up that appears when you die. Pressing F sends you back to Home Screen.
7
You died … press F to continue
27
53


Current round
7
Doors that can be interacted with after all enemies are killed and round ends
Gold dropped sometimes on enemy death. This gold can be spent either on permanent upgrades outside of game or in-game purchases
Card basic attack which automatically shoots and the player has to aim with their mouse, it can also go through more than 1 enemy depending on its upgrades
King enemy is melee
Health consumable that is sometimes dropped on enemy death and player can consume by pressing F after walking on it
Objects will collide with player and enemy movement
Queen enemy is throwing heart projectiles
27
53
Health bar lights up and decreases when player takes damage
HUD displaying ability cooldown on top left and gold count on top right and heath bar at the bottom



HUD displaying ability cooldown on top left and gold count on top right and heath bar at the bottom


7
Player presses E to use dice throw ability which are essentially bombs
Player can move with WASD
27
53


roulette ball weapon can bounce off walls (depending on upgrades)
player has many upgrades and is shooting both cards and roulette balls
- boss can shoot multiple projectiles, or spawn helpers, etc.
card weapons can pierce through enemies (depending on upgrades)
Boss name
upon entering a room, player can never go back
main character fighting a boss
boss is almost defeated
HUD
after defeating boss
+1 pierce for cards
-10
-10
Boss defeated, drops mystery boxes, auto open and collect after 2 seconds
+200
- rewards can be in-game only (such as health) or inter-game, such as in-game currency, free upgrade, etc.
Can also be negative (rare), such as take extra damage.
+200
no cards
HUD
each pathway adds beneficial & negative effects to the player (lasts until they die). Player will choose a pathway

- Home screen with buttons for playing, upgrading inter-game abilities, and exiting the game

    - Player defeating all of waves on round and then being prompted with gambling choice, after selecting their choice they will be prompted to go to the next room
    - Player running around room and shooting basic attack (cards), using basic ability at enemies
    - Player uses ability
    - Player shooting ball and it bouncing off walls/enemies/objects (depending on upgrades)
    - player throwing card and it goes through enemies (depending on upgrades)
    - Player shooting
        - Player can stack weapons to shoot, and each weapon can be upgraded depending on luck at the end of each room.
        - Playing Cards (go through enemies, will get blocked by walls/objects)
        - roulette ball (bounce)
        - Poker chips (no bounce)
    - Gambling mechanics
        - Upon killing a boss, the boss will drop a few mystery boxes. The mystery boxes will auto open after a few seconds. Each mystery box will have a perk, which can be beneficial or negative. For example, possible benefits include health, gold, upgrades, consumables, etc. Negatives can include taking increased damage, more enemy spawns, etc.

- Ultimate ability will wipe all enemies on the screen, with a longer cooldown. Jackpot animation, all enemies turn into coins. (3-5min cooldown)

Technical Elements:

Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI, physics.

1. Rendering
    a. Gore, Explosions, Jackpot effect
2. Assets
    a. Character/enemy sprites, audio effects
3. 2D Geometry Manipulation
    a. Collision with enemies/ walls.
4. Gameplay Logic
    a. WASD + E + F
        i. WASD for movement
            1. Shooting is done automatically as you move
        ii. E is ability (detailed in advanced elements below)
        iii. F or Left Click for interacting (spin mystery box, open door, etc)
    b. Normal weapon projectiles is done automatically as the character moves around
    c. Use mouse to aim your character when using abilities or shooting
    d. Consumables can be picked up by walking over them
    e. No inventory
    f. Enemy count per round, and health increases each round
    g. Survive as long as possible, no concept of beating a level.
    h. Upon enemy death, drops either coins or health (95%/5%)
    i.
5. AI
    a. Enemies have their movement controlled by AI. Pathing to player location. Ranged enemies will path toward players until they are within a certain radius, at which point they will stand still and start firing at the player.
6. Physics
    a. Projectiles kinematics / collision.
        i. Playing cards (players basic attack) can go through an enemy
            1. Upgrading the attack can make it go through more enemies
        ii. Roulette ball can bounce off walls (number of bounce starts at 1, can be upgraded during game)
        iii. Enemy projectiles will go through other enemies but will collide with walls and player.
        iv. Player will be able to walk through enemies but will take damage when colliding with them
        v. Projectiles will be able to fly “over” (through) objects such as tables.

7. Sound
    a. Music gets more fast paced as rounds progress
    b. Sound effects when new rounds start or bosses appear
    c. Weapon sound effects

Advanced Technical Elements:

List the more advanced and additional technical elements you intend to include in the game prioritized on likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative.

- **Swarm Behaviour**: We would like to have some form of swarm AI where waves of enemies will have intelligent behaviour to work together and not blindly follow the player.
    - For example, ranged enemies stay away from the player and space themselves out around the room, while melee enemies surround the player. They will also try to move around each other.
- Particle System: We wish for our abilities and attacks to display particles
- Precise Collision: We would like our projectiles to have precise collisions with the player and enemy hitboxes
- **Abilities** are either bought from the permanent upgrade shop or won via wheel of fortune. Each ability has its own cooldown and duration if applicable. We aim to have multiple abilities that the players can choose (time permitting) from but players are only able to equip one ability at a time :
    - Dice throw: throw a pair of loaded dice that act as grenades
    - Double down: temporarily doubles damage but also doubles damage taken
- Map Generation
    - We plan to make procedurally generated maps (rooms) using the techniques we will cover in lecture.
- Permanent Upgrades
    - purchase permanently increased base stats
        - health
        - speed
        - damage
        - luck
        - etc.
    - Use coins (in-game currency) that you collect from enemies/drops
- Sanity Level (Potential future mechanic)
    - number of lives/second chances one has
        - A player will start with full sanity which would mean they have 3 lives.
        - If a player dies in the game their sanity will go down by 1 and they will respawn.
        - With lower sanity, the player will have a lower luck chance and will encounter additional enemies and challenges
        - When the player hits 0 sanity, they lose.
    - Or alternatively

        - A status bar like health that indicates the player's sanity level, where full sanity grants slight damage multiplier, clear vision, slightly faster movement speed, faster abilities.
        - However, at low sanity, the player's vision may be impacted so they cannot see as far, may have slight damage reduction, slightly slower movement speed, and longer cooldown of abilities.
        - After a while of not being hit, the player will slowly regenerate sanity. Alternatively, the player can increase sanity with consumables.
        - The player can lose sanity by using select consumables, from being hit by select enemies/bosses
- Mana (Potential future mechanic)
    - Mechanic: Mana could be used as a currency for powerful abilities. For example, the player can trade a portion of their mana to gain powerful buffs or cast spells. This presents a trade-off: become more powerful but at the cost of losing grip on reality.
    - Gameplay Effect: Players can use mana to get ahead but risk permanent penalties or mental breakdowns if they push too far. Once mana drops too low, certain abilities may become unavailable, or the character may start misinterpreting the game's elements (e.g., friendly NPCs could appear as enemies).

Devices:

Explain which input devices you plan on supporting and how they map to in-game controls.

- Keyboard
    - WASD for movement
    - E + R for basic and ultimate ability
    - F to interact (open doors, pick and use consumables)
- Mouse
    - Cursor to aim

Tools:

Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.

We do not have any definite plans on libraries at the moment, however as we progress, we may revisit this section and look at using libraries if permitted for the following areas:

- Pathfinding / Enemy AI
- Text on screen
- Particle physics
- gdb or equivalent
- https://github.com/raizam/gamedev_libraries (repository of potentially useful libraries for C++/opengl game development)

Team management:

Identify how you will assign and track tasks and describe the internal deadlines and policies you will use to meet the goals of each milestone.

We will assign story points for each task that we have to complete. Tasks will be assigned based on preference first, and leftover tasks will be distributed as required. We will also attempt to distribute tasks in a way for all members to be able to touch many aspects of the code base as we all want to acquire experience in every part of the game developing process.

Development Plan:

Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code).

If we determine we are low on time, we will prioritize tasks to complete a barebones version of our game. For example:

- Implement just 1 weapon (probably the card throws)
- Implement just basic pathfinding AI for enemies
- Implement just one randomly generated room type, reusing the same obstacles (like tables, chairs, etc) scattered in different places.
- Implement just 1 consumable (healing consumable)
- Player movement and shooting enemies.
- Camera is functional.
- Player can enter new rooms upon clearing enemies.
- Player can interact with consumable(s).

Otherwise, if we have ample time:

- We will have multiple different types of weapons (cards, roulette balls, poker chips, etc)
- Different enemies will have different pathfinding, such as different pathfinding for melee enemies and ranged ones.
- We will have multiple different room settings
- We will implement everything outlined in Advanced Technical Elements.

Milestone 1: Skeletal Game

Who will do what: We will decide when time comes.

Week 1

- Main character sprite, background sprites (ground, wall, objects)
- Movement for main character implemented (with smooth interpolation)

- Collision with objects/wall
- Camera Controls

Week 2

- Reloadability/persist game state (upgrades)
- Have main character pivot/rotate following mouse cursor
- Have main character “fire” weapon constantly
- Have an enemy entity that moves

Milestone 2: Minimal Playability

Week 1

- Basic character animations completed/ New sprites
- Create Tutorial
- Switch to Mesh based collision
- Test Cases
- Fps Counter

Week 2

- Precise Collisions
- Working physics (Collision + Bouncing projectiles)

Milestone 3: Playability

Week 1 :

- All animations completed.
- Updated Tests
- Missing goals from previous milestone(s)
- External Integration
- Simple Path Finding

Week 2 :

- Swarm Behaviour
- Consistent game resolution and fps
- Handle memory management (no leaks or excessive memory hoarding)
- Bug fixes

Milestone 4: Final Game

Week 1 :

- Completed start to finish game

- At least 10 minutes of non-repetitive, smooth gameplay, where the player is constantly uncovering new content
- Major bugs fixed or addressed in some manner
    - No crashes or major glitches
- Consistent visual and graphics across devices
- Intuitive game design with minimal tutorials

Week 2

- All memory leaks are fixed or addressed and the game will not hog memory after exiting.
- Particle Systems implemented
- Fully implemented audio
- Bug list and future feature list established and prioritized.