Game Proposal: Cytotoxic Cataclysm

CPSC 427 - Video Game Programming

Team Members:

Keigo Lee - 11992385

Nick Zhang - 91901595

Nitahi Escolar Bach - 55607303

Payam Forouzandeh - 51597292

Ritik Keswani - 39633730

Shirley Du - 95559860

Story:

This game is an action top-down shooter (a.k.a. twin-stick shooter) in the narrative of an immunity cell. A new Cytotoxic immunity cell is born to 'rescue' a friendly cancer cell. This mission is written in its DNA.

The player as the Cytotoxic T immunity cell will fight against a bad infection in the body to deliver a cure to this friend cell that has become cancerous. The game consists of two phases where first is to fight against all the attacking bacterias on the way to finding the cure. The second phase is to deliver the cure to this friend cell after defeating the boss that is in front of it. Players should be motivated to stay active while fighting endless enemies; the different themes and items hidden in each area will motivate the player to explore the game space and ultimately finish the game by defeating the bosses.

Technical Elements:

Gameplay logic/AI:

- Player will take the initiative of the situational response according to the type of enemy he/she is facing.
- Throughout the game, different tools will be randomly generated upon investigating each section to benefit the player.
- A consistent indication will be given in the form of light intensity to help player navigate through the game space and locate the boss
- Player uses keyboard and the mouse to control the direction and weapons
- The enemies will be able to locate a player in an area and walk towards the player
- Player get stronger throughout by finding new weapons/abilities

Rendering:

- Repetitive/mono background textures
- Passive elements: weapons to be picked up by players
- Active elements: player and enemy objects

Assets (geometry, sprites, audio, etc.):

- All sprites are in pixel-art
- Background texture
- Enemies and player
- Medicine: used to cure the cancer cell
- Weapons: Sprites for the weapons that the player can take.
- Audio: background music, sound effects (hit, attack, reward)

2D geometry manipulation (transformation, collisions, etc.):

- Collisions
    - Between player/enemies
    - Between player and boundaries
    - Between melee/projectiles and player/enemy
- motion
    - Moving in in eight directions
    - 360 degree rotation for fighting
    - Dashing as an ability that can be unlocked
    - Enemies moving towards the player

Randomness:

- Everytime the game starts, the game map is randomly divided into different sections. The cure and the cancer cell are generated at random locations inside random sections of the map. Also different buff/abilities are randomly generated in different sections.
- Random enemies are spawned throughout the game. The type of the enemy and the position they are spawned are random.
- Random sacs (cyst since we are in the body) spawn in the map and breaking those will apply random (benign/malignant) effects onto the player.

Physics:

- Acceleration: player has inertia when starting to move and stopping
- Projectile movement: enemy/player being knocked back towards the direction that it gets hit
- Breaking a cyst may explode and knock back nearby objects.

Advanced Technical Elements:

Graphics:

- [advanced][3][Complex geometry][likelihood of inclusion: 100%][alternative: Only do the complex geometry for the boss enemies]:

The sword weapon will have a complex geometry matching the curve of the sword. Similarly, the 2 boss enemies are big and need complex geometry to match the shape. Skipping this item will affect the combat mechanic which is an important part of the game. Therefore, we will try to not skip it.

- [basic][1][Simple rendering effects][likelihood of inclusion: 100%][alternative: Only do the fading between sections]:
  When the player's character is hit, we want the sprite to flash red once or multiple times as a feedback for the player. If we decide to skip this, we will use a different sound effect for when the player is hit. Moreover, at the edges of each section, the background textures will fade into one another to improve the visuals.
- [advanced][5][Particle Systems][likelihood of inclusion: 80%][alternative: 2]:
  We will be using the SPARK Particle Engine library to add particle visual effects when the enemy dies and/or for each hit. Skipping this item will only affect the visual experience of the game slightly.
- [advanced][7][2D dynamic shadows][likelihood of inclusion: 70%][alternative: 6]:
  We will be using light intensity as a hint for the player to the points of interest (boss/collectibles). This light can cast shadow behind the player and/or enemies. Skipping this item will only affect the visual experience of the game slightly.
- [advanced][6][2.5(3D) lighting][likelihood of inclusion: 50%]:
  Depending on the material of the section the player is in, we may render a reflection of the characters on the ground to demonstrate the mucoid environment. Skipping this item will only affect the visual experience of the game slightly.
- [advanced][2][Parallax scrolling backgrounds][likelihood of inclusion: 10%][alternative: Use a solid dark background for areas outside the map]:
  We may use 2 extra layers of background for the area outside the boundary of the map. These layers will move at different speeds relative to the floor, creating the effect that we are suspended in one of the body organs. Skipping this item will not affect the game at all.

Physics & Simulation:

- [basic][8][Basic physics][likelihood of inclusion: 100%][alternative: Only apply momentum on a subset of moving objects]:
  Momentum will be applied on moving objects in the game. Collision between the objects (e.g. player bumping into an enemy or attacking an enemy with a weapon) will cause slowdown/knockback depending on the momentum of the objects.
- [advanced][10][Precise collisions][likelihood of inclusion: 100%][alternative: Collision with only simpler hitboxes]:
  Some objects in the game, such as the player, the weapons, and the bosses, will be represented with complex geometric assets that fit their exact shape. Collisions between these objects will be detected based on their precise geometry.
- [advanced][13][Physics-based animation][likelihood of inclusion: 80%][alternative: None. This feature is only to improve graphics and add more randomness]:

Breaking a cyst can cause smoke to be released after explosion, which will be implemented with particles.

- [advanced][12][Articulated motion][likelihood of inclusion: 50%][alternative: Make the position and geometry of the tentacle fixed on the boss]:
  The boss will have one or more arms which will attack the player. These arms will consist of several segments so the position of each segment will need to be calculated based on the positions of the player and the boss.

AI:

- [advance][16][Swarm behavior][likelihood of inclusion: 100%][alternative: 17][potential external lib:cpswarm/swarm_behaviors]:
  The same type of enemy will have the same behavior system and will aggregate together to attack the player. This allows the enemies to encirclement the player; this makes the player harder to escape and defeat them.
- [basics][14][Simple path finding][likelihood of inclusion: 40%][alternative: moving away from each other once the euclidean distance is smaller than a value][potential external lib: adityagupta1089 / Block-World-Planners; subh83/DOSL]:
  Depends on the need. A path finding algorithms can be used onto the enemy entities to avoid them from running into each other while having the same objective(ie. Chasing the player). Heuristic function might be used to speed up the search algorithm considering the large number of entities and the dynamically updated positions of them.
- [advance][17][Enemy group behavior Cooperative planning] [likelihood of inclusion: 40%][alternative: 18][potential external lib:pothitos/naxos]:
  This can work as an alternative or a supplement to [16]. By proposing a hierarchy level or tag to the enemies, this allows different types of enemies to behave according to each. For example, there is a healing type of enemy to help the other types of enemies while battling the player; or the low level enemies will make a path for the boss enemy to fight with the player.
- [advance][18][Cooperative planning][likelihood of inclusion: 10%]:
  Non-trivial collaboration may happen between different types of enemies to strategize the ways to attack the player.

Software Engineering:

- [basic][19][Reloadability][likelihood of inclusion: 100%]:
  We want the player to be able to exit the game and be able to continue from where they were when they play again. We are thinking of achieving this by saving key elements, like map state, player location/player data and other necessary information into a JSON. We will possibly use jsoncpp to make this easier.
- [basic][20][External integration][likelihood of inclusion: 80%]:
  We're thinking of using this to make particles easier in the game. We will possibly use freetype to render text, this will be used for story elements(in the case where we decide against this we will use images with text). We also might use jsoncpp for reloadability.

UI & IO:

- [basic][21][Camera Controls][likelihood of inclusion: 100%]:
- Make the camera follow the player or move it based on mouse or keyboard input.
- [basic][22][Audio Feedback][likelihood of inclusion: 100%]:
  Add audio feedback for at least three interactions in the game as well as background music with tones reflecting the journey of the game.
- [basic][22][Mouse Gestures][likelihood of inclusion: 30%]:
  Recognize gestures (patterns drawn with the mouse) to trigger jumps along an arc or other dedicated action.

Quality & UX:

- [basic][25][Game balance][likelihood of inclusion: 100%][alternative: none]:
  This will affect the overall difficulty and feel of the end product. At the very least some finetuning of game-balance will be done.
- [basic][24][Basic integrated assets][likelihood of inclusion: 90%][alternative: none]:
  Additional sprites will be added to the background to distinguish between different game sections.
- [advanced][27][Story elements][likelihood of inclusion: 80%][alternative: basic]:
  Story elements will be included in the form of text and triggered by game events as the player progresses. Alternatively, we will fall back to basic story elements.
- [advanced][26][Numerous sophisticated integrated assets][likelihood of inclusion: 70%][alternative: none]:
  Music may change in different scenarios. All art and sprites will have matching aesthetics/art style reflecting the game lore and genre.

Devices:

We will use mouse and keyboard as the primary means of controlling the character movement and aim.
Mapping:

- Keyboard ‘W’ ’A’ ‘S’ ‘D’ → character movement in 8 directions
- Mouse position relative to the location of the character → direction of character’s aim
- Mouse left click → attack
- Keyboard ‘Q’ → switch between the weapons once the secondary weapon is found
- Keyboard Space → dashing, once the ability is unlocked
- Keyboard Escape → pause, restart, a few settings like volume, and (potentially) game map

We may also add support for a standardized xbox game controller later if time permits. In this case:

- Left stick → character movement in all directions
- Right stick → character aim in all directions
- Right trigger → attack
- Right bumper → switch between the weapons once the secondary weapon is found
- ‘A’ button → dashing, once the ability is unlocked

- Start button → pause, restart, a few settings like volume, and (potentially) game map

Concepts:

Image 1: Fighting with sword in the liver.

Image 2: Swarm behavior of enemies in the urinary section.

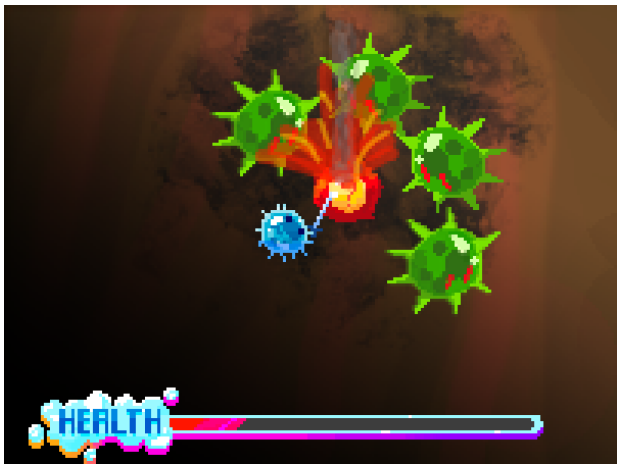

Image 3: Breaking a cyst mid-fight has random effects.

Image 4: Fighting the first boss which is guarding the cure.

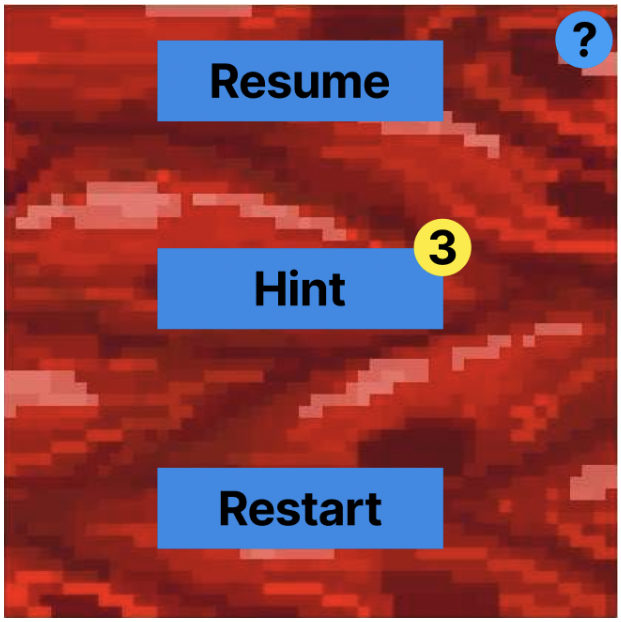

Image 5 (left): Game map. Sections are populated randomly in each round.

Image 6 (right): The game menu. This window pops up while the game is ongoing by user's choice.

Part of the art was created by Melissa (@coobeast) • Instagram photos and videos for this project(non-group member)and used with her permission

Tools:

We will likely be using the following tools and libraries

- Tools:
  - Krita → Used for drawing pixel art assets.
  - Audacity → Used to edit music/sound fx assets.
  - Microsoft Clipchamp → May be used for creating dialogues and start/end clip instead of handling in code.
  - Blender → To create complex geometry for the sword and bosses
- Libraries:
  - SPARK Particle Engine library → Used to add particle visual effects when enemy dies or is hit.
  - FreeType → If we decide to render the text and dialogue within the game, we will use this library for rendering the font.
  - swarm_behaviors → Potentially used for implementing the swarm attack on player.
  - Block-World-Planners / DOSL → Potentially used to prevent enemies from running into one another.
  - Jsoncpp to implement reloadability

Team management:

We are roughly following a Scrum approach in the team. We have a 1 hour planning meeting each week to identify and prioritize the tasks for the week, then will follow-up with each other regularly 3 times throughout the week like a standup. Throughout the week, we also have dedicated times for collaborative sessions and mutual bug fixing. Internal deadlines are different for each task and are set on the task board.
We will use Github's Project task board for tracking the status of each task and the bugs.

Development Plan:

Week 4: September 25 - **Skeletal Game**

- Layout basic class hierarchy/structure
- [Key-frame interpolation & Basic 2D transformations]: motion system
- [Keyboard/mouse control]: Player controls
- [Well-defined game-space boundaries]: Define the large circular boundary of the map

Week 5: October 2

- [Correct collision processing] Basic collision detection between simple geometry of character and boundary or enemy
- [Textured geometry]: the background and the player sprite
- [Random/coded action]: 2 sections randomly located in the map
- [Creative][basic][8][Basic physics]: Add inertia to character movement (positive and negative acceleration). Add momentum to the player and the enemies and calculate the result of collisions based on the momentum of these objects (slowdown/knockback).
- [Creative][basic][24][Basic integrated assets]: Different background sprites for each section

Week 6: October 9 - **MS1 Deadline**

- [Buffer time for potential delay]
- [Testing and Debugging] Up to Milestone 1 deadline
- [Sprite sheet animation]: Character movement animation
- [New sprite]: Adding one regular enemy and one boss sprites.

Week 7: October 16 - **Minimal Playability**

- [Player attack]: Player attacks using the gun. Each bullet hitting the enemy will reduce their health until they die.
- [New background asset]: Add music to the whole game.
- [Basic user tutorial/help]: Add instructions for movement and aim at the beginning
- [Game logic response to user input]: enemy follows the player until it gets close enough, then it does a close-range attack. Also, as the player gets close to the interest point in a section, increase the frequency of enemy spawn and vice versa.
- [Creative][advance][Complex geometry][3]: Implement complex geometry and collision for 1 boss.

Week 8: October 23 - **MS2 Deadline**

- [Creative][basic][Camera controls][21]: Implement camera following the player.
- [Start and pause menus]: Implement the pause, resume, home, and settings in the menu.
- [2 minutes of non-repetitive gameplay]
- [Buffer time for potential delay]
- [Testing and Debugging] Fixing any lag, glitch, or inconsistencies
- [Consistent game resolution]

Week 9: October 30

- [Cyst object]: Implement the cyst object and make it spawn randomly.
- [Cyst randomness]: Add 3 random logics that will happen once the cyst is broken.
- [Section interest points]: Add sprite for the interest points, then implement 3 random power-ups when the player finds these items and breaks them.
- [Creative][basic][Reloadability][19]: Save and load necessary game states

Week 10: November 6 - **Playability**

- [Creative][advanced][Precise collisions][10]: Implement complex geometry of the sword and its collision with enemies including the boss.
- [Creative][advanced][13][Physics-based animation]: Breaking a cyst may explode and knock back/kill enemies nearby. This will create a smoke afterwards which is simulated.
- [Enemy variation]: Add second type of enemy and second boss
- [Cancer cell object]: Add sprite and entity for the cancer cell with simple geometry.

Week 11: November 13 - reading break - **MS3 Deadline**

- [Creative][advanced/basic][27][Story elements]: Add dialogues and video clips to build the story.
- [Handle ANY user input]
- [Memory management]
- [>5 minutes of non-repetitive gameplay]
- [Buffer time for potential delay]
- [Testing and Debugging] Fixing any lag, glitch, or inconsistencies
- [Consistent game resolution]

Week 12: November 20

- Complete all random sections and assignment of their objects.
- [Game termination]: Identify losing or winning states of the game and finish the game.
- [Creative][basic][25][Game balance]: Adjust the properties for player and the enemies so that the game is neither too hard nor too easy to beat.
- [Creative][basic][26][Audio feedback]: Add sound effects for interactions in the game (e.g. hitting enemy, getting damaged, opening a chest, etc.)
- [Creative][advanced][7][2D dynamic shadows]: Add dynamic shadows for player and enemies based on their proximity and relative direction to points of interest
- [Creative][advanced][Swarm behavior][16]: Enemies aggregate to do swarm attack

Week 13: November 27 - **MS4 Deadline - Final Game**

- [Finish story]: Improve prologue and epilogue scenes and dialogues
- [Write report for user testing and changes done in response]
- [Tutorial for ALL necessary mechanics]
- [>10 minutes of non-repetitive gameplay]
- [Buffer time for potential delay]
- [Testing and Debugging] Fixing any lag, glitch, or inconsistencies
- [Consistent game resolution]