Game Proposal: UBZ: University of Bozos and Zombies

CPSC 427 - Video Game Programming

Team Members:

Luna Liu

Vanessa Ip

Sidharth Grover

William Gumboc

Lian Shao

Justin Koerner

Story:

Background:

Meet Bozo, a socially awkward yet extraordinarily bright university student with an unusual fascination: zombies. Bozo is the sole and founding member of the UBC Zombie Survival Club, a club yet to be officially recognized by the AMS (Associated Students of the University of British Columbia). He's not your typical college student. Rather than attending parties or social events, Bozo can often be found in a secluded, unsupervised janitor's closet in the biological sciences building. But this is no ordinary closet; it's the clubroom for his Zombie Survival Club, meticulously stocked with supplies and survival gear.

Bozo's academic excellence shines in his immunology classes, which he believes will be crucial for surviving a zombie apocalypse. Despite constant ridicule and dismissive behavior from his peers, Bozo remains steadfastly committed to his belief that a zombie apocalypse is inevitable.

The Inception:

Unexpectedly, Bozo's worst fears come true. A catastrophic virus outbreak occurs on the UBC campus, turning many student bodies into zombies. Soon after the outbreak, the authorities quarantine the campus, trapping the survivors—including Bozo—inside. For Bozo, this crisis is not an end but a beginning, an opportunity to put his years of preparation to the test.

The Journey:

Undeterred by the chaos and armed with immunology knowledge, Bozo embarks on a daring mission. He navigates through iconic UBC buildings like Buchanan, AMS Nest, ICICS, and Engineering to survive and uncover the mystery behind the sudden outbreak.

Technical Elements:

Core Technical Game Elements

- Rendering: OpenGL to create detailed side-scrolling environments that capture the UBC campus.
- Assets:
    - Geometry: 2D models of iconic UBC buildings and lecture halls.
    - Sprites: Characters, zombies, throwable objects, and obstacles.
    - Audio: A high-tempo soundtrack, along with sound effects for zombies, students, and environmental cues.
- 2D Geometry Manipulation: Platforms of varying height and motion, precise jumps, and throwables with differing arcs.
- Gameplay logic/AI: Zombies will chase the player at varying speeds, and NPCs can be thrown to slow down zombies.
- Physics: Consistent gravity, momentum, and knockback effects.
- Audio Feedback: sound effects when throwable items land, movement sounds (footsteps, door opening, etc.)

Advanced Technical Elements:

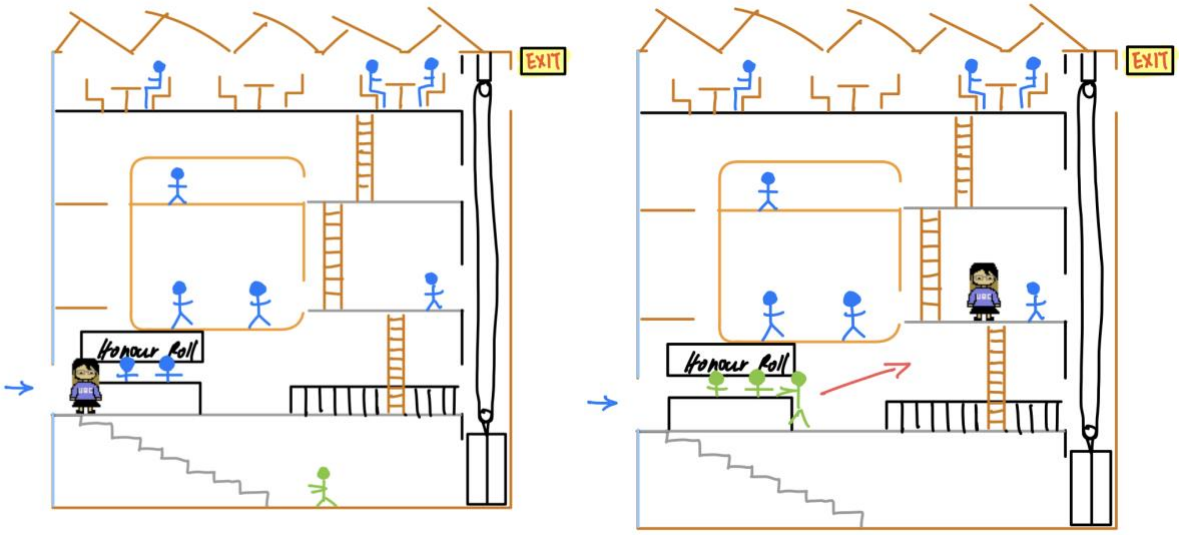
- Interactive Environment Elements:
    - Impact: Enhances gameplay strategy; players can use the environment to deal with zombies.
    - Alternative: More standard obstacles that don't require interaction.
- Advanced Zombie Types (Exploding, Jumper) with Unique AI:
    - Impact: Adds a layer of difficulty and strategy.
    - Alternative: Stick to generic zombie types.
- NPC Abilities and Characteristics:
    - Impact: NPCs have special abilities when thrown, adding a strategic layer. Saving them also gives you different tools/power ups.
    - Alternative: All NPCs have the same impact when thrown.
- Power-ups:
    - Impact: Adds gameplay layer.
    - Alternative: Make the game challenging enough without power-ups.
- Advanced Decision-making for Zombies and swarm behavior:
    - Impact: Enemy AI uses cooperation logic to take certain paths to chase down player more effectively
    - Alternative: Zombies chase the player independently of each other
- Particle Systems: For special effects like smoke, fire, and zombie disintegration.
    - Impact: exploding zombie heads, splashing water, objects on fire
    - Alternative: create effect with sprite textures
- Parallax Scrolling Backgrounds: To provide depth and immersion.
- Complex Physical Interactions: Like ropes or vines that sway when the player or zombies interact with them.
    - Impact: more versatile / complex level navigation
- Story elements
    - Simple cut scenes between level transitions
    - Impact: higher immersion, player motivation
    - Alternative: text-based story updates

Devices:

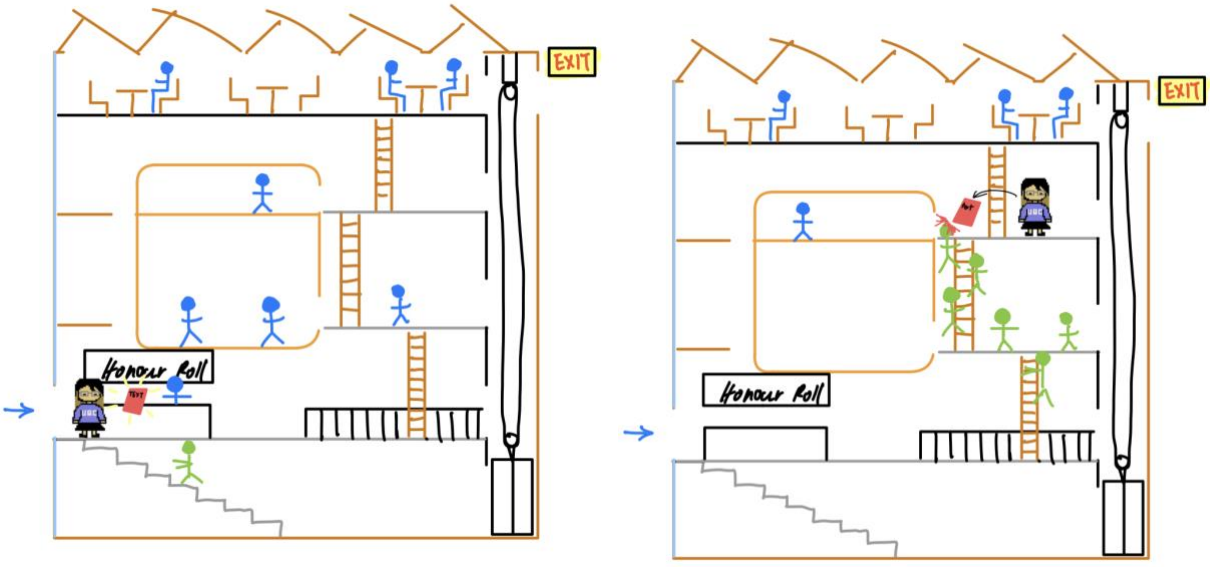
Keyboard and Mouse:

- WASD for movement
- Mouse to aim and throw objects
- Spacebar for special abilities
- Esc to open up the menu

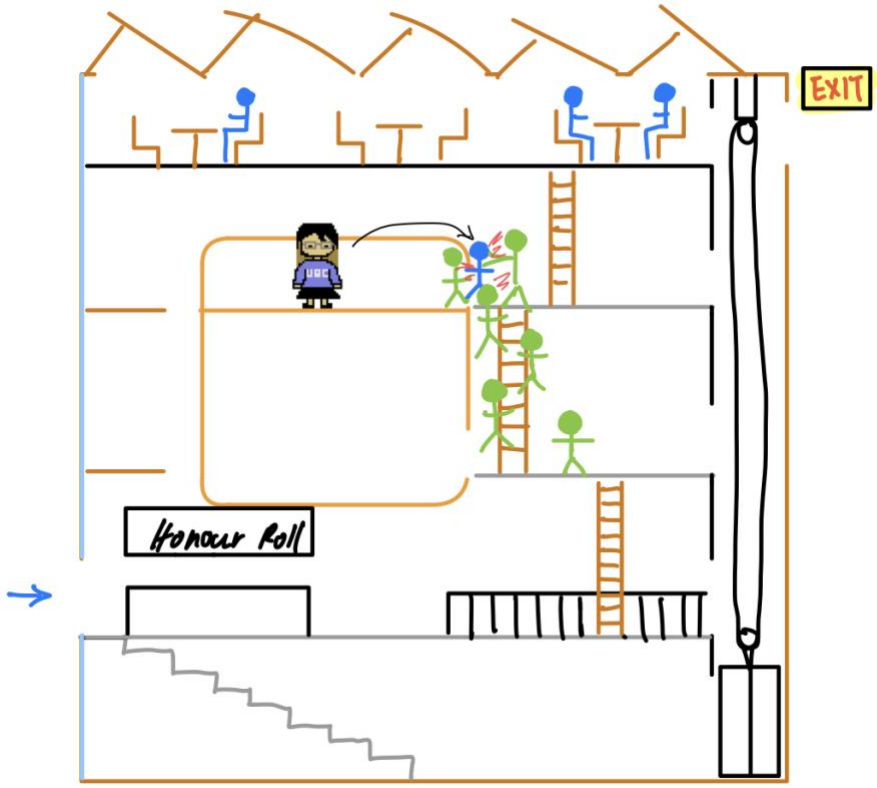
Concepts:



Player enters the map (Nest) at starting point and aims to reach the exit. Blue characters are NPCs, green characters are zombies. Each level starts with one zombie which will seek to infect the nearest human (player or NPC). Once an NPC is infected, it will do the same thing. Once the player is infected, the player loses.



There is the option to save NPCs when you approach them and there is chance that you'll receive useful throwable items but you won't be able to move for a few seconds (this may allow the zombies to get closer to you). throwing items at a zombie will slow them down/kill them.



but throwing an NPC at the zombies will draw all the zombies to it and slow them all down much more than an item, allowing you to escape.

Tools:

- OpenGL: For graphics rendering.
- GIMP: For 2D sprite design.
- Piskel: For pixel art assets
- FL Studio: For sound editing.
- GitHub Projects: For project management and task tracking.
- Box2D: Optional physics engine.

Team management:

- Task Assignment: Agile methodology with Scrum sprints.
- Internal Deadlines: Weekly sprints with retrospectives to evaluate progress and adapt plans.
- **Meeting: Team plans to meet weekly from 6:30pm to 8:00pm every Tuesday to discuss plans and progress**

Development Plan:

September 30

- Initial project setup with OpenGL
- Render a simple environment (map) with the main player and one zombie (geometry, 2D transforms, key-frame state interpolation).
- Implement simple state interpolation with character and enemy movement
- Sketch out some basic levels and puzzles (The first level will be turned into a tutorial later. classrooms to escape from, obstacles).
- Basic collision detection implemented. (The collision between player and zombie)

October 7

- Implement basic player controls (WASD for movement, mouse to shoot)
- Basic Gravity
- Randomized or hard-coded action of one or more characters/assets (5%)
- Introduce a simple throwable object.
- Implement boundaries through
- Begin work on NPC characters, basic movement, and abilities.

October 12 (Milestone 1 Deadline)

- Ensure skeletal game deliverable has continuous execution, graceful termination, and min lag

October 21

- Implement the first level with a simple escape objective.
- Create more backgrounds/maps/sprite assets
- Create animation/sprite sheet
- Decision tree
- Tutorial level
- Develop a scoring system. (creativity points)

October 28

- Introduce special zombie types (Exploders, Jumpers).
- Increase level content to a minimum of 2 minutes
- NPC characteristics - AI cooperation
- Power-up mockup.
- Implement more

October 30 (Milestone 2 Deadline)

- Ensure stability and graphics consistency

November 4

- Increase level content to a minimum of 5 minutes
- Add handling of all inputs from users
- Refine power-ups and NPC abilities.

November 11

- Implement save game state functionality.
- Finalize the main game menu.
- Basic sound effects and background music.

November 18

- Advanced zombie AI strategies.
- Implement swinging feature (physics based animation and geometric spline)

November 20 (Milestone 3 Deadline)

- Ensure 0 lag and proper memory management

November 25

- Increase level content to 10 minutes
- Implement robust tutorial
- Implement affordances
- Comprehensive bug fixing.
- Start final user testing for game balance and user experience.

December 2

- Implement additional optional features based on time and priority.
- The final round of user testing and bug fixing.
- Physics engine
- Integrate particle systems into graphics (zombie head explosion)
- Implement audio feedback for interaction

December 4th (Milestone 4 Deadline)

- Ensure stability