Game Proposal: Sugar Level: High

CPSC 427 - Video Game Programming
Winter 2 of year 21/22

Team Members:

Group 3:
Karen Lau (16524143)
John Shin (55242168)
Marcus Thian (29046612)
Galila Zhang (56781693)
Henry Zeng (22653159)
Lemuel Pranoto (36388908)

Story:

Describe the overall game structure with a possible background story or motivation.

After a terrible accident in the Willa Wonky candy laboratory, candies have come to life and gained control of the laboratory's logic-defying inventions. With their newfound power, the candy army has taken over the laboratory and set their sights on conquering the rest of the world next. While these terrifying events were taking place, children who received Silver Tickets were taking a tour of the laboratory. They are transformed into different-flavored candies when they encounter the candy army. With their tour guides and the rest of the laboratory's scientists defeated, the children must now fight to regain their human identities and destroy the candies' crazy plans. By collecting specific ingredients from different areas of the lab, the children will be able to make a potion to turn themselves back into humans.

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.

- **Rendering**: Render pixel art onto predetermined maps. Render changes to the map (e.g. terrain breaking apart) when these changes are triggered by player actions.
- **Geometric/sprite/other assets**: Playable characters, enemy characters, weapons (projectiles or melee) are sprites. Character animation (movement, actions), clickable buttons in the menu overlay (abilities, character 1/2/3…) are assets.
- **2D Geometry manipulation:** Player and enemy sprites will be translated to move around the map for each level. Collisions between sprites will be detected. Character and weapon sprites will be animated to show movement and attacks.
- **Gameplay logic:** Need logic to determine outcome of character movements and attacks, allow characters to interact with weapons and potion ingredients, and to determine when the player

team has succeeded or failed to complete the level. The order of the attacks will be determined based on the character's stats.

- **Physics:** Player and enemy characters can launch projectiles at one another, which will be affected by gravity and launch angle. Different terrains (e.g. chocolate, gummy) will affect characters' movement abilities on a particular map.
- **AI**: Need a search algorithm (BFS) and/or a basic decision tree for the enemy behavior. Enemies move to the closest player target in order to get into actionable range. Depending on stats like location and health, the enemy chooses appropriate action.
- **Audio:** Sound effects for player and enemy actions (e.g. attacks) or other gameplay events (e.g. beating a level).

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.

- Story elements
  - Animation for the storytelling with text dialogue
- Parallax scrolling background
  - Alternative: static scrolling background
  - Impact would be purely aesthetic if excluded
- Enemy group behavior cooperative planning
  - When multiple enemies are within a certain range, they will work together to "gang up" on a single player character
  - Alternative: enemies will act independently regardless of their proximity to one another
  - Gameplay would be relatively unaffected if this element is excluded; enemy teams may become easier to defeat since it would be possible for individuals to act against the team's best interests
- Complex physical interactions with the environment
  - Breaking/changing terrain during gameplay
  - Alternative: no terrain changes during gameplay
  - Gameplay would be less complex if this element is excluded, but the game would still be playable

Devices:

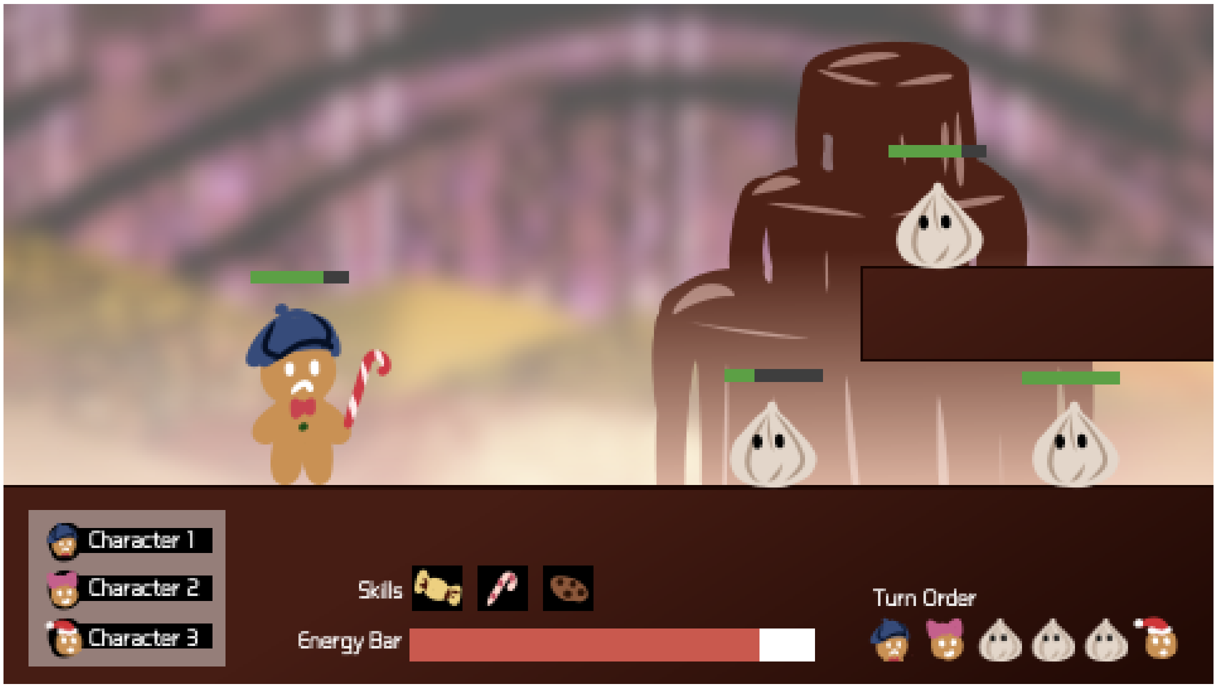
Explain which input devices you plan on supporting and how they map to in-game controls.

- Keyboard and mouse input
- W (climb up), A (move left), S (climb down), D (move right)
- Mouse left click to interact with menu overlay (like clicking on skills or characters)
- Use arrow keys to move the camera to see the other parts of the level
- Angle arrow when using a skill that fires a projectile will follow mouse placement. Left click to fire

Concepts:

Produce basic, yet descriptive, sketches of the major game states (screens). These should be consistent with the game design elements, and help you assess the amount of work to be done.

Main Screen



Tools:

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

- Pixel art: https://www.piskelapp.com/
- JSON library to read .json files: nlohmann-json
- Software to create textures: Clip Studio Paint

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.

- GitHub/Bitbucket for code repository
- Use GitHub Issues to track tasks (if repo allows)
- Possibly Trello for task management
- Communication via Discord
- Weekly meeting time: On milestone weeks, meet on the wednesday to finalize requirements to handin (or quickly fix/implement if needed)

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).

Week: January 24 - **Milestone 1 Sprint Start**

- First level design
- ECS model (Add a player character and a enemy character)
- Basic 2D rendering and rendering effects
- Basic character movement (AD)

Week: January 31 - **Skeletal Game (due on February 4, Friday) (for creative part, 2 basic or 1 advanced)**

- Basic physics (e.g. gravity)
- Enemies' movement is initially random
- Overlay for abilities (melee attacks only)
- Simple texture for one playable character and basic terrain texture (textured geometry)
- Key-frame/state interpolation of playable character movement
- Well-defined game space boundaries for the designed first level
- Correct collision processing
- Camera controls

Week: February 7 - **Milestone 2 Sprint Start**

- Sprite sheet animation
- Projectile abilities angle preview
- Simple decision tree for enemy abilities
    - One ranged attack, one melee attack
    - End turn if no player characters are in range for attacks
- Help/tutorial features for users

Week: February 14

- Basic character movement (WS)
- New integrated assets
- Basic user tutorial/help
- More level design
- Advanced physics and animation
    - Projectile attacks

Week: February 21 - **Minimal Playable Game (due on February 25, Friday)**

- Reloadability
- Debugging earlier milestones

Week: February 28 - **Milestone 3 Sprint Start**

- Debugging earlier milestones
- Additional levels/terrains

- Level selection
- Handle user input
- Parallax scrolling background

Week: March 7

- Memory management
- Real-time gameplay
- Audio effects

Week: March 14 - **Playable Game (due on March 18, Friday)**

- Debugging earlier milestones
- Consistent game resolution
- Story refinement
    - Introductory animations
    - Ending animations

Week: March 21 - **Milestone 4 Sprint Start**

- Debugging earlier milestones
- Physics-based animations
- Comprehensive tutorial levels
- Enemy group behavior

Week: March 28

- Main menu screens (main menu, level select, character select, character stats screen)
- Optimize user interaction
- Complete tutorial and additional levels
- Evaluate and improve UX

Week: April 4 - **Final Game (due on April 8, Friday)**

- Complete additional advanced features as necessary
- Debugging and minor improvements as necessary