Game Proposal: The Fast and The Furry-ous

CPSC 427 - Video Game Programming, Winter 21/22

Story:

- Background Context
    - Two groups of pets are neighbors and don't like each other, so they fight
- Major Levels
    - There will be a tutorial level but besides that
    - There are no levels; this is a turn-based game where players battle each other until one team is eliminated
    - There will be a few different maps
- Game Rules
    - Each game should only last around 5 - 10 mins
    - Team that runs out of pets first is eliminated
    - A pet is eliminated if their hp runs out
    - Each team has the same number of pets (3-5, depending on player-chosen settings)
    - Each turn one action (move, attack, defend) can be made on each team
        - Attack: Do one attack
        - Move: Move a certain distance
    - Each time has 30 seconds to choose their action
- Player Goals
    - Eliminate all of the pets on the opposing team

Technical Elements:

- Rendering
    - 2D side-view perspective
    - Two opposite environmental structures that serve as bases (houses)
    - Character action and projectile animation
    - Number of characters on each side
    - HP bar or number of lives
- Assets (Geometry, Sprites, Audio)
    - Pixel sprites for pets/units/weapons
    - Sounds for player actions (feedback)
    - Background music
    - Squares and triangles for shaping the environment/map terrain
- 2D Geometry Manipulation

    - Sprites can move, jump, and attack
- AI/Gameplay Logic
    - User can play against AI or another user
- Physics
    - Physics to calculate trajectory of ranged attacks (power, distance, direction)
    - Collision detection (projectile hitting a pet, pets not going through the terrain elements)
    - Universal gravity that affects everything (both projectiles and character movement)

Advanced Technical Elements:

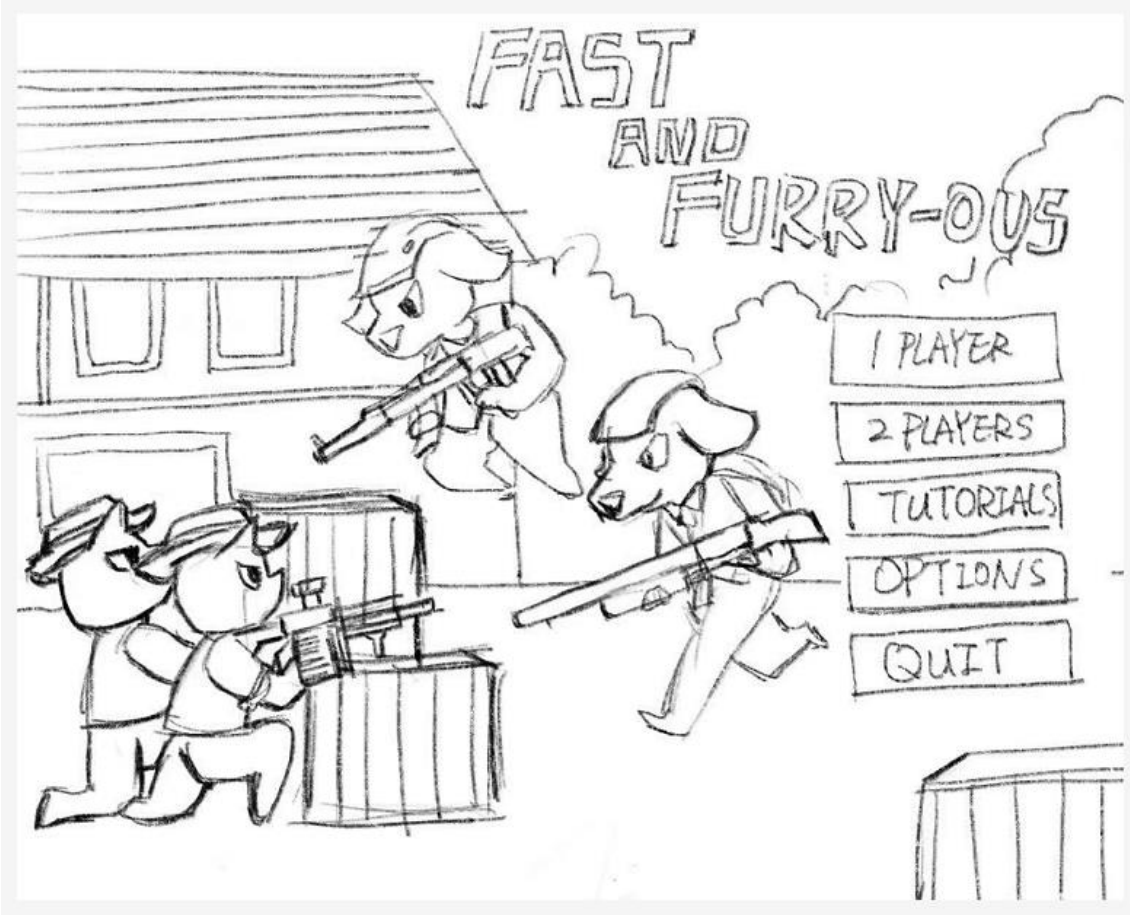
- Multiple pets to choose from for each team
    - There will be less customizability if this is skipped, but no major impact on gameplay
    - Instead of different pets, we could allow each player to choose a different colour to customize their team's base
    - If time permitting the different pets can have different weapons, with different properties i.e launch angle, destructive force etc
- Allowing pets to defend incoming attacks
    - If skipped, gameplay will be quicker
    - An alternative to this is allowing players to 'revive' pets
- Terrain or environmental changes due to collisions (base/terrain destruction)
    - There will be less variability in gameplay if this is skipped

Devices:

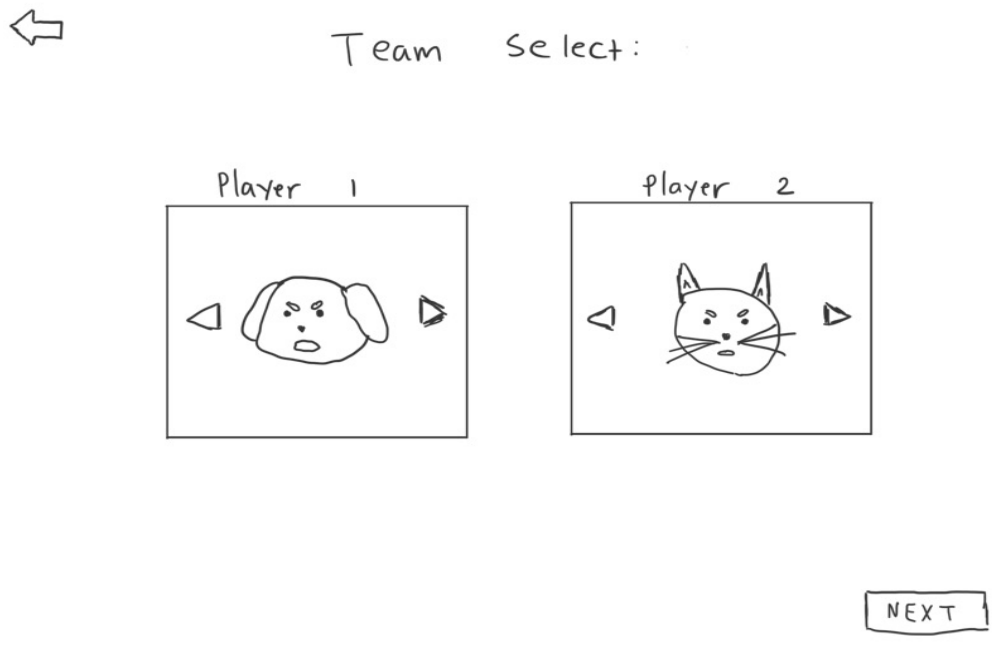
- Mouse and Keyboard
    - W/A/S/D will be used for character sprite movement in moving mode
    - M will be used for switching between moving mode and shooting mode
    - P will be used for switching between different character sprite on each team
    - W/S will be used for character aiming in shooting mode
    - The mouse will be used to aim and fire projectiles

Concepts:

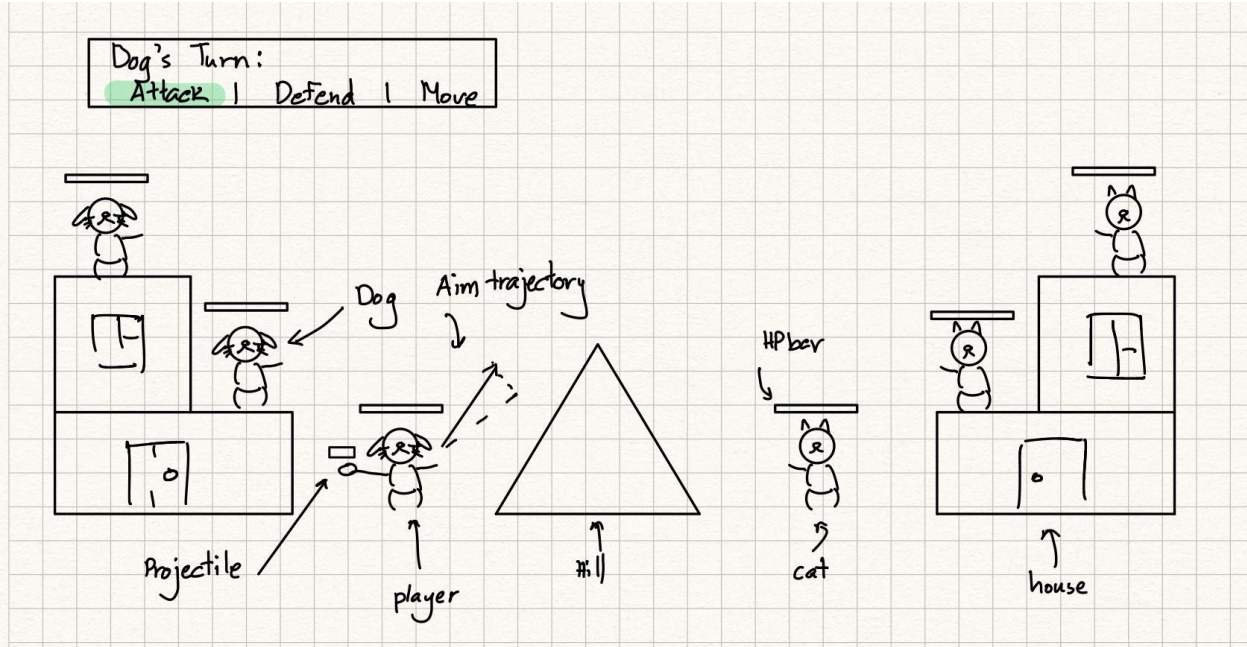
- Starting screen will include a list of choices for the player, and a game title with a game graphic in the background.
    - A list of choices includes a "1 player" button, a "2 players" button, a "tutorial" button, an "option" button, and a "quit" button



- In case of selecting the play button, the game will be directed to the team select screen for the players to choose their side.



- Once the teams are assigned to each player, the in-game screen will be rendered with 3 characters on each side.



Tools:

- No additional tools will be used

Team Management:

- Using workflow manager ClickUp to track tasks
    - Deadlines and task assignments will be tracked here, and it is expected that each team member will complete their assigned tasks on-time
    - Major deadlines are outlined below, internal deadlines will be decided on a sprint-to-sprint basis
- Using Discord for team communication

Development Plan:

Skeletal Game

Week 1

- Gameplay
    - Setting up an ECS architecture
    - Creating the player
    - Creating the blocks/terrain
- Rendering

    - Basic platforms for characters to stand on
    - Add character blocks

Week 2

- Gameplay
    - Moving characters
    - Object Collision - basic detection and avoidance
- Rendering
    - Add basic textures/sprites
- Testing & bug fixing

Minimal Playability

Week 1

- Gameplay
    - Create and implement a simple decision tree structure for user input
        - Player turn system
        - Character attacks
        - Character moves
        - Start Menu
        - Team/Character select
        - Victory conditions
- Animation
    - Create 2 sprite sheets (one for each type of pet)
- Assets
    - New sprites for pet types
    - Map & terrain textures

Week 2

- Gameplay
    - Add health and death state to characters
- Animation
    - Pets react to taking damage (sound effects)
- Assets
    - Background music

Week 3

- Gameplay
    - Implement aiming projectiles
    - Implement moving logic - players only allowed to move certain distances
- Help
    - Implement basic hints/user tutorial

- Testing & bug fixing

Advanced Game

Week 1

- Gameplay
    - Bug fixing
    - Work on stability and robustness as outlined in the doc
- Animations
    - Add fancier physics-based projectile animations
- Assets
    - Assets for all UI graphical info needs to be created
        - Turn #, player options, player turn
        - Start Screen, character select screen

Week 2

- Gameplay
    - Bug fixing
    - Work on stability and robustness as outlined in the doc
    - Make the terrain more complex using various geometric assets
- Animations
    - Add fancy idle animations
- Assets
    - Assets for weapons

Week 3

- Documentation
- Gameplay
    - Implement correct collision detection for the complex geometry
- Assets
    - Add different textures for different weapons

Final Game

Week 1

- Stability
    - Fix all previously identified bugs
- User Experience
    - Create tutorial explaining game mechanics
- Creative Components
    - Include advanced graphics (particle system, 2D dynamic shadows)
    - Implement advanced physics system (gravity)

Week 2

- Creative Components
    - Add audio feedback for all meaningful game interactions

Week 3

- Documentation
- Finish any other tasks or fix issues found in previous weeks