David Finol
Zesen Huang
David Woo

Game Tech Final Project Proposal: Team Ogre-Glider

Name of the game:

GalactiCombat

Objective of the game:

You are a spaceship, and your goal is to defeat the other spaceships, which include other players and/or pirates (AI bots). You use your energy to move and shoot the other spaceships. If you run out of energy (by moving or shooting too much), you can no longer move or shoot. To restore energy, just harvest into the minerals that are floating around the room. Be careful, though, the minerals will start zipping around, and if a mineral becomes larger than you, it will hurt you instead of providing you with energy. You lose mass whenever you get hurt (by minerals or other spaceships). Whenever you manage to hit another spaceship, as a reward, the size of your spaceship will then increase. Be the biggest spaceship in the space!

Start the game:

- To build the game:
  - ./buildit
- To make the game:
  - ./makeit
- To start a server:
  - ./GalactiCombatServer <number of players> <number of pirates>
- To run the game:
  - ./GalactiCombat

Overview:

When you start the game, you will be greeted a game description and the option to start a single player game or a multi-player game. Both games have the exact same gameplay, but the multiplayer type allows you to compete against friends! To connect to a server, type in <*hostname*,*clientname*>, where the *hostname* is the name of the machine that starts a server, *port* is the port number defined by the server, and *clientname* could be anything you want.

Details:

- Game Stages:
  - Welcome Screen with single/multi-player selection
    - If you select multi-player, you then need to type in connection info
  - Main Game
    - Move around the room
      - Running into minerals smaller than you increases the size of your ship
      - Running into minerals larger than you damages your ship, making

you flash and lose size
- Running into spaceships larger than you damages your ship, but the larger ships are unaffected
- Running into the boundaries of the space (walls) will decrease the size of your ship.

■ Every minute, the minerals undergo a Crazy Space Energy Injection, which shakes them all up
■ If your size goes below a certain level, your ship is destroyed
■ The game ends whenever there are no other spaceships left in the room
- There may be a time limit, too

○ End Game
■ Includes a summary of how you did
- If you were online, it also includes your server's leaderboard

■ If you are playing single-player game, you can either restart the game or quit the game if you lose. If you win, you can either move on to the next level or quit the game. If you are playing multi-player game, then you are returned back to the lobby

- Controls:
  - Mouse: rotate camera
  - Left-click: Shoot
  - WASD : horizontal movement
  - Left Shift: push you down
  - Spacebar: push you up
  - Escape: bring up the main menu [music options & quit]
- GUI:
  - In multiplayer games, other players' names appear over their ships.
  - Timer at the top of the screen indicates how much time has passed in the round
  - Energy bar at the bottom of the screen indicates how much running and gunning you can do
  - In the last ten seconds of every minute, a countdown is displayed at the center of the screen, warning the player about the upcoming Crazy Space Energy Injection, and showing the current leader of the game
- Music/Sound
  - The infinitely looping background music can be turned off/on after you have started a game by opening the main menu (ESC key)
  - A laser will sound when you shoot.
  - A bell chime will sound when you increase in size
  - A thunk will sound when you decrease in size

Networking:

- There are two applications used to play online: ./GalactiCombatServer and ./GalactiCombat
- ./GalactiCombatServer <number of players> <number of pirates> needs to be run before

you can play online, as it is responsible for creating the server instance that clients connect to (and it does nothing else)
    - It has no graphical interface
- ./GalactiCombat is the game client. If you select to play multiplayer, you must designate a server to join (there must be one running somewhere).
- Once you join, you will be placed in a lobby where a list of the other players will be displayed. The server will wait until all players click “Ready” to start the game.
    - The online game is exactly the same as the single player game, with the addition that all object positions are updated by the server
        - Essentially, the client sends input to the server, processes its own input, does its own physics, and then corrects its objects positions with the information specified by the server.
        - The other balls in the room would include other players on the server

Additional Extensions:

- Bug tracking:
    - At the end of the game, if you were playing online, the game may randomly crash with one of the following two errors:
        - Segmentation Fault
        - pthread_mutex_lock
    - If a TCP_Send fails, we don’t really have enough error-handling to deal with it. Either/Both server and client may have to be restarted.
- We want to extend the multiplayer aspect of the game by having players in the same instance of the game at the same time
    - We would definitely use a system of dead reckoning to do this
    - The packets from the server to the clients would focus around updating these three characteristics of each spaceship and mineral in the room:
        - Position
        - Velocity
        - Size
    - Server would use UDP packets to pass state information.
        - The clients would have to do some Lerping of both position and velocity.
- Enemy units: AI
    - Pirates
        - They will try to shoot at the players and other AI
- New Models
    - Players and pirates will be spaceships
    - Minerals
    - Bullets will be the regular sphere
- Single-player mode:
    - Levels
        - Level 1: One pirate. Destroy it to win.
        - Level 2: Three pirates. Same objective.

    - Level 3: One really fast boss pirate. Low energy consumption. Large initial size.
    - Maybe more levels.

Division of Labors:

- David Woo: Networking and Artificial Intelligence
- David Finol: Networking and Artificial Intelligence
- Zesen Huang: Class Prototypes, Models, GUIs, and Game Environment