LightBikes

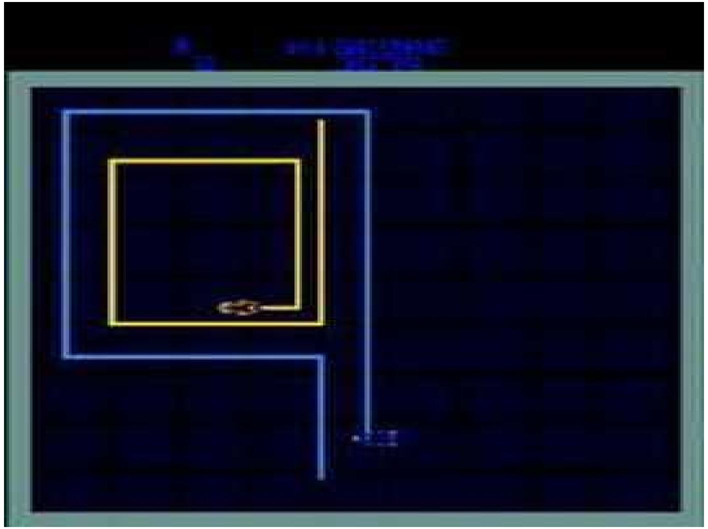
Final project presentation for RIT-ISTE-121.01

- ➔ Felice Aprile
- ➔ Justin W. Flory
- ➔ Malcolm Jones
- ➔ Timothy Endersby

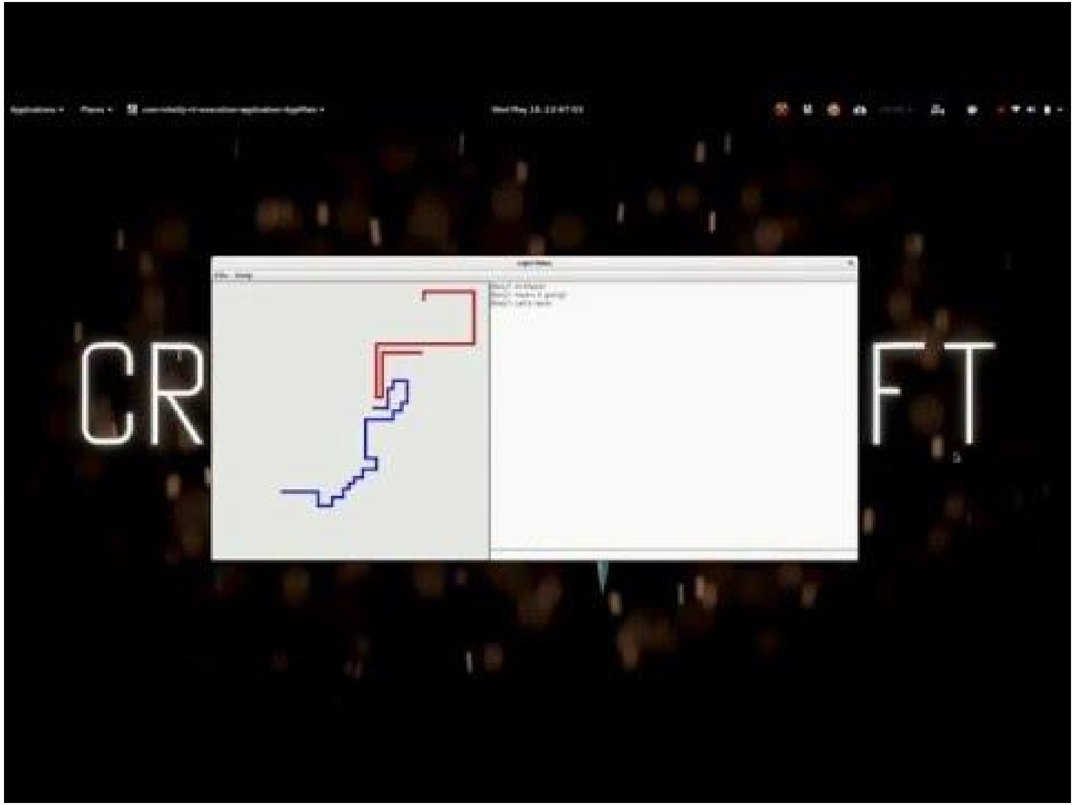
Introduction

Welcome to the world of Tron

- LightBikes based off of popular arcade game Tron (1982)
  - Two player game where players face off in arena with “bikes”
  - Players leave “walls of light” behind them as they move across arena
  - If a player hits a wall of light or arena border, they lose, game over
- tl;dr: Multiplayer Snake
- Our team replicated the original game



Excerpt from game



About project

Tools and resources used

- **V**ersion **C**ontrol **S**ystem (VCS)
  - Git
  - GitHub Desktop (desktop.github.com)
  - Atlassian SourceTree (sourcetreeapp.com)
- **I**ntegrated **D**evelopment **E**nvironment (IDE)
  - All major operating systems (Windows, OS X, Linux)
  - Different development environments
    - IntelliJ
    - Eclipse
    - JGrasp
- JavaDocs (for Java 8)

How does the code work?

- Written in Java
- Client and server are separated into packages
- Uses threading, networking, and a variety of GUI concepts learned from the course
    - Client and server GUI managed by Swing
    - Networked client connections run on their own threads
- Requires one server to host the game and two clients to connect to play the game

Best pieces

- Game logic was handled well
  - Movement for the Bike objects is simple and the aim was to send as little data to the server as possible
  - Intention to prevent network latency as much as possible, which could affect gameplay
- Even with some limitations of Swing, the GUI code is clean and efficient
  - Getting bike logic implemented inside of the GUI was difficult but was implemented well
- Designed with a flexible API
  - You could add more changes to the client and not have to update the server for handling those changes
  - Leaves project open for future changes or design decision changes

Worst pieces

- Game logic was limited to two players
  - Would have been better to leave the project flexible for more than two players (even though this was not how the original game was played)
- GUI isn’t dynamic
  - Difficult to resize the window or adjust it for different display settings
  - Future change could be to allow window resizing, but the size of the actual playing area must not be altered
- No acknowledgement that the client or server received packets / data
  - Would be better to have some kind of error sent or redundancy in place to handle failed communications that were not initiated by the player(s)
- Chat client could use more optimization and support for multiple users
  - Making it possible for someone to connect to a “game lobby” where two players are playing, but multiple people can chat (like having spectators for your match)

Faults & successes

Faults

- Gantt was used for project management but was not always followed in the beginning
  - Future projects, it would be best to keep other group members aware of their deadlines and tasks in the Gantt chart or another project management tool
  - Better plotting of tasks in the beginning would have improved workflow (i.e. using more specific tasks with smaller timeframes than larger tasks with longer timeframes)
- Merging the changes made by different group members proved difficult
  - Merging with git assisted with this, but the work for the project was done in a “modular” fashion
    - Everyone worked on specific components and then we all “tied” them together towards the end
  - Better communication and collaboration on integrating components would have made working on project smoother

Successes

- Using git allowed for easy collaboration asynchronously, letting us work on the project when we were not together without worrying about stepping on someone else’s toes
- Despite troubles with merging, the code was compartmentalized and integrated relatively well with the other parts of the project
    - Worked together to minimize the number of conflicts that would happen later
- Gantt chart became equally mapped to the work we were doing in the project closer to the end
    - Is a good representation and “deliverable” of work achieved during the span of the project

Questions?
Comments?
Concerns?