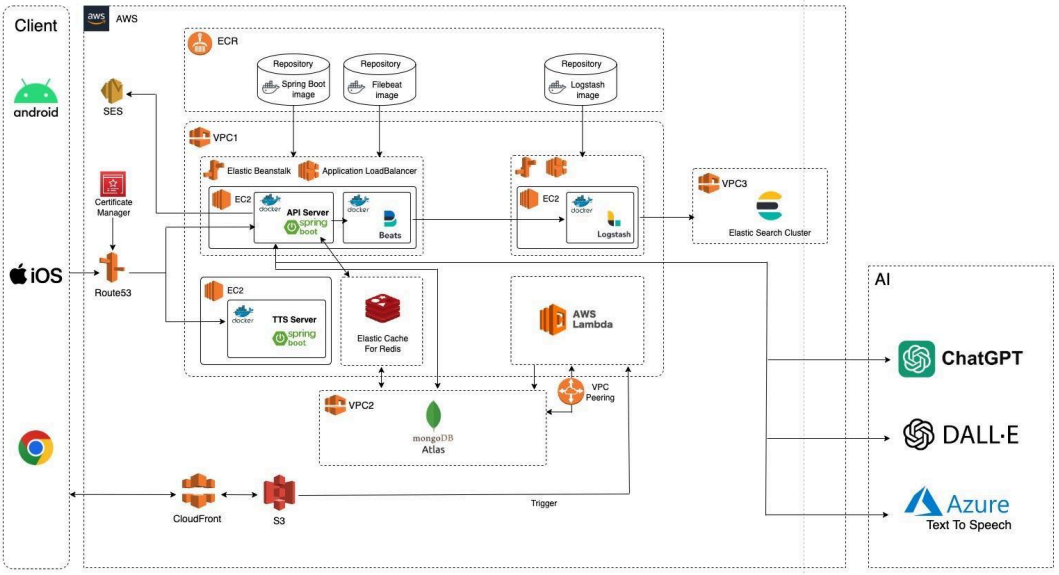
Jaehoon Choi Portfolio

Dotdotdot

OLOBO (2021.04 ~ 2024.03)

- Available for installation on app stores, including those in English-speaking regions such as the United States.
    - https://olobo.oopy.io/
        - https://youtube.com/shorts/vAr6MKOvHpg?feature=share
        - I implemented real-time rendering for character movement in the character list, character customization/shader improvements, synchronization of game design table data with UI logic, eye movement editing logic, and expanded timeline functionality to control scene components.
    - This feature automatically interprets the emotion of the inputted Text-to-Speech (TTS) or voice and expresses corresponding actions and facial expressions.
        - https://www.youtube.com/shorts/rjS0MZa4x0Q
        - https://www.youtube.com/shorts/zoilkjxCIxM?feature=share
        - I led the design for meaningful emotion types, variations in actions based on the presence of voice, animation combinations, and emotion extraction through GenAI.

- 서버 아키텍처



Designed and implemented a system where user-generated story text data is stored in MongoDB, while voice and image files are stored in S3 using pre-signed URLs. The API server is deployed on Elastic Beanstalk to ensure scalability. Subsequently, mentored a junior server programmer on system improvements, adding GenAI-related servers, and integrating signed URLs with CloudFront.

XLGames

Civilization Online (2010.08 ~ 2017.11)

- https://youtu.be/Uhy8fgxd6-M
- World


  - I integrated PhysX into the server, enabling ground detection for free fall and raycasting for collision detection.
  - I implemented the export of map data used by the client for server usage, allowing the assembly of maps. Multiple zone servers are employed to seamlessly cover the entire world (16 km x 16 km).
  - Mentored on the synchronization issues of NPC data at the boundaries of zone servers.
- Housing

  - I implemented basic building-related features, including persistent data.
  - I implemented real-time changes to the navigation mesh when constructing or destroying buildings.
    - Divide/conquer, edge swapping and node relaxation.
    - Splitting/merging of navigation mesh elements.
    - Forbidden area handling around housing.
  - Mentored on navigation mesh management and performance issues.
- Pathfinding


  - I implemented the prototype function for road construction and later provided mentoring on toll calculation based on road connections and road-based pathfinding implementation..
  - I designated a pathfinding specialist and provided mentoring on functionality improvements.
- Vehicle

- I implemented user synchronization, such as riding down on a rope from a semi-open vehicle. It supports detecting impacts on users inside non-enclosed vehicles.

Pet project

- Investigation and testing to create a character that lip syncs and moves according to user-provided data (2018)
  - https://github.com/goopymoon/goopymoon.github.io/blob/master/MMD.md
  - https://github.com/goopymoon/goopymoon.github.io/blob/master/DeepLearning.md
- Parsing and rendering Lego block data in text format using LDraw. (2017)
  - https://github.com/goopymoon/iBlock
- Windows network library for indie game development (2016)
  - https://github.com/goopymoon/RefLib