

Jerry Cai, Nick Clark, Kaden Gilbert, Gavin Sloan, Nova Solarz

What are we doing?

- Creating a 2D roleplay video game with a GUI, multiple "levels," items, enemies, background music, and a final boss.
 - Based off the original Legend of Zelda video game
- Using the incremental software development life cycle
 - Combines aspects of Waterfall and Agile

The Peeps

Team Lead - Kaden

- Created GitHub repository and Kanban project
- Completed Executive Summary, Timeline, etc.
- Coordinated with other team members to track progress

Code Lead – Gavin

- Led the game's implementation
- Coordinated with other team members to update them on the development process

Design Lead – Nick

- Created data flow diagram, problem frames, and UML diagram
- Generated ideas relevant to the game's plot, design, aesthetics, and features
- Coordinated with other team members to make sure the game's design matches what is being implemented

Documentation Lead - Nova

- Updated the physical documentation and repository as the project progressed
- Assisted with coding and created background music
- Coordinated with other team members to update them on the game's development

Security Lead – Jerry

- Completed testing on game elements and code
- Coordinated with Code Lead and Documentation Lead to make sure game components function properly

Roleplay Game Project Timeline

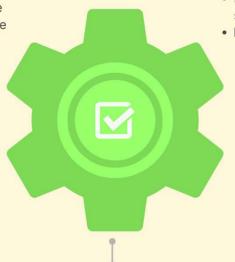
Week 2

- · Reevaluate requirements/scope
- Redesign diagrams accordingly
- Develop/test second prototype
- Milestone: Expanded prototype of roleplay game



Week One

- · Generate initial requirements/scope
- · Create initial diagrams
- Develop/test first prototype
- **Milestone:** Completed prototype of roleplay game



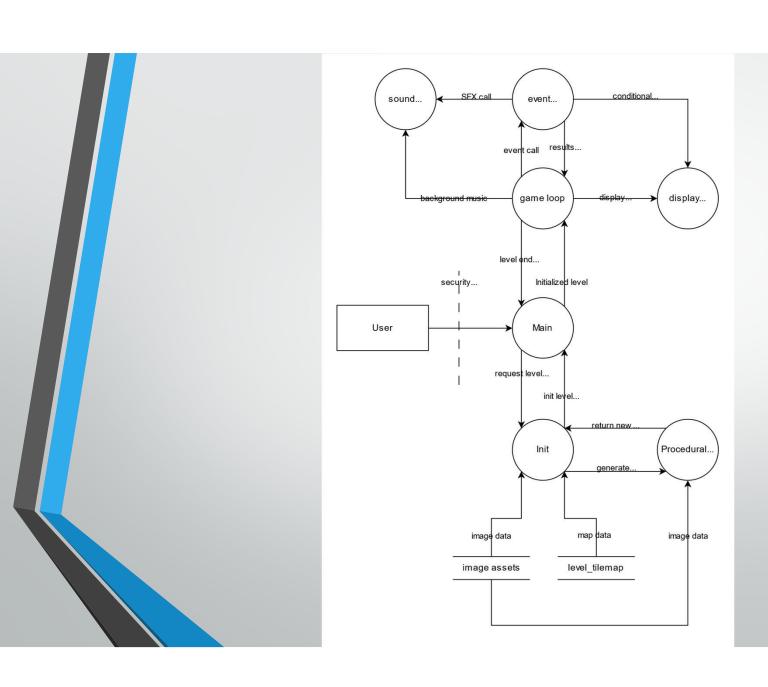
Week 3

- Finalize requirements/scope
- Finalize diagrams
- · Develop/test final prototype
- Milestone: Fully functional roleplay game

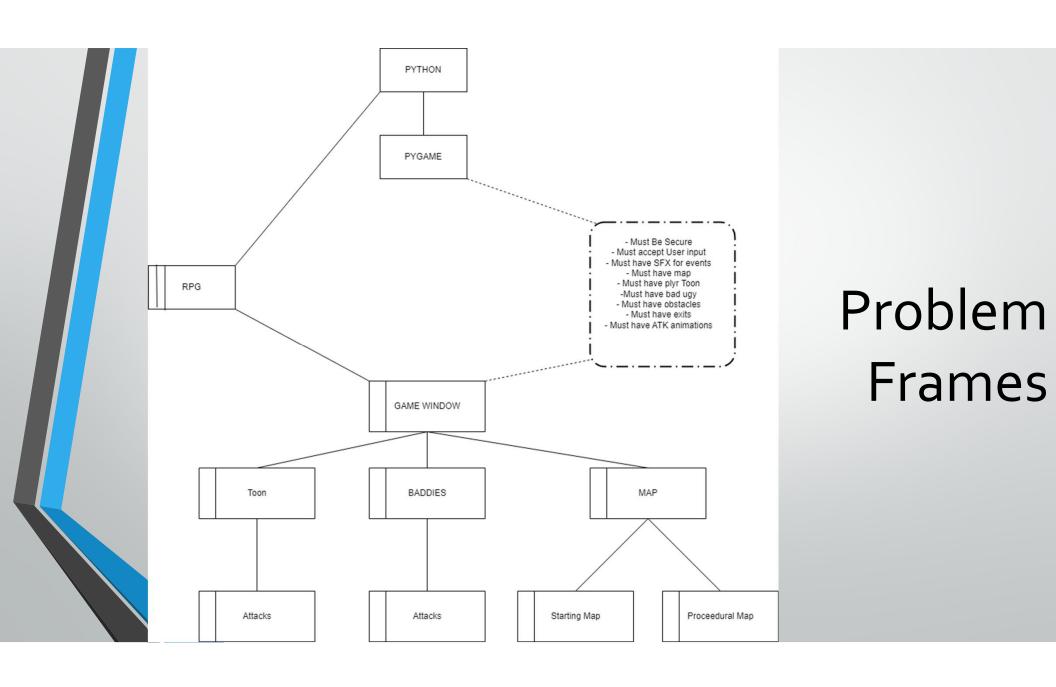
Final Product

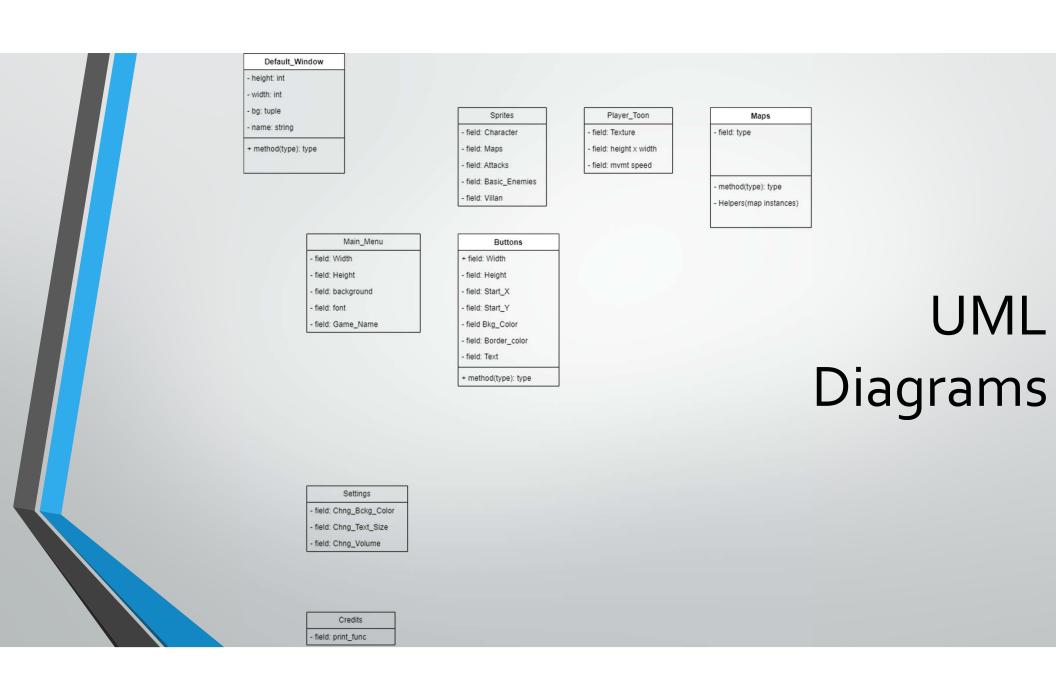
- Finalize documentation/repository for submission
- Milestone: Final product demonstration





Data Flow Diagram





Security/Testing

- Make sure we use private variables so outsiders can't access code
- Make sure we validate input so we get the correct input into whatever we need
- Make sure GUI syncs with the main code so there isn't any delay or lag between the two
- Make sure to have error checking ex: __private_variable = -1;
- Try and use try and catch exceptions to remove those pesky exceptions

Demo Time!

