

Tuxedo Amigos

SuperTux
Concrete
Architecture

Diana Balant, Matt Dixon, Ashley Drouillard,
Chris Gray, Sebastian Hoefert, Cesur Kavaslar

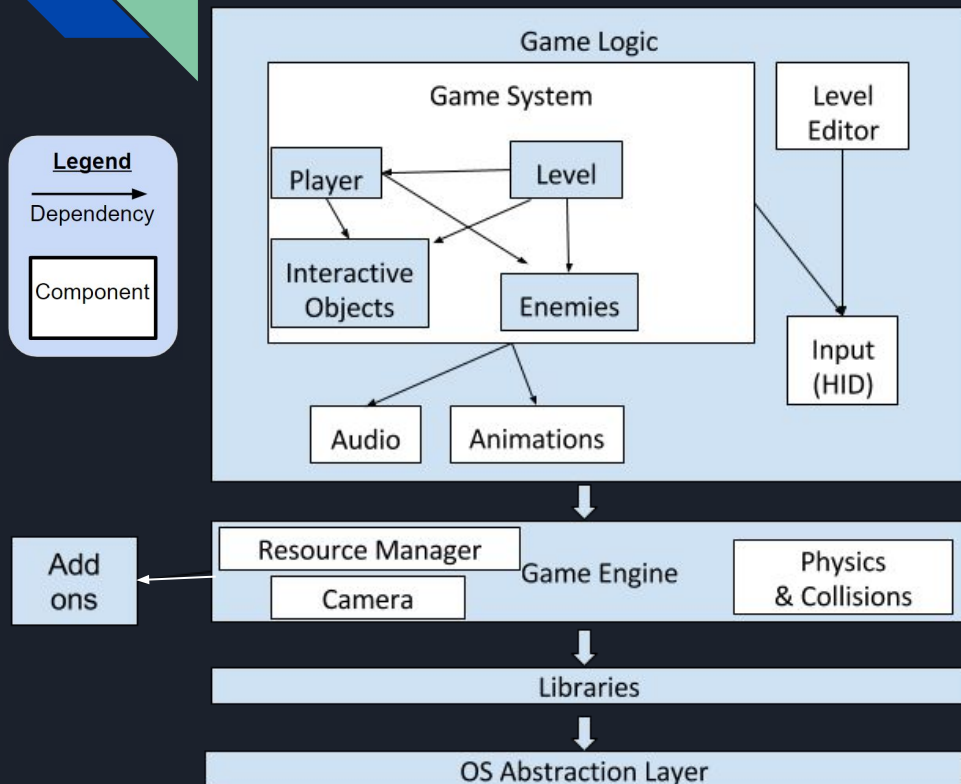




Overview

- Original Conceptual Architecture
- Revised Conceptual Architecture
- Updated Component Breakdown
- Mapping Rules
- Concrete Architecture
- Reflexion Analysis
- Sequence Diagram
- Concurrency
- Super Tux Team Issues
- Limitations & Lessons Learned
- Proposed Improvement
- Closing Statements

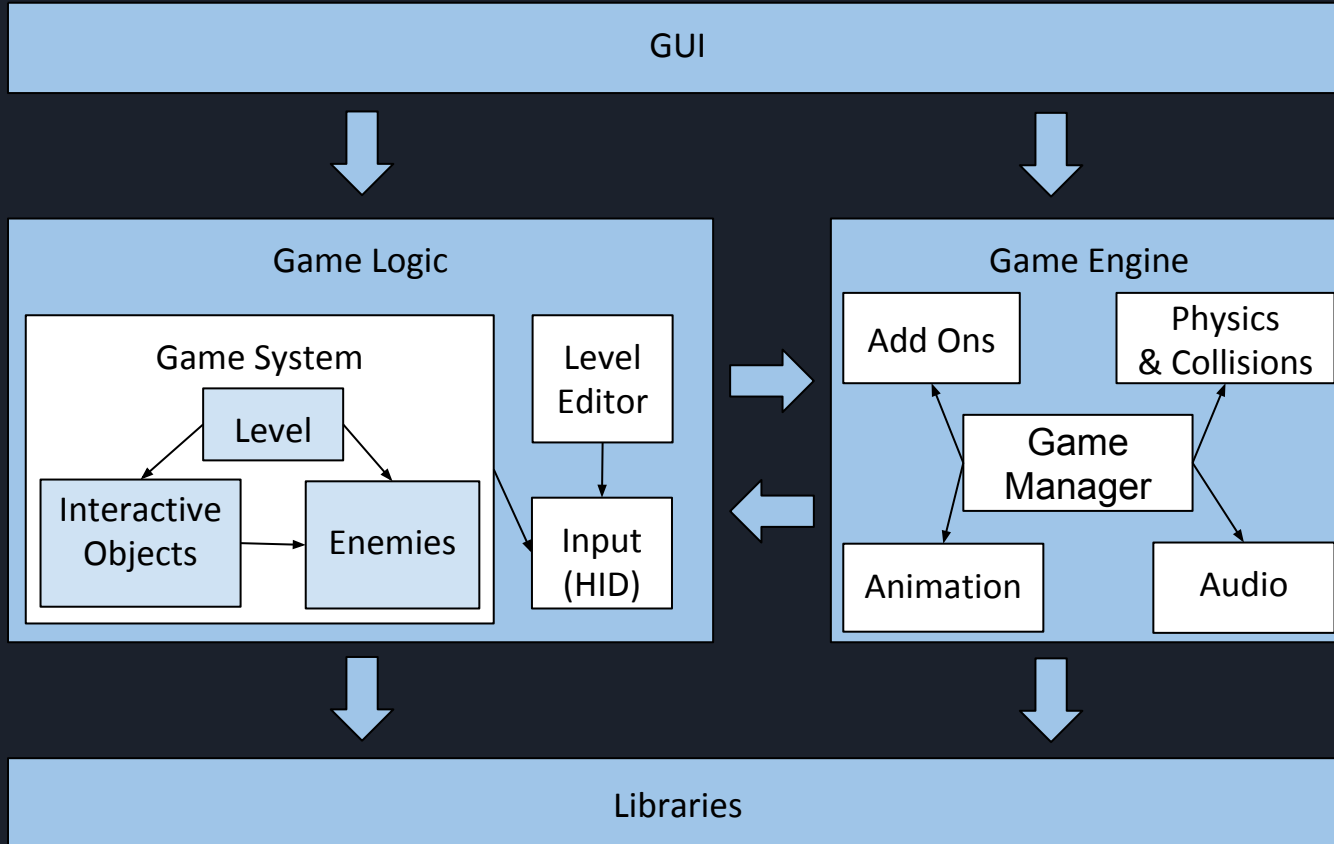
Original Conceptual Architecture



Some redundancies & missing pieces after looking at the source code:

- Audio/Animations and Resource Manager
- GUI subdirectory in the code
- Player and Interactive Objects could be merged
- Camera into Game Engine
- Removal of OS Abstraction Layer
- SuperTux Folder

Revised Conceptual Architecture



Updated Component Breakdown

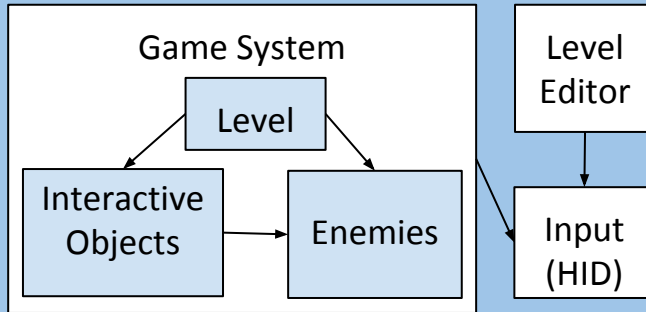
GUI

Legend

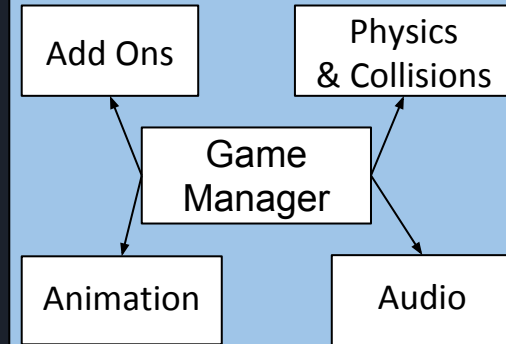
→
Dependency

Component

Game Logic



Game Engine

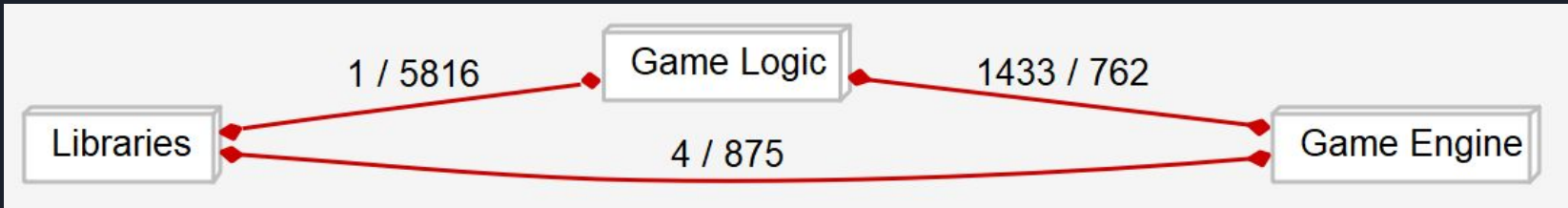
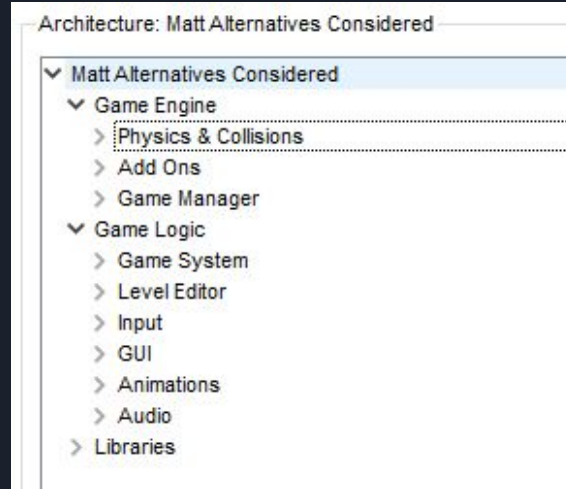


Mapping Rules

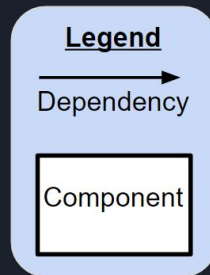
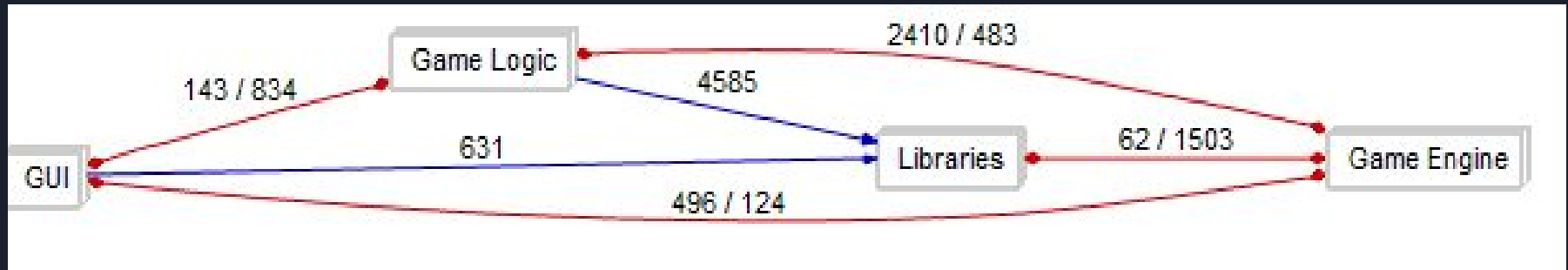
- Sort by general name
- SuperTux folder
- Go back and refine dependencies



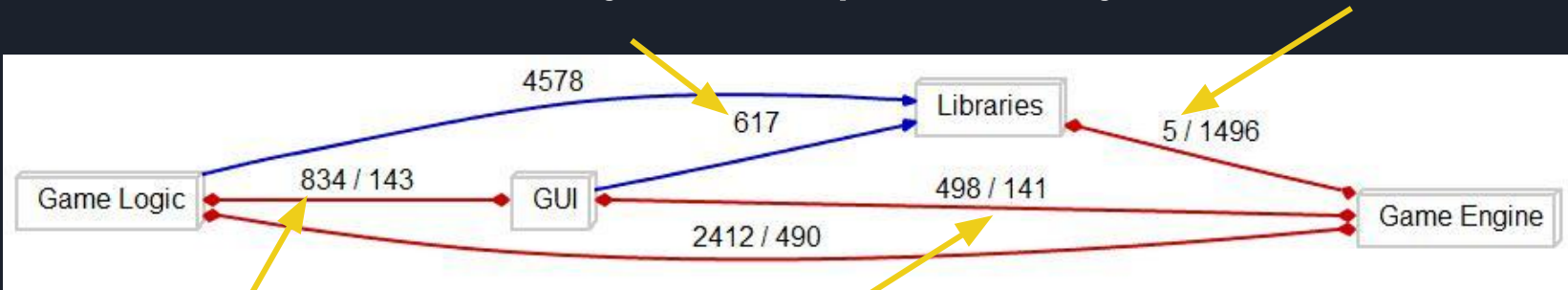
Alternatives Considered



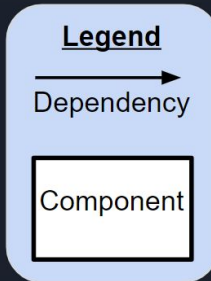
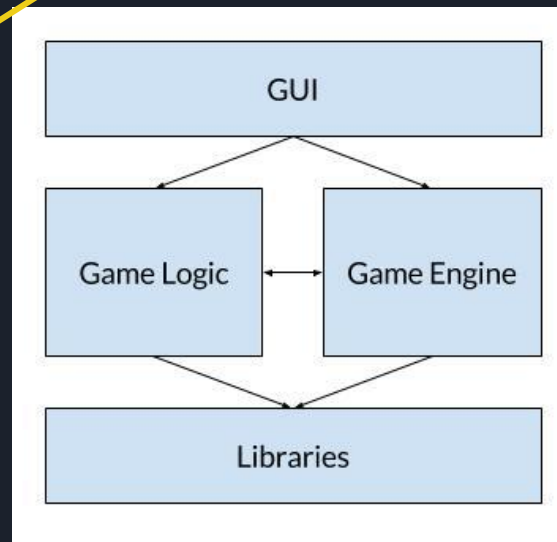
Concrete Architecture



Reflexion Analysis - Top Level Systems



- Does not present any layered architecture style
- Actually object oriented, everything is connected
- 2 way dependency between Game Logic/ Game Engine with GUI
- Game Engine has 2-way dependency with Libraries



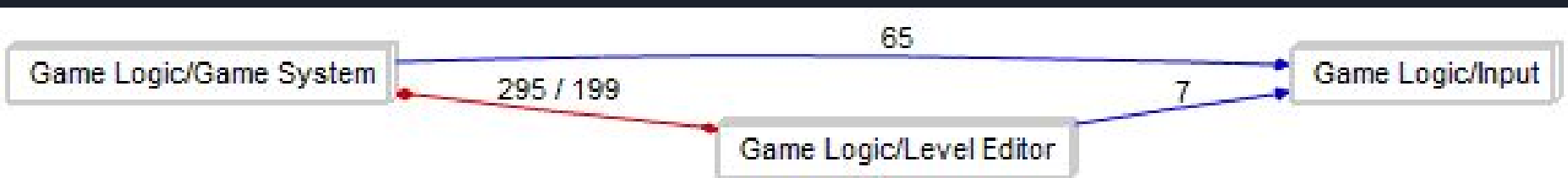
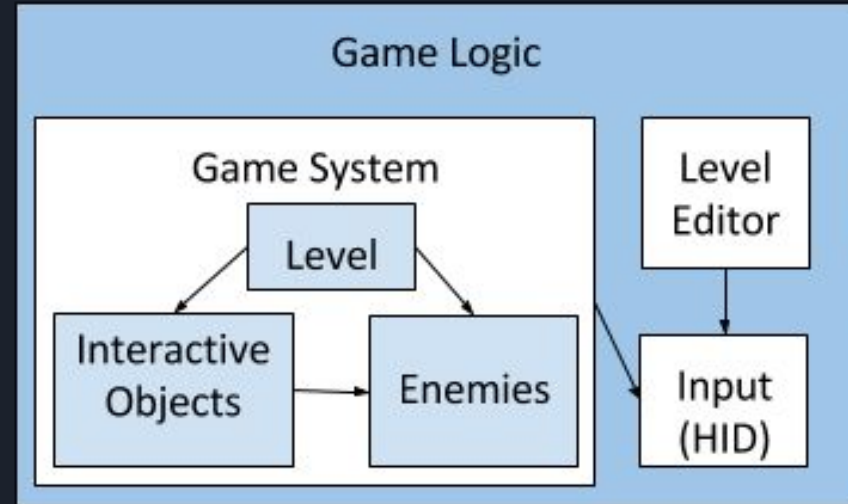
Reflexion Analysis - Game Logic

Legend

→
Dependency

Component

- Notice that Input has all incoming dependencies
- 295 incoming dependencies on Level Editor from Game System

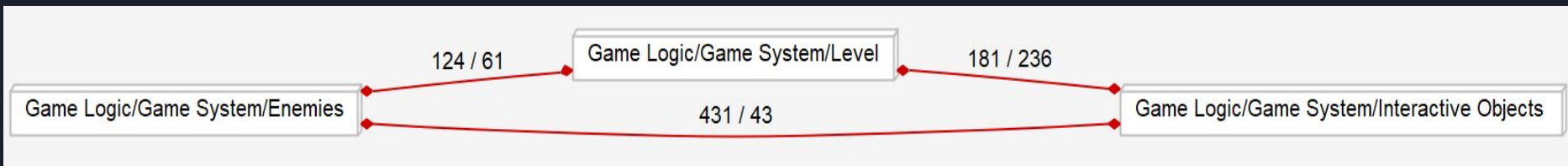
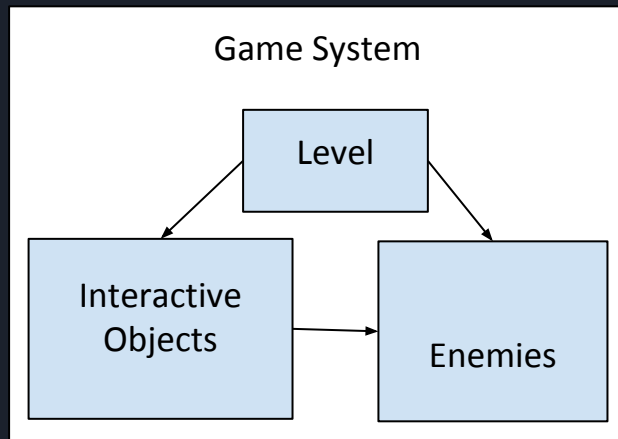


Reflexion Analysis - Game System

Legend
→ Dependency

Component

- Conceptual Architecture differs greatly from the Concrete
- Unexpected from Enemies to Level, from Enemies to Interactive Objects, and from Interactive Objects to Level
- This is caused by the object oriented nature of the code and the lack of formal documentation



Sequence Diagram - Pause Game

Legend

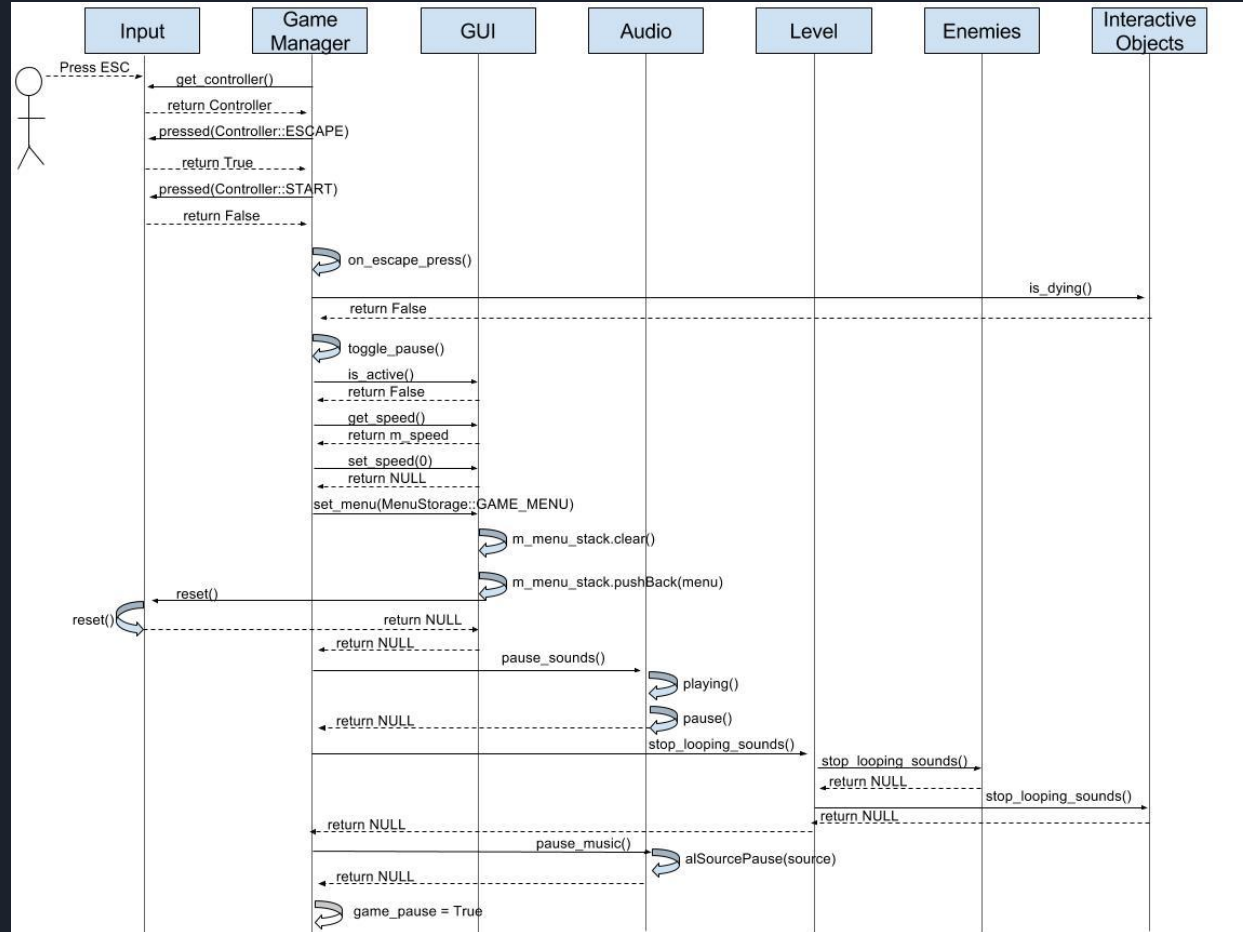
Function Call

Return Value

Component

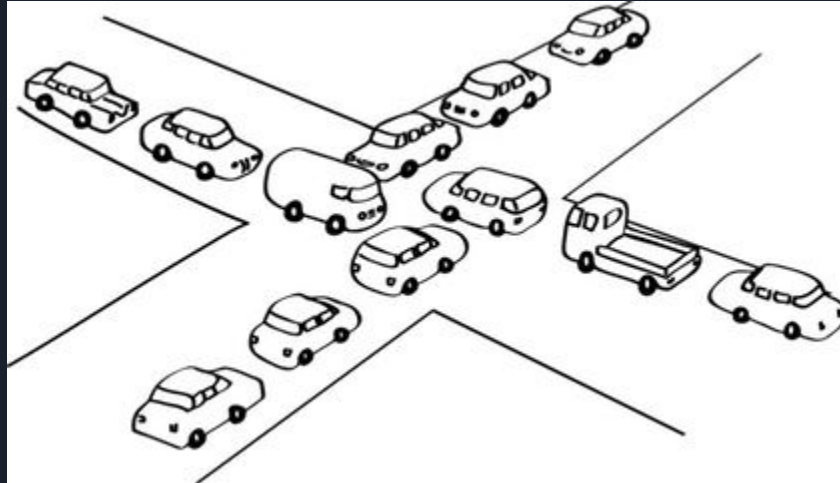
Internal
Function Call

Internal
Variable Set



Concurrency

- Allows for multiple operations to be run at once
- Enabled by Game Engine
- Game Manager calls on audio and animation



SuperTux Team Issues



Architecture Design

- Unclear Architecture: Issue for developers who want to contribute to the open source project
- High Coupling: Issue when updating a subsystem

Source Code Documentation

- Lack of comments: Issue for developers who want to contribute

Limitations & Lessons Learned

Limitations

- Working collaboratively with Understand
- Source Code Documentation

Lessons Learned

- Importance of having a clearly defined structure before software development
- Better knowledge of the Understand Tool
- It is best to document as you go



Enhancement Proposal: Score System

- Enemies Killed
- Coins Collected
- Completion Time
- Death Penalty



Closing Statements



Thank you!



Any Questions?