Tuxedo Amigos

SuperTux Enhancement Proposal

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

Overview

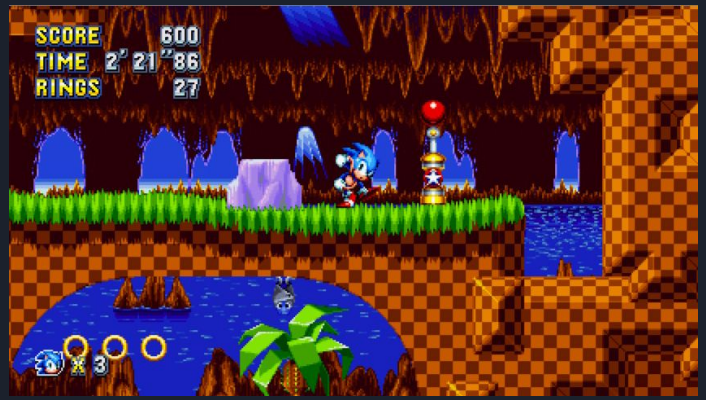
- Idea: Score System
- Motivation for Feature
- Implementations Considered
- SAAM Analysis
- Enhancement Impacts
- Potential Risks & Limitations
- Sequence Diagram
- Plans for Testing
- Concurrency
- Super Tux Team Issues
- Group Limitations & Lessons Learned

Idea: Score System

- Coins: 500 points per coin collected
- Badguys: 750 points per badguy killed
- Secrets: 1000 points per secret unlocked
- Time: Score is doubled if level is completed within the target time
- Score progress resets if the player dies during the level
- Score progress is saved at the checkpoint

Motivation for Feature

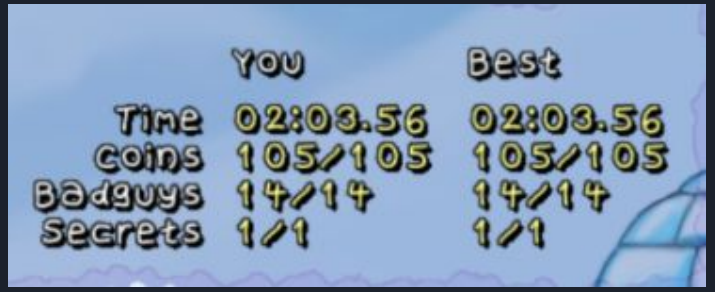
- Commonplace in other platformers
- Encourages players to replay levels
- Incentivises exploration
- Enables competition between players



Implementations Considered

Implementation 1:

- Calculate score at end of level using accumulated stats



Implementation 2:

- Continually track, calculate, and display score during the level

SAAM Analysis

Stakeholders & Related NFRs

User / Player	Developers
- Usability - Performance	- Maintainability - Evolvability / Modifiability - Testability

SAAM Analysis

Implementation One: Only calculate the score at the checkpoint and/or the end of the level.

Pros (+)	Cons (-)
Simple to Implement ● Much more simple to add to the game **Performance** ● Less frequent calculations **Maintainability** ● Easier to go back and correct errors or evolve the game in the future **Reliability & Testability** ● Less chance of making an error ● Fewer different ways to test	**Usability** ● No impact to the user throughout game play, however the user does not see the score during play and will only see it at the end

SAAM Analysis

Implementation Two: Update the score consistently throughout the level.

Pros (+)	Cons (-)
Usability ● Slightly more user friendly because the score would be displayed as the user plays **Simple to Implement** ● Rather straightforward to implement - would only require a few extra lines of code to existing functions.	**Performance** ● Potential to slow down game play because operations occur more frequently **Maintainability** ● More difficult to modify because code will be distributed across multiple locations **Reliability & Testability** ● Variety of locations increases the opportunity for small errors or miscalculations ● Also decreases our chances of finding the errors and makes testing more difficult

Impacts of Enhancement

- No changes to architecture or design patterns
- All changes occur in pre-existing files
- No new dependencies created

Files modified:

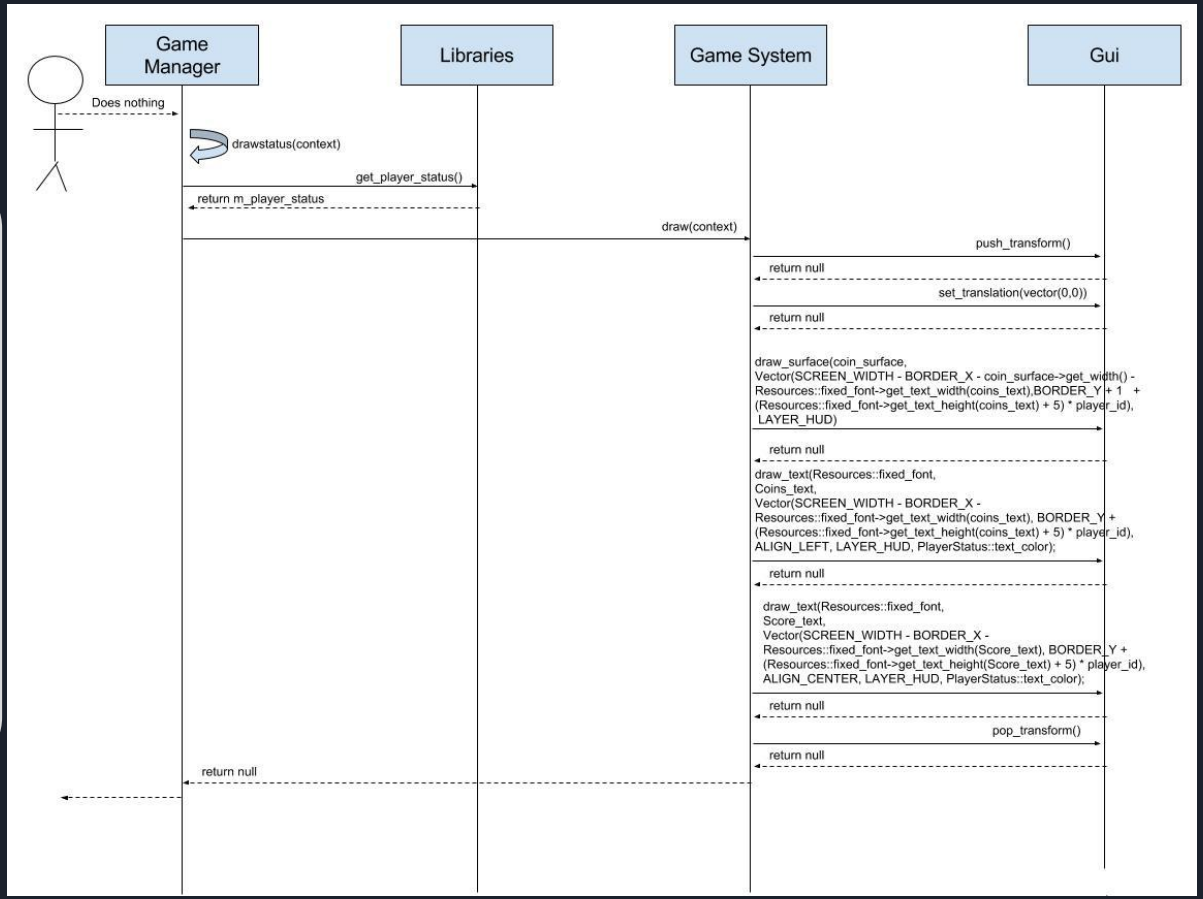
- supertux/Statistics.cpp
- supertux/LevelIntro.cpp
- supertux/game_session.cpp
- badguy/Badguy.cpp
- object/Coin.cpp
- supertux/player_status.cpp

Potential Risks & Limitations

- Adds another thing to be updated every frame
- Potential loss of performance (lowering of frame rate)
- Clutters the screen

Sequence Diagram

Score gets updated after tux picks up a coin



Legend

Function Call

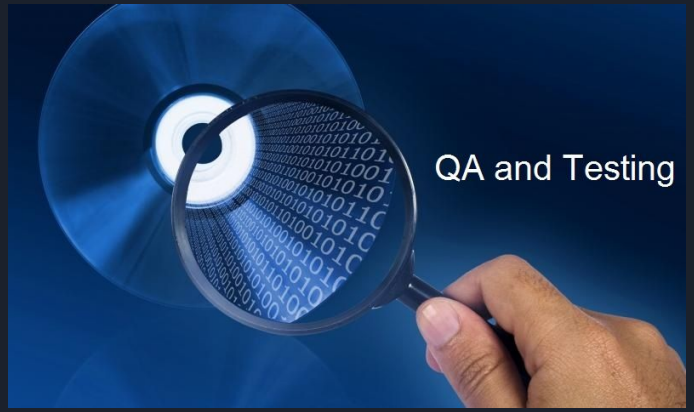
Return Value

Component

Internal Function Call

Plans for Testing

- Test every way to gain points
- Test every way to lose points
- Systematic white box testing



Concurrency

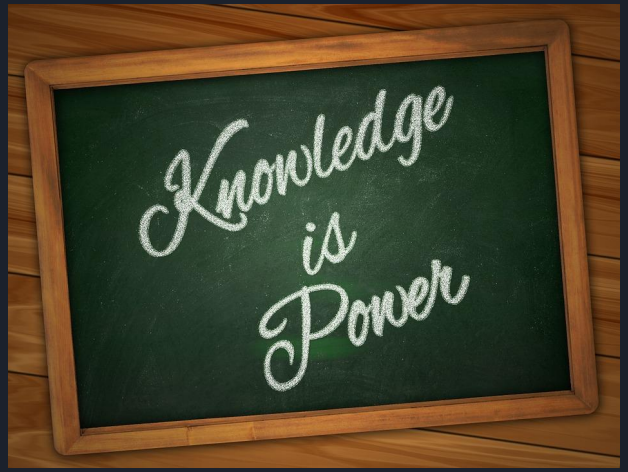
- Allows for a variety of different actions to be performed simultaneously by the Game Engine
- New score system will utilize this concurrency
- Allows for smooth game play

SuperTux Team Issues

- Would not need a large team or new members to implement this feature
- Need at least one team member that understand the system
- Open Source developers may add more complicated additions as ways to earn points

Group Limitations & Lessons Learned

- SuperTux is very unorganized and poorly commented
- Difficult to find where items get drawn and where files are invoked
- None of our team members know C++



- An enhancement doesn’t have to be large
- More time consuming to implement a feature than to map the architecture

Closing Statements

Thank you!

Any Questions?