




# Verification Games

Autumn kblack37, Zan zbalcom, Jake jchiang2,  
Alex ahd2112



# What is verification? Why is it important?

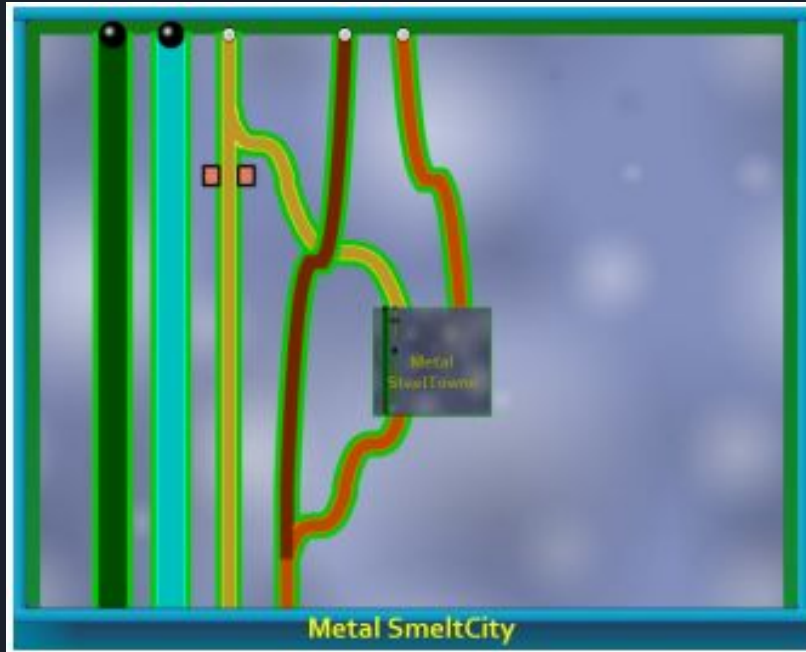
Verification is the act of assessing the correctness of a program with respect to a certain specification.

Verification lets you know truths about a program that can be valuable when considering the task a program is designed for.

1996, Ariane-5 rocket  
exploded 40 seconds after  
take off due to casting 64-bit  
floating point numbers to  
16-bit signed integers.

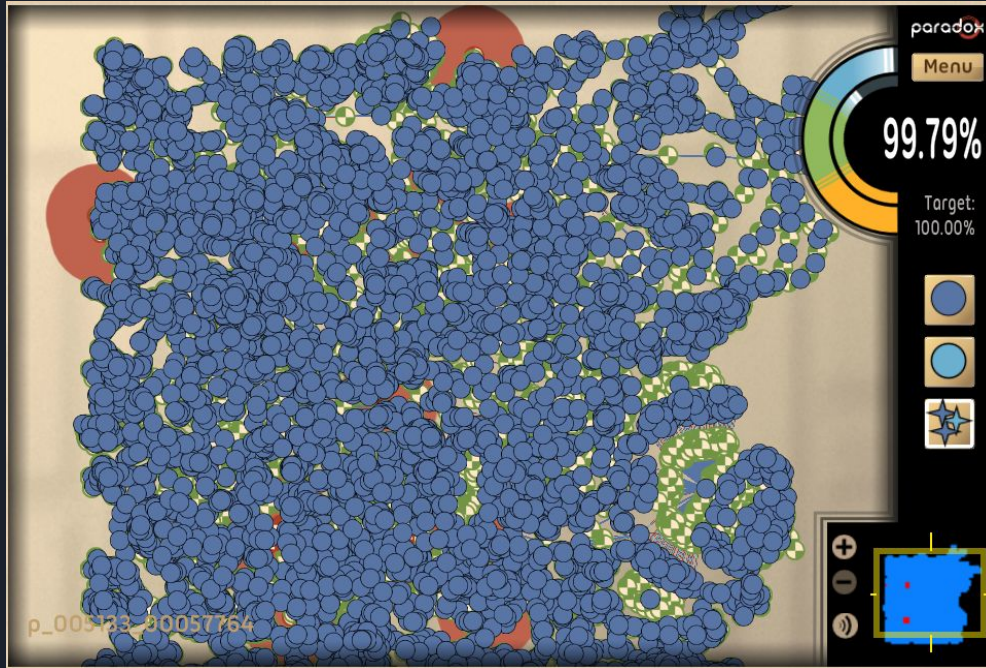


# Current Concepts



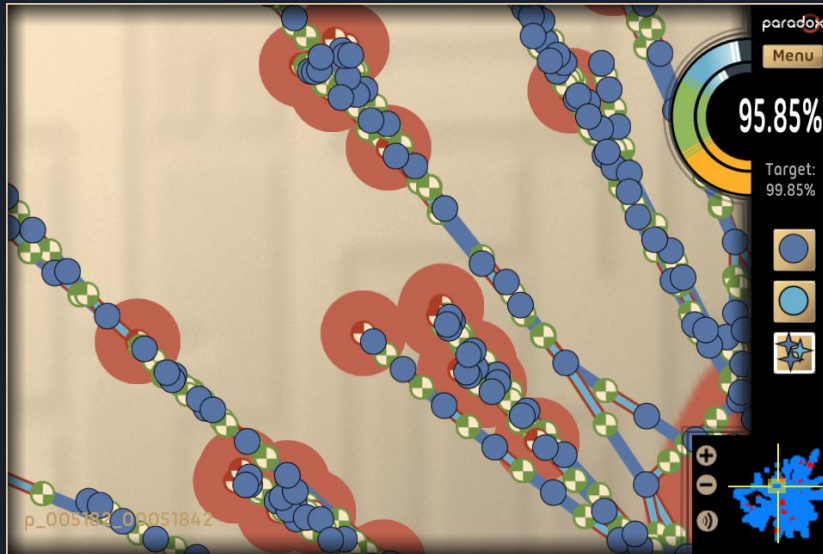
- ❖ Pipe Jam uses the color of the pipes to indicate the width of the pipe.
- ❖ Widths represent type annotations in the program being evaluated
- ❖ These annotations represent some specification you want to verify

# Current Concepts



- ❖ Paradox shows you the whole program at once
- ❖ The user changes variables to eliminate conflicts (marked in red)
- ❖ The goal is to have the smallest number of conflicts

# Current Concepts



- ❖ Paradox Close Up
- ❖ Paradox runs on html and is slow and unappealing when run in a browser
- ❖ Complexity has no abstraction



## Our Approach

- ❖ Port existing code to Haxe
- ❖ Assess the viability of PipeJam and FlowJam
- ❖ Restructure code to an entity-component-system design
- ❖ Add engaging aspects to the game such as leaderboards



# Port

The current code base is in ActionScript3 which runs in Adobe's Flash Player browser plugin which will not be supported after 2020

Haxe is an open source language with easy cross platform support for IOS, Android, and many others

The code for PipeJam and FlowJam are in the process of being ported into Haxe



# Assess Game Viability

Once we have two working prototype games working, PipeJam and FlowJam we plan on conducting player studies to assess which game format is more appealing to players.

This will be the game format we move forward with for the final product.





# Restructure

The current code base is in a state not consistent with a modular system architecture

Once the code is ported and operable we plan to restructure the code into a Component-Entity-System design for the purpose of future maintenance and the extendability of the game as a whole.



# Gamify

After the game is in its refactored state we want to add components to the game flow that will keep people engaged and returning to the platform.

- ❖ New Scoring System
- ❖ Leaderboards
- ❖ Badges



# Lessons

❖ Time: The porting

❖ Workflow : Team



Questions?