

## Motivation:

As programs get larger and larger, it becomes more and more difficult to prove the correctness of programs definitively, or run static analysis. Dynamic analysis can also take a lot of time and computing power on larger programs, and it relies on the tester to consider every possible domain of inputs. Static analysis, and specifically code verification, while not as versatile, can prove a program is correct for all inputs without ever having to run it, and thus is often highly beneficial. However, proving a program can be extremely costly, and thus is only commonly used on smaller code segments. In this project we would be scaling up and distributing this crowdsourced code verification game, with the goal of providing an easy way to generate a proof of larger programs, without even having to do it yourself.

## Approach:

Our approach would be to complete and expand the prototype code verification game Pipe Jam (detailed [here](#)) to accept more complicated programs, to more accurately translate those programs into game boards, and to be playable by a wide audience of people, either in a web browser, or on your phone. The theory behind the existing prototype is sound, but the code is unfinished, and it is not available to the public as of yet. Our goals would be to debug and complete the prototype game such that it works with programs of reasonable complexity (a standard which will be determined as part of the project), and to implement it in such a way that the general public has access to the game, and programmers have access to upload their programs as game board the users can play.



## Challenges and Risks:

Though most of the process of video game implementation has already been completed, it would still be no small feat to fix a game with such complex level-creation mechanisms, and to implement it on a public platform, in a relatively short time frame. Furthermore, the mere act of scaling the game to handle programs of sufficient complexity may prove much more challenging than expected. We will mitigate these risks by making a careful assessment of what aspects of

thee game must be fixed and enhanced before attempting to complete them, so as not to be taken by surprise. In addition, we will likely use some form of 3rd-party tool to make our game playable via a web browser or smartphone, to avoid the added development time of porting the game to those platforms. While it may seem difficult to accurately assess the complexity of this project, I believe it is well worth the risk to produce an entertaining product for the general public, as well as a widely beneficial tool for all programmers in the Paul G. Allen School and beyond.