

UNSCRABBLEING THE PROBLEM

- The main idea behind this project is that currently there is no easy way for middle school students to be able to be introduced to programming in a context that they have seen before.
- Most programming languages are either removed from anything that a student at that age would have seen or deeply related to math.
- Another issue with many programming languages is that children feel overwhelmed and discouraged by complex programming languages.
- Current language that exist to introduce programming language to younger students are not related to more complex programming languages used later and do not lead to a good transition.
- This led to development of the idea that there should be a programming language that students can relate to as well as have a enable good transition to more complex languages.
- Some of the main concepts that are lacking in these languages are functions, and types. This will make it more difficult for them to learn algorithms later in their education.
- One concept that most students can relate to are board games. Because of this context that the students have this led us to developing a programming language that would allow students to describe board games.

CODE IN SHEEP'S CLOTHING

Teaching a younger generation the fundamentals of programming through a specialized language focused on board games

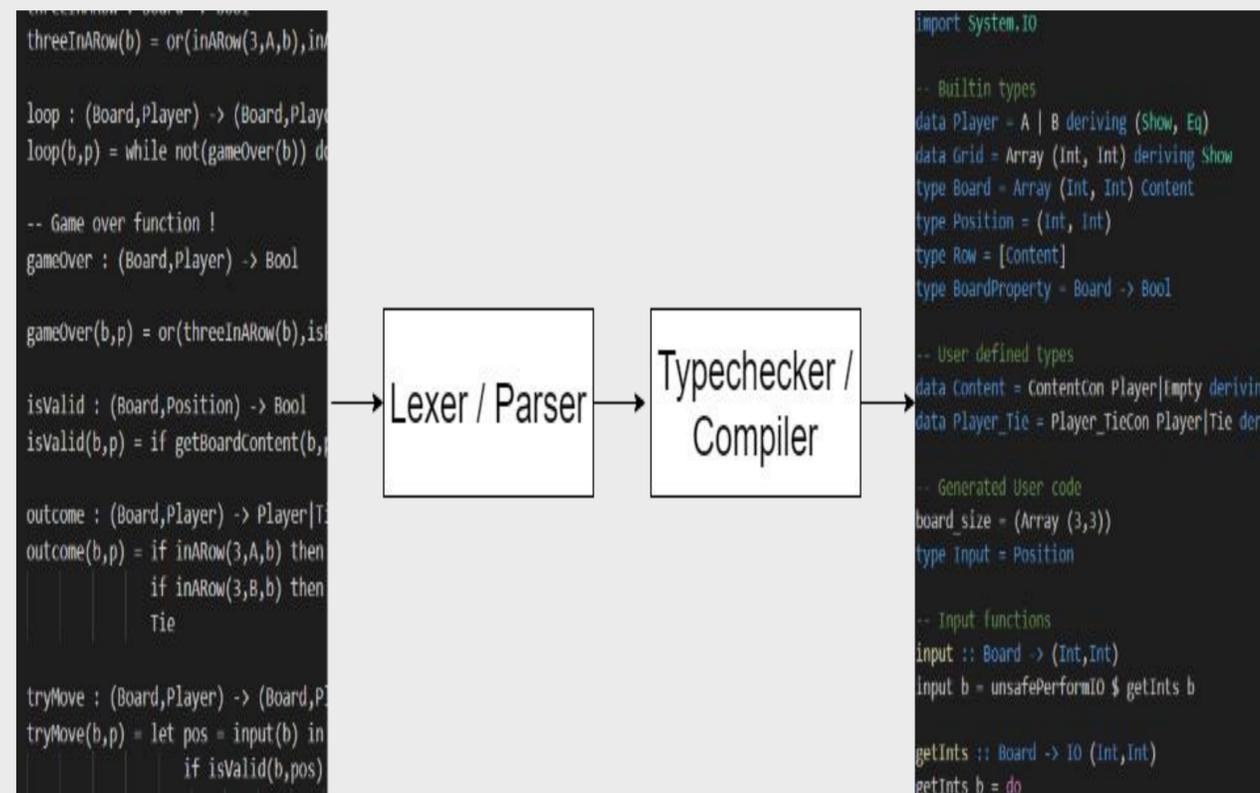


Figure: The process of taking the board game language to Haskell

TIC-TAC-TOEING THE LINE

- One of the main balances that our language needed to achieve was between ease of use and expressiveness.
- The main goals are to teach core programming concepts while also maintaining accessibility for middle schoolers.
- The risk of allowing students to have too much expressiveness is that they would not be able to learn the language.
- If the language is too simple it would not keep them engaged or let them to learn important programming concepts.

CANDY LANDING ON A SOLUTION

- We needed to produce a programming language that is a steppingstone to more practical existing language.
- Educators decided that Haskell would be he best language for students to learn next and therefore our language is like Haskell with a few key differences that limit the language while making it significantly simpler.
- We implemented our solution in Haskell using bottom up parsing using the lexing and parsing libraries Alex and Happy.

CODENAMES

Johannes Freischuetz –
freischj@oregonstate.edu

Ben Warschauer
warschab@oregonstate.edu

Fern Bostelman-Rinaldi
bostelmf@oregonstate.edu

Jackson Bizjak
bizjakj@oregonstate.edu

Tom Croll
crollt@oregonstate.edu



CLIENT

Martin Erwig, Ph.D.
Stretch Professor of Computer Science
eriwg@oregonstate.edu