

SOFTWARE DESIGN

- Our project is organized into an editor, an interpreter, and a GUI.
- It is written in Haskell & JavaScript.
- Editor: React web app which supports users in writing Spiel code.
- Interpreter: does lexical analysis, parsing, type checking, and evaluation.
- GUI: a web interface for the interpreter that allows users to play games by entering expressions or clicking on pieces.

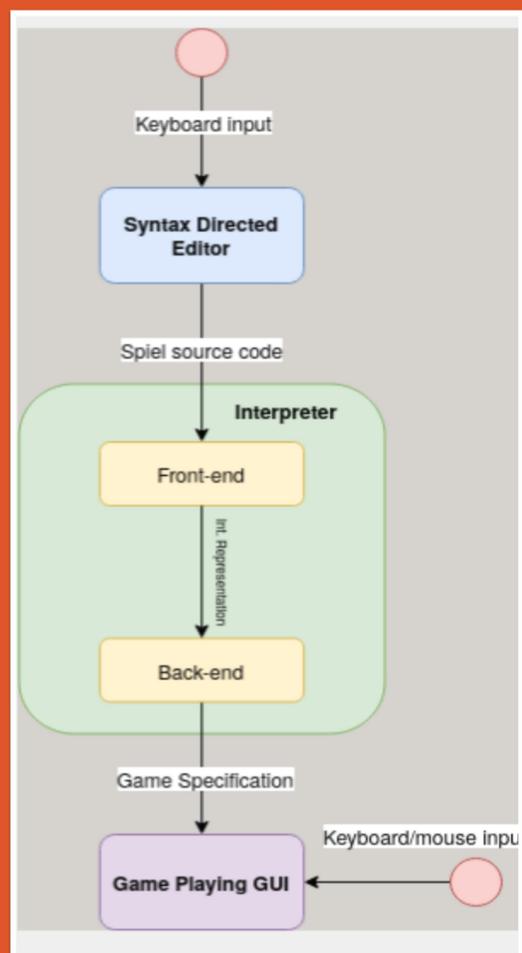


Fig 1: The block diagram of our project's top-level components

SPIEL BOARD GAME LANGUAGE

An educational domain-specific language for describing board games to teach computing in middle school science classrooms.

```

Spiel Language Home Free Editor Demo Themes ▾
1 game TicTacToe
2 type Board = Grid(3,3) of AnySymbol
3 type Input = Position -- currently ignored anyway!
4
5 initialBoard : Board
6 initialBoard(positions) = Empty
7
8 goFirst : Player
9 goFirst = A
10
11
12 -- Game ending: game-over condition and game outcome
13 --
14 gameOver : Board -> Bool
15 gameOver(b) = or(threeInARow(b), isFull(b))
16
17 outcome : (Board, Player) -> Player|Tie|Continue
18 outcome(b,p) = if inARow(3,A,b) then A else
19                 if inARow(3,B,b) then B else
20                 if isFull(b) then Tie else Continue
21
22 threeInARow : Board -> Bool
23 threeInARow(b) = or(inARow(3,A,b), inARow(3,B,b))
24
25
26 -- Predefined operations
  
```

Fig 2: The spiel web-interface provides a helpful editor for writing board game descriptions. It is also designed to support fill-in lesson files where the user can learn small portions of the syntax at a time.

PROBLEM

Teaching computer science and software engineering is difficult. While common first programming languages such as C++, Java, and JavaScript are powerful tools, they introduce unnecessary complexity for any beginner, and especially for the middle school student.

Despite the hard learning curve, computer literacy is one of the most important skills in the contemporary era. Practitioners of every field use software for their daily work and many of them require or soon will require a basic level of algorithmic ability, so it has personal and economic value.

Our goal is to facilitate computer science and software engineering education with a simpler programming language.

SOLUTION

- Our solution is Spiel, a domain-specific programming language for the specification and play of board games.
- Board games are easy to learn and familiar to middle school students. They are also very deterministic, meaning that they are great for teaching algorithmic thinking.
- We created an interpreter, editor, read-evaluate-print loop, and visual interface for playing the games.
- We also created example programs and corresponding curriculum which gradually teaches our language.
- It will be used in sixth and seventh grade classrooms at Linus Pauling Middle School.

LANGUAGE DESIGN

- Spiel has board-game specific prelude functions to support the domain.
- Spiel is a purely functional language with simple syntax.
- The result of any given function is only decided by its inputs.
- Students can focus on the intended algorithmic description conveyed in the functions without considering side-effects.

THE TEAM

Benjamin Friedman: friedmab@oregonstate.edu

- Backend Designer & Engineer
- Worked on versioning, continuous integration, and lesson plan concepts

Alex Grejuc: grejuca@oregonstate.edu

- Language Designer & Language Engineer
- Contributed to language design, abstract syntax, parser, type checker, evaluator, and curriculum

Calvin Gagliano: gaglianc@oregonstate.edu

- Front End Designer & Engineer
- Designer of the lesson plan editor

Kai Gay: gayk@oregonstate.edu

- Language Designer & Principal Language Engineer
- Contributed to language design, abstract syntax, parser, type checker, and evaluator

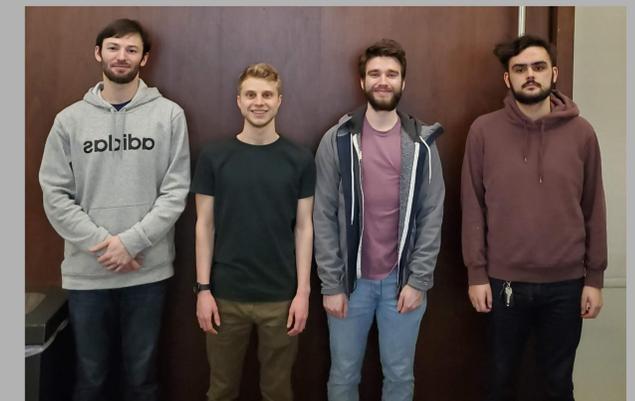


Fig 3: The Spiel team (left to right), Ben, Alex, Calvin, and Kai.

ACKNOWLEDGMENTS

- Professor Martin Erwig: erwig@oregonstate.edu, client, Stretch Professor at OSU
- Research funded by the National Science Foundation.
- Work done in parallel with alternative Board Game Language by team 41B.