Gridworld

BY ANDREW PHAN, LOREN WENG, KEVIN WENG, ZIKAI LIN
Our group

- Andrew Phan – Project Manager
- Loren Weng – Language Developer
- Kevin Weng – Language Architect
- Zikai Lin – Testing and Validation
Motivation

• Our group loves computer games but none of us really know how to make a game.
• We wanted to learn how games are designed.
• Want to simplify the process for new or inexperienced programmers.
  • Language has to be simple.
  • Language has enough features to form a game. Should tempt the user to actually want to use our language.
• Want to learn the necessary features for a game
What is Gridworld?

- Node-based language makes game design simple to understand.
- We have a starting point that leads to more nodes. Much like how a story, movie or game has a beginning and an end.
  - But it is also able to have multiple ending and storylines, which can be chosen by the player.
  - The user can design these game choices by node() and goto().
- Implements basic calculations, logical operations, and control structures.
- Easy to access global variables.
- In one sample game, saves around 400 lines of code.
## Supported Language Features

<table>
<thead>
<tr>
<th>Functions</th>
<th>Types</th>
<th>Control / Loops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>Int</td>
<td>If / Else / elif</td>
</tr>
<tr>
<td>Class</td>
<td>String</td>
<td>While</td>
</tr>
<tr>
<td>print</td>
<td>Bool</td>
<td>Return</td>
</tr>
<tr>
<td>Read (readStr, readInt)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choose</td>
<td></td>
<td></td>
</tr>
<tr>
<td>goto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>roll</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Language Architecture

- **Scanner.ml**: reads the source code and compares it to the tokens.
- **Parser**: parses the tokens.
- **Semantics**: analyzer.ml - to check the code when compiling.
- **AST**: ast.ml - abstract syntax tree.
- **Compiler**: grid.ml, compile.ml
- **Python Code**: execution in python, get results.
Testing

- Regression Test Suite is similar to the microc compiler.
  - Small test components to test each individual case.
  - Aimed for at least 40 simple tests.
  - Compare both expected output and the execution of compiled Python code
  - Long and tedious process.
  - Initially many things didn’t work. When we added functionality, other things would break. Example was adding our node based implementation broke many tests.
Conclusion

- Gridworld is a language designed to teach inexperienced programmers about simple game development.
- Ocaml is not for normal people.
  - Creating a language is difficult and consumes a lot of time.
  - Appreciate languages now and know why there are so many of them.
  - Horses for courses. Languages are good/bad for different applications.
    - Reason why people use C++ when they need speed.
- Group learned how to use Ocaml, debugging, Trello, Github and Latex for PDF creation etc.
- Hard to coordinate and manage jobs for people. And some advice for other:
  - Don’t code till the very last minute. Something always comes up.
  - Meet up every week especially if you are not busy.
  - Use your TA and professor.
  - Talk to other CS students.
Demo