Motivation

* My background – why I’m in school right now
* Want to study interdisciplinary approach between neuroscience and computer science
* Approached MBB professor, he suggested we do a project together
* He studies fear, but clinical methods of testing it are lame (give example of plane crash experiment from 60’s); he wants “video game-like” approach
* Intelligent agents needed
* Kew I wouldn’t have time to do something outside of class, so I tried to think of a way to incorporate it into my PLT project.
* Why not a language for modeling the behavior of intelligent agents?
* Prof. has since moved on – he couldn’t wait a semester for me to finish this project – but I may revisit it with him. Also, the need to model intelligent behavior has plenty of other applications.

Disclaimer

* Basic idea is sound, implementation of design is sound, but the part in between – the design of the language itself – is questionable. It doesn’t go much and what it does do is probably easier to do without Geppetto. I hope that doesn’t hurt our grade too much.
* Actually got in most of the basic language elements we wanted. Pretty pleased with that given limited resources.
* This (or any) language composed of: variables, functions, expressions, and statements. Given that all these elements are now working, we could probably repurpose Geppetto to do something else pretty quickly.

Overall Geppetto Language Structure

* Architectural block diagram
* “Rule-driven analysis engine”
* YACC-like structure: Declarative section that defines objects called entities, followed by rules that act on them, followed by section for supplementary code (functions)

Tools

* Git
* Eclipse
* JFlex
* BYACCJ
* JUnit?

Overall Geppetto Project Structure

* Package layout
* Most functionality encapsulated in domain classes (e.g., expressions know how to evaluate themselves, statements know how to execute themselves, etc)
* Parser is single biggest piece of code
* The main program and interpreter themselves are quite small (show screen shot)

Testing

* toString
* simple.gep