Next

Ernesto Arreguin (eja2124)
Danny Park (dsp2120)
Morgan Ulinski (mu2189)
Xiaowei Zhang (xz2242)
• A programming language used to create text-based games
  – RPGS
  – “Choose your own adventure” stories
• Provides the tools needed to implement a game, while still allowing programmers freedom to decide:
  – Locations, characters, items
  – Game plot, actions that take place
  – End conditions
• Data types
  – int, string
  – item, character, location

• Next-specific constructs
  – New operators and statements
    • exists, grab, drop, hide, show, kill
    • choose
      – take input from the player and uses that to decide what branch to execute
    • prob
      – inserts randomness into a game to make it more interesting
    • start
      – contains code associated with a given location
      – specify game end conditions
Next is a language where everything written is a declaration

Possible Next declarations include the two primitive Next data types:
  – int
  – string

Next also includes three complex data types:
  – item,
  – character
  – location

There is one more type of declaration, the start declaration, which declares the start and specifies what happens inside a location. Once a location is declared a start declaration must also be made for that location

The following example will help clarify the basics of Next:
Sample Program

int fin;

item object {(int size=3)}

character person {(string say ="Hello World!"),(object)}

location here{(),(object),(person)}
int fin;
item object {int size=3}
character person {string say ="Hello World!"},(object)
location here{()},(object),(person)}

start here end (fin == 1){
  if (exists here.object) then
    output person.say;
  else
    output object.size;
  fin = 1;
}
Translate Next to JAVA
Next code -> scanner -> parser -> first pass (checking) -> second pass (translating)
The Next language lets developers build text-based RPGs.

By providing abstractions of commonly used functionality in RPGs we have tried to simplify the RPG building process.
Lessons Learned

• Team dynamic can make or break the project
  – Since there are so many moving parts, the team has to be able to work well together

• Source control choice is key
  – Kept us sane, sometimes

• Regular Communication
  – Allowed the project to run smoothly throughout the semester

• Problem? Add another level of indirection
Thank You, Happy Holidays