SNL (Stage (null) Language) Project Proposal

Alex Liu, al3037 (Language Guru)
Andre Paiva, app2125 (Verification & Validation)
Daniel Maxson, dsm2157 (System Architect)
James Lin, jl3782 (Manager)

COMS W4115 - Programming Languages and Translators

Introduction

SNL is a language designed to model role-playing game scenarios based on the structure of state machines. It allows users to programmatically re-create sequential scenarios as "stories." Its syntax aims to be simple so that adolescents can easily understand how to use it. Where possible, SNL uses intuitive keywords in place of symbols, since symbols commonly known in the CS community can be confusing to those who are unfamiliar with programming. This will encourage children to learn how to write code in a controlled, fun environment while allowing them to exercise their creativity.

Motivation

Since our program will essentially model state diagrams, a "story" that can be expressed in terms of an initial Character input and transitions between states can be rewritten in SNL. This could be something simple and yet computational like the calculation of GCD or Factorial. The larger intent, however, was for users of SNL to implement creative scenarios, such as stories (especially Multi-Ending Chapter Books or Interactive Role Playing Games). A further application of SNL could be the generation of computerized surveys and studies within the social sciences.

Syntax

Types

Types are directly inferred from the data supplied for the variable. For example, if a string is assigned to a variable, then the variable is given an implicit string type. The primitive types are currently limited to integers, doubles, strings, and lists; these primitives being easy for children to understand and yet powerful enough to create robust logic. There are also two special object types: Character and Stages.

Operators and Keywords

- Basic Arithmetic (`+`, `-`, `*`, `/`)
- String Concatenation (`+`)
- Assignment (`is`)
- Conditional (`if`)
- Conditional Statements (`=`, `>`, `<`, `<=`, `>=`)
- Control Flow (`next`, `start`, `end`)
- Special Object Declaration (`:`)

Stages

In place of functions, SNL uses Stages, which contain variables and logic in order to determine the character's transition to another state.

Input/Output

In SNL the keyword `input` is used to denote any recently entered keyboard text.

Character

A special global object meant to loosely simulate the user, which carries attributes that can be modified by any Stage. There will be only one Character per program.

Sample Programs

Sample Program #1 - Goldilocks:

```
Character:
name is input
hair_color is "yellow"

start stage1:
show Character's name + " finds a house."
if input = "\n"
(next stage2)

stage2:
show "She enters the house and eats and sleeps."
if input = "\n"
(next stage3)

stage3:
show "The three bears find " + Character's name + ". She runs away."
if input = "\n"
(next stage4)

end stage4:
show "The End."
```

Sample Program #2 - Factorial:

```
Character:
num is 5
total is 1

start stage1:
```

```
if num = 0
(next last)

total is total * num
num is num - 1
next stage1

end last:
show total
```