

Matrix Entertainment Language (MatrEL): A Board Game Creation Language

Rochelle Palting
rcpalting@gmail.com
rcp2122@columbia.edu

Computer Science Department, Columbia University
CS4115 Programming Languages and Translators
Professor Stephen A. Edwards
Fall 2008

1 Introduction

Board games have been around for countless generations. There are numerous games that offer leisure that suits various minds and tastes from Tic Tac Toe, Connect Four, Minesweeper and Battleship. Matrix Entertainment Language (MatrEL) is a programming language that allows a designer to create a game with a square matrix board. With MatrEL, anyone can take any traditional matrix type board game and customize it to their liking. For the enthusiast, a totally new and original game can be fashioned with the ease of using MatrEL.

In this paper we outline the various features of MatrEL and provide a sample program that uses this exciting language.

2 Language Functionality

MatrEL will allow the programmer to create an interactive game using a square matrix as the game board. The typical use of the MEO programming language will be:

- initialize game board with desired size and contents
- initialize game rules and features
- start the game by prompting and accepting player inputs
- check to see if the player's input results in either a win or lose situation or neither
- prompt for another player input

The MatrEL programming language will make matrix board game creation and play simple and fun.

3 Language Attributes

3.1 Identifiers and Keywords

The following identifiers and keywords are reserved in MatrEL:

matrix
empty
containsAll
topDiagonal
bottomDiagonal

string
int
bool
while
if
then
else
input
full
print
map
row
col
gameOver

3.2 Data Types

GameRule – an expression that evaluates to a Boolean

int integer values
map a map of key/value pairs
string sequence of characters

3.3 Assignment Operator

= The '=' symbol will be used to set a value to an identifier.

3.4 Arithmetic Operators

+ addition operator
- subtraction operator
* multiplication operator
/ division operator

3.5 Equality and Relational Operators

=
==
>=
<=

3.6 Conditional Operators

while

3.7 Control Structures

if-then-else

3.8 Comments

// The line that begins with '//' will be treated as a comment.

4 Example Program

Below is the game Tic-Tac-Toe implemented using the MatrEL programming language.

```
// initialize game board to size 3 with each cell being empty
matrix game = 3
game[] = empty

// create win rules
GameRule threeInARow (rowNum, value) = game[rowNum][_] containsAll value;

GameRule threeInAColumn (colNum, value) = game[_][colNum] containsAll value;

GameRule threeInADiagonal (value) = game[topDiagonal] containsAll value or
                                     game[bottomDiagonal] containsAll value;

// set other game features
map players;
players.put(1, 'X')
players.put(2, 'O')
gameOver = false
int playerTurn = 1;

while(not gameOver)->
    print 'Enter selection.'
    input pos
    if threeInARow(pos.row, players.get(playerTurn)) or
       threeInAColumn(pos.col, players.get(playerTurn)) or
       threeInADiagonal(players.get(playerTurn)) then
        print 'Player wins.'
        gameOver = true;
    else if matrixFull then
        print 'Tie game, squares are all full.'
        gameOver = true;
;
```