

Kurs języka Lua 2017

Lista zadań nr 8

Na zajęcia 08-09.05.2017

Za zadania z tej listy można uzyskać maksymalnie 10 punktów.
Szczegółowe kryteria oceny zadań znajdują się na [stronie przedmiotu](#).

Zadanie 1. (2p) Napisz w C/C++ program, który wczytuje plik konfiguracyjny Lua i rysuje na ekranie podaną w nim funkcję.

Dane do wyświetlenia mają być umieszczone w globalnej tablicy TOPLLOT (patrz przykład). Zakładamy, że funkcja jest jednoargumentowa: przyjmuje liczbę i zwraca liczbę. Jeśli tablica nie posiada pola step, należy się zachowywać jakby miało ono wartość 1. Jeśli w pliku konfiguracyjnym znajdują się błędy, należy wyświetlić komunikaty w jednoznaczny sposób wskazujące ich przyczynę.

Wykres ma mieć formę gwiazdek wyświetlonych na konsoli (pozycje gwiazdek na osi *OY* zaokrąglaj do najbliższych liczb całkowitych, gęstość gwiazdek na osi *OX* jest wyznaczona przez pole step).

```
local function myfunction (x)
    return math.sin(x)*x
end
```

```
TOPLLOT = {function=myfunction, minx=-10, maxx=10, step=0.5}
```

Zadanie 2. (4p) Napisz w C/C++ program obsługujący grę w kółko i krzyżyk pomiędzy AI zapisanymi w Lua. Program powinien pobierać z linii poleceń argumenty w postaci nazwy plików graczy, uruchamiać programy graczy, nadzorować rozgrywkę pokazując użytkownikowi stan planszy i wykonane ruchy, a następnie zamykać wykorzystywane programy.

Skrypty Lua powinny udostępniać globalną funkcję AI, która jako parametry pobiera symbol gracza (string) oraz dwuwymiarową tablicę 3×3 zawierającą informację o zawartości planszy (patrz przykład).

Wszelkie błędy powstałe z powodu uruchomienia kodu gracza powinny być przechwytywane przez nadrzędną aplikację i wyświetlane w postaci przystępnych komunikatów.

(2p) Napisz AI które wygrywa z przedstawionym przykładem w co najmniej 80% przypadków.

```
local function rng() return math.random(3) end
```

```
AI = function(mysymbol, board)
    while true do
        local x, y = rng(), rng()
        if board[x][y] == ' ' then
            return x, y
        end
    end
end
```

```
-- print (AI('X', {{'X', 'O', 'X'}, {'O', 'X', 'O'}, {'O', ' ', 'X'}}))
```

Zadanie 3. (4p) W pliku Lua, pod zmiennymi globalnymi zaczynającymi się od prefiksu `level_` znajdują się dwuwymiarowe plansze pewnej gry.

```
level_0 = { {'W', 'W', 'w', 'w', 'L'},
            {'L', 'x', 'L', 'w', 'L'},
            {'f', 'f', 'L', 'L', 'L'}}
config_level_var = 128
level_deep = { {'01', '0a', '0a'},
               {'02', '1b', '00'},
               {'0x', '1x', '1r'},
               {'0y', '0r', '00'}}
local rnd = math.random
```

Napisz w C/C++ program, który wczytuje plik z planszami i w pętli rysuje plansze o nazwach podanych przez użytkownika (tzn. `deep` powinno wyświetlić zawartość `level_deep`). Oczywiście jeśli dana plansza nie znajduje się w pliku, powinna zostać wyświetlona stosowna informacja.

Update: Podanie przez użytkownika komendy **ALL* powinno spowodować wyświetlenie wszystkich zdefiniowanych plansz (w dowolnej kolejności).

Iterowanie po tablicach zawierających nieznane klucze najwygodniej przeprowadzić przy użyciu `lua_next`, natomiast długość sekwencji możemy uzyskać za pomocą `lua_len`.

Wskazówka: Znajdź sposób aby uzyskać dostęp do zbioru wszystkich zmiennych globalnych skryptu.