

# Programming web-course analysis: how to introduce computer programming?

Romenig Ribeiro<sup>1</sup>, Tulio Faria<sup>2</sup>, Leônidas Brandão<sup>1</sup> and Anarosa Brandão<sup>3</sup>

<sup>1</sup>Institute of Mathematics and Statistics - IME - USP

<sup>2</sup>School of Arts, Sciences and Humanities - EACH - USP

<sup>3</sup>Polytechnic School - USP

*{romenig, leo}@ime.usp.br, tuliofaria@usp.br, anarosa.brandao@poli.usp.br*

Frontiers in Education, October 2014





Institute of Mathematics and Statistics (IME)



Polytechnic School (POLI)



School of Arts, Sciences and Humanities (EACH)

# Overview

- 1 Introduction
- 2 Experiment
- 3 Discussion, conclusions and future work

# Introduction

## Context: logical reasoning and computer programming

Proposed as fundamental abilities and their introduction in early stages of education has been adopted in an increasing rate

## Problems: introducing it earlier may adress some challenges

- Challenges that are currently faced by teacher and students of STEM courses:
  - new software environment (programming environment)
  - new way of describing a problem and it's solution (programming language)
  - learn to solve problems computationally

## Proposals to overcome some of these problems

- There are some proposals of using visual systems to support the learning of introductory programming (e.g. Raptor [Carlisle], Greenfoot[Kölling], Scratch [Maloney et al.] and Alice [Cooper et al.] )
- Most of Visual Programming (**VP**) systems allows students to:
  - compile and execute (or interpret) their algorithm with a single click
  - build programs without “knowing by heart” the programming language syntax
  - consequently, they may focus their attention to the problem solution

## This paper approach

An experiment to evaluate the mental workload during the learning process (VP versus textual) in a web-course context.

To accomplish the analysis we:

- created two similar web-courses of Introductory Programming
  - students were separated into two groups: G1 (visual programming) and G2 (textual programming language)
  - delivered through Moodle
  - was an Opened Online Course, not a MOOC since it had 144 students enrolled
  - same instructional material with differences only concerning the tools
- created a NASA Task Load Index (NASA-TLX) Moodle module for evaluating mental workload



## iVProg

Visual Programming System to teach algorithms  
(procedural programming):

- firstly deployed in 2009 (based on Alice, from Carnegie Mellon)
- the second version of iVProg was developed using a framework for Interactive Learning Modules
- features:
  - new lines of code with context menu
  - drag-and-drop combined with pointing-and-click to order lines of code
  - edit-in-place for variable names and values
  - automatic evaluation
  - interpret students algorithm with a single click
- iVProg is a Free Software
  - <https://github.com/LInE-IME-USP/ivp2java>

# Background: iVProg, VPL and NASA-TLX

## iVProg

The screenshot displays the iVProg graphical programming interface. At the top, there is a toolbar with icons for settings, a list, undo/redo, and a checkmark. Below the toolbar, the main workspace is titled "Principal".

Under the "Variáveis" (Variables) section, three variables are defined: `num = 1.0`, `soma_pos = 0`, and `soma_neg = 0`. Each variable has a small icon next to it.

The main logic is a loop structure:

- An "Enquanto" (While) loop with the condition `(num != 0)`.
- Inside the loop, there is a "Se" (If) statement with the condition `(num >= 0)`.
  - If the condition is true, the instruction `soma_pos recebe (soma_pos + num)` is executed.
- Following the "Se" statement is a "Senão" (Else) block.
  - If the condition is false, the instruction `soma_neg recebe (soma_neg + num)` is executed.

Each instruction block has a "+ Instruções" (Add Instructions) button next to it. The bottom of the interface features a console window with a prompt `>` and the output `2.0` and `0.0`.

## Virtual Programming Lab - VPL

The VPL system is a Moodle module developed at Universidad de las Palmas Gran Canaria, Spain.

- is a Free Software for Moodle
- supports many languages (we used C programming language)
- automatic evaluation
- interpret students algorithm with a single click
- <http://vpl.dis.ulpgc.es/index.php/en/>

## Virtual Programming Lab - VPL

```
Arquivo  Editar  Opções  Ajuda
ex2-4.c
1  #include<stdlib.h>
2  #include<stdio.h>
3  int main() {
4      int N;
5      scanf("%d", &N);
6      int Soma;
7      while (N!=0) {
8          if (N%2==1) {
9              Soma=Soma+N;
10         }
11         scanf("%d", &N);
12     }
13     printf("%d", Soma);
14     return 0;
15 }
16
```



Console 1 - conexão fechada

```
1
2
3
0
4|
```

## NASA-TLX Protocol

- Objective: measure workload during the execution of an activity
- Workload: defined by Hart and Staveland (1988) as a hypothetical construct that represents the cost of someone finishing a task and reaching a certain level of performance.
- How: by filling up a questionnaire with six scales, giving them values and then conducting a pairwise choice between the scales, giving them weights
- Scales:
  - Mental Demand (MD)
  - Physical Demand (PD)
  - Temporal Demand (TD)
  - Own performance (OP)
  - Effort (EF)
  - Frustration (FR)

## NASA-TLX Protocol

LinE  Meu "Painel de Bordo"  Cursos

Você acessou como Romenig Ribeiro (Sair)

PÁGINA INICIAL / MEUS CURSOS / MOOC / PROG2014-T2 / 1. ALGORITMOS / NASA-TLX - COMO FORAM AS PRIMEIRAS ATIVIDADES?

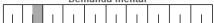
### Resposta ao questionário

Questionário sobre a tarefa - Parte 1

Clique em cada escala no ponto que melhor representa sua experiência durante a tarefa

**Demanda mental**

Quanto das atividades mental e perceptual foram exigidas (por exemplo: pensar, decidir, calcular, lembrar, observar, procurar, etc)? A tarefa foi fácil ou difícil, simples ou complexa, rigorosa ou tolerante?

Baixa  Alta

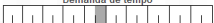
**Demanda física**

Quanto esforço físico foi exigido (por exemplo: empurrar, puxar, virar, controlar, ativar, etc)? A tarefa foi fácil ou exigente, morosa ou rápida, leve ou pesada?

Baixa  Alta

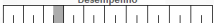
**Demanda de tempo**

Quanta pressão em relação ao tempo você sentiu no ritmo de execução da tarefa? O ritmo era lento e vagaroso ou rápido e frenético?

Baixa  Alta

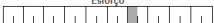
**Desempenho**

Quão bem sucedido você se considera ao realizar os objetivos da tarefa? Ficou satisfeito com o seu desempenho no cumprimento dessas metas?

Boa  Ruim

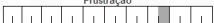
**Esforço**

O quão duro você teve de trabalhar o (mental e fisicamente) para alcançar o seu nível de desempenho?


Baixo  Alto

**Frustração**

Quão inseguro, desencorajado, irritado, estressado e aborrecido versus seguro, gratificado, satisfeito, relaxado e complacente você se sentiu durante a tarefa?


Baixa  Alta

[Continuar >>](#)

 ADMINISTRAÇÃO

- nasatix administration
  - Editar configurações
  - Funções designadas localmente
  - Permissões
  - Verificar permissões
  - Filtros
  - Logs
- Administração do curso
- Mudar papel para...
- Minhas configurações de perfil
- Administração do site

[Buscar](#)

 NAVEGAÇÃO

NASA-TLX 1<sup>st</sup> step: evaluating the demands for each scale after completing one task

## NASA-TLX Protocol

The screenshot shows a web application interface for the NASA-TLX protocol. At the top, there is a header bar with 'LiNE' and navigation links like 'Meu "Painel de Bordo"' and 'Cursos'. The main content area is titled 'Responda ao questionário' and contains a sub-header 'Questionário da tarefa - Parte 2'. Below this, there is a text prompt: 'Clique no fator que representa o fator de maior contribuição para a carga de trabalho da tarefa.' Two buttons are presented: 'Demanda mental' and 'Esforço'. To the right of each button is a descriptive text. The 'Demanda mental' button is associated with the text: 'Quanto das atividades mental e perceptual foram exigidas (por exemplo: pensar, decidir, calcular, lembrar, observar, procurar, etc)? A tarefa foi fácil ou difícil, simples ou complexa, rigorosa ou tolerante?'. The 'Esforço' button is associated with the text: 'O quão duro você teve de trabalhar o (mental e fisicamente) para alcançar o seu nível de desempenho?'. On the right side of the interface, there is a sidebar with two sections: 'ADMINISTRAÇÃO' and 'NAVEGAÇÃO'. The 'ADMINISTRAÇÃO' section includes links like 'nasatlx administration', 'Editor configurações', 'Funções designadas localmente', 'Permissões', 'Verificar permissões', 'Filtros', 'Logs', 'Administração do curso', 'Mudar papel para...', 'Minhas configurações de perfil', and 'Administração do site'. The 'NAVEGAÇÃO' section is currently empty.

LiNE Meu "Painel de Bordo" Cursos Você acessou como Romenig Ribeiro (Sair)

PÁGINA INICIAL / MEUS CURSOS / MOOC / PROG2014-T2 / 1. ALGORITMOS / NASA-TLX - COMO FORAM AS PRIMEIRAS ATIVIDADES?

### Responda ao questionário

Questionário da tarefa - Parte 2

Clique no fator que representa o fator de maior contribuição para a carga de trabalho da tarefa.

**Demanda mental**

ou

**Esforço**

Quanto das atividades mental e perceptual foram exigidas (por exemplo: pensar, decidir, calcular, lembrar, observar, procurar, etc)? A tarefa foi fácil ou difícil, simples ou complexa, rigorosa ou tolerante?

O quão duro você teve de trabalhar o (mental e fisicamente) para alcançar o seu nível de desempenho?

#### ADMINISTRAÇÃO

- nasatlx administration
  - Editor configurações
  - Funções designadas localmente
  - Permissões
  - Verificar permissões
  - Filtros
  - Logs
- Administração do curso
- Mudar papel para...
- Minhas configurações de perfil
- Administração do site

Buscar

#### NAVEGAÇÃO

NASA-TLX 2<sup>nd</sup> step: pairwise choice, giving each scale a weight that represents the importance of each scale while accomplishing a task

## The course

- The course had last for 4 weeks and was delivered fully online
- Since the environments were Java dependents, students had instruction on how to install Java by video tutorials for Windows and Linux, with Chrome and Firefox examples
- A small video (our production) with the history of computers were presented
- The course was organized in 4 blocks
- Every theoretical content for each block was delivered as hypertext with the very same textual content, but with different images concerning the environment (iVProg and VPL)



## Curriculum

Theoretical and practical content were distributed into the blocks as follows:

- **Block 1:** "Algorithms- definition of algorithms and basic concepts of programming (variables and their types, data input and output etc.)"
- **Block 2:** "Selection" and it is composed of comments about the previous module, definition of selection with examples.
- **Block 3:** "Looping Constructs" and it is composed of comments about the previous module, definition of the looping constructs while, for and repeat
- **Block 4:** "Closing" and it is composed of complex activities involving the content of the previous modules and discursive activities related to the course as a whole and a final NASA-TLX activity

## Enrollment

Volunteers registered through web. The propaganda was performed for a short period of time (4 weeks), and mainly restricted to the University of São Paulo (USP).

Group	System	With experience	Without experience	Total
G1	iVProg	31	41	72
G2	VPL	31	41	72
Total		62	82	144

## Enrollment: no show

The number of students that never accessed the system

Group	System	With experience	Without experience	Total
G1	iVProg	9	16	25
G2	VPL	7	14	21
Total		16	30	46

# Experiment setup

## Enrollment: last week of the course

A really small number of students had concluded the course and made the activities

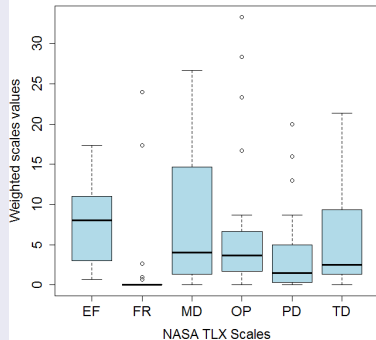
Group	System	With experience	Without experience	Total
G1	iVProg	3	3	6
G2	VPL	2	8	10
Total		5	11	16

## Data analysis

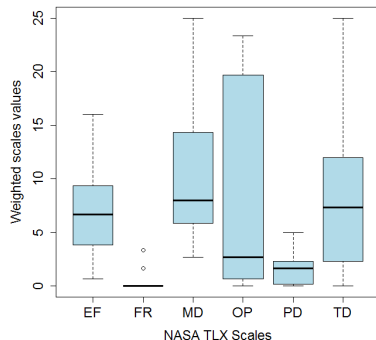
- NASA-TLX: Wilcoxon-Mann-Whitney (WMW) test and comparisons with medians
  - reason: low number of respondents, besides the apparent nonsymmetrical distribution of data
  - brief: WMW test consists of defining ranks based on the samples values. The higher the sample ranks, the higher is the values in the distribution
  - hypothesis:
    - $H_0 : \text{distribution}_{G1} = \text{distribution}_{G2}$
    - $H_1 : \text{distribution}_{G1} < \text{distribution}_{G2}$
    - exception: scale *Own Performance* (OP), where  $H_1 : \text{distribution}_{G1} > \text{distribution}_{G2}$
- Activities: quantitative analysis describing the number of attempts

## NASA-TLX Block 1

**NASA TLX Weighted scales G1 Block 1**



**NASA TLX Weighted scales G2 Block 1**



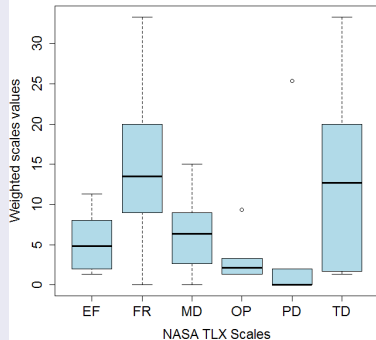
	EF	FR	MD	OP	PD	TD
<i>p</i> -value	0.6409	0.6676	0.1167	0.3002	0.8272	0.1132

## Attempts Block 1

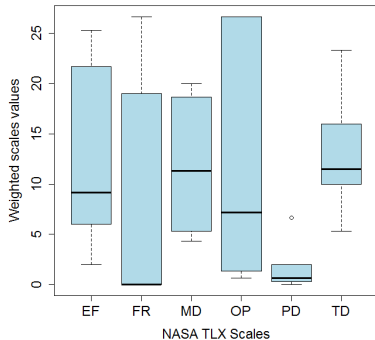
- G2 group: (C+VPL, textual programming), the maximum number of attempts were 15 times and it was not uncommon to find a number greater than five attempts
- G1 group: (iVProg) used 4 attempts at most (only 1 student), more than that, the most common situation was the student submit the correct answer in first trial

## NASA-TLX Block 2

**NASA TLX Weighted scales G1 Block 2**



**NASA TLX Weighted scales G2 Block 2**



	EF	FR	MD	OP	PD	TD
<i>p</i> -value	0.07441	0.8935	0.1473	0.2071	0.2023	0.532

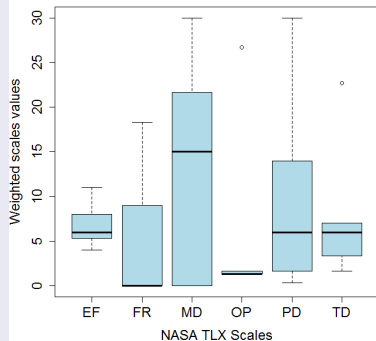


## Attempts Block 2

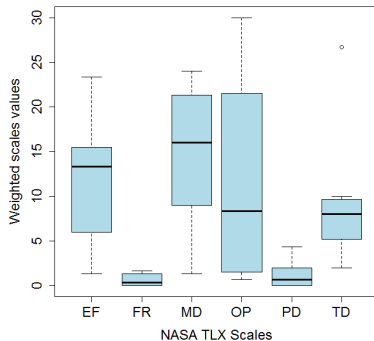
- G2 group: the maximum number of attempts to solve a problem was 12 and the most common situation is the use of 5 attempts
- G1 group: maximum number of attempts was 4 and the most common situation was students sending the correct answer in their first trial

## NASA-TLX Block 3

**NASA TLX Weighted scales G1 Block 3**



**NASA TLX Weighted scales G2 Block 3**



	EF	FR	MD	OP	PD	TD
<i>p</i> -value	0.1452	0.6028	0.4353	0.7435	0.9642	0.6028

## Attempts Block 3

- G2 group: the number of submissions was slightly higher than in G1 (6 students), however, the number of attempts reached the maximum of 17
- G1 group: regarding the number of submissions for G1 few students (5) did the activities, however, their submission was correct on only 1 attempt for all the students

## Discussion: NASA-TLX

The NASA TLX also showed that, in some cases, users have shown a little bit frustrated during the execution of the proposed tasks. In order to understand this phenomenon and, moreover, to collect qualitative data about the web-course we designed a simple online survey

- (unfortunately) the number of NASA TLX submissions for the forth block were not sufficient to make any comparison
- NASA TLX also showed that, in some cases, users have shown a little bit frustrated during the execution of the proposed tasks

To understand this phenomenon and, moreover, to collect qualitative data about the web-course we designed a simple online survey

# Discussion, conclusions and future work

## Discussion: online survey

- 1 If you have not accessed the course system or did not accomplished the module I could share the reason with us? If yes, fill out the form below telling us why.
- 2 If you did the activities and read the instructional material, do you have any suggetion of improvement to the environment or to the material?
- 3 What is your opinion about the tool used to create algorithms?

## Discussion: answers

The survey questionnaire was answered by 26 students (23 accessed the course and 3 did not). The main reason to frustrations were problems with the Java applet technology and the difficulties with its configuration.

## Discussion: perceptions and considerations

Considering all collected data, from NASA TLX, activities log, and the survey, we can observe that visual programming is a good model to teach algorithms and programming. However the low number of respondents do not allow stronger assertions.

## Future work

- Since the reduced number of enrolled students prevented us of any statistical conclusions, we intend to perform a new course edition, this time as MOOC
- Analyse if a new version of iVProg, now implemented using HTML5 technology can reduce the students frustrations with Java security issues
- Besides, this first course edition comparing visual with textual programming arose several questions that must be investigated in future.
  - One of them is how to compare the effective learning. Is it possible to compare both models?

Questions?

## Presenter

Leônidas de Oliveira Brandão

leo@ime.usp.br

DCC - IME - USP

Room 212 - Block C - 55 11 3091 9691

Rua do Matão, 1010 - Cidade Universitária - São Paulo - SP



# References



[1] Carlisle, M. C. (2009)

Raptor: a visual programming environment for teaching object-oriented programming

*Journal of Computing Sciences in Colleges*, Vol. 24, issue 4, April 2009, pp. 275-281.



[2] Kölling, M. (2010)

The Greenfoot Programming Environment

*Transactions on Computing Education (TOCE)*, Vol. 10, issue 4, November 2010.



[3] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010)

The Scratch programming language and environment

*Transactions on Computing Education (TOCE)*, Vol. 10, issue 4, November 2010.



[4] Cooper, S., Dann, W., Pausch, R. (2000)

Alice: a 3-D tool for introductory programming concepts

*Journal of Computing Sciences in Colleges*, Vol. 15, issue 5, May 2000, pp.107-116.



[5] Hart, S. G., Staveland, L. E. (1988)

Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research

*P. A. Hancock and N. Meshkati (Eds.) Human Mental Workload. Amsterdam: North Holland Press.*