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

Romenig Ribeiro[1], Tulio Faria[2], Leônidas Brandão[1] and Anarosa Brandão[3]

[1]Institute of Mathematics and Statistics - IME - USP

[2]School of Arts, Sciences and Humanities - EACH - USP

[3]Polytechnic School - USP

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

Frontiers in Education, October 2014

University of São Paulo

Institute of Mathematics and Statistics (IME)

University of São Paulo

Polytechnic School (POLI)

University of São Paulo

School of Arts, Sciences and Humanities (EACH)

Overview

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

Introduction

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 interprete) their algorithm with a single click
  - build programs without "knowing by heart" the programming language sintax
  - consequently, they may focus their attention to the problem solution

Introduction

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

Background: iVProg, VPL and NASA-TLX

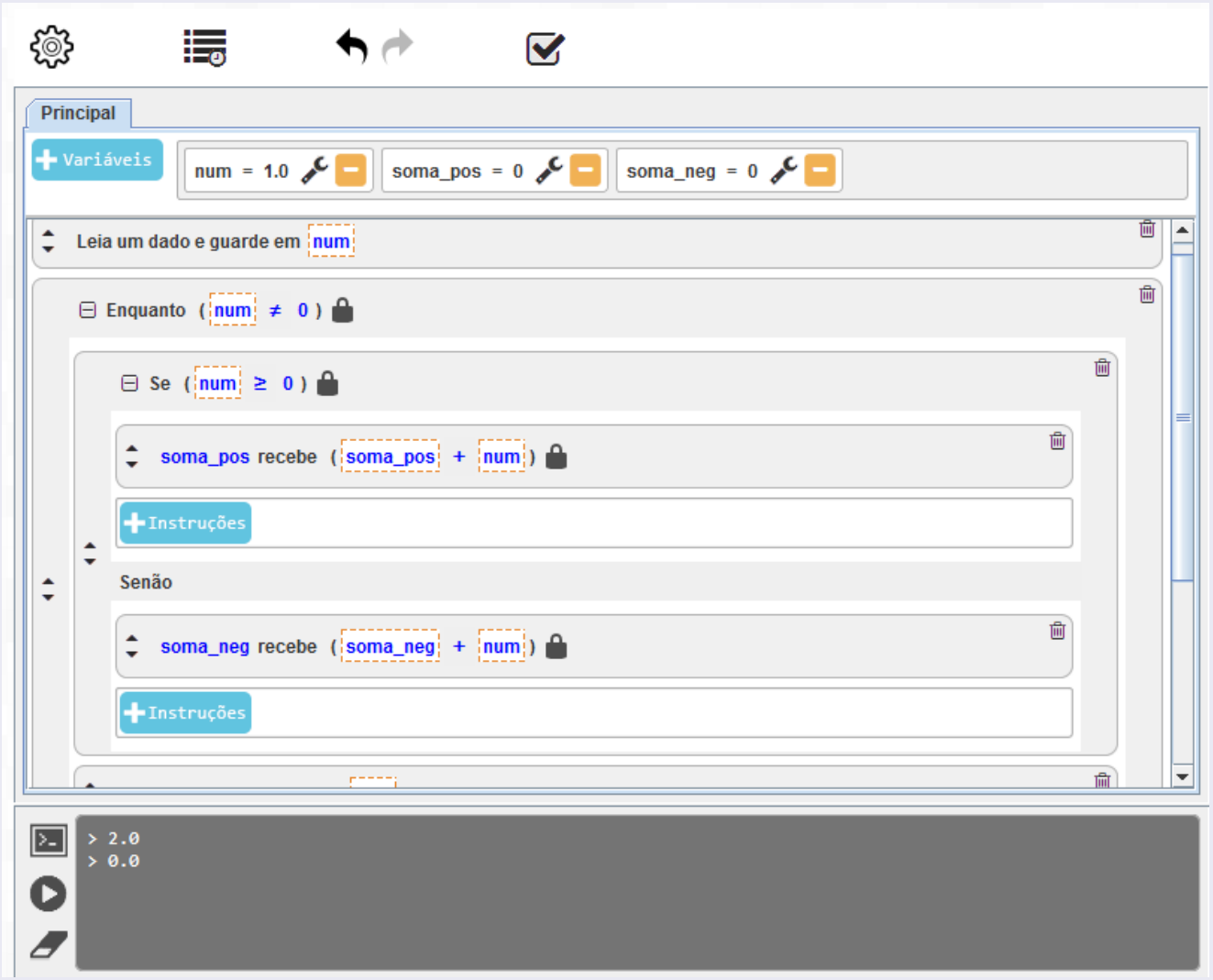
iVProg

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



Background: iVProg, VPL and NASA-TLX

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/

Background: iVProg, VPL and NASA-TLX

Virtual Programming Lab - VPL

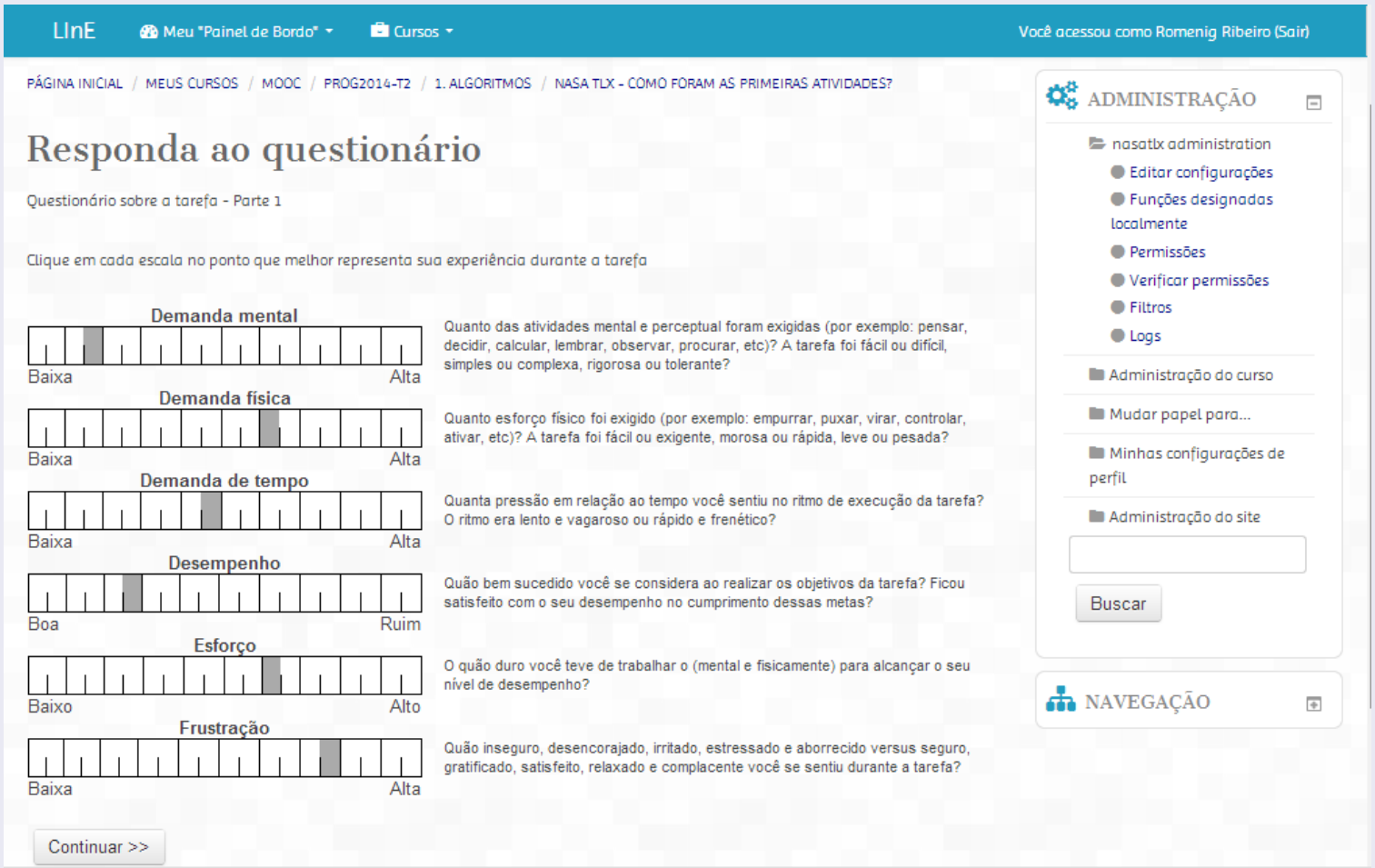
Background: iVProg, VPL and NASA-TLX

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 questionaire 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)
  - Teporal Demand (TD)
  - Own performance (OP)
  - Effort (EF)
  - Frustration (FR)

Background: iVProg, VPL and NASA-TLX

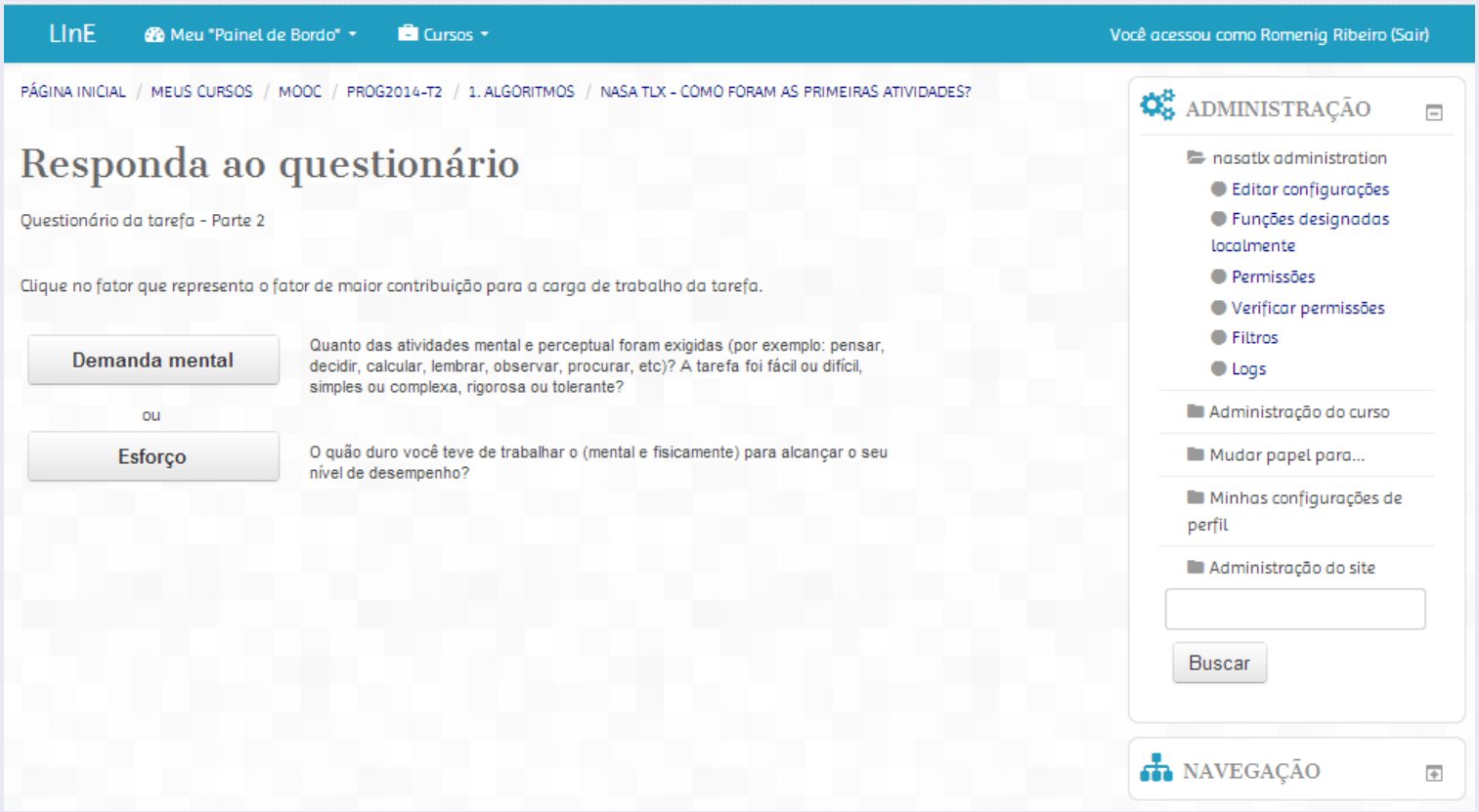
NASA-TLX Protocol



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

Background: iVProg, VPL and NASA-TLX

NASA-TLX Protocol



NASA-TLX 2^{nd} step: pairwise choice, giving each scale a weight that represents the importance of each scale while accomplishing a task

Experiment setup

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)

Experiment setup

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

Experiment setup

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

Experiment setup

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 os 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

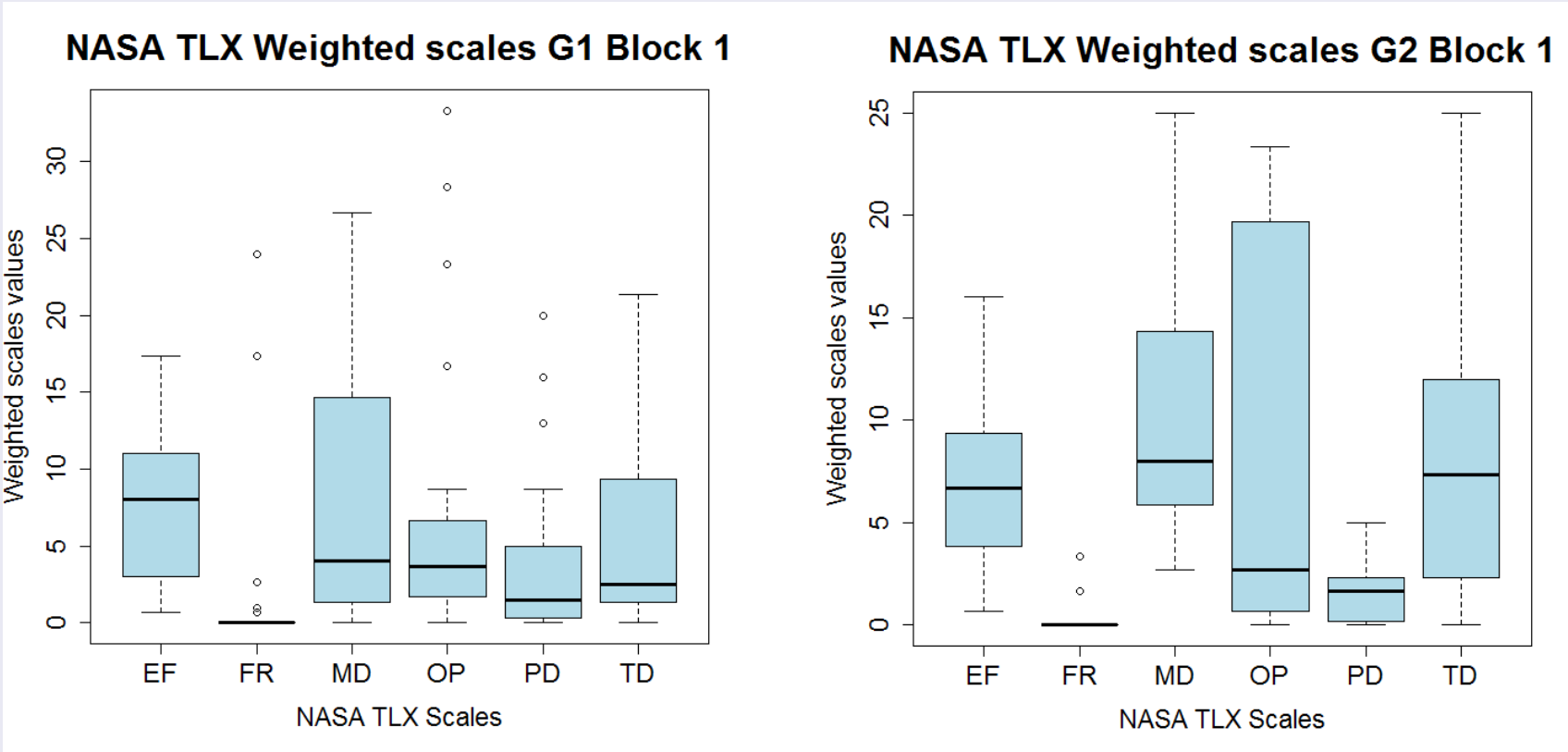
Methodology

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:
    - $H_0 : distribution_{G1} = distribution_{G2}$
    - $H_1 : distribution_{G1} < distribution_{G2}$
    - exception: scale *Own Performance* (OP), where $H_1 : distribution_{G1} > distribution_{G2}$
- Activities: quantitative analysis describing the number of attempts

Results

NASA-TLX Block 1



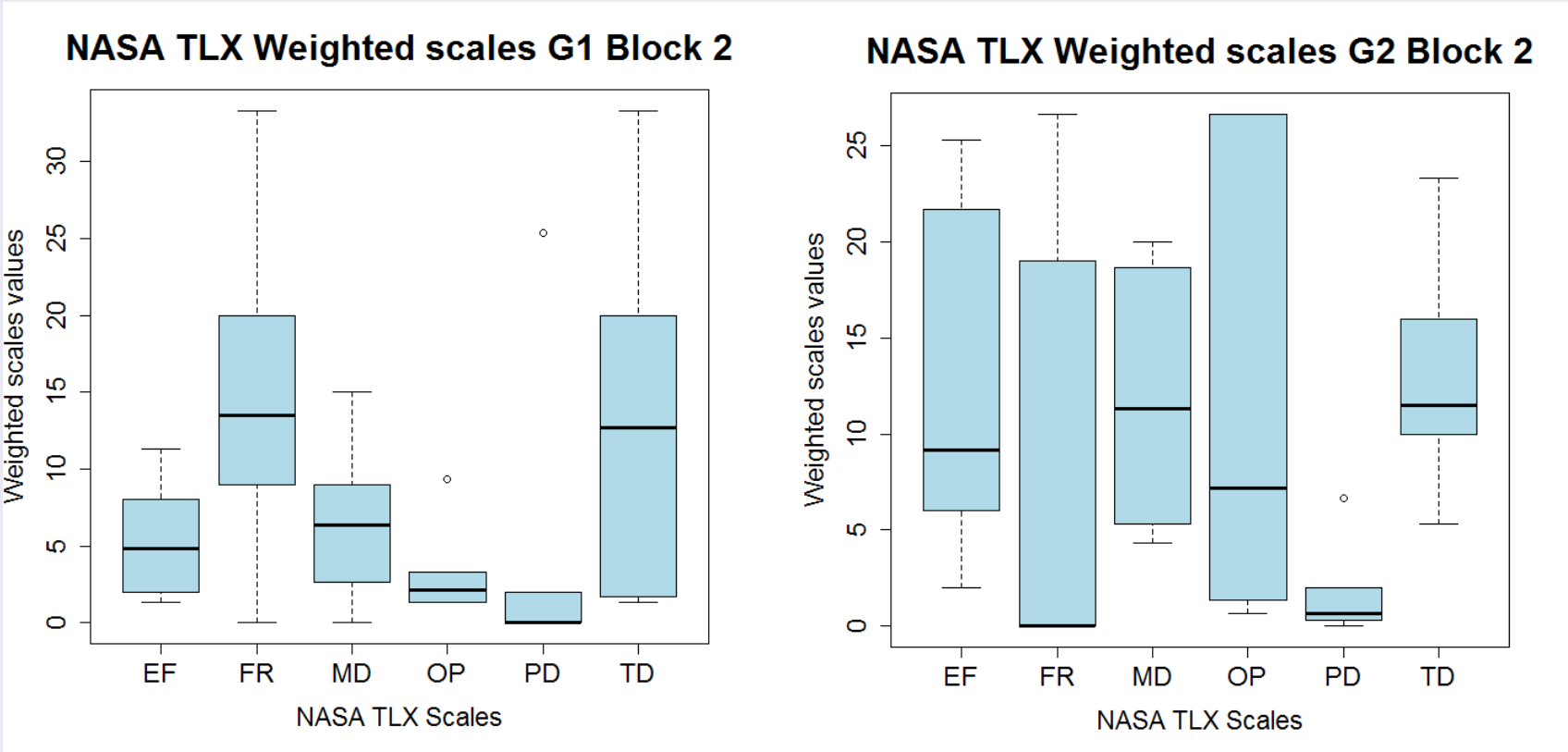
	EF	FR	MD	OP	PD	TD
p-value	0.6409	0.6676	0.1167	0.3002	0.8272	0.1132

Results

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



	EF	FR	MD	OP	PD	TD
p-value	0.07441	0.8935	0.1473	0.2071	0.2023	0.532

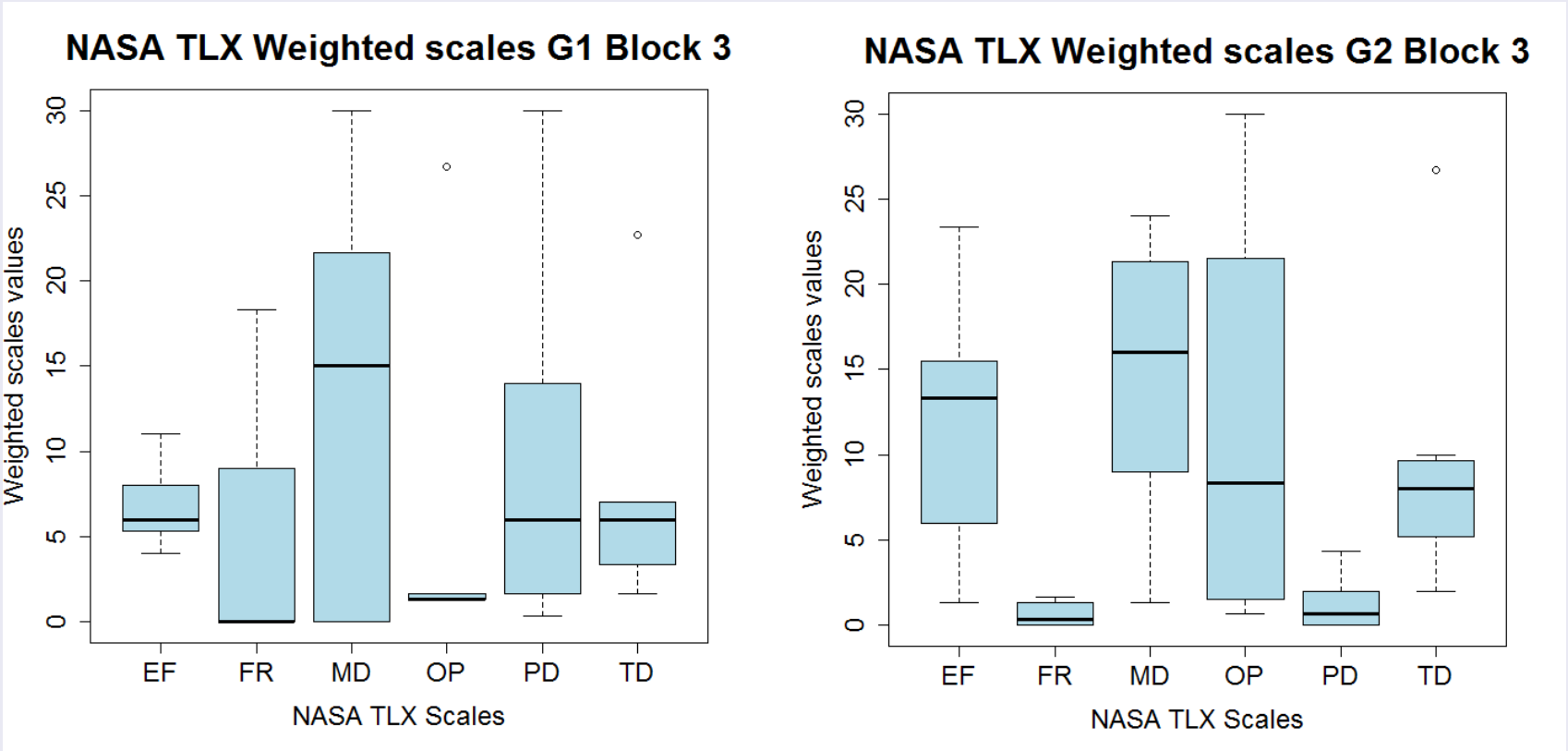
Results

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

Results

NASA-TLX Block 3



	EF	FR	MD	OP	PD	TD
p-value	0.1452	0.6028	0.4353	0.7435	0.9642	0.6028

Results

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, conclusions and future work

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, conclusions and future work

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.

Discussion, conclusions and future work

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?

Discussion, conclusions and future work

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.

References

[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.