Learning Computer Science Languages in Enki

José Carlos Paiva
CRACS & INESC-Porto LA &
DCC - Faculty of Sciences,
University of Porto
Porto, Portugal
up201200272@fc.up.pt

José Paulo Leal
CRACS & INESC-Porto LA &
DCC - Faculty of Sciences,
University of Porto
Porto, Portugal
zp@dcc.fc.up.pt

Ricardo Queirós
CRACS & INESC-Porto LA &
DI/ESEIG/IPP
Porto, Portugal
ricardoqueiros@eseig.ipp.pt

ABSTRACT

This paper presents an overview and main features of Enki, a web-based learning environment for computer science languages. Enki was designed to be a sort of entry level IDE, aggregating tools for navigating and viewing course materials, for solving exercises and receiving automated feedback, as well as promoting the learning process. Enki uses services from several other systems, namely for content sequencing and recommendation, exercise assessment, and gamification.

Keywords

Learning; Integration; Gamification; Educational Resources; Sequencing; Exercises; Programming Languages

General Terms

Languages, Design, Experimentation

1. INTRODUCTION

Enki is a web integrated environment for learning computer science languages, such as programming languages or diagrammatic languages. This environment is one of the GUIs of Mooshak 2.0 (the new version of Mooshak [2]), a framework for automated assessment of computer science languages. Enki was designed for a wide range of use cases, from introductory high school or college courses, to massive online open courses. It assumes that the students may have little or no help from a teaching assistant and that they may not have the necessary tools installed on their computers. In fact, students may be accessing the Internet through a tablet rather than a computer, where these tools would be virtually impossible to install. In spite of these constraints, it aims to prepare students to the development environments typically used in computer, namely IDEs.

Enki can be described as an entry level IDE, a scaffolding to support the progress of students towards more complex environments and tools. It has simplified versions of common IDE tools, such as editors or launchers, and new tools required by a learning environment, such as a resource browser or a content recommender. Enki borrows from IDEs both the role of a tool integrator and the characteristic graphic user interface (GUI).

A typical IDE, such as Eclipse or NetBeans, organizes its GUI in *regions*, each one containing several overlapping windows organized using tabs. The layout scheme for these regions locates them roughly in the main directions. Regions are resizeable and the windows they contain can be moved among regions, according to the needs and preferences of students. The remainder presents the main regions and tools as shown on the screen shot in Figure 1.

2. LEFT REGION

The left region of the GUI, with a purple highlight in Figure 1, contains a *resource browser*. In reference IDEs this region usually contains browsers that provide different views of the workspace, either organized in projects, packages and classes, or organized in directories and files. In Enki this region is used for browsing pedagogical resources using a tree widget that mimics those used by IDE browsers.

The resource browser is a particularly important part of Enki's GUI as it drives student interaction by presenting both the course structure and content. The view of course unfolds as the student progresses throughout the course. This view is mediated by Seqins [3], a learning content sequencing engine. Tree leaves may hold educational resources of different types: text (HTML or PDF), multimedia and activities (exercises). They are presented in the tree with an icon reflecting its type and a color depending on its state relative to the student: available, solved/seen, unavailable or recommended.

3. CENTER REGION

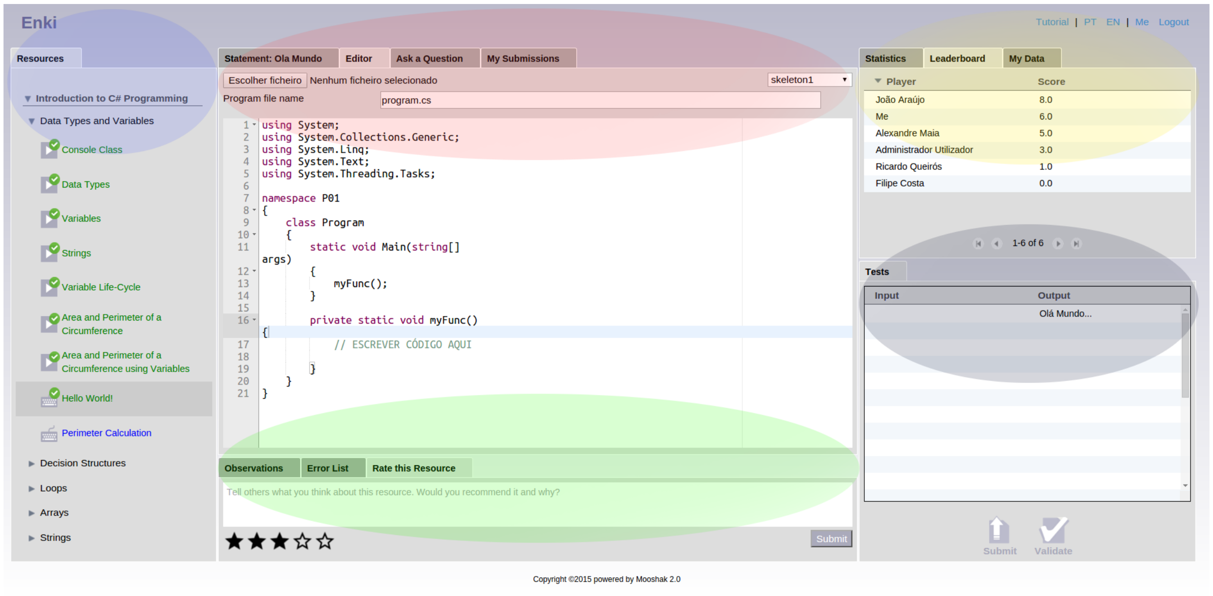
The center region, highlighted in magenta in Figure 1, is the main region of Enki's GUI. By default, the expository and evaluative resources selected in the browser will be open here in one or more tabs.

Expository resources are presented in specialized viewers. For instance, a sequence of steps can be illustrated by an embed video hosted in YouTube, and reference material can be presented in PDF or HTML formatted pages.

Evaluative resources open several windows: to show the exercise statement, to edit a solution and to evaluate it. Currently Enki supports two editors for different kinds of computer science languages: a code editor for programming languages such as Java or C#, and a diagram editor for diagrammatic languages such as EER or UML.

ITiCSE '16 July 09-13, 2016, Arequipa, Peru

ACM ISBN 978-1-4503-4231-5/16/07.
DOI: http://dx.doi.org/10.1145/2899415.2925504



Nenhum ficheiro selecionado

skeleton1

Program file name: program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace P01
{
    class Program
    {
        static void Main(string[]
args)
        {
            myFunc();
        }

        private static void myFunc()
{
            // ESCREVER CÓDIGO AQUI

        }
    }
}
```

Tell others what you think about this resource. Would you recommend it and why?

★★★☆☆

Player	Score
João Araújo	8.0
Me	6.0
Alexandre Maia	5.0
Administrador Utilizador	3.0
Ricardo Queirós	1.0
Filipe Costa	0.0

1-6 of 6

Input	Output
	Olá Mundo...

Figure 1: Interface of Enki for students with different regions highlighted in different colors

4. UPPER RIGHT REGION

The upper right region is highlighted in yellow in Figure 1. It aggregates gamification related windows, such as the leaderboard and the achievements, and windows that summarize global or personal progress, such as problem statistics and profile data. These windows are not always present, it depends on the selected resource type.

The gamification related windows use the gamification features provided by Odin [1], a gamification service inspired in the Google Play Game Service (GPGS) which authenticates institutions rather than end-users, and the others may use services provided by Seqins [3], a sequencing tool of digital educational resources which includes a flexible sequencing model that fosters students to learn at different paces.

5. BOTTOM RIGHT REGION

By default, the bottom right region is just available, in exercises and activities. In this case, it contains the window to write input test cases to execute with the exercise, where corresponding outputs are later on presented, and buttons to validate or submit the solution.

The validation of an exercise consists in taking the inputs provided by the student, and/or public inputs (written by the exercise author), and running the program with these inputs. This kind of evaluation is not graded. If the test case is public, it will be highlighted in green (if the output matches) or red (if it does not match). If the input is provided by the student, the output obtained with the program with that input is written in the grid.

A submission follows a similar process but uses a complete set of test cases provided by the exercise author. It sends the program for assessment and returns the a grade and a feedback. Unlike a validation, a submission is considered for statistics and can unlock new resources of the course.

6. BOTTOM REGION

The windows presented in this region (highlighted in green in Figure 1) depend on the selected resource type. In exercises and activities, three windows are shown in different tabs: the observations window which contains automated feedback on an exercise, the error list window that summarizes the compilation/syntax errors and warnings, and a window that allows the student to rate and/or comment a resource. This last window is also shown on educational resources, as well as one with links to related resources.

7. ACKNOWLEDGMENTS

Work funded by the ERDF and FCT Project NORTE-01-0145-FEDER-000020 and POCI-01-0145-FEDER-006961; and by IPP trough Project BInt-ICD/IPP-BST/KMILT/01/2015.

8. REFERENCES

[1] J. P. Leal, J. C. Paiva, and R. Queirós. Odin: A service for gamification of learning activities. In *to appear*, page to appear. Springer, 2015.

[2] J. P. Leal and F. Silva. Mooshak: a web-based multi-site programming contest system. *Software: Practice and Experience*, 33(6):567–581, 2003.

[3] R. Queirós, J. P. Leal, and J. Campos. Sequencing educational resources with seqins. 2014.