SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects
A.Sc.2 - Development

Contents

2016-2017

Version 1.0
Last update: 29/11/2016
Use: Students/Staff
Author: Samuel CUELLA

TABLE OF CONTENTS

1. Project Overview

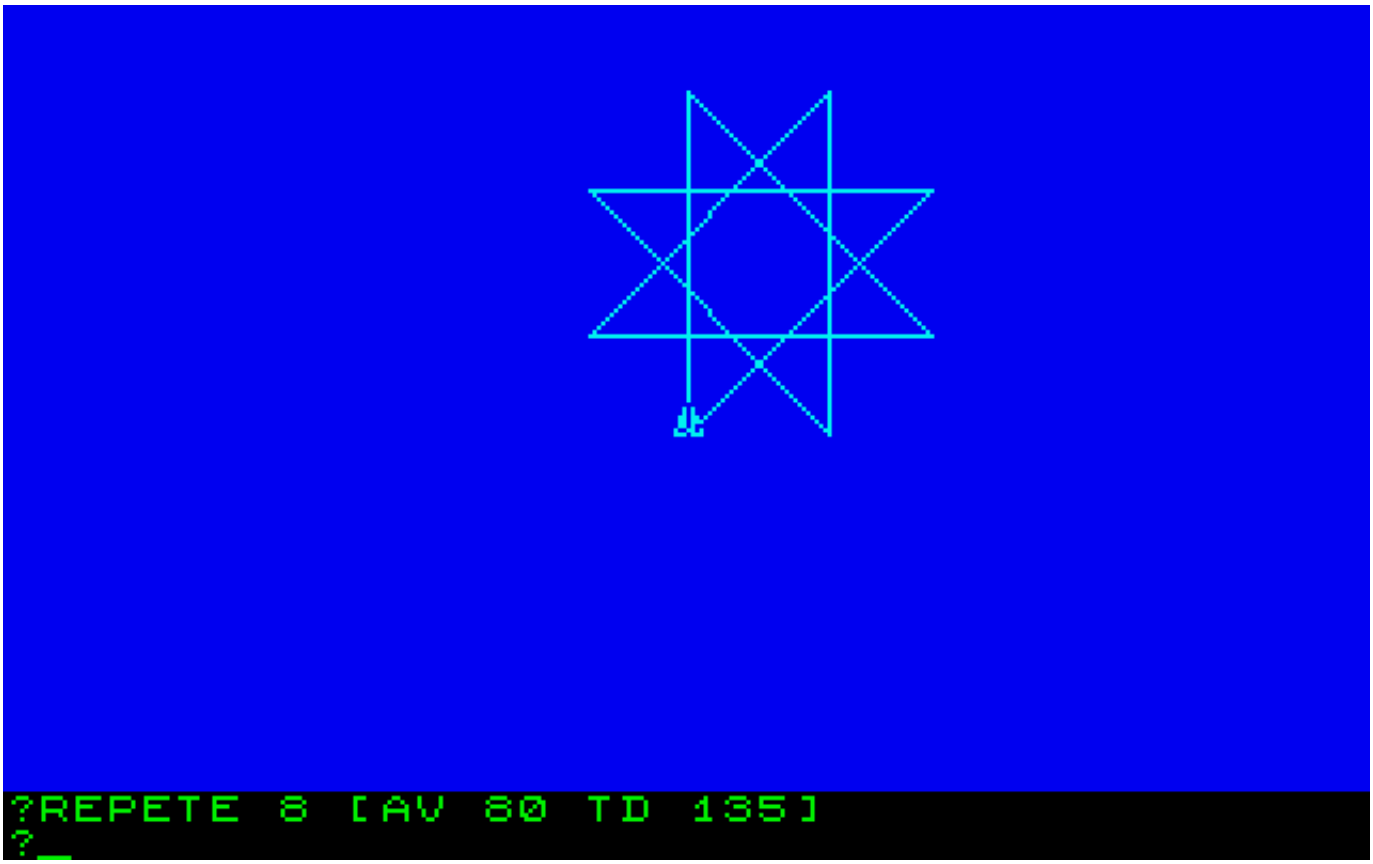
"The Good Old Days" is an entertainment company specialized in all times video games classic revivals. They now want to rewrite a really fun programming language: LOGO.

Your team has been chosen among several subcontractors to do the development and you are free to use whichever language/library you want, such as Python/Pygame or C/SDL. Your implementation must work on the three major platforms: Linux, Windows and Mac OS X.

2. Functional Expression

2.1. The Language

LOGO is a language that let users order a turtle around. Except that the turtle is a triangle. You can enter commands for the turtle to perform. The result of your commands are usually a nice picture... or a complete mess :



The screen is divided in two parts:.

- Graphic window
- Interactive shell

The user type his commands in the interactive shell while the turtle draws the result in the graphic window. The shell prompt is '?'. Example:

```
?REPETE 8 [AV 80 TD 135]
```

LOGO being a rather complex language, you're only required to implement a subset of commands which is:

Table 2.1. Commands to implement

Command	Feature
AV *pixels*	The turtle moves *pixels* foward
RE *pixels*	The turtle moves *pixels* backward
TD *degrees*	The turtle turns *degrees* to the right
TG *degrees*	The turtle turns *degrees* to the left
FCC *color*	Change the trace color to *color* in RGB format as #FF0000 for red.
LC	Pen up (no trace)
BC	Pen down (trace active)
VE	Clears the screen and put the turtle at the center, facing upwards.
CT	Hide the turtle
MT	Show the turtle
REPETE *times* [*commands*]	Do the *commands times* times.

When starting the program, the turtle is centered on the screen and faces upwards.

2.2. Bonus

As a bonus, you can implement the ability to define procedures that can be used later. Procedures are defined as follows:

```
?POUR name :param
>instruction
>instruction
>FIN
```

Name and paramters being defined by the user. For example:

```
?POUR square :size
>REPETE 4 [AV :size TD 90]
>FIN
?square 100
```

3. Deliverables

Students should include the following elements in their final delivery:

- A zip archive with the project source code. The source code must also come with the build system used (Project file, autotools...), if any.
- Project documentation, based on the template.
    - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
        - Grammar recognition
        - Loop handling
    - Game manual

The first document is an academic document. Address the reader as a teacher, not a client. This document can be in French or in English, as you wish.

4. Graded Items

The project will be graded as follows, on a 25/20 scale:

- Interface (5 points)
  - The interace has both a text shell and a graphic window (1.5 points)
  - The user can enter commands in the shell (1 point)
  - The graphic window show the step-by-step execution of the entered commands (2.5 point)
- Language (15 points)
  - All the primitives work as expected (10 points)
  - The repete construct works (5 points)
- Bonus features (5 points)
  - Bonus features done by the students (5 points)