

vsep: Simple educative programmms

Manuel Gutiérrez Algaba

21st November 2000

Version 0.1.0

Abstract

vsep is a set of programmms that runs simple exams. It's composed of several independent modules.

0.1 jpypsitechnic

This is a Java applet that performs IQ tests. Currently it supports the 'how-many-clocks' IQ test. It uses Java 1.1. You need appletviewer or a Java compliant browser (Netscape).

0.2 Old vsep

It's ready for exams of whatever you may think of. Actually you can practice math and Spanish, but the current examples are *only* examples, it depends on new additions (of people) to be more usable.

It uses *tcl/tk* so you can use it in Linux and in most platforms you can imagine.

1 Synopsis

vsep-simple-math

vsep-spanish

gipsytechnic

2 Description

gipsytechnic : The shell script that launches the appletviewer of jpypsitechnic. You can launch it directly pointing your browser to: `/usr/lib/vsep-0.1.0/IQ.htm`. (or `/usr/local/lib/...`). Basic instruction/description:

- You can see several rows of clocks.
- The last row has only a clock. That clock is the pattern.

- You have to guess how many clocks in the other rows have the same time that the 'pattern'.
- So, you click the button whose label is the number of clocks similar to the pattern.
- Once you clicked a number labelled button, the answer is stored, and a new game is shown (or the same if there's no more games).
- If you press the 'next' button. You skip the current test.
- If you press the 'end' button, you'll see how many tests you have guessed correctly.
- The quit button kills the buttonry windows.
-

vsep-simple-math : Runs a short number of exercises of Math.
vsep-spanish: Runs a short number of exercises of Spanish.

In both cases, you'll see a window with some buttons. At the top row you'll find the "problem" to be solved, and in the second row and in the vertical column you can choose a possible solution.

The different exercises are presented and solved till there's no more exercises and you quit the game. You can't get back to a previous exercise.

3 Files

3.1 jpypsitechnic

- It's made of the shell script that launches it.
`/usr/bin/gypsitechnic`
- The jar of the applet.
`/usr/lib/vsep-0.1.0/IQ.jar`
- The HTML of the applet.
`/usr/lib/vsep-0.1.0/IQ.htm`

Or `/usr/local/lib/...`

3.2 old vsep

As a matter of fact *vsep-simple-math* and *vsep-spanish* are just simplified versions of the following command :

```
tcl /usr/lib/vsep/vsep.tcl -g simple_math  
tcl /usr/lib/vsep/vsep.tcl -g spanish
```

So you can invoke them directly.

The exams are in that directory or they should if have installed this as a .rpm, otherwise, run it depending on your installation.

3.3 Local files

jpypsitechnic

None.

old vsep

If you have a look at the files `game1.tcl` and `game2.tcl`, you'll see that there is contained the data of the exams(math and spanish, respectively). You can have a

`/home/myuser/game1.tcl`

for having more exercises added by yourself. If such files doesn't exist then it will only use the default files.

4 Program internals: Only for programmers

4.1 jpypsitechnic

Well, there two aspects here. The Java and the python. The Java that is the game itself.

- `Answers.java` `Draw.java` `Reloj.java` `StepGame.java`
- `Buttonry.java` `GameIQ.java` `RowClocks.java`
- `Clock.java` `HandsMaths.java` `SolutionRecorder.java`
-

The Python part, that is `IQ.py`, it's a program that generates the file `GameIQ.java`. That is, `IQ.py` is the generator of the game, so whenever you run `python IQ.py`, a new game (`GameIQ.java`) will be written, you'd have to compile it and make the Makefile for that.

4.2 old vsep

The script is composed of :

- `vsep.tcl` : It parses the args and choose the files for the games.
- `vsepmain.tcl`:It contains all the subroutines for the user interface.
- `l_exer.tcl` : Establish the format the exercises are.
- `game1.tcl`: List of exercises of math.
- `game2.tcl`: List of exercises of spanish.

4.3 Format of an exercise

The scheme is rather flexible, and it allows many possibilities. Right now, you can choose among two primitives:

```
gs 16 - 4 >12 11 6
```

and

```
vs "La casa está" "-" "-" ">en obras" "moviéndose" "comiendo"
```

gs It takes 6 parameters, param1, param2 and param3 will appear in the top row (problem description), the params 4 5 6 will appear in the second row (possible solutions). So the above example is telling , the problem description is “9 + 3” and the possible solutions are “12”, “11” and “6”, the right solutions (one or more) are marked with the > symbol prepended to the solution (leave no spaces, all together > and solution) .

vs vs is identical, except that it uses “vertical” column of solutions.

In the future (and if people helps) there'll be more exercises.

5 Wishlist

Well, comments, money, ... Well, I suppose a great set of exercises, more complex behaviour. Statistics, login of users...

6 Author

Manuel Gutierrez Algaba, algaba@seul.org. Released under GPL license. I have no responsibilities for misuse, harm, wrongdoing,... that may arise or be caused by the use of this software. If you use it, you take all the risks and responsibilities for it.