

Graphical Programming Language Application This assignment is to use what you have learned to produce a fairly complex program. It involves version control, unit testing, the use of design patterns and automatic documentation generation. The idea of the program is to produce a simplified environment for teaching simple programming concepts. You are to create a simple programming language that has the basics of sequence, selection and iteration and allows a student programmer to explore them using graphics.

1 Management --20 marks total

Set up version control (5 marks)

Early commit, project description

At least five commits of software

Unit Tests (10 marks)

Fully implemented as unit tests.

Appropriately documented

Agile Development (5 marks)

Evidence of Agile

2 Implementation -- 50 Marks total

Command parser class

Reads and executes commands on command line one at a time

Reads a program and executes it with a "run" command

Saves and loads a program (5 marks)

Syntax checking

Checks for valid commands

Checks for valid parameters (5 marks)

Basic drawing

Position pen, pen up/down, pen move

Draw basic shapes rectangle, circle, triangle (5 marks)

Draw advanced shapes polygon, texture, 3D (5 marks) -

Programming commands

Simple repeat command with plus or minus parameter (5 marks)

E.g. repeat 4 circle + 10 (draws 4 circles centered at the same point getting bigger by 5 pixels each time) This should work for all shapes

Variables - allow loops to define variables to pass as parameters to draw commands (5 marks)

Loop command (10 marks)

Repeats everything between Loop on the first line and "end" on a later line.

If statement (10 marks)

5 marks for one line

5 marks for block with "endif"

3 Design and Implementation Standard --20 marks total

Use of design patterns - factory class (10 marks)

All shape classes should use appropriate inheritance but should also use the factory design pattern.

Code documented with XML tags, XML documentation produced (5 marks)

Use of exception handling (including user generated exceptions) (5 marks)

4 Additional functionality --10 marks total

Here you can come up with your own functionality. Here are some suggestions but you are free to come up with your own, however you should discuss them with your tutor first.

Additional commands, one example might be to transform/rotate shape, more complex shapes and the drawing of shapes.

Methods. Allow your program to define and call methods.

Full user interface with graphical "Visual Studio-like" from end for graphic programming that allows drag and drop but write the code.

Command Examples

The pen position is stored in the drawing object. Commands should not be case sensitive.

DrawTo x,y

MoveTo x,y

Circle <radius>

Rectangle <width>, <height>

Triangle <base>, <adj>, <hyp>

Polygon [points,...]

Complex commands

repeat 4 circle radius +10 (would draw 4 circles with progressively 10 pixel bigger radii centered at the current pen position)

Radius = 20

Width = 20

Height = 20

Loop 4

Circle radius

Radius+10

Rectangle width, height

Width+10

Height + 10

Endloop

For a loop you should declare a standard variable "counter" which takes the iteration number and can be used in the program inside the loop

If counter = 5 then radius+25

| | |
|---|--|
| Part 1 Management 20 marks | |
| Version Control | |
| early commit | |
| good number | |
| good descriptions | |
| Unit Tests | |
| Appropriate tests | |
| Well documented | |
| Agile Dev | |
| design evidence | |
| Part 2 Implementation 50 | |
| Command Parser | |
| Reads and executes commands on command line one at a time | |
| Reads a program and executes it with a "run" command | |
| Saves and loads a program | |
| Syntax checking | |
| Checks for valid commands | |
| Checks for valid parameters | |
| Basic drawing | |
| Position pen, pen up/down, pen move | |
| Draw basic shapes rectangle, circle, simple triangle | |
| Draw advanced shapes polygon, texture | |
| Programming commands | |
| Simple repeat command with plus or minus parameter | E.g. repeat 4 circle + 10 (draws 4 circles centred at the same point getting bigger by 5 pixels each time) This should work for all shapes |
| Variables | allow loops to define variables to pass as parameters to draw commands (5 marks) |
| simple loop command | some sort of loop command |
| Full loop block | Repeats everything between Loop on the first line and "end" on a later line. |
| If statement one line | 5 marks for one line |
| If statement block | 5 marks for block with "endif" |
| Part 3 Design and Implementation Standard 20 marks | |
| Use of design patterns - factory class (10 marks) | All shape classes should use appropriate inheritance but should also use the factory design pattern. |
| Code documented with XML tags, XML documentation produced (5 marks) | |
| Use of exception handling (including user generated exceptions) (5 marks) | |
| Part 4 Additional functionality 10 marks | |

