

KPL, A Kid's Programming Language

An Announcement From Morrison Schwartz

KPL was designed and developed based on the principle that programming is fun. KPL is composed of:

- A readable and modular programming language
- A kid-usable but functionally complete IDE (integrated development environment)
- A growing collection of entertaining and educational sample programs

KPL (Kid's Programming Language) was developed by software development and consulting company Morrison Schwartz and is available as a freeware download from

<http://www.k-p-l.org>

Since the KPL release 6 months ago, volunteers have translated KPL into 15 additional languages, as well as provided translations of the website, tutorial materials, and sample programs. For code compatibility reasons, the programming language keywords remain English-only. KPL has been selected by the Washington Software Alliance as a 2006 Consumer Product of the Year finalist. Significantly, WSA does not have an award category for educational software.

1. The Programming Language

KPL is designed to be as simple and readable as possible, while conveying all the important principles of structured, modular and class-based programming. KPL lets beginners see eye-catching and immediate results from their programs while teaching them fundamental concepts like variables, data types, loops, decision structures, methods, functions, classes, and even interactive debugging. KPL's data types include integer, decimal, string, boolean, array, user-defined structures, and user-defined classes. Code Sample 1 shows a simple KPL program example.

The IDE automatically applies color to text based on language syntax as shown. One of KPL's most important features is the high level of abstraction provided in its library of classes, methods and functions, as demonstrated here for displaying and animating a graphical sprite across the screen. Maximum fun result from minimum simple code – this is a primary goal for KPL.

The Integrated Development Environment

KPL presents a modern and full-featured integrated development environment (IDE), whose features include:

- Windows-standard menus, toolbars, and mouse-over tooltips
- Many code editor features, such as syntax color coding, that assist learning as well as productivity
- Informational tooltips on mouse-over of variable and method names
- Multiple program files presented on tabs and shortcutted on the Window menu
- A program explorer pane for viewing and navigating the code hierarchically
- Dockable and pinnable IDE panes
- Collapsible and expandable code regions for all methods and functions—on mouse-over, collapsed regions show underlying code in a tooltip
- Message pane for trace, debug, and status messages

- C# and VB.NET code can be generated from KPL source with a button click

Code Sample 1. KPL source code for animating a sprite Program UFO

```
Method Main()  
  
LoadSprite( "UFO", "UFO.gif" )  
MoveSpriteToPoint( "UFO", 0, 0 )  
ShowSprite( "UFO" )  
  
Define Location As Integer  
For Location = 1 To 300  
    Delay(10)  
    MoveSpriteToPoint( "UFO", Location,  
                      Location )  
  
Next  
  
End Method  
  
End Program
```

The KPL IDE offers many usability advantages, and is carefully designed to prepare a beginner to “graduate” to a modern and professional IDE such as Eclipse or Visual Studio.NET.

In the screenshot shown in Figure 1, the mouse is over the method call to ColorRGB and a tooltip indicating the system method's signature is displayed. Also, the right-side pane has been unpinned, and is collapsed to the edge of the window. The program has just been checked for errors, as is shown in the Messages pane.

2. Entertaining and Educational Sample Programs

Learning is best achieved when the learning material is fun, intuitive and interactive—this was the basis for the collection of sample programs included with KPL. We are certain that this has been an important element of KPL's success, and that the contribution of dozens of additional educational or game examples from the global KPL community prove this point. Showing students immediate and lively graphical results from their programming work is a great way to get them (and keep them) excited and interested.

Of the many fun and exciting graphics capabilities KPL supports, Sprites offer the best graphical bang for a student's programming buck. Graphics files in many different formats can be used to create a Sprite, which can then be displayed, moved, animated, rotated, scaled, and can interact with other Sprites. This allows a student to create an interactive program without having to learn technical details such as window creation, file handling, device

contexts, alpha blending, collision detection, or any of the other details most programming languages and environments require.

In our experience, starting off a new programmer with their first program displaying a spaceship or other Sprite running around the screen—in only a few simple lines of code—is a great way to catch their interest, make them believe they can program, and make them want to learn more.

The sample program shown in Figure 2 is a fully functional interactive game with a scrolling screen, animated target Sprites moving at varying speeds, and a keyboard-controlled spaceship. This particular sample program, as functional as it is, is a KPL program of only 210 lines of code, plus considerable in-line commenting. Such a small but functional game demonstrates the graphical leverage and power offered by KPL—and it provides an excellent teaching code base that a student can modify and expand on to build more complex games.

KPL v 2, which will release in March 2006, adds the ability to do KPL-simple 3D programming. Our canonical 3D example shows a beginner how to interactively fly a 3D spaceship model through space, with only 35 KPL instructions.

We, and the global KPL community, are constantly implementing further sample programs that demonstrate the usefulness of KPL for teaching subjects other than computer science—in a way which is extremely engaging, interactive, and entertaining. For example, Figure 3 shows one math-oriented program result. KPL's graphical engine can be used effectively to teach concepts from geometry, algebra, trigonometry and physics. Even topics like reading comprehension, spelling and literature can be taught using fun and interactive KPL games. One of the secondary goals for KPL is to increase the availability of educational software by allowing teachers, parents and precocious students to develop that software themselves, no computer science degree required.

3. Call to Action

KPL is a freeware product that has served nearly 50,000 downloads in its first 6 months, with only word-of-mouth marketing. KPL includes a constantly-growing library of sample programs, image files usable as sprites, and sounds usable as program sound effects. Translators have made KPL available in 15 additional languages to date, and have translated website, tutorial, and example program code.

Our first call to action is for educators, parents, and students to download and work with KPL. Feedback is always welcome. We hope you will find it as exciting, engaging, and full of potential as we do. If you do, we ask that you help us spread the word about it.

KPL has been created and funded to this point by Morrison Schwartz, and will continue to be as we work to spread word about it and build interest in it. As a freeware educational product, KPL is actively looking for grant or other assistance that will help to pay for further enhancements, distribution, and user support.

KPL welcomes educational partners who will assist us with reviewing, enhancing, and further development of KPL's sample programs—in all subjects, not just computer science.

KPL is currently available in 16 languages, and we welcome assistance with further KPL language translations. KPL has implemented language support in a way that allows for fairly simple translation of the user interface strings—so the first level of language translation would require only a few hours of effort from the translator.

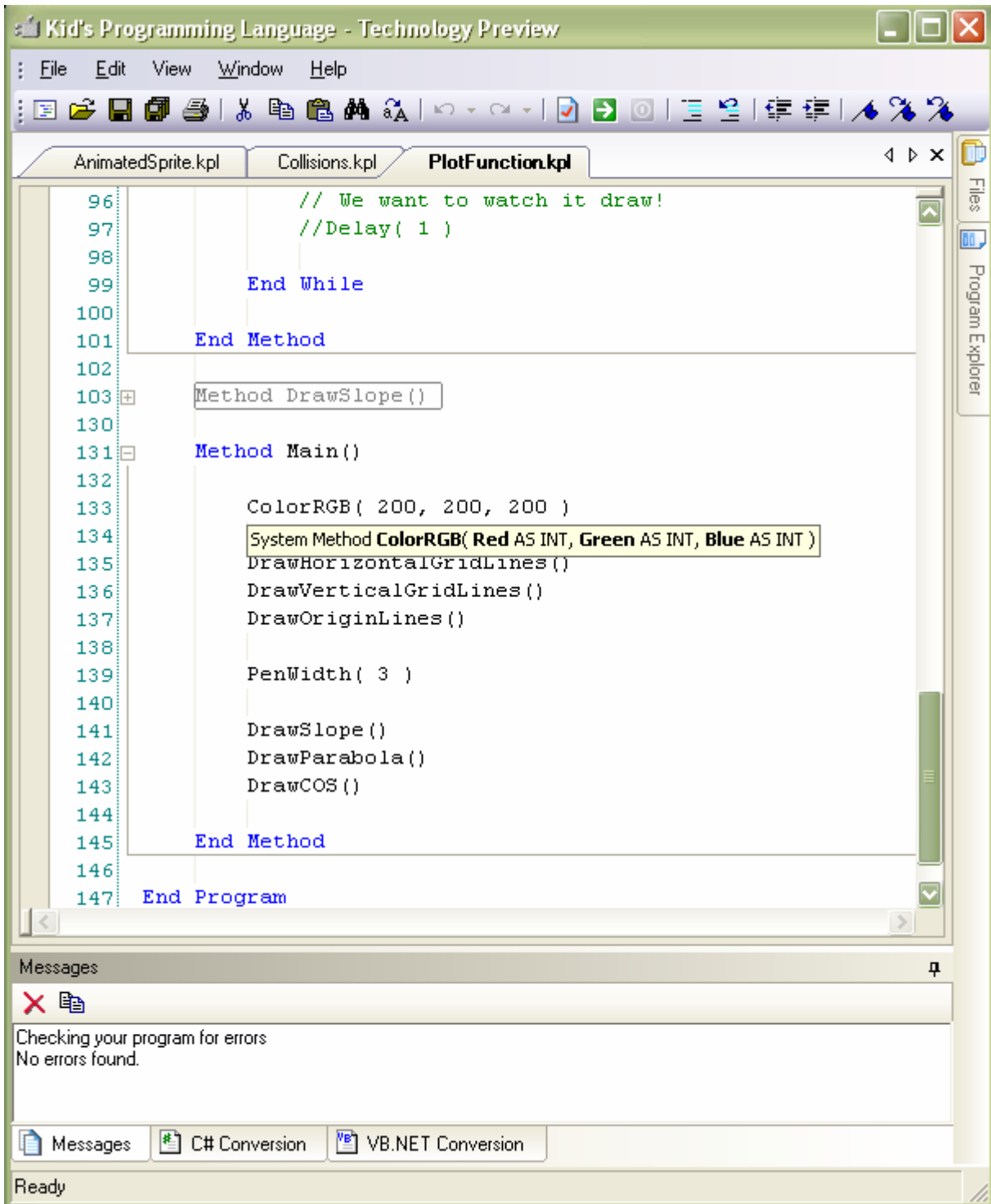


Figure 1. A screenshot of the KPL Integrated Development Environment

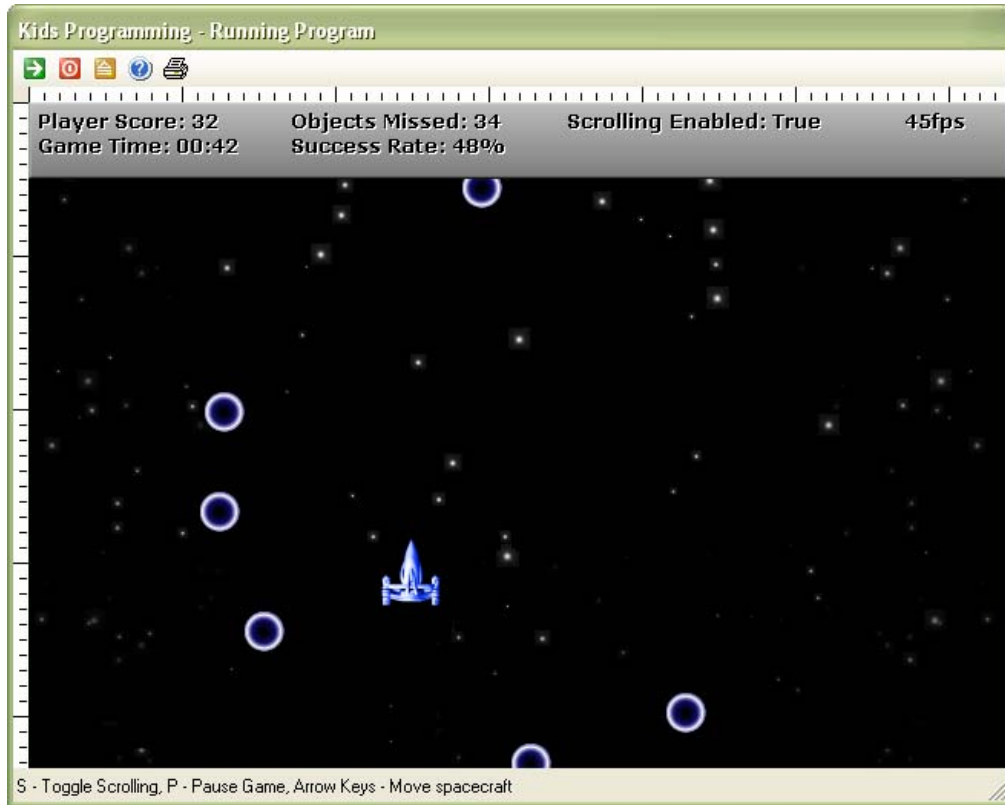


Figure 2. Screenshot of a running KPL sample program

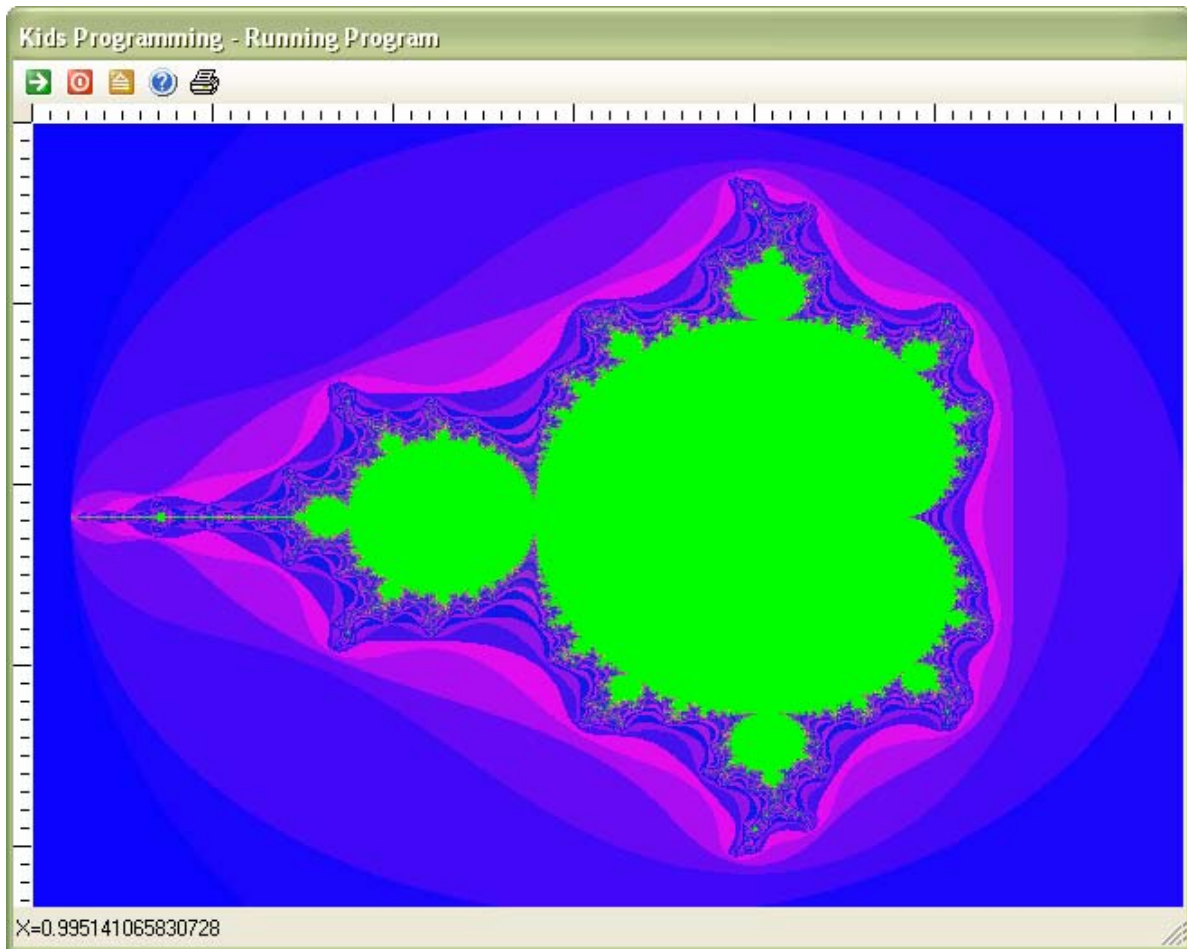


Figure 3. A KPL-rendered Mandelbrot set, plotted with 70 lines of KPL code