

FUNDAMENTALS

CREATE GAMES AND LEARN TO PROGRAM WITH DIGIPEN'S PROJECT FUN



Project FUN Is Live

Programming Is FUN!

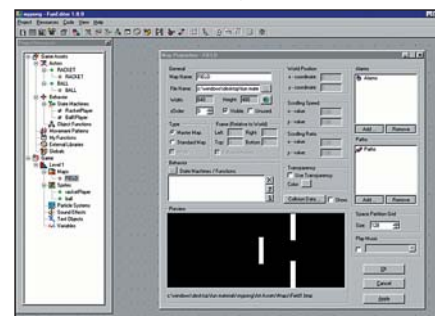
The wait is over. DigiPen Institute of Technology's Project FUN is now live at www.projectfun.digipen.edu. The new online club provides tools and instructions that give members the power to create games and learn how to program. It's for anyone who has access to a personal computer and the desire to learn about programming and designing games—from complete novices to people with programming experience. Club members will download DigiPen-developed tools, finished projects and instructions that detail all of the steps needed to create real, working games. The club encourages its members to practice creativity and

experimentation. Once members learn the programming basics, they can develop games from their own designs, share them with friends who don't have



Project FUN—Club, Website, Community
Log on to www.projectfun.digipen.edu to check out the Project FUN website, download playable games and learn more about the club.

the Project FUN software and upload their creations to the website for other club members to enjoy.



Creating Games Can Be FUN
Project FUN members can download game development tools and step-by-step instructions on how to make their own games.

The NP Connection

Nintendo Power subscribers have the opportunity to join the Project FUN club for a full year at a rate of \$15 U.S., in addition to normal Nintendo Power subscription fees. Non-subscribers and subscribers interested in renewing can join Project FUN for an annual fee of \$30 U.S. and receive a Nintendo Power subscription as part of the deal. Signing up for the club is a snap. You'll find all the information you need on the website. When you register for the club and include your Nintendo Power subscription number, the site will confirm your subscriber status and offer you club membership for the \$15 U.S. Nintendo Power subscriber rate. Nintendo Power's new monthly Fundamentals column will provide you with the latest news about the Project FUN club and offer advice about the current projects, including instructions for creating additional fun features.



Signing Up Is Fast and Easy
When you log on to the Project FUN website, follow the links to the new member registration area then follow a few quick steps to sign up.



New Members Can Start Right Away
Once you have your Project FUN user name and password, you can enter and explore the site's members-only areas.

FUN Facts

- The DigiPen Institute of Technology's Project FUN is an online club that gives members an opportunity to create video games on a personal computer.
- The Project FUN Game Factory offers instructions and assets that give members the power to create games. A new assignment is introduced every month.
- Nintendo Power subscribers can join Project FUN at a rate of \$15 U.S. for an entire year. Details are available at DigiPen's website: www.projectfun.digipen.edu.
- The Nintendo Power Fundamentals column offers information and advice on the latest Project FUN projects, with helpful hints and bells and whistles.

Connect with the Community

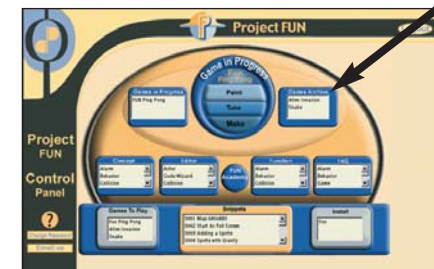
Your connection to Project FUN begins at the Project FUN website. Before you join the club, you can download FUN games created by DigiPen Institute of Technology freshman students at www.digipen.edu/downloads. After you sign up, you'll have access to Project FUN tools and information about using the tools to create games. You'll find current projects in the Game Factory, starting with FunPingPong, a simple paddle-and-ball Ping Pong game. The project includes pre-made game assets (art and sound effect files) and step-by-step programming instructions. If you don't want to create the game from scratch, you can start with the finished project and make adjustments by following the Paint or Tune set of instructions. The Beginner, Advanced and Expert instruction sets walk you through the project from the beginning. If you follow the Beginner instructions for FunPingPong, you'll end up with a single-

player game (one paddle and several backgrounds). The Advanced instructions take off from where the Beginner instructions end and introduce a second paddle for a two-player game. The Expert instructions guide you through the creation of a single-player game that applies artificial intelligence. The instructions are broken up into easily digestible sessions that each take about 20 minutes to complete. The Game Factory will feature new assign-

ments every month. If you've done everything that you want to do with the current project, you can create your own game or find another project in the site's Archive section. More instruction is available in the online documentation and the Snippets section. Snippets are exercises that demonstrate a programming concept that you can apply to your own projects. All in all, the site offers plenty of examples that you can learn from.



Navigation Control
All of the Project FUN features are immediately accessible from the Control Panel. Jump to the Game Factory, online documentation and more.



Plenty of Projects
In addition to the current FunPingPong project, there are two more game projects in the Archives.

Download Project FUN Tools

As a Project FUN member, you will have access to the versatile Project FUN editor, a tool that gives you the power to develop nearly any style of sprite-based game

imaginable, from arcade classics to RPGs. Follow the links to the download area and select the Project FUN tools. When you download the tools, they will automatically install onto your computer's hard drive and create a shortcut

on the Start Menu. All you have to do to run the tools is select the FUN Editor from your list of available programs. You can also download current projects, which include art assets and programming instructions.



Download the Tools Quickly and Easily
One of the first things that you will want to do as a new club member is download the Project FUN tools.

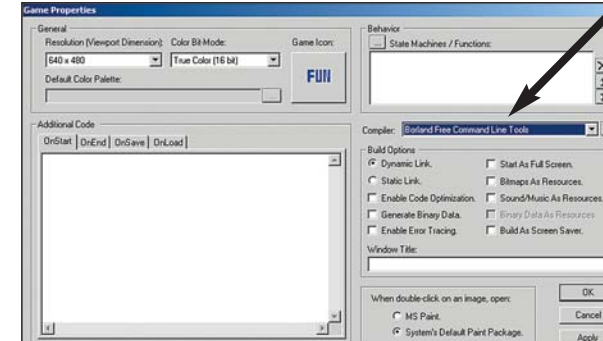
System Requirements

Project FUN members must have access to an Internet connection and a personal computer that meets the following recommended hardware specifications:

- Windows 98/ME/2000/XP
- Internet Explorer 5.0
- 128 Meg RAM
- 16 Meg Video RAM
- Pentium II, 350 MHz
- Soundblaster compatible sound card
- 4X CD-ROM drive
- 150 Meg hard drive space
- C/C++ compiler (see DigiPen's website for more info)

Additional Software Required

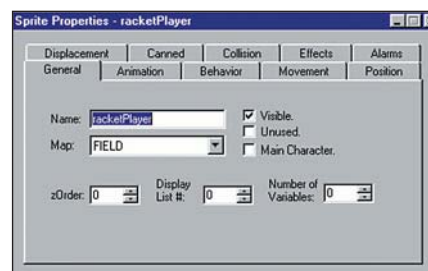
You'll need a C/C++ compiler to bring your Project FUN games to life, as the FUN tools do not include a built-in compiler. Several C/C++ compilers are available, including a free compiler from Borland. Information on how to download the compiler and make it work with the Project FUN tools is available on the Project FUN website. You'll be able to set up the compiler in a few quick steps.



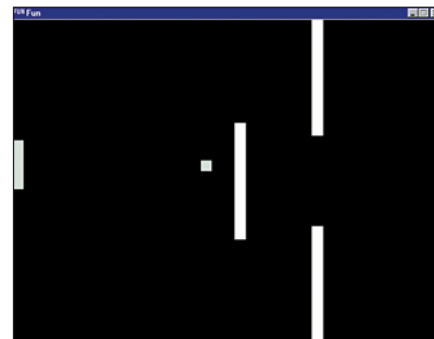
Set It Up
The Project FUN website gives instructions on how to download a free C/C++ compiler. Select the compiler and set the path in the Game Properties section.

First Project: FunPingPong

The Project FUN Game Factory offers step-by-step instructions that show how to create a video game. The first project is FunPingPong, a Ping Pong-style game with a paddle and a ball. As you go through the steps to create FunPingPong, you'll learn how to add a background and sprites, how to create a state machine and how to compile and run a working game. It's amazing how much you can learn while you're having FUN.



Add the Elements and Give Them Properties
Every object has several properties that relate to how it behaves within the game.



Put It All Together and Play
After you've gone through all of the steps, you can sit back and enjoy the game.

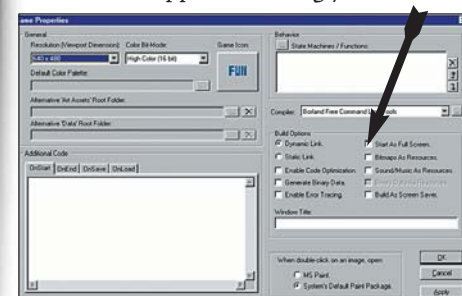
Game Development Hints

Save, Compile and Run

The FunPingPong instructions tell you to save your project, compile your code and run the game often. It's important to go through the process whenever you make significant changes to your game. If you have to find the root of a problem, it's better to have made only a few changes since your last compile than a long list of complex additions.

See Your Game Full-Screen

When you run your game for the first time, you'll see that the game appears in a window that occupies about a third of the screen. You can make the game appear on the entire screen with an easy adjustment. Right-click on the Game Node of your project tree and select the Properties option. The Game Properties window will appear, showing you several changes that you can make to the game as a whole, such as a Build on Full Screen option. Check the box next to the option and close the window. When you run the game, it will fill up the screen.



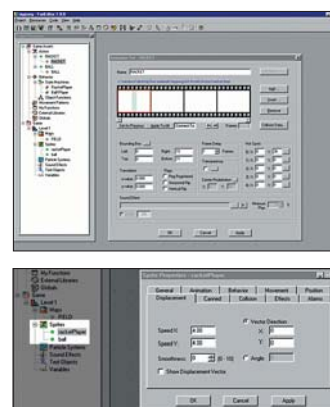
You'll find several Build Options on the Game Properties screen, including an option to maximize the size of your game screen.

Add and Manipulate

Much of the programming process involves adding new elements and giving those elements new properties. You'll discover that whenever you want to add an element, you will right-click the appropriate node on the project tree—right-click the Actor node to add an actor, for example. Once you have added the new element, you'll left-click the item to bring up the window that changes its properties.

What's an Actor? What's a Sprite?

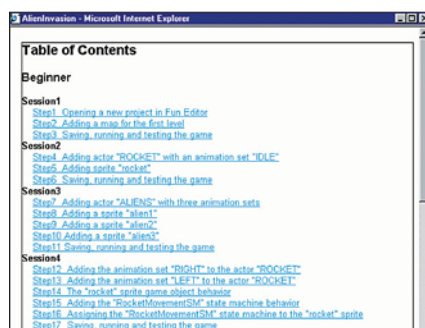
The terms "actor" and "sprite" sometimes seem like they're interchangeable. They're actually two different things. An actor is an animation set, or a series of animation sets, such as the single frame racket actor in FunPingPong. A sprite is an object in the game that uses an actor to display animation. It has many properties that an actor does not have. Multiple sprites, such as two rackets in FunPingPong can use the same actor.



An actor is an animation set, or a collection of animation sets. A sprite is an object in your game. You'll use an actor to create every sprite.

Experiment and Have FUN!

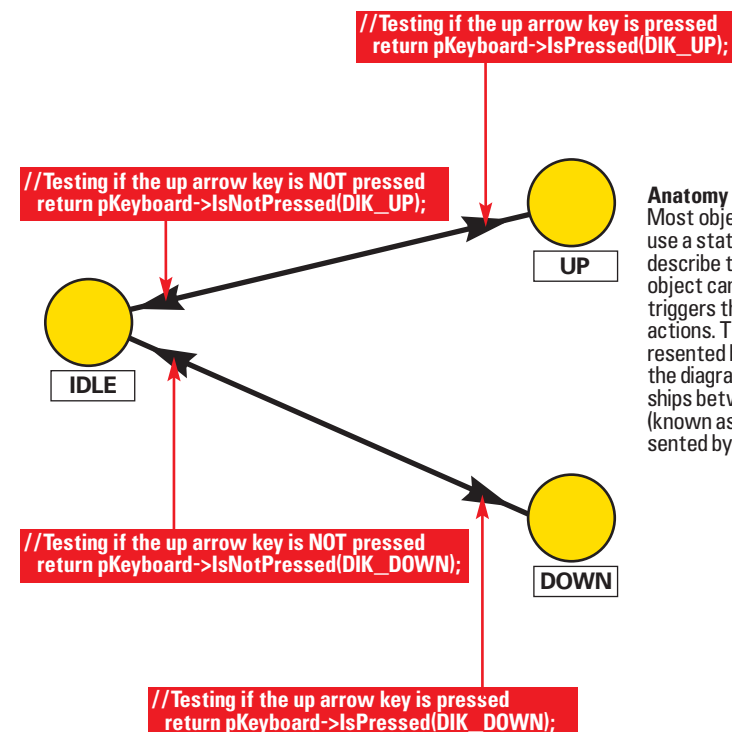
When you begin programming, you'll want to follow the instructions carefully to make sure that you get everything right. After you have some success getting your project up and running, you'll find that there are many ways to accomplish programming tasks. Use your newfound knowledge to experiment with the program and develop new ideas—the possibilities are endless.



The best way to learn is to follow the instructions as they are written, then experiment with new ideas. You will make some interesting discoveries.

State Machines

Your game development experience will really take off when you make your first state machine. State machines allow the objects in your game to transition from one action to another. In FunPingPong, your racket can perform three different actions. It can sit idle, move up or move down. Those actions are represented by states in a state machine diagram. When the player presses the Up Arrow key on the keyboard, the racket transitions from the idle state to the moving up state. When the player releases the Up Arrow key, the racket transitions from the moving up state back to the idle state. The transitions are represented by arrows that connect the states. Most states and arrows have C/C++ programming code attached to instruct the object how to behave. After you make your first state machine diagram, you'll find that state machines are easy to understand and incredibly powerful. When you come up with new ideas for object actions, you can design them with state machines in mind.



Anatomy of a State Machine
Most objects in your game use a state machine to describe the actions that the object can perform and the triggers that set off those actions. The actions are represented by circular states in the diagram and the relationships between the actions (known as "edges") are represented by arrows.

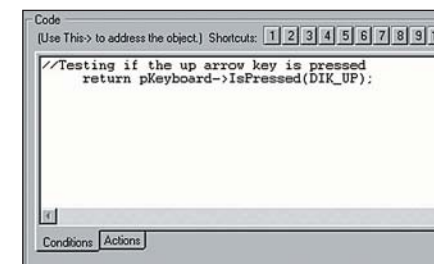
Programming Basics

Be Exact

When you're typing code into your game project, you must key it in to exactly match the instructions. Every capital letter and semicolon has meaning. If it's not perfect, you'll have errors.

Comments, Please

Some lines in the instructions begin with two forward slashes. Those lines are programmer comments that are not part of the compiled code. They can help you understand how the program works.



//Programming Commentary
It's considered good programming practice to include plenty of comments while coding. They help you organize your thoughts.

Learning to Code

The first lines of code that you will enter for the FunPingPong project tell the computer to follow prewritten instructions known as functions. Functions are sub-programs that perform specific tasks. The function called VectorDirection, for example, tells the computer which direction the sprite should follow. When you advance to the programming steps for the FunPingPong ball, you will begin to enter actual C/C++ programming code—using the conditional "if" statement.

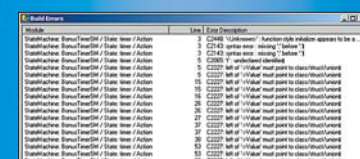


C Programming Begins

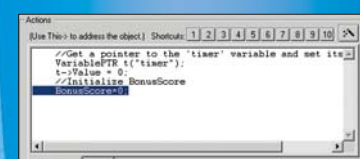
The first actual C code that you will enter checks to see if a particular condition has been met. If so, it will follow the next line of code.

Compiling Corrections

The computer follows your instructions exactly as written. If you compile code that contains typos, you'll experience easily fixable errors.



When the computer alerts you to a compiling error, some compilers allow you to double-click on the error message to skip to the code.



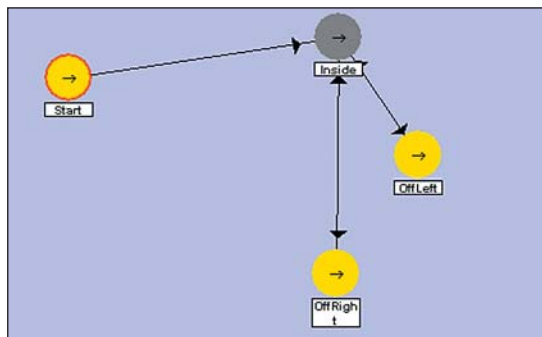
More often than not, a compiling error is the result of a misspelled word or missing punctuation.

Bells & Whistles

FUN with Physics

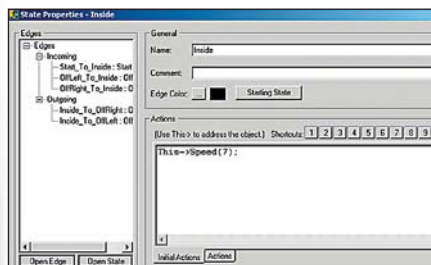
You can learn a lot about programming by changing parts of a finished project and discovering how your changes affect the game. Assuming that you have gone through all of the Beginner steps to create a single-player game, you will have the basic building blocks that you need to make some interesting adjustments. You can start by changing the speed of the ball and racket. The ball speed is set in the BallPlayer state machine. Open the state machine and double-click on

the Inside state. You'll see a function call that sets the speed of the ball to 5. It reads This->Speed(5); You can change the ball speed by changing the number inside the parentheses, but make sure that it is a positive number. A larger value will make the ball move faster. A smaller number will make it move more slowly. After you make the change, compile and run the game. Your

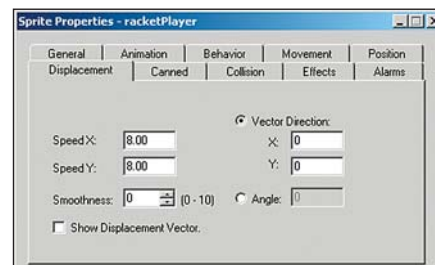


Open the Ball's State Machine
The Inside state of the BallPlayer state machine sets the ball's initial speed.

adjustment will apply to every single-player level in the game. The racket speed is set within each racket sprite's properties. Double-click on the racketPlayer sprite entry to open up its sprite properties. Select the Displacement tab and adjust the SpeedX and SpeedY properties. By increasing the numbers, you will increase the speed of the racket. If your project includes two rackets, you can give one racket an advantage by allowing it to move faster while the other racket moves at its default speed.



Change the Code
The single line of the ball's Inside state includes a number. Change the number to change the speed.



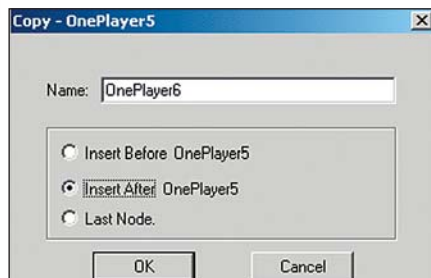
Adjust Racket Displacement
The speed of an individual racket is set in the displacement values for the racket's sprite.

Create New Levels

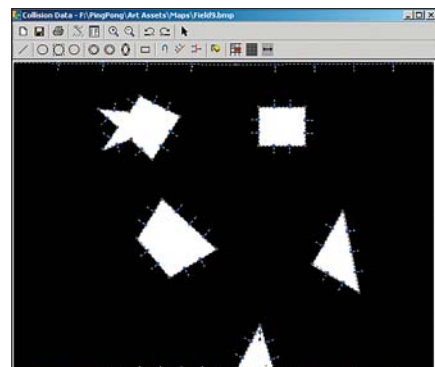
After you go through all of the Beginner steps, you'll have a single-player game with several levels. Adding new levels is quick and easy. First, use a basic art program, such as Paint, to create a new background file with the same dimensions as the other background files in the game. You can start with one of the existing files and change it as you like or start with a different art file. You can even use a photograph. Open the project in the FUN Editor, right-click on any level entry and insert a copy of the level. You will be able to place it immediately before or

after the level that you are copying, or add it as the last level in the game. The copy will include all of the features that are part of the level that you copied, including the map. Double-click on the FIELD entry under the map node and change the file name to the BMP file for the new background. Then click on the Collision Data entry and use the line tool to create collision data around any obstacles that you want the ball to bounce off

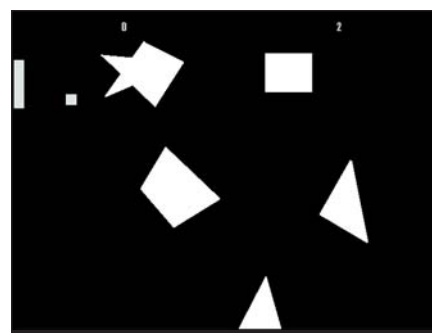
of. By working around the obstacles in a counterclockwise pattern, you will ensure that the ball bounces off the obstacles. After you save the changes, compile the code and run the game. The new level will automatically become part of the game. You can add any number of levels with different types of backgrounds and obstacles. You can also replace the backgrounds in existing levels by pointing their FIELD entries to new BMP files.



Make a Copy
Right-click on a level entry, then choose to insert a copy of the level.



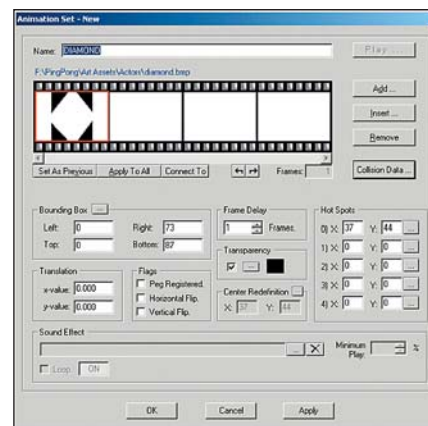
Add Collision Data
After you add a new BMP file to your FIELD entry, add new Collision Data.



Play the Game
Compile and run your game. The new level will appear where you placed it.

Add an Obstacle

Level 5 of FunPingPong introduces the concept of moving obstacles. You can develop another moving obstacle level with a few easy steps. Create a BMP file that displays an empty, black field then right-click on the Level 5 entry and insert a copy that appears after the level. You can call it Level 6. Since it is a copy of Level 5, it already includes coding that supports a moving obstacle. Change the FIELD to the blank BMP file and add collision lines at the top and bottom of the map. Next, use your art program to create an object in the shape of a dia-

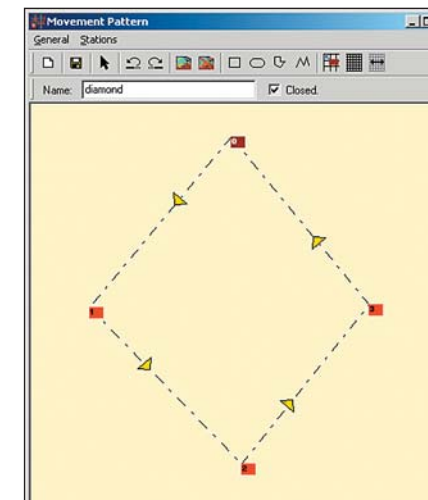


Create an Actor
Your new obstacle will start as an actor. Add the shape of the object as a single frame animation.

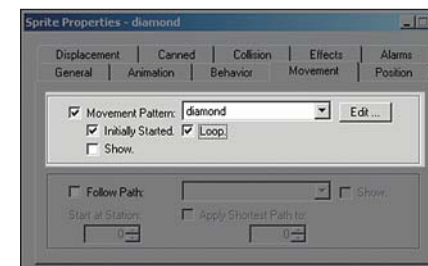
More Fun

The DigiPen developers are just getting started. Two new projects are already in the Archive, and more projects are on the way. Every project is designed to introduce new programming concepts and

new possibilities for Project FUN game development. Future projects include platform games, shooters and arcade-style action games. As you follow the steps to develop each project, you'll learn more about programming and you'll gain the tools that you need to create your own original games. The sky is the limit. Create the projects, experiment, develop your own ideas and have fun!



Make It Move
Add a movement pattern by using the rectangle tool and moving the nodes.



Make it Loop
When the movement pattern is in place, connect it to the sprite and make sure that it loops.



Alien Invasion
The Alien Invasion game introduces color and multiple collision detection algorithms, an important concept for games with a lot of individual sprites.



Snakes
When you follow the steps to create the Snake game, you'll learn how to create your own program functions using the C programming language.



Bust the Boxes
September's project will cover advanced collision concepts and deal with a large number of sprites in various shapes and sizes.