Chapter 12. Fun and Games

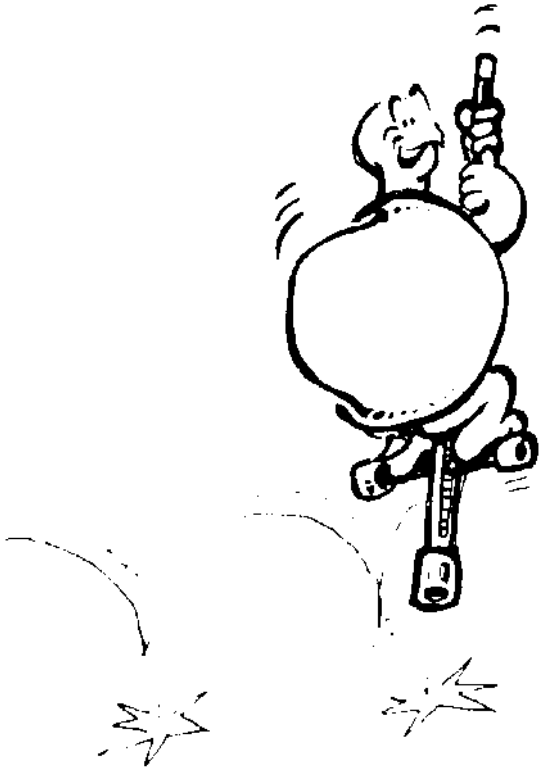
By now you've seen quite a bit of MSW Logo. It's time to go off on your own a bit. There is much, much more left to discover. We'll help you get there.

Designing Your Own Games

Designing your own games isn't as hard as it sounds. You've probably played lots of computer games.

There are role-playing games where you play one of the characters in the story. There are games of skill, strategy games, card games, and all sorts of others. What do all these games have in common?

What makes some games fun and some not so much fun? Do fancy graphics and sound effects make a good game? Do they make a game good?



Why not get a group of friends together and talk about the games you like…and some you don't like. See if you can come up with a list of things that make a game fun to play.

What makes a game fun to play?

1. **You have to be able to win! That's probably the most important part of a game. If you can't win at least some of the time, why play?**

2. **There has to be a challenge. If winning is too easy, it's no fun. If winning is too hard, that's not much fun either.**
3. **There has to be some competition. You can play another team. You can play another person. You can play against the computer. Some games even make you play against yourself.**
4. **The game has to be logical. Players have to be able to figure out what to do next. The rules of the game and the action have to make sense.**

This is probably the most important thing to think about when designing your own game.

It has to be logical! You have to "think of everything."

This means that it takes careful planning. And this is where your journal can come in real handy.

1. **What's the purpose of the game?**
   Is it a sports game? An action game? A strategy game? A puzzle? A dice game?
2. **How does the game open?**
   What is the challenge? How will you tell the players about the game? How will you draw them into the game so they will want to play it?
3. **What about the middle game?**
   What will the players do? How will they move? How will they play?
   Will there be surprises? New challenges?
4. **How will the game end?**
   What will players do to win? What happens when they win?

These are some of the things to think about when designing games. It's really not so bad. It just takes some

careful, logical thinking. And you've been doing that all the way through this book.

Logo Sports

You've already been introduced to Logo Football and Logo Baseball. Ever played Logo basketball? There's a version of basketball on the disk that came with this book, BASKET.LGO. Take a look at it.

BASKET.LGO is a good example of what you can do with variables and random selections. This procedure sets up many of the conditions you run into in a typical game of basketball…jumps, passing, shooting, fouls, rebounding, 2 point and 3 point shots. Too bad it's not a graphic game.

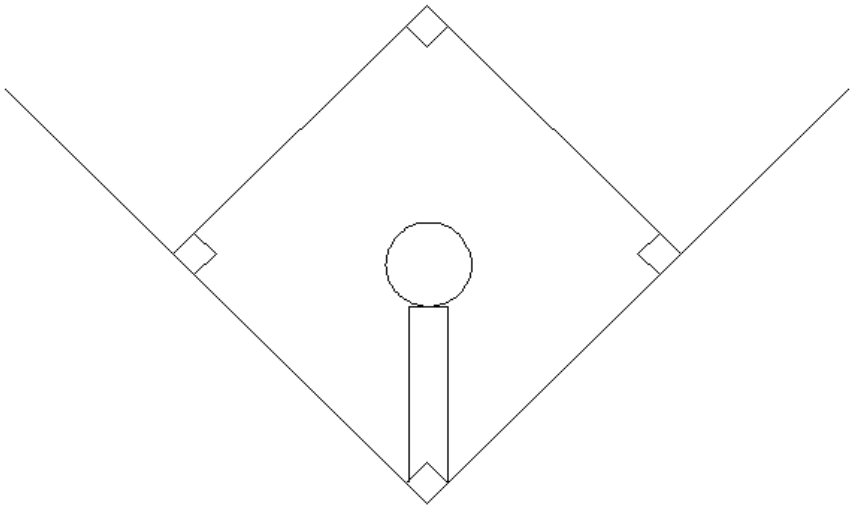
Your Challenge

Why not add graphics to the game? Create a basketball court and then move two or ten players around according to what's going on with the text procedures.

Logy and Morf created a graphic game. But rather than try to use ten players…ten turtles…they used two. The two players were basketballs of different colors, each of which moved according to the action.

No…this isn't arcade quality graphics. But it's a start.

Remember Logo Baseball?



This one is easy to turn into a graphics game. Here's some ideas.

1. **Have the pitcher throw a turtle that looks like a ball.**
2. **Use Random numbers to test whether the batter got a strike or a ball.**
3. **Set up a procedure to keep track of balls and strikes.**
4. **When the batter gets a hit, have the turtle select a random heading.**

   If the heading is more than 315 or less than 45, the ball is fair. Otherwise it is a foul.
5. **Create some targets in the outfield. A large one for a single, a smaller one for a double, still smaller for a triple, and smaller yet for a homerun.**

   If the ball doesn't hit one of the targets, the batter is out.
6. **Set up multiple turtles to act as players who get on base and advance when others hit the ball. Or maybe they're out? How would you set that up?**

Then there's Logo Football. What can you do with that one?

Here's a procedure that will randomly draw defense players on the screen…eleven circles with x's in each. It's called FTBALL2.LGO.

Your Challenge

Modify this procedure so that the eleven defense players are always far enough apart to allow the turtle to move through them without touching.

To play the game, write a procedure that will send the turtle from the left goal line to the right goal line without touching any of the defensive players.

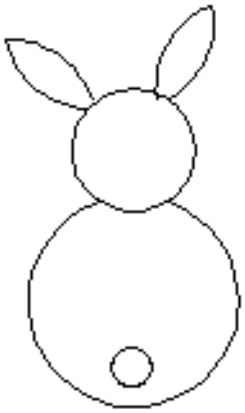
```
TO CIRCLE :CENTER :RADIUS
LOCAL "AMT
MAKE "AMT (:RADIUS * 2) * PI/360
PU SETPOS :CENTER
SETX XCOR - :RADIUS SETH 0 PD
REPEAT 360 [FD :AMT RT 1]
RT 90 FD :RADIUS * 2 BK :RADIUS
RT 90 FD :RADIUS BK :RADIUS *2
PU SETPOS :CENTER PD
END

TO DEFENSE
REPEAT 11 [PLAYER]
END

TO FOOTBALL
CS HT PU BK 100 LT 90 FD 240 RT 90 PD
REPEAT 22 [REPEAT 2 [FD 10 RT 90 FD 40 RT 90] ~
      FD 10]
RT 90
REPEAT 10 [FD 40 REPEAT 2 [ FD 40 RT 90 FD 220 ~
      RT 90]]
```

```
FD 40 RT 90
REPEAT 22 [REPEAT 2 [FD 10 LT 90 FD 40 LT 90] ~
        FD 10]
RTGOAL LTGOAL DEFENSE
END

TO LTGOAL
BK 110 LT 90 FD 480 RT 90 FD 110 LT 45 FD 50 RT 45
FD 30 LT 45 FD 70 BK 70 RT 45 BK 60 LT 45
FD 70 BK 70
END

TO MARK
RT 90 FD :RADIUS/10
BK :RADIUS/10 LT 90
END

TO PI
OP 3.14159
END

TO PLAYER
PU HT SETXY ( RANDOM 200 )( RANDOM 95 ) ~
        - 90 PD
MAKE "CENTER POS
CIRCLE :CENTER 10 PU HOME
END

TO RTGOAL
LT 90 FD 40 RT 90  BK 110 LT 135 FD 50 LT 45 FD 30
RT 45 FD 70 BK 70 LT 45 BK 60 RT 45 FD 70 BK 70 LT
45 FD 30 RT 45 BK 50 LT 45
END
```

Who's Who in the Zoo

Logy and Morf ran a contest a few years ago to find the best animal procedures. These are on the disk that comes with this book.

Some of them are pretty simple. Others get pretty complicated. They also use turtle positions and coordinates in one way or another.



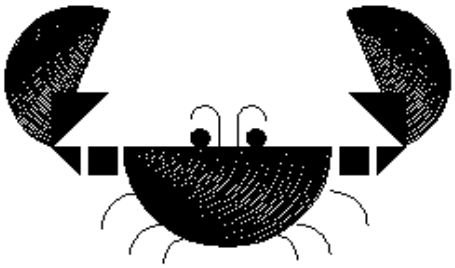
You can start off with one of the simpler ones, MORPH.

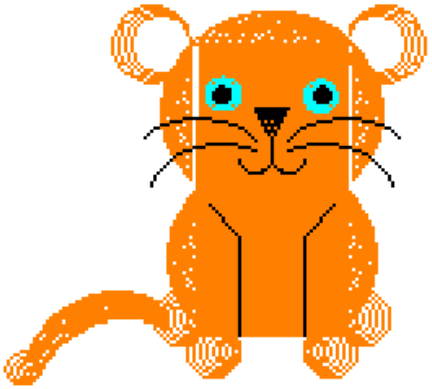
Did you know that the scientific name for rabbit is "lagomorph?"

That sounded a lot like Logomorph. So Morf just shortened the last part and came up with a new name.

Some friends from New Jersey worked on a series of animal drawings for the contest.

I bet this is one of the cutest crabs you ever saw. It comes from one of the Ron Ebberly books. Take a look at CRAB.LGO to see the procedures that created the crab.



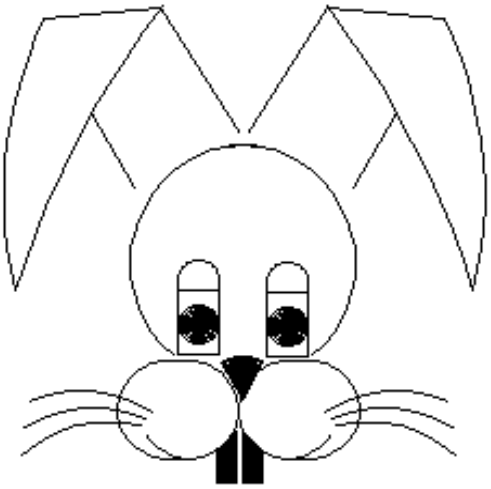


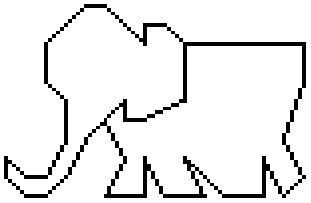
The young people that designed the crab and the tiger cub (CUB.LGO) used a version of Logo that has no FILL command. So they wrote their own.

Take a look…a nice use of recursion to create a drawing.

Want to learn another language?

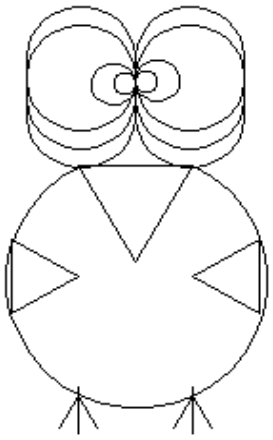
The HAAS procedure has procedure titles written in Dutch. Can you figure out what those titles mean by what they do?





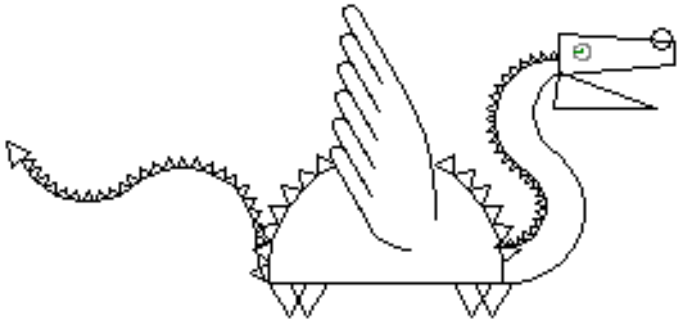
Here's a hint about the elephant. This is it! It was made using SETXY commands.

Did you ever see a DODOBIRD? That seemed like a good name for this crazy bird.



Now…what kind of animal can you dream up? Why not do some sketches. Then put them on the computer .

Puff, The Magic Dragon, may give you an idea or two. Puff was sent in as an entry to a Logo animation contest Logy and Morf had some years back. As you will see in the introduction to animation, Puff breathes smoke and spits fire.

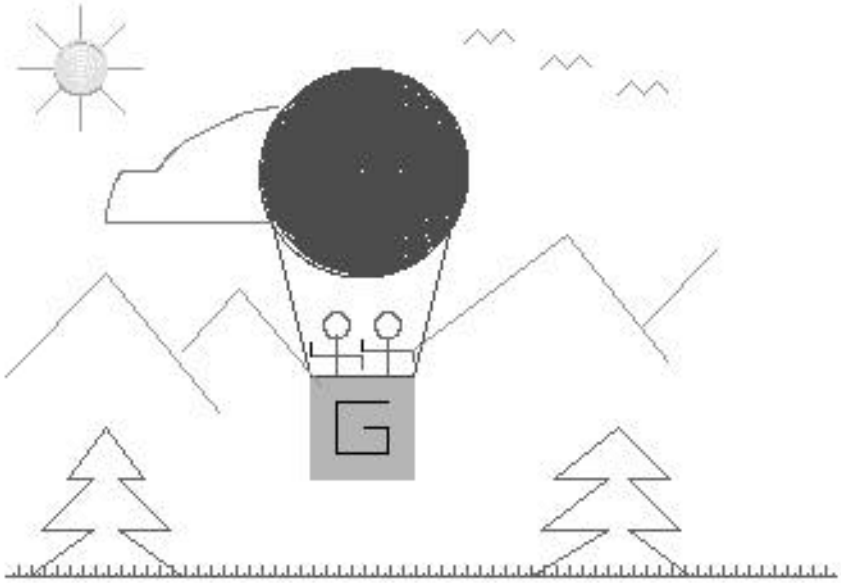


Up, Up, and Away

A very popular contest that Logy and Morf held was called the Up, Up, and Away Contest. It celebrated the birthday of the first flight of a hot air balloon .

Young People were told to let their imaginations run wild and create a drawing that featured a hot air balloon.

Here's Gretchen's balloon (BALLOON.LGO). It was one of the winners, mainly because it was the only one that included animation. The two people in the balloon wave at you.



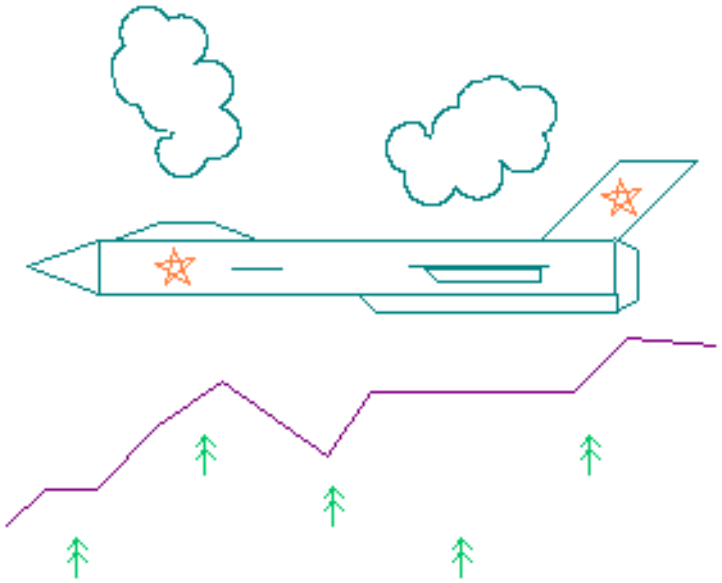
Why not try this yourself? Start with a basic balloon shape like this…

Then see what you can do with it.

How about making the balloon into a turtle that floats over the landscape?

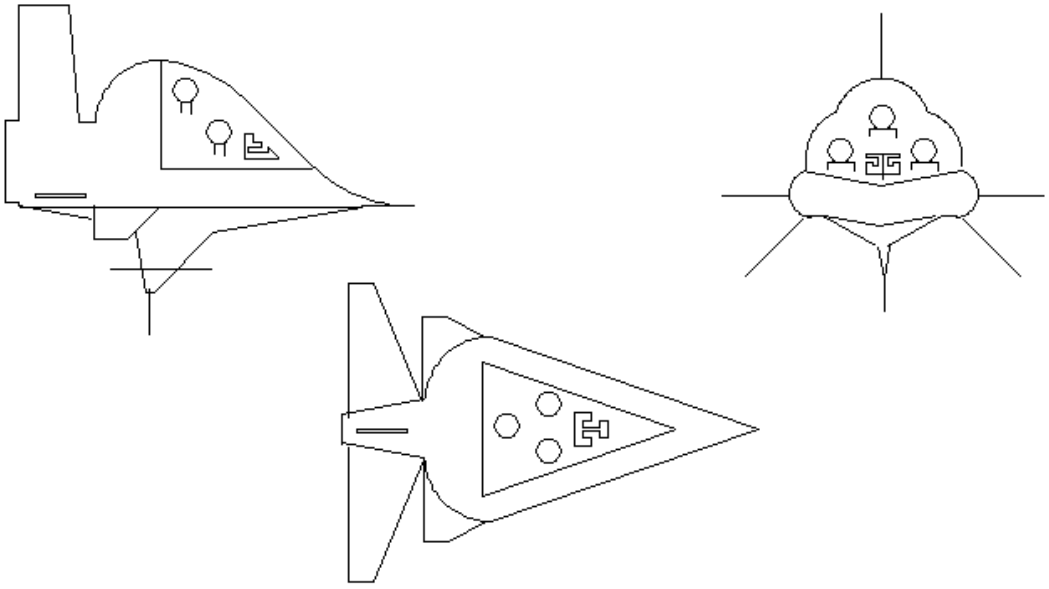
Fly, Fly Away

The Up, Up, and Away Contest was so successful, Logo and Morf held a Fly, Fly Away Contest to see who could create the best aircraft artwork .



FIREFOX.LGO is a translation of one of the many procedures sent in by young people at the Utah State School for the Deaf.

On the next page are the three views of X2.LGO, a very creative procedure from an 11-year-old from San Diego .



Logo Geography

It started with the idea of Wrapping.

Some young people in an elementary school computer club got curious as to what happened when the turtle went off one edge of the screen and then showed up on the opposite edge.

Did it travel behind the screen?

This is the same group that explored the soccer ball. They already understood the idea of flattening three-dimensional objects on the two-dimensional screen. So it was easy enough to picture flattening the world into a two-dimensional map…that if they travelled off one edge of the map, they would automatically "wrap" to the same spot on the opposite edge.

NO...that's a different kind of wrapping!

Fence and Window

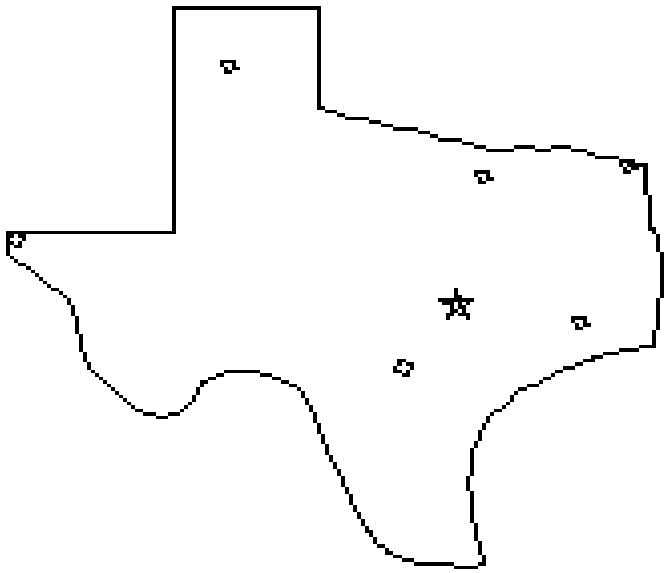
But how can you stop the turtle from wrapping?

WINDOW is the command that tells the turtle not to wrap…just keep on going off into the world beyond the screen.

FENCE is the command that stops the turtle at the edge of the screen. It stops the action and tells you that the turtle is "Out of Bounds."

Let's get back to geography.

After talking about all sorts of maps, the students decided to create their own maps on the screen. Here's the picture of Texas they developed. It's part of the TEXAS.LGO procedure.



The map of Texas includes the cities of Amarillo, Austin, Dallas, El Paso, Houston, San Antonio, and Texarkana. And since game design was a very important activity with this group, they made a game out of it.

Here's the GAME procedure.

```
TO GAME
PR [Logy, What direction do we turn?]
PR [How many miles do we have to go?]
WAIT 80
PR [Can you give Logo a hand?]
PR [What direction do they turn?]
MAKE "DIR RW
PR [How far?]
MAKE "TURNS RW
IF :DIR = TURNS  [LT :TURNS]
IF :DIR = "RT  [RT :TURNS]
PR [How many miles do they have to travel?]
MAKE "FAR RW  PU
FD :FAR * .187
CHECK
END
```

It took a lot of discussion but the group finally figured out that the actual miles on the screen map were the number of turtle steps times 0.187.

550 miles * 0.187 = 103 turtle steps
(Actually, it's 102.85. But 103 is close enough.)

The game was good practice in guessing directions and drawing mileage to scale. After working with Texas, the group did several other states. Each time, they kept adding new features.

1. **Take a look at the Texas procedure and see if you can develop the same type of procedure for your state.**
2. **Add other landmarks from your state…maybe lakes, rivers, parks, and things.**
3. **If a player moves from one place to another correctly, have them answer a random question about that place.**

Think up five or six questions for each location you put on your map.

Can you think of other ways to explore Logo geography?

Logo Animation

Graphics are much more fun when they come to life. It gives you the chance to play movie producer, movie director, script writer, and actor all at the same time.

There are several different ways you can produce movies using the turtle. Let's start with the simplest method. Even in this day of advanced computer graphics, this method is still used to animate characters.

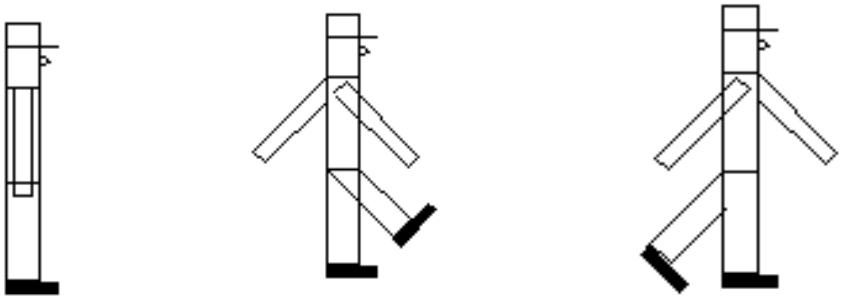
This procedure draws a stick figure that appears to wave its arm as if exercising.

```
TO ANIMATE :N
HT LEGS BODY HEAD
ARMS MOVE :N
END

TO ARMS
LT 90 FD 20 LT 90 FD 80 BK 160 FD 80
END

TO BODY
RT 165 FD 120 LT 90
END

TO HEAD
REPEAT 60 [ FD 4 RT 6 ]
END
```

```
TO LEGS
RT 165 FD 120 BK 120 RT 30 FD 120 BK 120
END

TO MOVE :N
IF :N = 0 [STOP]
MOVE.UP MOVE.DOWN
MOVE :N - 1
END

TO MOVE.DOWN
PE FD 80 BK 160 FD 80 PENPAINT
RT 10 FD 80 BK 160 FD 80
END

TO MOVE.UP
PE FD 80 BK 160 FD 80 PENPAINT
LT 10 FD 80 BK 160 FD 80
END
```

This procedure draws the arms in one position, erases them, moves them, erases them, moves them, and so on. But it gives you a look at how animation can begin.

Another way to create animation is to use different turtles, each showing part of an action. Here's three very crude drawings of a soldier walking.



Look at the SOLDIER.LGO procedure. If you have a fast computer, run soldier and watch the soldier seem to walk.

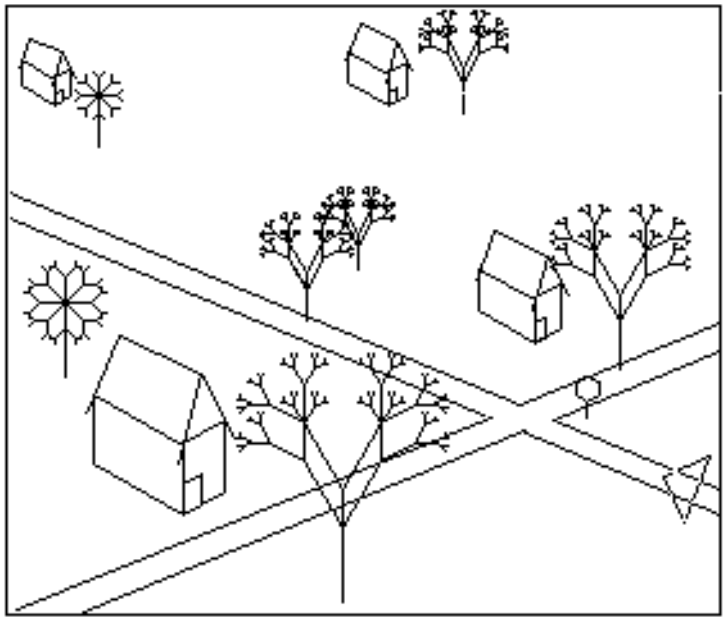
Read the BITMAP Functions section of the MSW Logo Online Help. Also, take a look at the MSW Logo DEMO procedure. This will tell you how to cut the soldier pictures as bitmaps that can then be used as individual turtles.

Now you can walk the soldier across the screen.

You can add additional views to make the action smoother.

You can create pictures using MS PAINT or another graphics program and then animate those pictures.

Take a look at RUNNER.LGO. Here's another approach to animation.



RUNNER is a cute little procedure. The turtle comes up to the stop sign, stops, looks both ways, beeps, and then moves up the road.

Take a look at the tree procedures. They are classic examples of recursion in action.

Some challenges…

1. **Add color to the drawing.**
2. **Using a random number, have the turtle turn onto the cross road. At some random point, have the turtle turn back onto the left-to-right road.**
3. **The turtle beeps at the Stop sign. Change that to a short tune.**

For another example of animation, take a look at PUFF.LGO. Watch her blow fire and smoke.

Logo Physics

Physics? This is fun and games, not science!

You're about to see how science can become fun and games. Did you ever play a lunar module game, where you land a spaceship on the moon or another planet?

You were dealing with physics.

Did you ever play an artillery game where you fired your cannon at the enemy?

You were working with physics.

The laws of physics describe how things work…how planets stay in orbit…how different types of force work including the force of gravity.

Most of physics is a bunch of mathematical equations. Unless you're a mathematician, those equations don't mean much. You need to see physics in action to understand how things work. And if you're going to see things in action, why not have some fun doing it!

Logo Gravity

When you drop a stone from a tall tower, gravity pulls it to the ground. It accelerates as it falls, moving faster and faster.

Then splat! It hits the ground.

Just what is acceleration? What's the difference between acceleration and speed?

Speed vs. Velocity

Speed and velocity mean the same thing. Speed is easier to spell.

Speed measures the rate at which your position changes. If you run 100 yards in 10 seconds, your rate of speed is…

Distance/Time = Speed

100 yards/10 seconds = 10 yards per second

This is your average speed over the 100 yard track. But you started at 0 yards per second. Let's take what we know and figure the acceleration.

Acceleration

Speed measures the rate at which your position changes. Acceleration measures the rate at which the speed changes.

Change in speed/Time = Acceleration

Another way of saying this is…

Final speed - Beginning speed/Time = Acceleration

Let's say that the speed crossing the finish line was 50 yards per second. This gives us...

50 yards per sec. - 0 yards per sec. / 10 seconds = an acceleration of 5 yards per second per second.

Now let's put this information into a procedure to demonstrate what acceleration is all about. When you drop a rock from a tower, the laws of physics tell us that the acceleration is going to be 32 feet per second per second.

That's too fast to see on the computer. So we're going to slow this down a lot. If this is too slow or too fast for your computer, you can change it. In the FREEFALL procedure, there's a line that says…

MAKE "ACC 0.2

Change that 0.2 to whatever works well on your computer.

```
TO GROUND
HT PU SETPOS [-100  -200] PD
SETPOS [100 -200]
PU HOME PD
END

TO FREEFALL :HEIGHT
CS PD GROUND PU SETY :HEIGHT
MAKE "VEL 0
MAKE "ACC 0.2
MAKE "TIME 0
ST PD ACCELERATE
PRINT "
PRINT [The turtle has landed.]
END
TO ACCELERATE
( PR "TIME :TIME "VELOCITY :VEL "DISTANCE ~
        ( :HEIGHT - YCOR ) )
```

```
BK :VEL + :ACC / 2
MAKE "VEL :VEL + :ACC
MAKE "TIME :TIME + 1
IF YCOR < -200 [STOP]
ACCELERATE
END
```

To run the procedure, type FREEFALL and a height from which the turtle will fall.

Artillery Practice

One of the first…if not the very first…computer game was an artillery game played long before personal computers were even thought of. Here's a procedure that describes the basic movement of an artillery shell. (Because SHELL is a command, we use the term SHEL.)

To fire an artillery shell, you need to know three things: the velocity of the shell out of the gun barrel, the angle at which it was fired, and the local gravity (or acceleration) factor. This gets more complicated than we need to worry about here. Let's just say that as you go higher, the effects of gravity get smaller.

Try this to start…

```
SHEL 10 45 .4
```

Then experiment with some different numbers.

```
TO ACCEL :SX :SY
IF YCOR < -50 [STOP]
SETH 90 FD :SX
SETH 0 FD :SY - :ACC/2
ACCEL :SX (:SY - :ACC)
END
```

```
TO GROUND
PU SETPOS [150 -50] PD
SETPOS [-150 -50]
END

TO SHEL :VEL :ANGLE :ACC
HT GROUND
MAKE "SX :VEL * COS :ANGLE
MAKE "SY :VEL * SIN :ANGLE
ACCEL :SX :SY
(PR [Range = ] XCOR + 150)
END
```

When you fire an artillery shell, it flies through the air. And air produces resistance. So let's add in some air resistance to our procedure.

Have you ever heard of the sound barrier? As a jet fighter flies through the air, it pushes air out in front of it. The faster it goes, the more air it pushes.

When a jet flies at the speed of sound, it actually pushes through that big bubble of air that it has pushed out in front of it. If you're under that airplane, you'll hear a sound like thunder.

Our artillery shells are not going to break the sound barrier. But they are going to push air out in front of them. And that air is going to slow them down.

In our procedures, we can add an "air friction factor" that will slow the shell down… :FRIC. Watch and see.

```
TO ACCEL :SX :SY
SETPOS LIST XCOR + :SX YCOR + :SY
IF YCOR < -50 [STOP]
ACCEL (:SX - :FRIC * :SX) (:SY - :FRIC * :SY - :ACC)
END
```

```
TO GROUND
PU SETPOS [150 -50] PD
SETPOS [-150 -50]
END

TO SHEL :VEL :ANGLE :ACC
MAKE “FRIC 0.1
HT GROUND
MAKE "SX :VEL * COS :ANGLE
MAKE "SY :VEL * SIN :ANGLE
ACCEL :SX :SY
(PR [Range = ] XCOR + 150)
END
```

This gives you enough information to design your own artillery game. Why not see what you can do to design a game where two artillery units fire at each other.

Think about it. There’s all sorts of possibilities.


MY JOURNAL
OR HOW I DID IT!