TOONTALK

An Animated Programming World

ToonTalk 3 Reviewer's Guide

Prepared by:
Ken Kahn
Animated Programs
August 14, 2007

Welcome to the reviewer’s guide to ToonTalk 3.

ToonTalk is an educational computer software product that enables kids to learn and do computer programming. Inside ToonTalk, you will be able to create computer programs and see animations of your creations running. ToonTalk is uniquely designed to allow children of different ages, abilities, and styles to learn about programming. A child is introduced to ToonTalk in three ways: playing the puzzle game, watching narrated demonstrations, or free play. Each way fits a different type of learning style. Children are encouraged to try all three and to move between them. This guide presents only one possible path through these many choices.

We encourage you, if possible, to also try out ToonTalk with a child (10 years old, plus or minus 5 years).

1. Installing ToonTalk

ToonTalk can be downloaded from www.toontalk.com/English/trialdwn.htm. If the documentation if viewed with Internet Explorer then the Microsoft Agent ActiveX Control will be able to read the ToonTalk web pages. If you have any difficulties installing ToonTalk please send email to support@toontalk.com.

2. Getting Started

When you click on the ToonTalk icon, you are presented with seven choices:

1. **Set Options.** The default settings should be adequate, however, you may wish to change the screen resolution or voices.
2. **Help.** This launches your web browser on the ToonTalk HTML help files. This shouldn’t be necessary in order to use ToonTalk but there is a wealth of information about ToonTalk available there.
3. **Play Game.** This starts the ToonTalk puzzle game which introduces ToonTalk objects and design techniques. We recommend you start here. Later in this guide, we provide help for going through the first nine puzzles.
4. **See Demos.** There are eight different narrated demos illustrating different uses of ToonTalk. We recommend you view the “Introductory Tour” after trying “Play Game”. The demos are described in more detail later in this guide.
5. **Free Play.** Here you are free to create software that ranges from games, to silly sentence generators, to math explorations and computer animations. We recommend you try this last. Later in this guide you will be guided through an example of free play.
6. **Visit Playground.** The European Union funded Playground project created a great deal of ToonTalk material aimed at helping young children build computer games.
7. **Explore Weblabs.** The European Union funded WebLabs project created ToonTalk material to help older children explore mathematics and build scientific simulations and games.

3. Playing the puzzle game.

To start ToonTalk, choose “Play Game”, and enter your name. ToonTalk maintains a user model and treats you differently depending upon what you have done with ToonTalk in the past. Choose your appearance, (you can choose from long hair, a hat, or bald) and you’ll see the ToonTalk opening screen. Click your mouse button to proceed to the credits screen. Click your mouse button again to proceed to the next screen. Now, you’ll see a brief story providing a narrative context to the game. This story provides an explanation for why Marty the Martian knows everything, but can’t do anything himself. You will need to make things on your own, but can go to Marty for more verbal help.

When the introduction ends you find yourself standing in front of a rocket that has crashed, with a house nearby. In the beginning, this is all you will see. If you try to go into the house you will find the door is locked. If you enter the door of the rocket, Marty greets you. In the puzzle game you interact with Marty by entering his rocket and listening to either the description of what he needs now, or why what you just brought wasn’t what he needed, or a bit of advice about how to proceed, or congratulations if you brought what was needed. If you want him to give you the next hint, either walk close to him or leave the rocket and return. The puzzle game consists of over 60 puzzles in 5 levels. We recommend you try to solve at least the first 9 puzzles.

After hearing what Marty needs, you need to walk into the house and click your mouse button to sit down. To solve the puzzle, grasp (with a mouse click) the desired object, and press ‘Esc’ to stand up. You’ll automatically put the object in your pocket. Walk back to Marty. If you need help, you can walk back to Marty empty handed.

Puzzle #1 – Marty needs a box with a 1 and 2 in it. This introduces ToonTalk boxes and numbers. Walk into the house and click your mouse button to sit. Move your hand over the 1, click your mouse button to grab the 1, move it over the left hole of the box, and click your mouse button to drop it in the first hole. Then pick up the 2 and drop it in the second hole. Then grab the whole box, press the ‘Esc’ button to stand up, and walk out the door, and back to Marty.

Puzzle #2 – Marty needs a 4. This puzzle introduces Bammer the Mouse who does arithmetic and other basic operations. Walk back into the house and click your mouse button to sit again. Pick up a 2 and drop it on top of the other 2. Wait for Bammer the Mouse to combine them. Pick up the 4, press ‘Esc’ to stand, and walk back to Marty.

Puzzle #3 – Marty needs a box with 8, 16, and 32 in it. This introduces a way to combine boxes to make bigger boxes. Drop the 16 on a 16 that's in a box. Then drop the box with 8 on the left side of the one with 16. Make sure they overlap just a little bit. Then drop the box with 32 on the right side of the other box. As with all of these puzzles, if you make a mistake you can pick up the bomb and press the space bar to start the puzzle over again.

Puzzle #4 – Marty needs a number bigger than 1,000. This puzzle introduces the ToonTalk magic wand which is used to copy ToonTalk objects. Pick up the wand, point the tip of the wand at the 1, and press the space bar to copy the 1. Drop the 1 on the original 1 by pressing the space bar. Copy the 2 and drop it on the 2. Keep doing this until the wand runs out of magic. The result should be exactly 1,024.

Puzzle #5 – Marty needs a zero. This puzzle introduces Dusty the Vacuum. Pick up Dusty, turn him on by pressing the space bar, and move him over all the stuff on the floor. Press the space bar again as soon as you see a number pad with 0 on it. Pick up the 0 and bring it back to Marty.

Puzzle #6 – Marty needs a –1. Point to Dusty's button, wait for it to start wiggling, and then click your mouse button. Dusty's button should now show R for reverse. Point to any other part of Dusty, click again to pick him up, and then press the space bar to turn him on. When you see him spit out the '-1', press the space bar again to turn him off. Then pick up the –1 and bring it back to Marty.

Puzzle #7 – Marty needs a blank number pad. Push Dusty's button until it shows E for erase. Point to a different part of Dusty and click your mouse button to pick him up. Move Dusty over the number, press the space bar, and then set Dusty down. And then pick up the blank number and bring it back to Marty.

Puzzle #8 – Marty needs a box with two zeroes. Pick up the box and drop it on the robot. You will enter the robot's thoughts. Then train the robot to copy the box by pointing the wand at the whole box and pressing the space bar. Drop the copy right next to the box. Make sure they overlap a little bit. You will automatically leave the robot's thoughts. Then give the trained robot the box and wait for him to copy and join the boxes. Pick up the box and bring it back to Marty.

Puzzle #9 – Marty needs a number greater than a billion. Give the robot the box. Move the magic wand over the 1 until it wiggles and then press the space bar to copy it. Drop the copy on the 1 to double it. You will automatically leave the robot's thought bubble. Pick up the box and drop it on the newly trained robot. He'll stop after doubling the number. Dusty will run into your hand. Set him down and click your mouse button until the button on his nose shows 'E' for erase. Then move his mouth over the 1 in the robot's thought bubble. Press the space bar to turn him on. Set Dusty down and give the robot the box again. Wait until he stops, pick up the number in the box, and bring it back to Marty.

What is Learned.

"Behind the Scenes of ToonTalk's Puzzle Game" describes what programming concepts are learned as a player solves these puzzles. Briefly, these first nine puzzles introduce *data structures*, *data types*, *operations on data*, the technique of doubling by copying, and *program construction and testing*. The next nine puzzles of the first level introduce *recursive data structures*, *incremental data structure construction*, and *binary arithmetic*. The next level introduces *characters* and *text strings* and how to manipulate letters and words. The goal of the third, fourth, and fifth level is to fix the ship's clock. Doing so requires computing conversions between different units of time, using sensors to measure the passage of time, using trucks to *spawn new computational processes*, using birds and nests for *communication* *and synchronization between separate computations*, and using scales to express *inequalities* for expressing programs with more flexible *iterations*.

4. Watching the Demos.

If you click on 'See Demos' after starting ToonTalk you will see a selection of eight demos. We recommend you watch the one called 'Introductory Tour' which takes less than 15 minutes. Time permitting you may wish to see some of the other demos.

1. **Exploding Houses** (less than 5 minutes). This is the silliest of the demos and is probably the most popular.
2. **Swap if Bigger** (less than 3 minutes). This demo illustrates two very basic programming techniques: comparison conditionals and exchanging the values in two locations.
3. **Build Ping Pong** (each part is less than 20 minutes). This demo shows two kids trying to build a Ping Pong game. They don't know what they are doing and the demo shows the *process* of exploring, trying things, finding bugs, fixing bugs, getting help, and the like.
4. **Fibonacci Numbers** (less than 4 minutes). This demo illustrates the kind of mathematical exploration that is possible within ToonTalk.
5. **Factorial** (less than 6 minutes). This demo shows a programming technique that is critical for building large programs in ToonTalk out of pieces that work together. The technique is a generalization of defining and calling procedures.
6. **Bank Account** (less than 15 minutes). This demo shows how advanced computer programming and computer science concepts can be made accessible within ToonTalk. It also illustrates another important programming technique needed for large-scale programming – concurrent objects with state.
7. **Append Lists** (less than 8 minutes). This demo illustrates another advanced programming technique (list manipulation) and illustrates a connection between ToonTalk and logic programming.

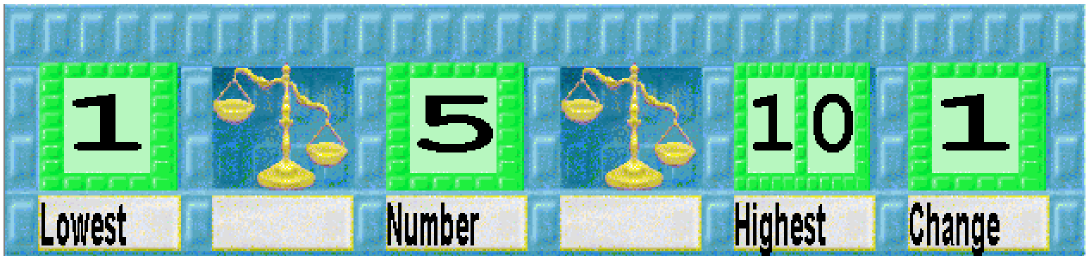
5. Being Creative in Free Play.

If you select 'Free Play' after starting ToonTalk, you'll find yourself flying a helicopter. Hold down the left mouse button until you land near a house. You'll find Tooly the Toolbox is following you around. When you enter a house and sit down, Tooly opens up and you have a full selection of ToonTalk objects to use to build whatever you like. You can explore math or language, build games, computer animations, simulations, and plenty more.

Marty is around to help you. If you pick up an item, Marty will suggest things you can do with it that you have never done before. You can also get Marty to give you "directed help" by sending him away by pressing the 'F1' button, and then pressing it again while holding or pointing to the item you want him to tell you about.

To give you a sense of what free play is like, we will now walk you through a simple example. You will train a robot to change a number so that it grows until it reaches a limit and then shrinks until it reaches another limit. You will then use this robot in a variety of different contexts.

1. Start ToonTalk and click on ‘Free Play’. Use your mouse to land the helicopter near a house. Walk in and click your mouse button to sit down. Take things out of Tooly to create a box that looks like this:

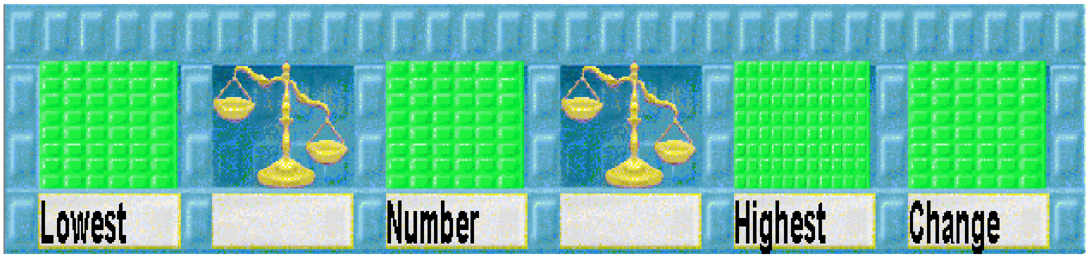


You can construct this box by placing boxes next to each (so they overlap a little bit) and filling them with the appropriate numbers (you can use the keyboard to change a number you are holding or pointing to). The labels (“Lowest”, “Number”, etc.) are not strictly necessary but can be added by pointing to the appropriate place and typing. If the box is too wide to see all at once, you can use Pumpy to make the box narrower by clicking on ‘N’ while holding Pumpy over the box.

The idea behind this box is that the compartment labeled “Number” will change on each cycle by “Change” so long as the “Number” is greater than the “Lowest” and less than the “Highest”.

2. Give this box to a fresh robot from the toolbox. Train the robot to pick up the magic wand, point to the 1 labeled “Change”, and after it starts wiggling press the space bar to copy the number. Move the wand so that the copy is over the 5 and click your mouse button to drop the number. Press the ‘Esc’ button since that is all this robot needs to do.

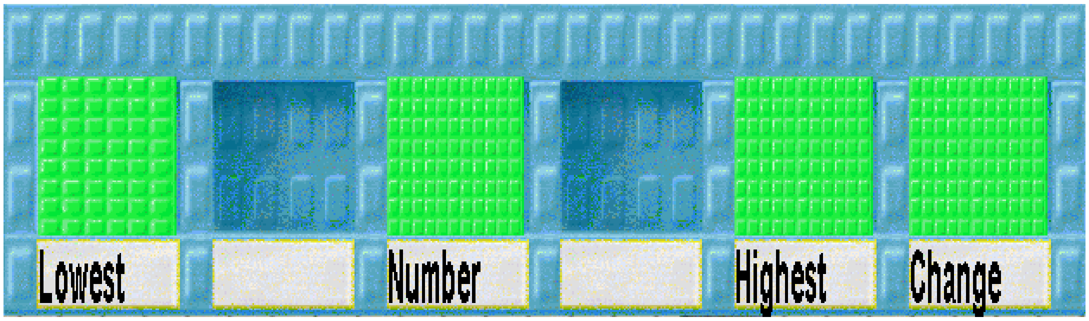
3. Next click on the button on Dusty’s nose until it shows ‘E’ for erase. Move Dusty over one of the numbers in the box in your robot’s thought bubble. Hit the space bar and move Dusty back and forth until all the numbers have been erased. Now your robot will work for all numbers so long as the scales tilt the right way. The box in your robot’s thought bubble should now look like this:



4. You are now ready to try out your robot. Pick up the box and give it to him. You should see him add to ‘Number’ until it reaches 10.

5. We now need to train another robot to switch the direction that the number is changing. Take out a fresh robot and give him the box. Train this robot to pick up the number in ‘Change’, type ‘-‘ to make it negative (or to make it positive if it already is negative), and put the number back. Pick up the magic wand and copy the ‘Change’ number and drop the copy on ‘Number’. That’s it, so press ‘Esc’.

6. We are going to want this new robot to work whenever the first robot isn’t working. One way to arrange this is to erase the numbers in the new robot’s thought bubble as before and also to vacuum away the scales. This robot will want to run no matter what numbers are in the box he is given. So the box in the new robot’s thought bubble should look like:



7. Now you are ready to make a team out of your robots. Since the first robot in the team gets first crack at working on a box, you need to be sure that your new robot goes behind the old robot. To do this, just drop the new robot on the old one.

8. It is good practice to save things in the notebook as you work. Find a blank page in the notebook, and drop the team of robots on the right side of the notebook. Pick up a text pad, type some reminder of what these robots do (like “Oscillates”), and drop the text pad on the left side of the notebook.

9. Now you can have fun with your robots. Find a picture you like in the notebook on page 2. While holding the picture press ‘f’ to flip it over. A notebook of remote controls for that picture will fly out. Take out the remote control labeled ‘Width’. Remove the number that was in the ‘Number’ slot of your box and drop the width remote control in its place. Change the ‘Lowest’ to 100 and the ‘Highest’ to 900. Now pick up the picture, type ‘f’ to flip it back over, and set it down. Take out a copy of your team of robots from the notebook and give them the box. You should see the width of your picture change very slowly. It is so slow because robots go slow when they are watched. Now flip the picture back over, drop the team of robots on the back, and then drop the box on the back. Flip the picture back over and drop it on the floor. You should see it get fatter, then thinner, then fatter, and so on.

10. You can flip your picture over and have the robot change other values like the position or appearance. You can even have more than one team on the back so that multiple aspects of the picture can change at once. Take out other pictures, flip them over, and add your robots and copies of the box. You can drop pictures on pictures to form composite pictures where each one can be oscillating in ways you control.

11. When you have an animation you like, you can save it in your notebook. You can also pick up the picture, press the ‘Pause’ key, and select ‘Come back later’. A Java applet of your creation should appear in your web browser soon afterwards.

Congratulations, you just programmed your first ToonTalk computer animation.