Guide to Promoting Independent IT-Related Problem Solving in Middle School Youth

Table of Contents

Introduction

To be successful in a world where new technology is increasing exponentially, youth need to be able to adapt quickly and think independently. They need "critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources" (International Society for Technology in Education's [ISTE] Educational Technology Standards for Students, 2007). Educators can help youth develop these capacities by promoting independent problem solving rather than giving students the right answers when they get stuck and ask for help.

The purpose of this guide is to share lessons learned about how to promote independent problem solving in youth information technology (IT) education programs. The ultimate goal is to increase students' skills and confidence so they are not afraid to explore and, as a result, become producers, not just users, of information technology. By teaching students how to problem solve and explore without fear of making a mistake or getting lost, educators are helping to build intrepid (fearless) explorers, which is, according to Sherry Turkle, "a key advantage in the world of computer science" (cited in Margolis & Fisher, p. 29). Intrepid explorers like to be challenged and have the confidence to deal with setback and errors (Margolis & Fisher, 2002). Intrepid explorers are resilient when they encounter barriers or mistakes; they see challenges as opportunities to learn. The concept of "Intrepid Exploration" is discussed in more detail in an article by Denner and Bean (2006) cited in the list of references.

Because independent problem solving is important in any IT setting and can be used with any software for any IT project, the material in this guide is meant to be used broadly. Materials pertaining to specific software, such as Macromedia's Flash, Stagecast Creator, or Storytelling Alice, are included as samples only.

This guide is based on the experiences of ETR Associates (ETR) developing, implementing and researching out-of-school IT programs for middle school youth, specifically two programs for girls called Girls Creating Games (GCG) and Girl Game Company (GGC) funded by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This guide discusses the problem solving process, suggests some ways to foster it in middle-school youth, and provides examples of some problem solving strategies. Because the optimum amount and type of assistance will vary depending on individual student needs and program factors, it also provides examples of how to assist students in various situations. It concludes with a tool to assess what level of assistance an educator should provide.

Please note, many problems are beyond the students' ability to solve – problems like security or network issues or software limitations that only an adult with technical experience or new versions of software can fix. This guide focuses on problems that are within students' zone of proximal development (Vygotsky, 1978), or what Brooks (1987) refers to as "essential difficulties" – problems that students are capable of solving with training and practice.

The Problem Solving Process

For most adults, being able to analyze and solve problems is an implicitly learned skill and one that not every adult does very well (Jonassen, 2000). That is, few of us are formally trained in a problem solving process. We learn through trial and error and practice. As we experience successful problem solving outcomes, we gain confidence, the ability to "think outside the box" and the skill to apply what we have learned to other situations. Because middle school youth are relative beginners to this way of thinking, they need explicit instruction in how to problem solve. They also need as many opportunities to practice and experience successful problem solving outcomes as possible.

Most models of the problem solving process are variations of the following steps:

1. Define the problem – what is the problem?
2. Determine possible causes for the problem – why does the problem exist and what are the possible causes? What is the best way to find out?
3. Propose and test solutions – how can the problem be fixed?
4. Apply the best solution – what is the best way to fix the problem?

   (See Appendix A for a handout of the above process. Also for more information, see "What is Problem Solving Instructional Strategies" and "Key Instructional Strategies" cited in the references.)

1. **Define the Problem**
   First describe the problem. What is not happening that you want to happen? What is happening instead? Next, focus on the symptoms by listing all of the characteristics of the problem. Pare down until you have a clear, specific definition – the more specific the better.

   Asking strategic questions to help students arrive at a specific definition of the problem will help them to not only think more critically but also determine if the problem is something they are capable of solving. What does a student mean, for example, when she says that her computer does not work? Does it start up, but she cannot open a document? Does it not start up at all? These are two very different situations with different causes and, most likely, different solutions.

2. **Determine Possible Causes of the Problem**
   Decide why the problem exists by examining the evidence collected in the first step and considering possible causes. A problem can be analyzed in many different ways, and the most effective strategy will depend on the situation. Some strategies we have used in game development classes for middle school students are:

   a. **Reduction – look at the solution to a similar problem**. This technique involves comparing the problem to one with a known solution (Armoni, Gal-Ezer & Tirosh, 2005). It is a particularly useful technique for left brain types of problems such as those involving programming, logic or reasoning. For example, in a two-week game development class for middle-school youth that used a 3D animation programming environment called *Storytelling Alice* (SA), pairs of students were able to solve their own problems by comparing their code to that of working samples (see Appendix B), code which is easily exported and printed in SA.

b. **Process of elimination**. A standard problem solving technique that is often employed by mechanics, engineers and programmers is to narrow down the problem by examining possible causes from most likely to least likely and eliminating them one by one. If, for instance, a student can't find the work she thought she posted in a file, you can walk her through the possible reasons why it's not where it's supposed to be, starting with the most likely – it was put in the wrong folder.

c. **Analogy – look for similarities in incongruent (different) elements.** Quite often, inspiration and solutions to problems can come by comparing seemingly unrelated elements or changing your perspective. This strategy can be particularly useful for problems involving the creative right side of the brain, such as design. For example, an architect may get the inspiration for a new kind of houseboat marina from the layout of a major airport. Arranging the houseboats in circular patterns like the hubs at many airports could provide a much more communal feel and better views than the traditional, side-by-side configuration. In the same way, asking students to look for ideas in unlikely places can spark students' imaginations. In the Girl Game Company program, when girls are having problems coming up with ideas for their games, they are encouraged to get inspirations from what they are learning in school. In one game, the player enters medieval Japan and has to successfully play the game to get home. In another, Amelia Earhart must find different parts of a plane to get home.

d. **Collaborate with peers.** Studies have shown that collaboration can increase individual problem solving and motivation (Barron, 2000; Inkpen, Booth, Klawe & Upitis, 1995; Littleton & Light, 1999.) Collaboration promotes critical thinking because the collaborators must articulate their ideas and negotiate solutions. Also, the "acknowledgement of mutual ignorance" removes the burden and shame associated with the problem solving task (Schoenfield, 1995, as cited in Denner & Werner, 2007.) In Girl Game Company, collaboration is built into the process because the students work in pairs and are encouraged to first check with their partners and then with other girls – particularly the more experienced girls who are official peer educators – before asking the teacher for help. On the creative side, students often collaborate with each other by drawing inspiration from each others' games.

e. **Brainstorm:** This technique can also be useful for both creative and technical problems, but students need to be walked through the process and given opportunities to practice. It's important to emphasize that there are two distinct parts of the brainstorming process: 1) first come up with and record as many ideas as possible as quickly as possible without judging their merit, and 2) narrow down the ideas until you arrive at the best possible answers.

f. **Work backwards:** Trace and test your steps from the end product back until you find where the problem no longer exists. This technique can be particularly useful for technical problems like incorrect code.

g. **Start over:** Sometimes the best solution is to throw out problem-ridden work and begin again. It's important that students know that it's okay to start over; it does not mean they are failures. On the contrary, a great deal can be learned from "going back to the drawing board," and it is an option that professionals frequently take.

3. **Propose and Test Solutions**
   Consider and test possible solutions. Test by applying the most simple and direct solution first and moving on until you find something that works. Several of the techniques used to analyze the problem can be applied in this step, including brainstorming and starting over.

4. **Apply the Best Solution**
   Often there is more than one solution to a problem. In this stage, the problem solver weighs the possible solutions and picks the one that seems to be the most effective or expedient.

No matter what strategies students use to analyze a problem and find a solution, learning is most effective when it moves from explicit teacher-centered instruction to implicit student-centered instruction in the following way:

1. Set the stage for learning by explaining that this is one of many problem solving techniques that people use in everyday life, and it will get easier with practice.
2. Explain what to do.
3. Model the process. For example, if you determine that a process of elimination is a good way to find the source of a student's problem, you can begin by examining a possible cause out loud and asking the student to do the same thing with another possible cause.
4. Guide students through hands-on practice. The level of guidance you provide will depend on individual student needs and other factors as discussed in the next section, How to Assist Students to Problem Solve.

How to Assist Students to Problem Solve

The following outlines the range of assistance that is recommended for youth who encounter problems. It’s important to note that the possible range of assistance is broader than what is noted here. For example, on the far left, minimum-assistance end of the scale, you could offer no help whatsoever. The opposite extreme would be to take over the mouse and perform the task yourself. To promote independent problem solving, you should avoid either extreme, and stay within the following recommended range of assistance. You should also train adult and student helpers to do the same.

Recommended Range of Assistance

Require students to try to figure out problem themselves using available tools such as sample code, flow charts, or online help.	Ask students to get help from peers.	Provide coaching/ modeling — assist students in finding solution by taking them through the problem solving process.	As a very last resort, provide the answer.

Minimum Assistance → **Maximum Assistance**

The minimum level of assistance that is recommended is to encourage students to solve their own problems and help them by providing necessary tools and information. In Girls Creating Games, for example, students are encouraged to first check their binders for the appropriate flow chart or step-by-step instructions before asking for help from other students or the teacher. For this strategy to be effective, students must be confident and motivated enough to do their own detective work. Also, tools to help them must be readily available, and they must be trained in how to use them. (See Appendix C for a sample flow chart for Flash from Girls Creating Games and one for Stagecast Creator from Girl Game Company.)

The next level of assistance would be for a student to ask a fellow student for help. If a girl in the Girls Creating Games program is still having a problem, for example, she is encouraged to seek help from her partner, another girl who is trained to help, and/or other girls in the class. Peer assistance has a two-way benefit. The helper gains a new perspective on the problem by having to explain it to someone else, and the person being helped gains confidence from knowing that someone his or her own age can do it.

If the problem is still not resolved, you can provide the next level of assistance – walking a student through the problem solving process – as explicitly as time permits. That is, explain to the student what you are doing to try to solve the problem and why. Also, involve her or him in the decision-making as much as possible so she or he can become more familiar with the process.

Providing the answer should be the last resort but is sometimes necessary due to lack of time and other factors explained in the next section. Even so, it’s important that students know that with time and experience, they will be able to solve their own problems.

Variables to Consider
When Deciding Level of Assistance

To foster independent problem solving, the ideal would be to give students as little assistance as possible so they learn to figure things out themselves instead of relying on you and/or others. However, insisting that students follow a rigid minimum-assistance protocol, especially near the beginning of a class or program, may do more harm than good.

Before deciding how much help to give to a particular student or pair of students, you need to weigh a number of variables. Some of these variables can be evaluated before classes begin because they are global and stable; they affect all students in the program and do not change over time. Other variables must be assessed on a case-by-case basis — often on the spot — and revisited frequently because they will vary among students and most likely change over time.

Global and Stable Variables

These are factors affecting all students, but their impact must be weighed against individual factors when you are deciding on the level of assistance.

1. **Program Length and Available Help**: Giving students the right answer takes a lot less time than helping them do their own problem solving. There may simply not be enough instructional hours or trained adult and/or peer assistants to accommodate a minimum level of assistance.

2. **Goal of Program**: Is the main goal of your program to foster independent thinking or to encourage student interest in the subject matter and have fun? If the latter, you may turn some students off by forcing them to go through a time-consuming process that takes them away from hands-on tasks, especially when they are used to getting immediate answers from the Internet.

3. **Setting**: Students participating in an extended learning program in a school setting are more likely to expect and accept a more rigorous problem solving process than those who signed up for a less "official" class in a community setting.

In summary, if you have the time and available help, your primary goal is to promote independent thinking, and the setting is school-related, your overall program would fall near the left-hand, minimum assistance end of the scale. On the other hand, if you are conducting a summer or short afterschool class, have little or no instructional assistance, and the goal is to promote interest in the subject matter and have fun, then it may be advantageous to offer more assistance to students.

Individual and Fluctuating Variables

Weighing the various global variables will help you determine the general level of assistance you can and should offer, but to determine how much help to give to each student or pair of students, you must also look at individual and fluctuating variables – the factors that will vary across students. Because these variables will most likely change over time, they must be regularly reassessed for each student or pair.

1. **Frustration Tolerance or "Intrepidness"**: Students who are patient and see problem solving as a fun challenge require a much lower level of assistance than those who are impatient and/or easily frustrated.

2. **Personal Definition of Success:** Students who define success in terms of their ability to problem solve will need a lower level of assistance than those who view starting over as a sign of failure or tend to give up whenever they get stuck.

3. **Personal Investment in Class/Program:** Is this class one of many activities, such as sports or theater, that the student is involved in? If so, the class may lose the students' attention and interest if they are forced to follow a rigid problem solving process.

4. **Confidence Level**: Those who have little or no confidence in their ability to problem solve are more likely to ask for help and to expect to be "rescued."

5. **Experience and Skill Level:** Fortunately, confidence increases and the need for higher levels of assistance decreases over time as students experience successful problem solving outcomes and increase their skill levels.

6. **Physical Comfort Level:** We are all less patient when we are uncomfortable, so it's no surprise that students will have less tolerance for a problem solving process if the room is too hot, crowded or noisy.

Case Studies

Students Needing Minimum Assistance

In general, students who fall into the low intervention, minimum assistance group are very focused, are comfortable working on the computer, have low levels of frustration, and are highly motivated. These students also tend to have high attendance, and if they are working in pairs, they often have the same partners, friends they work well with. When students in this group encounter a problem, they tend to try and solve it on their own before reaching out for any other support. As a next step, they will ask other high performing students for help. When they do ask for help from an instructor, it is often simply a matter of pointing them in the right direction and leaving them alone. Many of their problems stem from limitations of the software or lack of knowledge by the staff about how to make their ideas come to life.

A Minimum Assistance Case Study

Janie and Cheri are halfway through a year-long afterschool computer programming class and have been partners from the very beginning. They rarely miss a class session, and when they arrive, they immediately get to work on their latest project. Because they are highly skilled and enjoy solving problems, other students often ask for their help, which they are happy to give. One day, they decide that an interesting obstacle in the game they are working on would be a moving platform of fire. They know how to make the platform move but not how to make it look like it is on fire. After discussing the issue and testing various options, they ask their teacher, Mrs. Flores, for help.

Mrs. Flores: What's the problem?

Janie: We want to make this look like it's on fire, but it doesn't look right and we don't know how to make it flicker.

Mrs. **Flores**: Have you looked at other fire animations and compared them to yours?

Cheri: No. But that's a good idea.

(Janie goes to a bookmarked clip art web and searches for fire animations.)

Mrs. Flores: So what's the difference between what you've drawn and what you see?

Janie: These flames are "spikier" than ours and more orange.

Mrs. Flores: Right. And what have they done to make it look like it's flickering?

Cheri: Oh, I see. This one just turns the drawings on and off.

Janie: This one goes back and forth between two or three drawings. We got it. Thanks, Mrs. Flores.

Mrs. Flores: Good. If you need help, check the worksheet on parameters in your folder.

(Mrs. Flores walks away.)

Students Needing Medium Assistance

Students who fall somewhere near the middle of the Assistance Continuum usually attend regularly and are motivated, but they tend to be more into the social scene than task-oriented, so they often need to be reminded to stay on task. These students often ask for help before looking for any other resources. After helping them with their problem by showing them where to find the answer or taking them through the problem solving process, you need to stay with them and make sure they follow through.

A Medium Assistance Case Study
Javier and James are enrolled in a summer-long computer science program at the community center. They have both attended regularly and enjoy the class, but they like to joke around a lot. They were especially boisterous when they were working with their previous partners, their best friends, and the teacher, Mr. Johnson, had to frequently remind them to get back on task. In the hopes of getting them all to settle down, he recently switched the partners. Shortly after the switch, Javier and James are working on a game with various levels of play, and they want to add programming that will take the main character to the next screen. They immediately raise their hands to ask for help.

Mr. Johnson: Need some help?

James: How do we get to a new screen?

Mr. Johnson: Do you remember when we went over that last week?

Javier: Yeah, but we forgot how to do it.

Mr. Johnson: Okay, there’s a flow chart in your folder that will walk you through it. Look for the one called “Doors.”

(Javier looks in folder and finds the chart.)

Mr. Johnson: James, why don’t you read the directions to Javier while he works on the computer? You can trade off in half an hour.

James: Okay.

(Mr. Johnson stays and watches as the pair goes through the first several steps in the process. He comes back to check on them periodically and to remind them when to switch jobs.)

Students Needing Maximum Assistance

These students tend to have little or no technical experience and lack confidence. Often they are quiet and prefer to work alone. Instead of asking for help, they will focus on tasks they feel comfortable with and, as a result, make little progress. Because of their lack of engagement, it is easy to overlook their need for help. It is therefore important to monitor their progress closely, provide as much help as time permits, and, if necessary, walk them through the problem regardless of the difficulty. Because these students need to build their skills and their confidence, they need as much encouragement and successful problem solving experiences as possible. Often, receiving help from peers can help to build their confidence and make them feel more a part of the group.

A Maximum Assistance Case Study
Stephanie enrolled late in a two-week game design offered at a local Boys and Girls Club. She does not know anyone in the class and declined an offer to partner with another girl. For the past week, she has been quietly working by herself. While most of the rest of the class is almost finished with the adventure game they were assigned, Stephanie is still working on designing the background for the introductory screen. She is engrossed in painting flowers when the teacher, Miss Foster, comes over.

Miss Foster: So Stephanie, how are you coming on your adventure game?

Stephanie: (not looking up) Fine.

Miss Foster: You've done a great job on the background. It's beautiful.

Stephanie: Thanks.

Miss Foster: How are you doing on the game itself? Have you programmed your character to walk?

(Stephanie shakes her head.)

Miss Foster: Since you don't have a partner, how about if I ask Kayla to help?

Stephanie: Okay.

Miss Foster: (motioning for Kayla to come over) Kayla, would you please be the navigator and help Stephanie make her character walk to the door?

Kayla: Sure.

(Miss Foster watches until she's satisfied that Kayla is helping without taking control. She checks back later and praises Stephanie on her progress.)

Problem Solving Assessment Tool

Step #1: Assess where your program's global and stable variables fall on the Assistance Continuum.

A. Before your program begins, examine each of the global and stable variables and assign each one a number value by marking where you think it falls on the following Assistance Continuum Grid. For example, if the setting for your program is a drop-in, community center, put a dot in the 9th or 10th quadrant on the right-hand, maximum assistance end of the Setting variable.

Assistance Continuum Grid for Global and Stable Variables:

	1	2	3	4	5	6	7	8	9	10	
Length: 1 year or more											**Length:** 5 or 6 weeks
Help: Several qualified aides											**Help:** Little or none
Goal: Promote problem solving											**Goal:** Promote interest in IT; have fun
Setting: After school/ extended learning											**Setting:** Drop in community center

→

Minimum Rescue **Maximum Rescue**

B. Add all of the numbers together and then divide by four to calculate an average for all four global/stable variables and determine where your program generally falls on the Assistance Continuum.

Step #2: Assess the individual and fluctuating variables and determine an initial level of help for each student or pair.

A. After you have become familiar enough with each student or pair to be able to make an initial assessment, determine an initial number value for each individual and fluctuating variable, using the following Assistance Continuum Grid:

Assistance Continuum Grid for Individual and Fluctuating Variables:

	1	2	3	4	5	6	7	8	9	10	
Frustration Level: Patient; highly intrepid											**Frustration Level:** Impatient; easily frustrated
Definition of Success: Values problem solving; doesn't give up											**Definition of Success:** Sees starting over as failure; gives up easily
Investment in Class: Has few or no other interests											**Investment in Class:** Has several other interests competing for attention
Confidence: Highly confident											**Confidence:** Has little or no confidence
Experience/ Skill: Highly skilled; experienced											**Experience/ Skill:** Has little or no skill; experience

→

Minimum Rescue **Maximum Rescue**

B. If the values for each variable are fairly uniform (i.e., they are all within the same or nearby quadrants), use the average value to determine an initial level of assistance for each student or pair. If there is more than a three-point difference between any variables, use a paired weighting process as outlined below to determine the relative value of each variable.

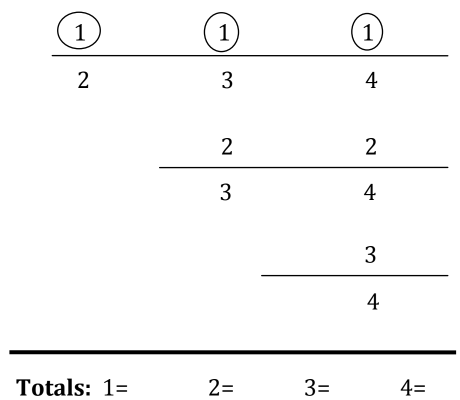
Paired Weighting Process:

Purpose: To compare each item on a list of relatively equal or difficult to rank items to each other item and decide which is more important or should come before the other.

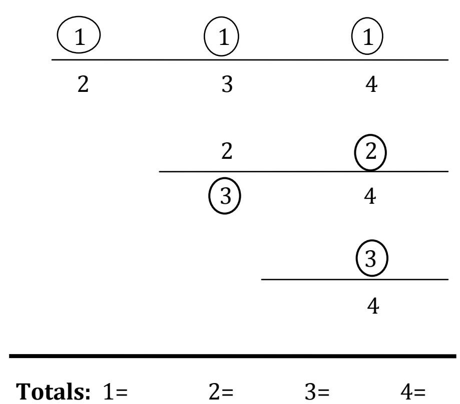
Instructions:

1. Assign a number to each item. For example, the items on a list of types of food to have for dinner might be:
   1. Mexican
   2. Chinese
   3. Italian
   4. American

2. Compare Item #1 to each of the other items and on the scorecard, circle the number of the item that you feel is more important in each comparison.



3. Compare item #2 to #3, #4, etc. and circle the number of the item that you feel is more important. Continue until you have chosen between all the items and circled your choices on the scorecard.



4. Tally all 1s, 2s, 3s, etc.

(1) / 2	(1) / 3	(1) / 4
	2 / (3)	(2) / 4
		(3) / 4

Totals: 1= 3 2= 1 3= 2 4= 0

5. The highest number is top choice; the lowest number is bottom choice. In this example, the choice for dinner is #1: Mexican food since that received the highest number.

If necessary, use the paired weighing worksheet in Appendix D to assess the relative value of individual and fluctuating variables.

Step #3: Assess comfort levels before and during each session.

A. Use the following grid to assess the overall comfort level of the classroom before and during each session. Then use that assessment to modify individual assessments as necessary.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Physical Comfort Level: Room is comfortable and quiet											**Physical Comfort Level:** Room is hot, crowded, and noisy

Minimum Rescue → **Maximum Rescue**

Step #4: Reevaluate on a regular basis.

References Cited

Armoni, M., Gal-Ezer, J., & Tirosh, D. (2005). Solving problems reductively. *Journal of Educational Computing Research*, 32, pp. 113-129.

Brooks, F.P., Jr. (1987). No silver bullet: Essence and accidents of software engineering. *Computer Magazine*, 4, pp. 10-19.

Denner, J., & Bean, S. (2006). Girls, games, and intrepid exploration on the computer. In E.M. Trauth (Ed.), *Encyclopedia of Gender and Information Technology*, pp. 727-732. Hershey, PA: Idea Group Reference

Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research*, Vol. 37(2), pp. 131-150.

Inkpen, K., Booth, K.S., Klawe, M., & Upitis, R. (1995). Playing together beats playing apart, especially for girls. *Proceedings of Computer Supported Collaborative Learning (CSCL)* (pp. 177-181). Hillsdale, NJ: Lawrence Erlbaum Associates.

International Society for Technology in Education's (ISTE) Educational Technology Standards for Students, www.iste.org/Content/NavigationMenu/NETS/ForStudents/2007Standards/NETS_for_Students_2007.htm

Jonassen, D. (December 2000). Toward a design theory of problem solving. *Educational Technology Research and Development*, Vol. 48(4), pp. 63-85.

Key Instructional Strategies, www2.ups.edu/community/tofu/lev2/intlessons/keystrat.htm

Littleton, K., & Light, P. (Eds.). (1999). *Learning with computers: Analyzing productive interaction.* London: Routledge.

Margolis, J. & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT Press.

McKenna, P. (2004). Gender and black boxes in the programming curriculum [Electronic version]. *ACM Journal on Educational Resources in Computing (JERIC)*, 4(1).

Schoenfeld, A. (1985)). *Mathematical problem solving*. Orlando, FL: Academic Press.

Turkle, S., & Papert, S. (1992)). Epistemological pluralism and the revaluation of the contrete. In I. Harel & S. Papert (Eds*.), Constructionism* (pp. 161-191). Norwood, NJ: Ablex.

Vygotsky, L.S. (1978). *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press

What is Problem Solving, Instructional Strategies Online, olc.spsd.sk.ca/DE/PD/instr/strates/psolving/index.html

Additional Resources

Laughlin, P., Hatch, E., Silver, J. and Boh, L. Groups Perform Better Than the Best Individuals on Letters-to-Numbers Problems: Effects of Group Size. University of Illinois at Urbana Champaign; *Journal of Personality and Social Psychology*, Vol. 90, No. 4. Abstract available at: http://www.apa.org/releases/group042306.html

Phillips, P. (2009) *Computational Thinking: A Problem-Solving Tool for Every Classroom*. Includes information about computational thinking and numerous links to demonstration sites. http://csta.acm.org/Resources/sub/HighlightedResources.html

Problem Solving Techniques. www.increasebrainpower.com

Real-Life Problem Solving: A Collaborative Approach to Interdisciplinary Learning Edited by Beau Fly Jones, Claudette M. Rasmussen, and Mary C. Moffitt. 1997. American Psychological Association. Describes an approach to interdisciplinary learning in which students and teachers investigate open-ended problems.

Schur, Y. and Kozulin, A. Cognitive Aspects of Science Problem Solving: Two Mediated Learning Experience Based Programs. *Journal of Cognitive Education and Psychology*, Vol. 7, No. 2, 2009, pp. 266-287(22). Results of programs to improve middle school students' problem solving skills.

TRIZ, 40 Principles for Innovative Problem Solving, http://www.triz.org.uk/papers/FortyPrinciples.pdf
TRIZ, the "theory of inventor's problem solving," is a method for generating innovative ideas and solutions.

Winerman, L. Where did that idea come from? *Monitor on Psychology*, American Psychological Association, Vol. 36, No. 10, Nov 2005, p. 34. Online version available at: http://www.apa.org/monitor/nov05/idea.html. Research into the seat of creativity and problem-solving ability in the brain.

Winger, W. (2005) Double-Entry A-Ha! Method. http://www.winwenger.com/deam.htm
An idea-generating and problem-solving method

Vardi, I. and Ciccarelli, M. Overcoming problems in problem-based learning: a trial of strategies in an undergraduate unit. *Innovations in Education & Teaching International*, Vol. 45, No. 4, Nov 2008, pp. 345-354(10). Reports on a range of strategies to overcome common problems in problem-based learning environments.

Contacts for More Information

Questions about the Programs:

Jill Denner
ETR Associates
831-438-4060 ext. 264
jilld@etr.org

or

Steve Bean
ETR Associates
831-438-4060 ext. 176
steveb@etr.org

Questions about this Guide:

Pat Rex
ETR Associates
831-438-4060 ext. 106
patr@etr.org

Appendix A

Problem Solving Process

Step 1: **Define the Problem**
What is the problem?
What do you want to happen that is NOT happening?
What IS happening?

Step 2: **Determine Possible Causes of the Problem**
Why does the problem exist?
What are the possible causes?
What is the best way to find out?

Step 3: **Propose and Test Solutions**
How can you fix the problem?

Step 4: **Apply the Best Solution**
What is the best way to fix the problem?
Why is it the best?

Appendix B

Sample Storytelling Alice Code

Created by: staff

World

Events

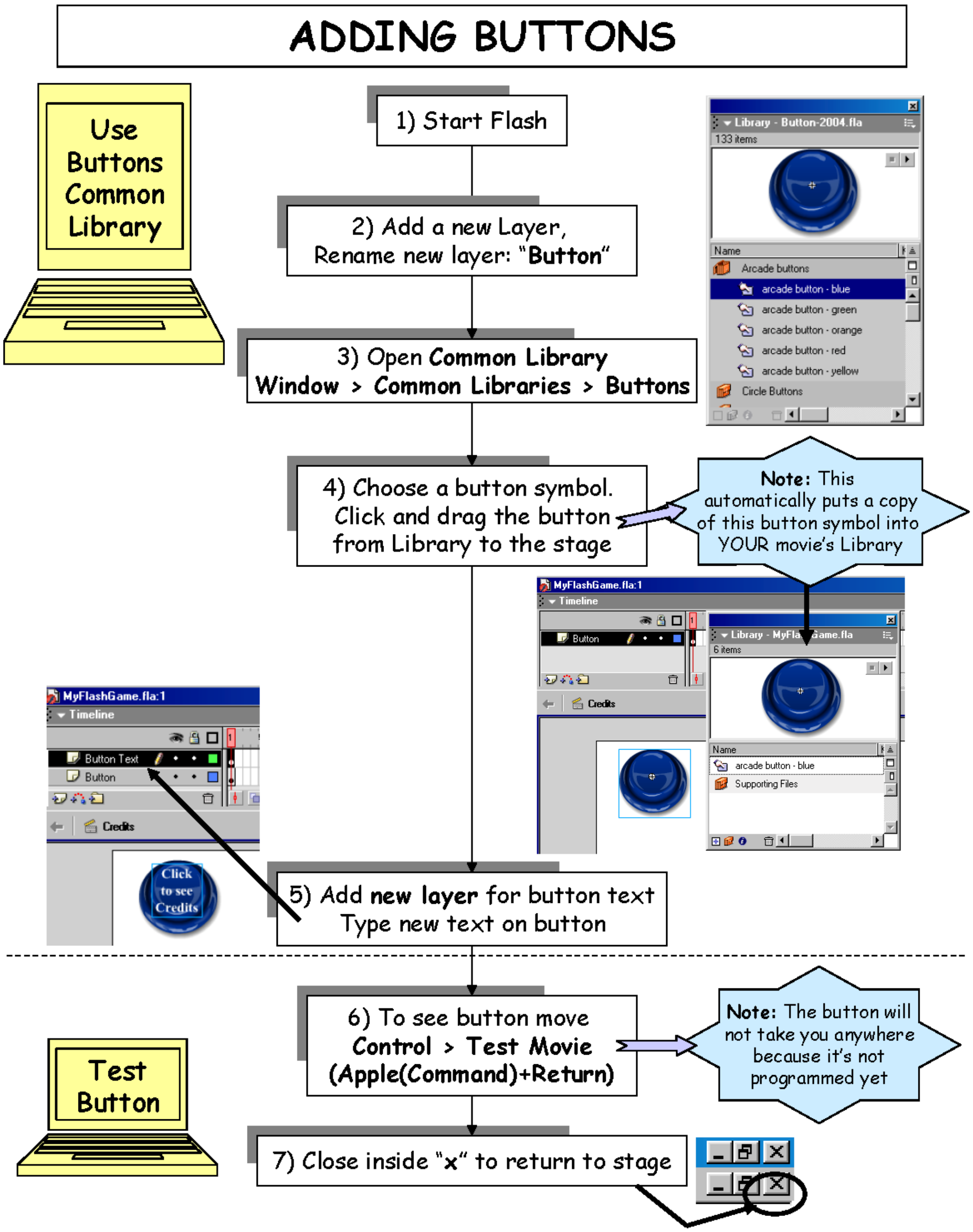
When the world starts	
Do:	**World.scene 1 method**

Methods

World.scene 1 method ( )
No variables

- **Harry the Lion say I am now going to roar.**
- **Harry the Lion.Roar**
- **Harry the Lion say Now I'm going to disappear.**
- **Harry the Lion set isShowing to false** duration = **3** seconds
- **Harry the Lion set isShowing to true** duration = **3** seconds
- **Harry the Lion say Now I'm going to attack!**
- **Harry the Lion.Attack**
- **Harry the Lion say goodbye**
- **Do together**
  - **Harry the Lion turn right 0.25** revolutions
  - **Harry the Lion.Walk** Distance = **10**

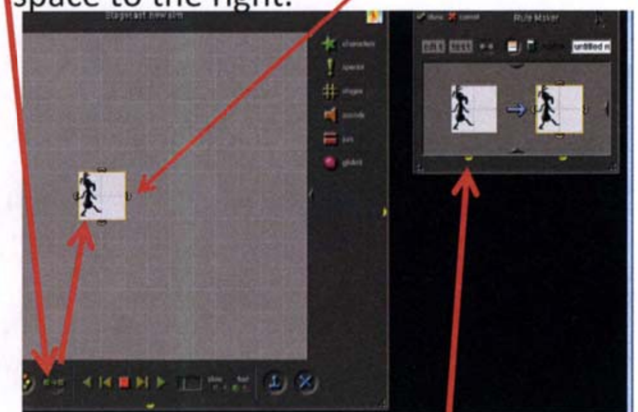
Appendix C



Developed by ETR Associates, as part of grant 0217221 from the National Science Foundation

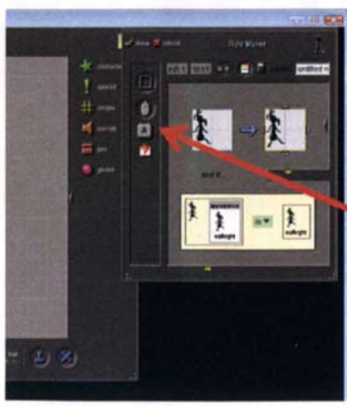
Creator Handout # 8
Moving Characters with Arrow Keys

Step 1: Click on the Rule Tool, and put it onto the character. Stretch the window one space to the right.

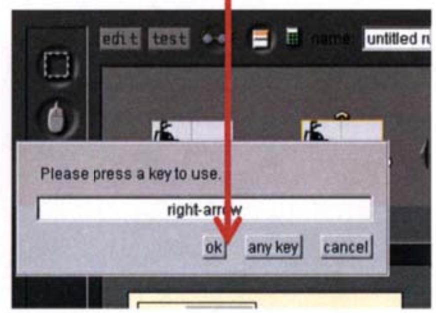


Step 2: Click on the yellow tab under the rule.

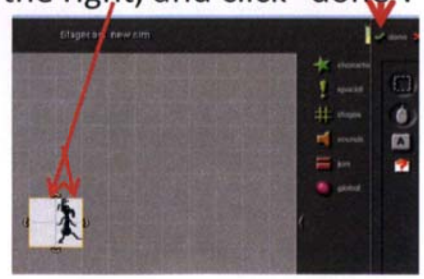
Step 3: Click on the "Use Keys" button – the little "A".



Step 4: Press the right-arrow key, on your keyboard, to tell Creator you want him to move "right" when that key is pushed. Then click "OK".



Step 5: Move your character one space to the right, and click "done".



Step 6: Go through the same steps for "left", "up", "down."

Appendix D

Paired Weighting Worksheet for Individual and Fluctuating Variables

Variables:

1. Frustration or tolerance level
2. Personal definition of success
3. Personal investment in class/program
4. Confidence level
5. Experience and skill level

1	1	1	1
2	3	4	5
	2	2	2
	3	4	5
		3	3
		4	5
			4
			5

Totals: 1= 2= 3= 4= 5=