Chapter 5

Instructional Software for Student Learning

Learning Outcomes

After reading this chapter and completing the learning activities, you should be able to:

5.1 Identify characteristics, sources, and roles for instructional software in order to evaluate them to facilitate adoption. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

5.2 Describe the selection criteria, benefits, challenges, and integration strategies of drill and practice functions for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

5.3 Describe the selection criteria, benefits, challenges, and integration strategies of tutorial functions for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

5.4 Describe the selection criteria, benefits, challenges, and integration strategies of simulation functions for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

5.5 Describe the selection criteria, benefits, challenges, and integration strategies of game and gamification functions for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 4—Collaborator, 5—Designer; 6—Facilitator; 7—Analyst)

5.6 Describe the selection criteria, benefits, challenges, and integration strategies of problem-solving functions for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

5.7 Describe the selection criteria, benefits, challenges, and integration strategies of personalized learning system functions (PLS) for relevant instructional situations. (ISTE Standards for Educators: 1—Learner; 2—Leader; 5—Designer; 6—Facilitator; 7—Analyst)

Technology Integration in Action: Math by Design

GRADE LEVEL: Middle school

CONTENT AREA/TOPIC: Geometry and measurement

LENGTH OF TIME: Two weeks

PHASE 1 Analysis of Learning and Teaching Assets and Needs

Step 1: Analyze problems of practice (POPs)

Ms. Igwe was a veteran middle school mathematics teacher who was always able to get a high percentage of her students to pass required state tests, yet she continued to struggle with engaging them in geometry skills. She was always looking for examples to show her students practical applications of geometry concepts in everyday life, but each year that she taught, she found students were increasingly disinterested in them. Although they were learning geometry skills and passing tests, she doubted that they would remember the concepts or be able to connect these skills to practical problems and situations in the world around them.

Step 2: Assess technological resources of students, families, teachers, and the school

Ms. Igwe knew that most of her students had Internet access and used the web to play games or used social media at home on computers, tablets, or phones. A few students did not have computers or tablets at home but still had access to phones. Her classroom had six newer computers that she often used for center-type activities or individual computing activities. If needed, she could check out some tablets from the library. She often searched for interesting web-based mathematical representations or instructional materials to support her teaching..

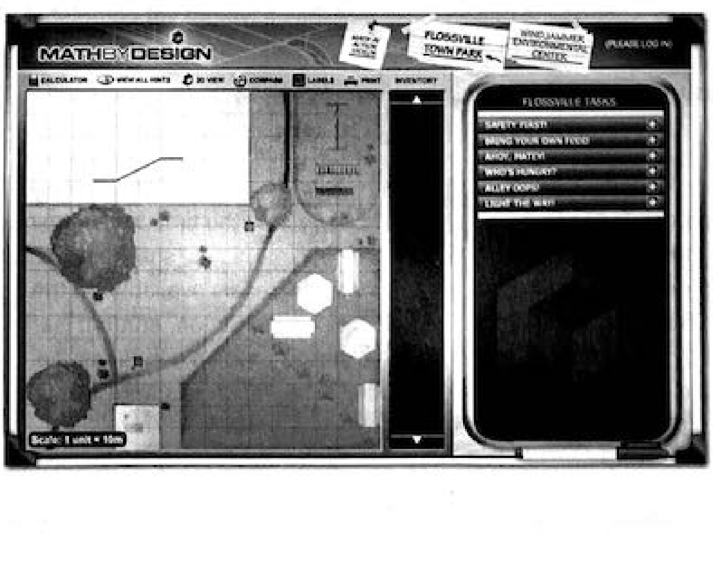
Step 3: Identify technological possibilities

While perusing the state's websites to support teaching science, technology, engineering, and mathematics (STEM) subjects, Ms. Igwe learned of Math by Design in which students solved geometry and measurement problems by becoming "junior architects" and redesigning a park in an imaginary town called Flossville (see Figure 5.1). She thought that these hands-on activities that required applying geometry skills in a fun environment might stir students' interest in her geometry instruction more than just seeing examples.

Ms. Igwe spent several hours reviewing the website's materials and working through each of the unit's design problems. Although she was a competent geometry teacher, she realized she would have to teach each of the geometry skills in a way that would help students connect them with Flossville design tasks. It made her think about how to sequence and assess concepts in a much different way than she was used to doing, but the website's teaching hints and sample lessons were very helpful.

Figure 5.1 Flossville

SOURCE: Reprinted by permission of Maryland Public Television, © 2009. http://mathbydesign.thinkport.org



PHASE 2 Design of the Integration Framework

Step 4: Decide on learning objectives and assessments

Ms. Igwe also realized that she wanted her students to achieve more with these materials than just passing tests. She wanted the experience to be interesting and challenging in ways that would change their attitudes about geometry. She developed the following outcomes, objectives, and assessments that would help her determine later whether she had accomplished her aims:

Outcome: Select and use appropriate geometry concepts to solve graphic design problems.

- **Objective:** All students working in groups will create correct solutions for assigned Flossville design problems.
- **Assessment:** Score at least 20 of 25 on a teacher-designed rubric assessing group work and design solutions.

Outcome: Transfer geometry skills from Flossville to teacher-designed situations.

- **Objective:** Working individually, all students will achieve a score of at least 80% on a post-unit test requiring transfer of geometry concepts covered in Flossville to novel design scenarios.
- **Assessment:** Teacher-designed, multiple-choice test of 12 scenario items (two scenario items per geometry concept).

Outcome: Demonstrate achievement of state geometry standards.

- **Objective:** All students will achieve at least a passing score (80%) on the state-required post-course geometry test.
- **Assessment:** State multiple-choice, post-course geometry exam.

Outcome: Enjoy learning geometry skills and appreciate their value.

- **Objective:** At least 90% of students will demonstrate improved attitudes toward learning geometry skills.
- **Assessment:** A teacher-designed, 10-item survey that students complete anonymously.

Step 5: Design integration strategies and determine relative advantage

Ms. Igwe felt that students would find the Math by Design materials motivating if they completed the design tasks themselves using the hints and tutorial explanations, so she would provide very little direct instruction. Even though the site was really a problem-solving environment, it had a gamelike feel to it, and she felt they would do well with this project after she had taught several of the prerequisite geometry skills. With these aspects in mind, she created the following activity sequence:

Day 1: Students complete a pre-assessment of geometry attitudes using the survey she had created. Then, Ms. Igwe introduces the environment and its design tasks and illustrates on the whiteboard how to access the hints, calculator, and other features. With the classroom computers and tablets, students work in pairs to solve an assigned design task. Each of the groups becomes "expert" in one of the design tasks and the skills and procedures required to complete them. Then, each individual student works individually on all design tasks with the "student experts" available for assistance.

Days 2–4: The groups begin their work completing the first design task in the classroom.

Days 4–9: All individual students work on the design tasks that they have not yet completed. Some students act as experts and assist other students with the design tasks.

Day 10: The class completes the post-project assessment and geometry attitudes survey.

Relative Advantage

After planning her integration approach but before teaching it, Ms. Igwe took the time to determine the relative advantage of using Flossville to meet the learning needs of her geometry learners. So, she RATified her proposed lesson using her knowledge of teaching geometry in the past, the educator materials available on the site, and her experience exploring the environment. Refer to Figure 5.2 for the aspects of instruction, student learning, and curriculum that she felt would be impacted by her students working on the design problems in Flossville. She was pleased to find that the lesson held many transformative elements and felt there was much relative advantage to do the lesson.

(Continued)

Figure 5.2 Ms. Igwe's RATified Lesson

	Instruction	Learning	Curriculum
Replacement Technology is a different means to same end.			• Aligned with current mathematics (e.g., proportional reasoning, volume of cylinder, area of composite figure) • Aligns with ISTE standards
Amplification Technology increases or intensifies efficiency, productivity, access, capabilities, etc., but the tasks stay fundamentally the same.	• Computers and tablets facilitate individual and small group, problem-based inquiry in the environment		
Transformation Technology redefines, restructures, reorganizes, changes, and creates novel solutions.	• Scaffolded through hints, tutorials, and examples within Flossville • Progress feedback available at any time	• Situates learners as architects (real professionals) • Students apply geometry and measurement to simulated real-life tasks to build a park • Additional challenges to apply developing knowledge	• Supports reading in the content areas

Step 6: Prepare instructional environment and implement the lesson

Ms. Igwe knew that working with technology materials required some additional planning, so she did the following to make sure that everything needed for the project was in place.

Materials:

Ms. Igwe made notes on which geometry skills were required for each task so she could be sure that she covered these in her instruction and reviews before students began working with Math by Design. She made a chart showing the student groups assigned to the first of six design tasks. She set up a schedule designating which students would work on which computing devices each day.

Computer scheduling:

The library tablets were much in demand among the teachers, so she met with the librarian and explained the project to her before requesting use of the tablets for 7 days.

Parent letter:

The Math by Design website provided a letter about the project to send home to parents. She thought it was a good idea to let parents know what students were doing, so she printed the letters for students to take home. She also posted a copy of the letter on her page on the school's website.

Backup plans:

In case any technical issues arose, she made some "just-in-case" plans. Her backup plan was to show the whole class one or more of the short Math in Action videos on the Math by Design website, hold a discussion using the questions the site provided, and follow up by having students work the practice problems the site offered to support each video.

PHASE 3 Post-Instruction Analysis and Revisions

Step 7: Analyze lesson results and impact

Ms. Igwe was delighted with the students' obvious enthusiasm for the Math by Design activities. They liked completing the first design task with a partner and then serving as an "expert" as other students completed all tasks individually. In this way, all the students were experts and learners. Some students began to bring in and share their own examples of geometry principles and examples they had identified outside the classroom, and the geometry attitudes survey reflected a definite pre- to post-lesson improvement. Several students asked if she had materials on careers in architecture. The only real problem she noticed was that one of the design tasks was more difficult than the others and the "student experts" assigned to it required more teacher assistance than others. Also, on the post-assessment that measured the transfer of skills to design situations similar to Flossville, she found that some geometry skills transferred more than others. There was no difference in the number of students passing the end-of-course geometry test, which established that the lesson was as efficacious as her past instruction.

Step 8: Make revisions based on results

Ms. Igwe made a note to remember to make the following changes:

- Do more pre-teaching on the concepts in the more difficult Flossville design task.
- Provide more review and discussion (aided by Math in Action videos) before post-assessment.
- Expand the Flossville project to include other Math by Design environments.
- Use the bulletin board in the computer lab to display printouts from the Flossville environment and describe the geometry standards that students had achieved.

Step 9: Share lessons, revisions, and outcomes with other peer teachers

Ms. Igwe felt that the lesson had successfully improved students' knowledge of geometry's role in real-world situations. She tweeted out a link to the Math by Design website and pinned it on Pinterest for her math teacher followers. She followed up with the school librarian who added the site to her website that logged useful content resources for future teacher consultations.

Introduction

This chapter introduces you to some of the oldest and most well-researched technology-based strategies in the field. Each decade gives these strategies a fresh face using the newest technologies, yet their underlying functions and benefits remain the same. Instructional software is a general term for computer programs or apps used specifically to deliver instruction or assist with the delivery of instruction on a topic. Instructional software includes pre-programmed content material that often is instructionally sequenced. Instructional software is used solely to support instruction and/or learning and therefore are designed for specific K–12 content areas and learner developmental levels.

Instructional software is introduced according to its teaching function: drill and practice, tutorial, simulation, games and gamification, problem solving, and personalized learning. However, remember that these teaching functions meet certain learning needs, so a specific software app could serve more than one function. For example, an app can offer a tutorial sequence on chemical compounds and then let students safely simulate making the compounds. Integrating instructional software products means matching their underlying teaching functions to what and how students need to learn. Look for five things as you read about each function of instructional software in this chapter: defining characteristics, selection criteria, benefits, challenges, and integration strategies.

Introduction to Instructional Software

In the 1960s and 1970s, educators and software developers alike began to pursue the idea that computers could be programmed to teach. Some believed that education would be more efficient if computers took over the traditional role of teachers. Some 50 years later, we talk less about computers replacing teachers and more about helping teachers transform the teaching process. This chapter shows how software empowers teachers rather than replaces them. We begin with some basic definitions and terms for instructional software and the functions that these products can play in teaching and learning.

Definition of Instructional Software

Instructional software is a general term for computer programs or apps used specifically to deliver instruction or assist with the delivery of instruction on a topic through demonstrations, examples, and explanations. Unlike software for productivity activities (described earlier in this text), instructional software includes pre-programmed curricular material that often is instructionally sequenced. Instructional software is used solely to support instruction and/or learning.

In the early days—when the purpose of instructional software was primarily tutoring—it was called **computer-assisted instruction (CAI)** or **courseware**. These terms are still in common use, but some kinds of instructional software such as simulations, instructional games, and problem-solving software are designed with more constructivist purposes in mind; they support rather than deliver instruction. Therefore, teachers also can hear instructional software referred to as computer-based instruction (CBI), computer-based learning (CBL), or computer-assisted learning (CAL) or in more generic terms, such as software learning tools.

Teaching Functions in Instructional Software

Software serves the following teaching functions: **drill and practice, tutorial, simulation, game** or **gamification, problem-solving**, and **personalized learning**. Many of today's software packages fulfill several different functions. For example, a language-learning system could have a number of drill activities that have problem-solving and game functions. Although software teaching functions are distinct, developers tend to use the terms for these functions interchangeably. Some developers refer to a drill program that gives extensive feedback as a tutorial. Others refer to simulations or problem-solving functions as games.

Instructional software reflects the same six functions, but in light of current trends toward multiple-function software systems, teachers must analyze software carefully to determine which instructional function(s) it serves to ensure that it supports their specific teaching and learning needs. Teachers may not be able to categorize an entire software as a drill or a simulation, but it is possible and desirable to identify whether it provides, for example, science vocabulary skill practice (drill and practice function) and/or opportunities for studying plant growth in action (simulation function). Each software function serves a different purpose during learning and, consequently, has its own appropriate integration strategies.

The first instructional software products reflected the behavioral and cognitive learning theories that were popular at the time. Some software functions (e.g., drill and practice, tutorial) remain focused on directed strategies that grew out of these theories, delivering information to help students acquire and retain information and skills. Later, instructional software was designed to support more constructivist aims of helping students explore topics and generate their own knowledge. Therefore, some software functions (e.g., simulation, games) can be used in either directed or constructivist ways, depending on how they are designed or used.

Application Exercise 5.1 Instructional Software Functions

Selecting Appropriate Instructional Software

It can be difficult for teachers to identify new instructional software options or to know which ones that are available are worthwhile. The following section describes sources that help review software and introduces evaluation criteria for teachers to use to determine whether specific software is appropriate for classroom needs.

SOFTWARE SOURCES The number of commercial instructional software products has grown so much that new sites have emerged to help teachers, parents, and schools select ones that meet criteria for quality and alignment to Common Core State Standards

(CCSS) or state standards. Some sites include edshelf, Learning List, What Works Clearinghouse, Lea(R)n Community Product Library, EdSurge Product Index, and Common Sense Education (Molnar, 2013). All of these sites are free except for Learning List, which Molnar says provides additional analysis of both digital and print resources. We recommend using these review sites along with our recommendations for evaluating appropriate instructional software.

CRITERIA FOR EVALUATING SOFTWARE When you start to search for instructional software to meet your teaching or student learning needs and find some possibilities, you will need to determine whether it is appropriate. Using a range of resources that provide suggestions for finding and evaluating software and apps (Ed Tech Rapid Cycle Evaluation, n.d.; Hirsh-Pasek et al., 2015; Karolcík, Cipková, Hrusecký, & Veselský, 2015; Lee & Cherner, 2015; Ok, Kim, Kang, & Bryant, 2016), we have developed the following selection criteria that may assist you in determining the instructional strength of the resources you have discovered. We emphasize that you should have specific learning need(s) or knowledge objective(s), which are developed in Steps 1 and 4 of the TTIPP model, in mind when choosing any software.

- **Content**—The software should contain rigorous disciplinary content aligned with content standards that matches your students' learning needs at their developmental level. Ask whether the content is accurate and current and has enough content to meet a range of learner levels to provide differentiated instruction across the timeframe of use.
- **Instruction**—Aim for meaningful learning to be accomplished, such as learning that is purposeful, is personally relevant, connects with learners' prior knowledge, and provides practice that applies knowledge to authentic, real-life situations (Hirsh-Pasek et al., 2015). Assessment elements should track student progress and provide meaningful information that assists in determining students' learning gains, needs, and progress and that is accessible to both the student and teacher (and sometimes parents).
- **Integration model**—Consider the underlying instructional model within the software, such as directed or constructivist instructional approaches, and determine whether that approach matches or complements how you will integrate the software within your intended instruction and overall lesson.
- **Special needs features**—Ok et al. (2016) developed an extensive rubric for selecting apps for students with learning disabilities. These authors emphasize seven research-based elements that contribute to effective instruction: inclusion of clear objectives, strategies for developing skills and subskills, three or more examples for each skill, five or more skill practice opportunities, error feedback with correct answer, error analysis, and progress tracking. The Adapting for Special Needs feature box highlights what to consider for students who need physical accommodations.
- **Learner involvement**—Examine the extent of student agency by determining how learners are involved in the software. Software should be responsive to users' actions, but Hirsh-Pasek et al. (2015) distinguish different types of active learning, such as mind-off physical activity (swiping or tapping), mind-on mental activity (solving puzzles), and interactivity with a resource (choosing a character or path or using an in-game microscope).
- **Learner engagement**—Software features can impact engagement through a range of elements including behavioral (motivating persistence and effort in activities), emotional (generating affective reactions to the experience), cognitive (offering problems that challenge but do not frustrate), assessment (providing immediate and differentiated feedback and positive praise for achievement), and instructional (providing open-ended tasks) (Hirsch-Pasek et al., 2015).

Box 5.1: Adapting for Special Needs

Instructional Apps, Software, and Web Resources

When selecting apps, software, and web resources for the classroom, teachers, technology specialists, and administrators should be mindful of special needs that can impact student access and engagement. For example, some students might not be able to use the standard mouse and keyboard because of physical disabilities that impair gross and fine motor movement. In these situations, schools need to provide items such as the following that allow an individual with a special need to operate the computer with adaptive equipment:

- Expanded keyboards, such as Key Technologies' Intellikeys
- Keyboards with guards, such as the Beyond Adaptive website
- Adaptive switch interfaces, such as those listed in the Enablemart catalog

Options are available for students with sensory impairments who are unable to access software that has not been properly designed to be accessible. Students who are blind use software to read the information on the screen to them. Be sure to ask the vendor whether the software is compatible with screen readers such as Job Access for Windows and Speech (JAWS). Students who are deaf are not able to access information presented in audio format unless the audio is captioned or a text transcript is available. Look for JAWS at the Freedom Scientific website.

Finally, students with cognitive disabilities could encounter difficulty using apps, software, and web resources that have extensive commands and complex interfaces. In these situations, look for software such as the following that offers simplified user interfaces:

- Scholastic Keys at Tom Snyder, Inc.
- Inspire Data at Inspiration, Inc.

For more about accessible software, see the Microsoft's Developer Networks Designing Accessible Applications website.

—Contributed by Dave Edyburn

- **Technical and implementation aspects**—Determine the software's technical ease of use for both the learners and the teacher. Will learners understand and navigate through the software successfully and is it accessible for all learners? Hirsh-Pasek et al. (2015) found that younger learners control learning better on touchscreens because they reduce the need for mouse motor control skills. Determine whether the software functions within your existing technical resources. What user support or documentation exists? Can you try it for some period of time for free to ensure its applicability?
- **Design and aesthetics**—Examine the multimedia elements, such as graphics, text, and sound, to determine whether they are connected to the learning tasks, not gratuitous additions. The graphic elements and design need to match the developmental age and interests of target learners to ensure that they find the resource engaging.
- **Evidence of effectiveness**—Investigate whether strong evidence exists that this instructional resource leads to the learning gains in the knowledge areas that you are targeting. Is the evidence from an independent source, not the product developer?

These criteria can assist with your initial review of instructional resources. In the following sections, you will find additional criteria to consider when reviewing instructional products that have one or more of the specific functions: drill and practice, tutorials, simulations, games and gamification, problem solving, or personalized learning. The Adapting for Special Needs feature box highlights more considerations for choosing software for learners with special needs.

✓ **Check Your Understanding 5.1**

Characteristics of Drill and Practice Functions

Drill and practice software provides exercises that have students work example items, usually one at a time, and receive feedback on their correctness. Programs vary considerably in the kind of feedback that they provide in response to student input. Feedback can range from a simple "OK" or "No, try again" to elaborate animated displays or verbal explanations. Some programs simply present the next item if the student answers correctly. Types of drill and practice are sometimes distinguished by the way the program tailors the practice session to student needs. Common types of drill functions follow:

- **Flash card activity**—This is the most basic drill and practice function, arising from the popularity of real-world flash cards. A student sees a set number of questions or problems presented one at a time and chooses or types an answer, and the program responds with positive or negative feedback depending on whether the student answered correctly.
- **Chart fill-in activities**—In this kind of practice, students are asked to complete a whole set of answers (e.g., multiplication facts) by filling in a chart, usually on a timed basis to test for fluency. Then they receive feedback on all the answers at once.
- **Branching drill**—In this more sophisticated form of drill and practice function, a branching drill moves students to advanced questions after they get a number of questions correct at some predetermined mastery level; it can also send students back to lower levels if they answer a certain number of questions wrong. Some programs automatically review questions that students get wrong before going on to other levels. Students probably will not realize that branching is happening because the program does it automatically. Sometimes the program can congratulate students on good progress before proceeding to the next level, or it can allow them to choose their next activities.
- **Extensive feedback activities**—In these drills, students get more than just correct/incorrect feedback. Some programs give detailed feedback on why the student gave a wrong answer. This feedback is sometimes so thorough that the software function is often mistaken for a tutorial. (See the next section for a description of tutorial functions.) However, the function of a drill is not instruction but practice *after* instruction. Consequently, drill and tutorial functions have different integration strategies.

Figure 5.3 Vocabulary Practice Uses Drill and Practice

SOURCE: Images reprinted by permission of BrainPop. http://www.brainpopesl.com BrainPop © 1999–2011.

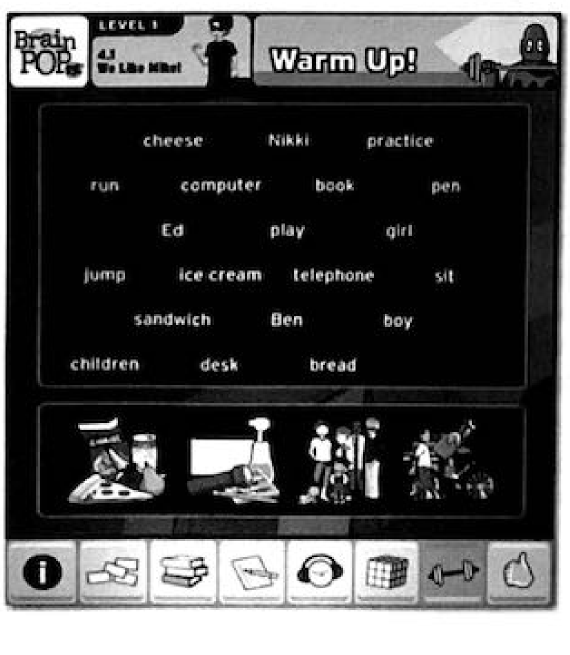


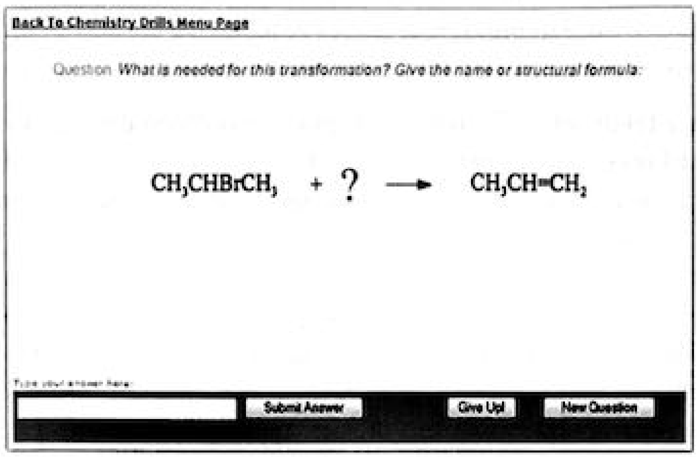
Figure 5.3 shows a screen of a software that has a drill and practice function in which learners practice word identification by dragging each word into the correct category.

Selecting Appropriate Drill and Practice Software

In addition to meeting general criteria for instructional software, well-designed drill and practice programs should also meet specific criteria:

- **Control over the presentation rate**—Unless questions are part of a timed review, students should have as much time as they wish to answer and examine the feedback before proceeding to the next questions. Students usually signal their readiness to go to the next one by simply pressing a key.
- **Answer judging**—If programs allow students to enter a short answer rather than simply choosing one, a good drill program must be able to discriminate between correct and incorrect answers.

Figure 5.4 *Chemistry Formulas* Software Uses Drill and Practice
SOURCE: Images reprinted by permission of Meta-Synthesis. http://www.chemistry-drills.com.



- **Appropriate feedback for correct and incorrect answers**—If students' responses are timed or if their session time is limited, learning can be undermined if students find more motivation in simply moving quickly to the next question to meet time demands rather reading information about the correctness of the response. When drills do give feedback, they must avoid two common errors. First, avoid elaborate displays for feedback, which over time cease to motivate students. Instead, feedback should be simple and display quickly. Second, some programs inadvertently motivate students to get *wrong* answers by giving more exciting or interesting feedback for wrong answers than for correct ones.
- **Characteristics tailored to young learners**—Luik (2011) offers advice specific to programs for young learners. Recommendations from her study focus primarily on keeping instructions and procedures simple and avoiding screen elements that could distract the students.

Figure 5.4 provides a screen of software with a drill and practice function; learners practice creating correct chemical formulas by filling in the missing information. The learners control how quickly they answer the questions.

Benefits of Drill and Practice

The benefits of drill and practice software have been well established by research. Indeed, their effects were so well documented in the early days of computer-based learning that little current research focuses on it. It became clear long ago that drill activities can provide the effective rehearsal that students need to establish newly learned information into long-term memory (Merrill & Salisbury, 1984; Salisbury, 1990). To help them master higher order skills more quickly and easily, students must have what Gagné (1982) and Bloom (1986) call **automaticity**, or automatic recall of lower order prerequisite skills. Teachers of students with learning disabilities have found drill programs useful (Graham, Bellert, Thomas, & Pegg, 2007).

Schoppek and Tulis (2010) found that fluency in basic math skills is essential for mathematical problem solving. Their results with third graders showed that even a moderate amount of individualized practice with drill software greatly improved both arithmetic skills and problem solving, an impact that continued in follow-up testing 3 months later. The researchers felt that individualized practice with drill software was a more efficient use of time than other kinds of practice. Drill software has been found to yield equivalent or better benefits when compared to paper-pencil practice, and drill software is both more efficient and often more appealing to students.

Focus on educational accountability and meeting standards in the Every Student Succeeds Act (ESSA) has led to more use of directed teaching strategies. Although curriculum also emphasizes problem solving, higher order skills, and deeper learning, teachers still assign students practice for many skills to help them learn and remember correct procedures. Drills and tutorials support directed strategies that are ideal for preparing students for skill-building or tests. The following are acknowledged benefits of drill software as compared to paper exercises:

- **Immediate feedback**—When students practice skills on paper, they often do not know until much later whether they did their work correctly. If they continue to provide incorrect answers, students could be memorizing the wrong skills or information. Drill and practice software informs them immediately of whether their responses are accurate so that they can make quick corrections. This helps both "debugging" (identifying errors in their procedures) and retention (placing the skills in long-term memory for future access). The drill and practice downloadable app shown in Figure 5.5 provides learners immediate feedback if they are identifying the correct musical note when practicing keyboard note identification.
- **Increased motivation**—Many students refuse to do the practice they need on paper because they have failed so much that the whole idea is abhorrent, they have poor handwriting skills, or they simply dislike writing. In these cases, computer-based practice could motivate these students to do the practice they need. Computers don't get impatient or give any looks when a student gives a wrong answer.
- **Saving teacher time**—Because teachers do not have to present or grade drill and practice activities, students can practice on their own while the teacher addresses other students' needs. The curriculum has dozens of areas in which the benefits of drill and practice apply, for example
    - Math computation
    - Typing skills
    - English and foreign language vocabulary
    - Place-based knowledge, for example, about countries and capitals
    - Preparation for SAT, ACT, TOEFL, and other high-stakes tests
    - Understanding musical keys and notations

Figure 5.5 *Name the Note* Software Uses Drill and Practice

Source: Images reprinted by permission of Rifftech. http://www.musicdrills.com

Video Example 5.1 Using Drill and Practice Software

In this video, a principal describes classroom uses of drill and practice software across many content areas. She describes teachers' use of drill and practice software when students need to practice math skills that are prerequisites for higher learning, which saves classroom time.

Challenges Related to Drill and Practice

Although drill and practice can be extremely useful to both students and teachers, it is also frequently criticized for three reasons:

- **Instructional overuse or misuse**—Some criticize teachers for presenting drills for overly long periods or for teaching functions that drills are ill suited to accomplish. For example, teachers give students drill and practice software as a way of introducing new concepts rather than for practicing and reinforcing familiar ones.
- **Criticism by constructivists**—Because drill and practice software is identified so closely with directed instructional methods, critics claim that introducing isolated skills and directing students to practice them contradicts the trend toward restructured curriculum in which students engage in deeper learning by using skills in an integrated way within the context of their own projects that specifically require the skills (Office of Educational Technology, 2016; U.S. Department of Education, 2013).
- **Inequity in use**—Researchers have found that drill and practice software is often used more by students of color and/or who are in low-income families. For example, in a 7-year longitudinal study of Florida schools, Hohlfeld, Ritzhaupt, Wilson, & Dawson (2016) found that students in low-income schools were more likely than their peers in affluent schools to use software for directed instruction, such as those with drill and practice features.

Despite these criticisms, it is likely that some form of drill and practice software will be useful in many classrooms for some time to come. Teachers should seek to identify needs that practice and feedback can meet and use the software in ways that take advantage of its capabilities.

Integration Strategies Using Drill and Practice

Teachers mainly take advantage of drill and practice software to give students practice using skills. Teachers can consider the following strategies and guidelines for integrating these software:

STRATEGIES FOR USING DRILL AND PRACTICE Strategies for integration of drill and practice software include the following:

- **Supplement and/or replace worksheets and homework exercises**—Use when students lack automaticity in skills that are prerequisite to higher order ones. The motivation, immediate feedback, and self-pacing can make practicing skills on the computer more productive for students than on paper.
- **Prepare for tests**—Use when students need to prepare to demonstrate mastery of specific skills in important examinations (e.g., for end-of-year grades or college entrance). Common Sense Education (2015) noted seven best apps from its independently rated and reviewed learning resources for students to prepare for standardized tests, some of which included IXL Math Practice (PK–12) and Spelling City (K–6).

An example integration strategy for drill functions is shown in Technology Integration Example 5.1.

GUIDELINES FOR USING DRILL AND PRACTICE Consider the following four guidelines for instruction as you integrate software into your teaching:

1. **Use only after teaching the concepts**—Never use drill to introduce new topics. Use only to assess students' understanding and help them retain their grasp of familiar concepts.
2. **Set time limits**—Limit the time for drill assignments to 10 to 15 minutes per day to ensure that students will not become bored and that the drill and practice strategy will retain its effectiveness. Also, teachers should be sure that students have been introduced previously to concepts underlying the drills.
3. **Assign individually**—Take advantage of self-pacing and personalized feedback by allowing individual use rather than group use. If technology resources are limited and all students in a class will benefit from practice in a skill, make the software available at several learning centers so that all students can cycle through.
4. **Use learning stations**—If not all students need the kind of practice that a drill provides, assign the software to serve those with identified weaknesses in one or more key skills while others can be assigned other learning tasks.

Technology Integration

Example 5.1

TITLE: Using Ratios

CONTENT AREA/TOPIC: Mathematics—Ratios and Proportions

GRADE LEVELS: 4–5

ISTE STANDARDS•S: Standard 1—Empowered Learner; Standard 3—Knowledge Constructor

CCSS: Math.Content.4.NF.A.1, Math.Content.5.NF.A.1, Math.Content.6.RP.A.1

DESCRIPTION: One of the activities at Cynthia Lanius's Mathematics Lessons website provides practice in recognizing pictured examples of ratios and proportions. After students learn about ratio concepts in the classroom, the teacher has students access "All About Ratios" on the site on classroom computers. Students get a 10-minute period to practice the concepts by completing the items, and they check their own answers by clicking an on-screen tab to reveal each correct answer.

SOURCE: Based on online resources at Cynthia Lanius's Mathematics Lessons website.

Check Your Understanding 5.2

Shared Writing 5.1 Using Drill-and-Practice Functions

Characteristics of Tutorial Software

Tutorial software provides a complete instructional sequence on a topic similar to a teacher's classroom instruction. This instruction usually is expected to be a self-contained instructional unit rather than a supplement to other instruction. The software should enable students to learn the topic without any other help or materials. Unlike other types of instructional software, tutorials are complete teaching materials. Gagné, Wager, and Rojas (1981) said that good tutorial software should address all **Nine Events of Instruction**. Today's technologies (e.g., Camtasia, Youtube) make video demonstrations often referred to as *tutorials* possible. Instructional software tutorials require instructional sequencing of content and practice and feedback capabilities. However, demonstrations alone do not fulfill software tutorial functions. For example, when Sal Khan first began making content explanation videos, they were not tutorials but were archived instructional content available on the web. When Khan Academy expanded to include practice questions and feedback in addition to the instructional content videos, the software became a tutorial.

A well-designed tutorial sequence emerges from extensive research of content knowledge and how to teach the topic well. Designers must know what learning tasks the topic requires, the best sequence for students to follow, how best to explain and demonstrate essential concepts, common errors that students are likely to make, and how to provide instruction and feedback to correct those errors.

Some people confuse tutorial and drill activities for two reasons. First, drill software can provide elaborate feedback that reviewers could mistake for tutorial explanations. Even software developers sometimes claim that a package is a tutorial when it is, in fact, a drill activity with detailed feedback. Second, a good tutorial should include one or more practice sequences to check a student's understanding. Because this kind of checking is a drill and practice function, teachers reviewing tutorial software can become confused about its primary purpose.

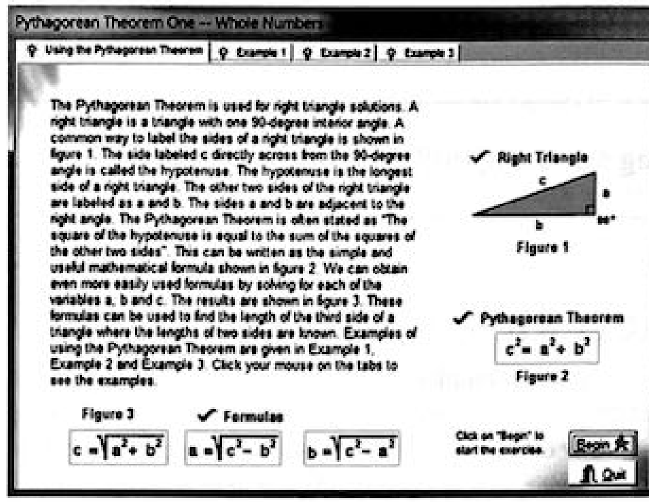
Tutorials often are categorized as linear or branching (Alessi & Trollip, 2001):

- **Linear tutorial**—A simple, linear tutorial gives the same instructional sequence of explanation, practice, and feedback to all learners regardless of differences in their performance.
- **Branching tutorial**—A more sophisticated version, a branching tutorial directs learners along alternate paths depending on how they respond to questions and whether they show mastery of certain parts of the material. Branching tutorials can range in complexity according to the number of paths they provide and how fully they diagnose the kinds of instruction a student needs. Complex tutorials can also have computer-management capabilities; teachers can place each student at an appropriate level and get progress reports as each goes through the instruction. Cognitive Tutor is a tutorial for mathematics learning in grades 9–12 that has many layers of branching within content instruction and feedback features.

Tutorials are usually geared toward learners who can read fairly well and who are older students or adults. Because tutorial instruction is expected to stand alone, it is difficult

Figure 5.6 *Trigonometry Challenge* Tutorial

SOURCE: Images reprinted by permission of ETCAI Products. http://www.etcai.com



to explain or give appropriate on-screen guidance to a nonreader. However, some tutorials aimed at younger learners have found clever ways to explain and demonstrate concepts with graphics, succinct phrases or sentences, or audio/video directions and illustrations. Figure 5.6 shows a screen from a tutorial software for trigonometry concepts, which shows linear sequences of screens that provide information on concepts such as the Pythagorean Theorem followed by practice items with feedback.

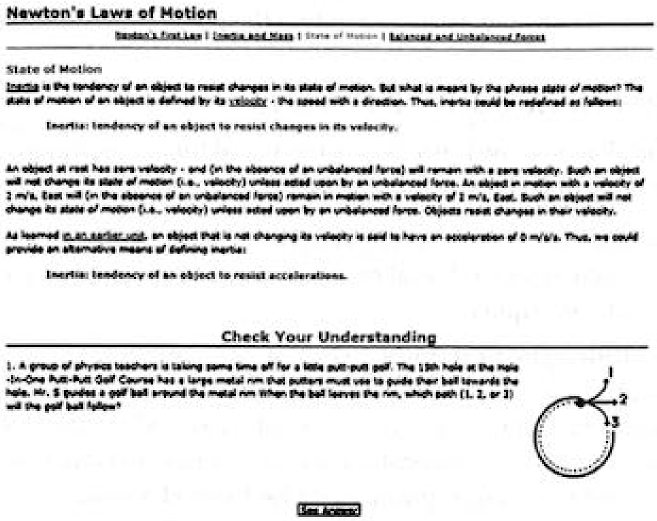
Selecting Appropriate Tutorial Software

Being a good teacher is a difficult assignment for any human, let alone a computer. However, to fulfill tutorial functions, teaching is exactly what tutorials are designed to do. In addition to meeting general criteria for good instructional software, well-designed tutorial programs should meet the following standards:

- **Extensive interactivity**—Tutorials should give frequent and thoughtful responses to questions and supply appropriate practice and feedback to guide students' learning. The most frequent criticism of tutorials is that they are "page-turners"—that is, they ask students to do very little other than read, click, or swipe.
- **Thorough user control**—Students should always be able to control the rate at which text appears on the screen. A program should not go on to the next information or activity screen until the user has signaled readiness. The program also should offer students the flexibility to review explanations, examples, or sequences of instruction; to move ahead to other instruction; and to have frequent opportunities to exit the program.
- **Appropriate pedagogy and content**—The program's structure should provide a suggested or required sequence of instruction that builds on concepts and covers the content adequately. It should provide sufficient explanation and examples in both original and remedial sequences. In sum, it should compare favorably with an expert teacher's presentation sequence for the topic.
- **Adequate answer-judging and feedback capabilities**—When possible, programs should allow students to answer in natural language and should accept all correct answers and possible variations of correct answers. Tutorials should also give appropriate corrective feedback as needed after only one or two tries rather than frustrating students by having them keep trying indefinitely to answer.
- **Appropriate graphics and/or video**—Most experts say that graphics should be used sparingly and not interfere with the purpose of the instruction. When graphics

Figure 5.7 *Laws of Motion* Tutorial

SOURCE: Images reprinted by permission The Physics Classroom. http://www.physicsclassroom.com/



are used, they should fulfill an instructional, aesthetic, or otherwise supportive function. Video-based tutorials should provide clear, uncluttered demonstrations of procedures.

- **Adequate recordkeeping**—Depending on the purpose of the tutorial, teachers might need to track student progress. If the program keeps records on student work, teachers should be able to get progress summaries quickly and easily.

Figure 5.7 shows one of a sequence of screens in a tutorial that provides instruction in physics concepts such as Newton's Laws of Motion followed by practice items. Included animated demonstrations are appropriate because they fulfill an instructional purpose.

Benefits of Tutorials

Because a tutorial includes drill and practice activities, helpful features include the same ones as for drills (immediate feedback, motivation, and time savings) plus the additional benefit of offering a self-paced instructional experience. Many successful uses of tutorials have been documented over the years. For example, see Ellington and Hardin (2008), Gobert et al. (2015), Offner and Pohlman (2010), Criswell (2011), and Wilson and Wilson (2013).

A more sophisticated tutorial software is the **intelligent tutoring system (ITS)**, a branching tutorial that adapts the sequence of instruction to the needs of each learner. For example, MathSpring Math Tutor is an ITS that prepares middle and high school learners for standardized math tests. Steenbergen-Hu and Cooper (2013) conducted a meta-analysis that examined 26 studies on ITS. They found that student use of ITS had no negative and a small positive impact on mathematical learning. They also found more positive learning outcomes when students used ITS for less than a year and with students who were not low achieving. This study could indicate that long-term reliance on ITS to teach concepts to low-achieving students is not as effective as using classroom teachers or human tutors and that most effective use could be for short-term instruction. However, many schools rely on online tutorial software, such as Programmed Logic for Automatic Teaching Operations (PLATO) for students who have failed courses and need to recover credits (Anderson, 2016). Some schools use a "hybrid" approach in which a teacher is available for consultation as students work through the online instruction.

Some tutorial software is using artificial intelligence and data mining to improve instructional sequencing of content and assessment practices. For example, Gobert et al. (2015) developed an inquiry ITS that uses data mining to assess students' science inquiry skills, specifically designing and conducting experiments. Critical thinking skills, such as inquiry, have been more difficult to teach and assess in tutorial software, but technological advancements are expanding the possibilities to do so.

Challenges Related to Tutorials

Tutorials can fulfill many much-needed instructional functions, but like drill and practice, they have their share of criticism, including:

- **Use of directed instruction**—Constructivists criticize tutorials because they deliver directed instruction rather than allowing students to generate their own knowledge through hands-on inquiry.
- **Lack of well-designed products**—The difficulty and expense of designing and developing true tutorial functions make such programs more scarce than other kinds of software. Although programs identified as "video tutorials" have emerged in recent years, most are demonstrations and explanations that are or are not well researched, and few contain practice and feedback elements.
- **Limit to one instructional approach**—True tutorial programs that have adequate feedback are difficult to design because teachers frequently disagree about what should be taught for a given topic, how to teach it most effectively, and in what order to present the learning tasks. For instance, a teacher could choose not to purchase a tutorial with a sound instructional sequence because it does not cover the topic the way he or she prefers.

Although tutorials have considerable value and are popular in military and industrial training, schools have never fully tapped their potential as teaching resources. However, recent trends to combine a tutorial approach with new technologies (online and streaming video) are bringing tutorial functions into more common use.

Integration Strategies Using Tutorials

Tutorial software can serve several classroom needs. This section describes integration strategies to meet each of these needs and offers guidelines and practical tips on how to integrate these strategies in the classroom.

Self-instructional tutorials are becoming more useful in light of new strategies such as the **flipped classroom** model and the need for credit recovery to support student advancement. These tutorials are easier to develop as the result of new technologies such as **screencasting** or video captures of actions on a computer screen, usually accompanied by narration (Stagg, Kimmins, & Pavlovski, 2013). The tutorial's unique capability of presenting an entire interactive instructional sequence can assist in several classroom situations.

STRATEGIES FOR USING TUTORIALS Three strategies for integration of tutorial software include:

- **Self-paced reviews of instruction**—These are most appropriate for students who are are slower than the rest of the class to understand concepts, need to spend additional time learning them, learn better in a self-paced mode rather than a whole-class pace, and need a review before a test. Technology Integration Example 5.2 illustrates a teacher's use of tutorials for students' review of concepts.
- **Alternative learning strategies**—Tutorial software should be used when students at advanced levels prefer to structure their own learning activities and proceed at their own pace or before meeting with a teacher for assessment and/or further work assignments.
- **Instruction when teachers are unavailable**—Tutorial software is used when students surge ahead of their class and the teacher cannot leave the rest of the class to

Technology Integration

Example 5.2

TITLE: Minds on Physics

CONTENT AREA/TOPIC: Physics—Newton's Laws of Motion

GRADE LEVELS: 9–12

ISTE STANDARDS•S: Standard 1–Empowered Learner; Standard 3–Knowledge Constructor

CCSS: ELA/Literacy RST.11-12.7, WHST.9-12.7, Mathematics MP.2, Math.Content.HSN.Q.A.1, Math.Content.HSA-CED.A.4

NSTA: HS-PS2-1, HS-PS2-4

DESCRIPTION: After presenting the topic in the classroom using usual instructional strategies, the teacher assigns the Minds on Physics (MOP) module at the Physics Classroom website as a mastery learning activity. Students complete the assigned online materials as homework or classwork and submit their encrypted "success codes" to their teacher. The teacher checks that students have completed them successfully by using a decryption page provided on the site. The teacher then assigns one or more of the online labs to illustrate application of Newton's Laws and uses the scoring rubric provided to assess students' write-up(s) of their results.

SOURCE: Based on The Physics Classroom materials developed by Tom Henderson. http://www.physicsclassroom.com/mop/join.cfm

provide advanced instruction or when no teacher is available for the comparatively few students who need a lower-demand course or to repeat a course that is not being taught.

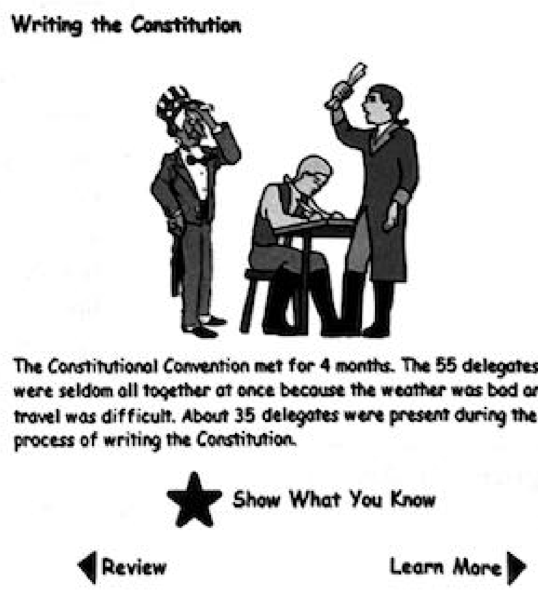
GUIDELINES FOR USING TUTORIALS When teachers decide to adopt a tutorial for use in their classroom, they should consider the following guidelines:

- **Assign to individuals**—Like drill and practice functions, tutorial functions are designed for use by individuals rather than groups of students.
- **Use at learning stations for individual assignments**—Because tutorials should be used individually, they can be used on a classroom computer or assigned to individual students on their individual devices in a one-to-one setting. Sometimes teachers working with other students send individual students to learning stations with tutorials to review previously presented material.

Figure 5.8 is a tutorial screen that provides instruction on aspects of U.S. government. Individual students can learn how the government works in sequences of instructional screens. Assessment items are provided to review concepts and check comprehension.

Figure 5.8 *The Constitution* Tutorial

SOURCE: A special thanks to the Dirksen Congressional Center for permission to use information and screenshots from Congress for Kids. Images reprinted by permission of the Dirksen Congressional Center. http://www.congressforkids.net



Check Your Understanding 5.3

Characteristics of Simulations

A **simulation** is a computerized model of a real or theoretical system designed to teach how the system works. Unlike tutorial and drill and practice activities in which the teaching structure is built into the package, learners using simulations usually must

Video Example 5.2 Honeybees' Hive Hunting Model

In this video, observe how this simulation supports learners' understanding and exploration of insects' physical activity that would be difficult for a class to observe in real life because of limited access to and danger of bees.

https://youtu.be/wwJuLxbjLg8

choose tasks to do and the order in which to do them to explore how changing input variables impacts the outcomes within the simulated system. Alessi and Trollip (2001) identified two main types of simulations: those that teach about something and those that teach how to do something; they also divide the "how-to" simulations into procedural and situational types.

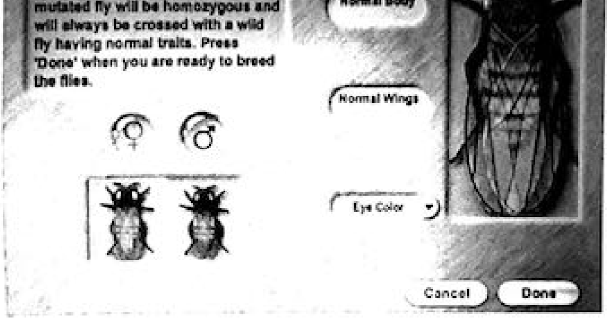
Simulations That Teach about Something

Alessi and Trollip (2001) divide "simulations that teach about something" into two categories: physical and iterative. These subcategories are based on how users interact with the simulation.

- **Physical simulations**—These allow users to manipulate things or processes represented on the screen. For example, students might see selections of chemicals with instructions on how to combine them to see the result or how various electrical circuits operate. More recent investigations of physical simulation software include the use of three-dimensional models (Kim, 2006). For example, BeeSmart (Center for Connected Learning, n.d.; Guo & Wilensky, 2014) simulates honeybees' hive-finding behavior, which represents a complex system concept. Students can observe, hypothesize, and verify conjectures. This model uses free software called NetLogo, and there are hundreds of scientific and social science models in the NetLogo Models Library (Wilensky, n.d.).
- **Iterative simulations**—These speed up or slow down processes that usually happen either so slowly or so quickly that students cannot see them unfold. For example, a simulation can show the effects of changes in demographic variables on population growth or the effects of environmental factors on ecosystems. Alessi and Trollip (2001) refer to this type as iterative because students can run it over and over again with different values and observe the results each time. Biological simulations, such as those on genetics, are popular because they help students experiment with natural processes. Genetic simulations let students pair animals with given characteristics and see the resulting offspring. SimAnimals is a forest simulation in which users try to make animals and plants flourish, a process that actually takes years.

Figure 5.9 *BioLab Fly* Simulation

SOURCE: BioLab Fly. Copyright 2011 by Bob Doltar, All Rights Reserved. http://www.biolabsoftware.com/bls/fly.html



The image in Figure 5.9 is part of a simulation that allows learners to replicate the breeding of fruit flies. Students learn genetics concepts through repeatable experiments.

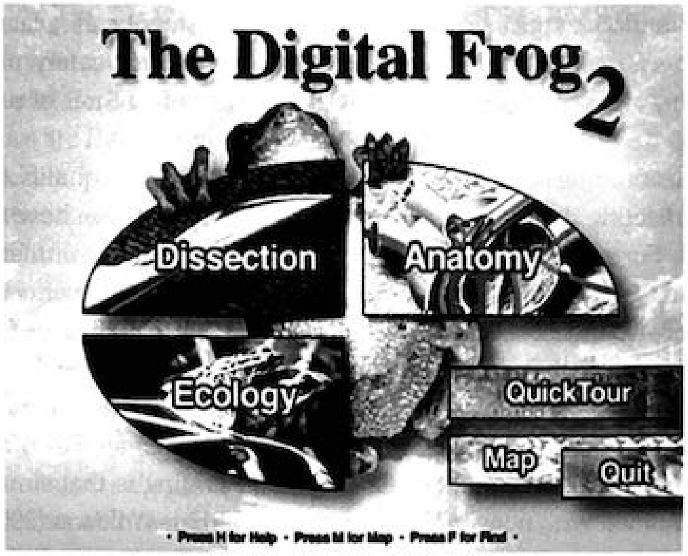
Simulations That Teach How to Do Something

Alessi and Trollip (2001) divide the "how-to" simulations into procedural and situational types. Again, these categories are based on how users are able to interact with them.

- **Procedural simulations**—These teach the appropriate sequences of steps required to perform certain procedures. They include

Figure 5.10 *Digital Frog* Simulation

Source: The Digital Frog. Reprinted by permission from Digital Frog International, Inc. http://www.digitalfrog.com



diagnostic programs in which students try to identify the sources of medical or mechanical problems and simulators in which students practice piloting an airplane or other vehicle. For example, Vfrog and Digital Frog (see Figure 5.10) simulate the procedures to dissect a virtual frog.

- **Situational simulations**—These programs give students hypothetical problem situations and ask them to react. Some simulations allow for various successful strategies, such as letting students play the stock market using Stock Market Simulation (National SMS, n.d.) or operate businesses. Others have most and least desirable options, such as choices when encountering a potentially volatile classroom situation. Motion Math: Pizza! simulation positions learners as owners of a pizzeria. Using a range of math skills, they design and set up their pizza business.

These descriptions clarify the various forms a simulation might take, but teachers need not feel they should be able to classify a given simulation into one of these categories. Simulations usually emphasize learning about the *system itself* rather than learning general problem-solving strategies.

Selecting Appropriate Simulation Software

Simulations vary so much in type and purpose that a uniform set of criteria is not possible. For some simulations, a realistic and accurate representation of a system is essential, but for others, knowing only what the screen elements represent is important. Because the screen often presents no set sequence of steps, simulations need good accompanying documentation or scaffolding—more than most software. These help the teacher learn how to use the program quickly and then to show the students how to use it. In a study of a projectile motion simulation, Tsai, Kinzer, Hung, Chen, and Hsu (2013) found that scaffolding provided before and during simulation use led to more learning than if no scaffolding was provided or was provided only during actual use. Lee and Guo (2008) believe that real systems are often preferable to simulations, but a simulation is useful when the real situation is too time consuming, dangerous, expensive, or unrealistic for a classroom presentation.

Benefits of Simulations

Simulations are more prevalent in science than any other area (Brunsell & Horejsi, 2012; D'Angelo et al., 2013; Eskrootchi & Oskrochi, 2010; Evagoroua, Nicolaoub, &

Constantinoub, 2010; Jaakkola & Nurmi, 2008; Lalley, Piotrowski, Battaglia, Brophy, & Chugh, 2010; Urban-Woldron, 2009; Wilensky, n.d.), but they are also popular in teaching social science topics (Bartels, McCown, & Wilkie, 2013; Iannou, Brown, Hannafin, & Boyer, 2009). Simulations are currently available across content areas; nearly all are now accessed online. Some updated products combine the control, safety, and interactive features of computer simulations with the visual impact of pictures of real-life devices and processes.

Research comparing simulations with "real" activities often qualify the simulation benefits by indicating that the impact of simulations depends on how and by whom they are used. For example, in a study of a simulation of environmental concepts, Eskrootchi and Oskrochi (2010) report that for simulations to be effective, a teacher must provide instructional structure for students' use. Gelbart, Brill, and Yarden (2009) found that some students learn more than others from simulations designed to teach genetics concepts. Research on teaching "systems thinking" skills to elementary school students by using environmental simulations (Evagoroua et al., 2010) reported good impact on some skills but not others. One common finding is that simulations work best when combined with nonsimulation activities. Urban-Woldron (2009) found that physics simulations were most useful as a follow-up to hands-on activities, and Jaakkola and Nurmi (2008) reported that simulations in electrical concepts had more positive benefits with elementary school children when accompanied by hands-on learning. A meta-analysis (Merchant, Goetz, Cifuentes, Keeney-Kennicutt, & Davis, 2014) also found that students learned more from simulations when they were used for practice after having already learned content concepts than when used as a stand-alone instructional approach. Merchant et al. (2014) also found increased learning when students were assessed immediately versus a delayed assessment after simulation use, meaning that the learning achieved through simulations might not lead to long-term retention.

However, other studies report unqualified positive results from simulations. Iannou et al. (2009) reported that a problem-based simulation designed to teach world issues made students more motivated to learn social studies than did text-based materials. The review of research by D'Angelo et al. (2013) found that the use of science-based simulations led to increased student science achievement in comparison to instruction

Video Example 5.3 Simulated Virtual Science Lab

In this video, a principal describes how a biology teacher uses a simulated lab to supplement physical labs.

without simulations. Furthermore, simulations with scaffolding or feedback enhancements showed even more learning effects. Lalley et al. (2010) compared the effectiveness of simulated and physical frog dissections on learning, retention, and student satisfaction and found no significant differences on any measure although they noted that students and teachers might prefer the simulation for ethical reasons.

Wieman and Perkins (2005) reported research indicating that interactive simulations in physics are frequently much more effective than seeing actual demonstrations. These researchers believe that real-life demonstrations and labs include peripheral information that is not central to the concept being learned. An expert can easily filter out the "extra" information to focus on the phenomenon of interest, but a novice does not even know what to filter out. Thus, the extra information can produce confusion and much higher cognitive loads for novice learners. Roschelle et al. (2010) used SimCalc software as part of a specific instructional approach that involves student action, teacher explanation, and teacher-led discussion to engage math learners in observing animations of motion, building mathematical functions to control animated characters' motion, observing results, and writing stories to coordinate with animations. Their research demonstrated that SimCalc contributed to learning gains in advanced mathematics in a wide variety of settings with hundreds of teachers.

Depending on the topic and the way it is used, a simulation has potential to provide one or more of the following instructional benefits:

- **Compress time**—This feature is important when students study the growth or development of living things (e.g., pairing animals to observe their offspring's characteristics) or other processes that take a long time (e.g., the movement of the sun across the sky). A simulation can make something that normally takes days, months, or longer happen in seconds so that students can cover more variations of the activity in a shorter time.
- **Slow down processes**—Conversely, a simulation can also model processes normally invisible to the human eye because they happen so quickly. For example, physical education students can study the slowed-down movement of muscles and limbs as a simulated athlete throws a ball or swings a golf club.
- **Involve students**—Simulations can capture students' attention by placing them in charge of things and asking, "What would you do?" The results of their choices can be immediate and graphic. Users can also interact with the program instead of just seeing its output.
- **Make experimentation safe**—When learning involves physical danger, simulations are the strategy of choice. This is true when students are learning to drive vehicles, handle volatile substances, or react to potentially dangerous situations. They can experiment with strategies in simulated environments that might result in personal injury to themselves or others in real life (D'Angelo et al., 2013).
- **Make the impossible possible**—Very often, teachers simply cannot give students access to the resources or situations that simulations can. For example, simulations can show students what it would be like to walk on the moon or how to react to emergencies in a nuclear power plant. They can see cells mutating or hold countrywide elections. They can even design new societies or planets and see the results of their choices. One researcher at the University of California–Davis recreated a mental health treatment ward in a virtual world and gave each of his students a simulated experience of schizophrenia in the real world (Yellowlees, 2008). As students' virtual characters walked the hallways, they were overcome by hallucinations of the floor disappearing, objects suddenly appearing, or hearing voices. More recently, journalist Anderson Cooper attempted to do his daily work while listening to an audio simulation of hallucinated voices that a schizophrenic might hear (Sedor, 2014). The NGSS standards acknowledge that data generated in simulations can play a role in allowing students to provide evidence and explanations across many content standard areas.

- **Save money and other resources**—Many school systems are finding dissections of animals on a computer screen to be much less expensive and just as instructional as using real animals. (It is also easier on the animals!) Depending on the subject, a simulated experiment can be just as effective a learning experience as an actual experiment is but at a fraction of the cost.
- **Allow repetition with variations**—Unlike real-life situations, simulations let students repeat events as many times as they wish and with unlimited variations. They can pair any number of fruitflies or make endless spaceship landings in a variety of conditions to compare the results of each set of choices.
- **Allow observation of complex processes**—Real-life events often are so complex that they are confusing—especially to those seeing them for the first time. When many things happen at once, students find it difficult to focus on the operation of individual components. Who could understand the operation of a stock market by looking at the real thing without some introduction? Simulations can isolate parts of activities and control background noise. This makes it easier for students to see what is happening when, later, all the parts come together in an actual activity.

Challenges Related to Simulations

There are some concerns as to the instructional usefulness of simulations related to the following:

- **Use of virtual lab as supplements**—Although modern simulation software makes it possible to complete simulated labs for topics in biology and chemistry and is encouraged and supported by science organizations (National Science Teachers Association (NSTA), 2016) and within standards (NGSS Lead States, 2013), both the American Chemical Society (2014) and the National Science Teachers Association (Davis, 2009; NSTA, 2007) have come out strongly against replacing hands-on, in-class labs with virtual ones, saying that simulations should be used only as supplements to, not substitutes for, regular labs. The College Board has specifications for hands-on lab requirements for advanced placement (AP) courses although virtual, interactive labs that simulate investigative practices similar to practicing scientists tend to qualify.
- **Accuracy of models**—When students see simplified versions of systems in a controlled situation, they could have inaccurate or imprecise perspectives on the systems' complexity. For example, students could think that they know all about how to react to driving situations because they have experienced simulated versions of them. Many educators feel strongly that such simulations must be followed at some point by real experiences, a position with which organizations such as the NSTA and the American Chemical Society concur. In addition, many teachers of very young children believe that learners at early stages of their cognitive development should experience things first with their five senses rather than on computer screens.
- **Instructional misuse**—Sometimes, simulations are used to teach concepts that could just as easily be demonstrated on paper, with manipulatives, or with real objects. If students can master the activities of a simulation without actually developing effective problem-solving skills, such applications can actually encourage counterproductive behaviors.

Integration Strategies Using Simulations

Simulations are considered among the most potentially powerful computer software resources; as with most software, however, their usefulness depends largely on the program's purpose and how well it fits the purpose of the lesson and student needs. Teachers are responsible for recognizing the unique instructional value of each simulation d using it to its best advantage. The following integration strategies and guidelines p teachers to make best use of simulations.

RATEGIES FOR USING SIMULATIONS The following are strategies for integrat- simulations:

- **Use in place of or as supplements to lab experiments**—Use as replacements for labs when adequate lab materials are not available or for experiments that would be unmanageable or too dangerous in person. Use as supplements to prepare students for actual labs or as follow-ups with variations of the original experiments without using consumable materials.
- **Use in place of or as supplements to role-playing**—When students either refuse to role-play in front of a class or when the topics studies are sensitive (e.g., discrimination), a level of masking students' identification within a simulation can facilitate learning.
- **Use in place of or as supplements to field trips**—Use when a desired location is not within reach of the school and a simulated experience at all or part of the location is the next best thing. Simulations also provide good introductions and follow-ups to field trips.
- **Use to introduce and/or clarify a topic**—Use to provide a nonthreatening (ungraded), get-acquainted look at a new topic and build students' initial interest in it. Some software helps students see how earlier learning relates to the topic. The Technology Integration Example 5.3 shows how a teacher used a simulation for students to be introduced to President Lincoln's decision making.
- **Use to foster exploration and process learning**—Use to emulate in-class science labs and to illustrate and provide practice in using scientific methods.
- **Use to encourage cooperation and group work**—Use to interest students in working together on a project. For example, a simulation on immigration or colonization might launch a group project in a social studies unit.

UIDELINES FOR USING SIMULATIONS Consider the following guidelines when tegrating simulations into your teaching:

- **Provide usage instruction and guidelines**—Because simulations are unstructured, students need to know how to make them work and what they are to do with them. Carefully plan how to integrate the simulation's content into your curriculum.

Technology Integration

Example 5.3

TITLE: Crisis at Fort Sumter

CONTENT AREA/TOPIC: U.S. History—Civil War period

GRADE LEVELS: 10–12

ISTE STANDARDS•S: Standard 1–Empowered Learner; Standard 3—Knowledge Constructor; Standard 4—Innovative Designer; Standard 6—Creative Communicator; Standard 7—Global Collaborator

CCSS: ELA-LITERACY.RH.9-10.5 ELA-LITERACY.RH.11-12.2

SSCS: Theme 2-Time, Continuity, and Change

DESCRIPTION: Introduce students to the historical simulation website Crisis at Fort Sumter. The simulation outlines the decisions that President Abraham Lincoln had to make concerning Fort Sumter. Tell students to choose one decision on which they disagree with the President, outline what they would have done differently, and write a defense of their alternative solution. Students post the description on the discussion board, and the class discusses the merits of the various proposals and the impact each might have had made on the situation.

SOURCE: Based on an exercise idea submitted at the MERLOT website for use with the Crisis at Fort Sumter simulation website.

- **Use with either groups or individuals**—Because they can prompt discussion and collaborative work so well, simulations usually are considered more appropriate for pairs and small groups than for individuals. However, individual use certainly is not precluded; Merchant et al. (2014) did not find any significant learning differences when students used simulations with cooperative groups or individually.

Check Your Understanding 5.4

Shared Writing 5.2 Using Simulation Functions

Characteristics of Games and Gamification

Technology-based games bridge the worlds of entertainment, gaming, and education in an attempt to deliver motivating and effective learning. Young et al. (2012) define digital learning games as those that focus on the acquisition of knowledge or higher order skills and are academically useful. More simply defined, **instructional games** are software products that combine game rules and/or competition to learning challenges. They often have the following features:

- An artificial game environment with objects, tools, and characters
- Game mechanics designed to govern rules of game play
- A content-based conflict, contest, or challenge sometimes situated within a narrative story assigned to or chosen by the user
- Feedback and assessment that marks progress toward resolution of the learning challenge sometimes including awarding of **badges**

Tobias and Fletcher (2012) note that people also refer to instructional games as *computer games*, *video games*, *serious games*, and *educational games*. **Serious games**, which have garnered more focus in the last decade, have explicit instructional content and sequencing carefully infused into a game play experience through thoughtful pedagogy (Tsekleves, Cosmas, & Aggoun, 2016). Many serious games involve collaborative features in which other learners are involved in the game play. Instructional games now come in many digital forms, such as computer-based games, console games, mobile computing games, and web-based games. Some software also incorporates **gamification** in which motivational aspects of games, such as levels of play and badges, are built into nongame activities.

Although many writers and researchers tend to conflate the use of games and simulations, games are considered a separate software function because they involve structured rules, an explicit goal to win (or lose), mechanisms to compare player performance (such as a leader board), and entertaining formats (Young et al., 2012). These elements generate a set of mental and emotional expectations in students that make game-based instructional activities different from nongame ones. Students expect a fun and entertaining activity because of the challenge of the competition and the potential for winning.

Devaney (2013) notes that although interest in video games seems to be growing, their adoption in schools has been slow. One reason is that effective educational video games take a long time to develop, few good models are available for teachers to see

Figure 5.11 *Lure of the Labyrinth* Game

SOURCE: Lure of the Labyrinth. Reprinted by permission from Maryland Public Television. Copyright © Maryland Public Television. http://www.labyrinth.thinkport.org

and try out, and there is no central repository for them (Tsekleves et al., 2016). However, video games of various kinds are emerging for elementary school to high school levels, all designed to immerse young people in alternate worlds for the purpose of learning various content and skills. Devaney cites examples such as Minecraft, SimCity, Surge EpiGame, and Satisfraction. Other available products include those from the Dig-It! Games company, which let students learn about ancient civilizations. Finally, she notes that Educade is a free site that links standards-aligned lesson plans with various interactive activities including video games. Young et al. (2012) mention 3-D virtual world games such as Quest Atlantis and Alien Rescue and games for subject areas such as Physicus and Virtual Cell (science), DimensionM (math), My Spanish Coach (Spanish), and Civilization and Oregon Trail (history). In the Lure of the Labyrinth game (Figure 5.11), students apply pre-algebra skills to locate a missing pet in the labyrinth.

It is important to note that our definition of instructional games excludes some games because learning about content topics is not an explicit goal or built into the game, such as the popular game Minecraft, which is a multiplayer 3-D sandbox game that allows players to explore landscapes and create structures with blocks. Because teachers or learners must bring instructional context and content to Minecraft, we define it as a web-based creativity tool, and it will be described later in the book. However, the same company recently released Minecraft: Education Edition, which has some pre-designed lesson activities and supports a secure classroom environment with a role for teacher support (Minecraft in the classroom, 2017).

Many people now view video games as important learning resources when they are designed with compelling educational content and immersive and interactive features. To illustrate this, in 2014, the White House sponsored an educational **game jam** in which game developers, learning scientists, teachers, and students came together to conceptualize and build new games for learning.

Video Example 5.4 White House Education Game Jam: Who Killed Lincoln?

Watch this video trailer, which introduces an educational game created at an educational game jam sponsored by the White House in 2014.

https://youtu.be/AY9vUost7KU

Selecting Appropriate Instructional Games

Several researchers have worked to develop criteria, categorizations, and rubrics for assessing the quality of games or serious games for education (Borji & Khaldi, 2014; De Lope & Medina Medina, 2016; Hong, Cheng, Hwang, Lee, & Chang, 2009). With these assessment and categorization approaches in mind, we developed the following criteria that teachers can use to choose effective instructional games:

- **Game development**—Who developed a game can give clues to its longevity and the developer's commitment to learning. Does the developer or company have a long history of creating educational products? Who is involved in development? Pay special attention to whether developers involve content and teaching experts, such as professors or teachers in the design process. Has any independent research (i.e., not sponsored by the company) been conducted about the game?
- **Curricular value**—Teachers should examine instructional games carefully for their educational value. Is learning the curricular content central to the game's objectives? By playing the game, will students learn high-quality content and be able to apply it in nongame situations? Are there timely feedback and scaffolding mechanisms to support the player's learning? Do the content and its representations match the learning level of target students? Is the content topic meeting a curricular need that other learning resources cannot?
- **Pedagogical framework**—Determine whether the game is built on directed or constructivist instructional theories. Are the learning objectives explicit and observable through modeling? Does the game acknowledge learners' existing knowledge on the topic? Does it assist learners in moving from simple to more complex ideas? Does it adapt based on learner actions (e.g., have branching)?
- **Assessment capabilities**—Consider how assessment is built into the game and how it helps the learner and the teacher. What types of assessment, such as formative, summative, individual, or group, are involved? Do built-in assessments provide indicators of learning progress to the player ? Does the game track information about the player/learner and provide that to teachers? Is tutor scaffolding available as in the form of a pre-programmed pedagogical agent or an in-game character that the teacher assumes?
- **Social, societal, and cultural considerations**—Games may be inappropriate for children if they are not designed with a respectful outlook. For instance, games that call for violence or combat require careful screening, not only to prevent students' modeling of this behavior but also because girls often perceive the attraction of these activities differently than boys do. In addition, games could present girls and various ethnic and cultural groups in stereotypical roles. Teachers should choose games that do not perpetuate stereotypes but highlight positive messages (e.g., peace and friendship) rather than unnecessary violence (e.g., aggression).
- **Playfulness and motivation**—The most popular games include elements of adventure and uncertainty as well as levels of complexity matched to learners' abilities. Does the game elicit different emotional states within the player? Does the game use applicable, age-appropriate graphics, sounds, and scenarios to immerse the learner within the game? Do interactive elements stimulate the learner manually and intellectually? Is the player motivated to pursue challenges and achieve them? Does the game have replayability? Does it offer multiplayer capability?
- **Technical considerations**—Teachers should ensure that the game is technically optimal for their computing setting. What are the computing system requirements for the game, such as specific operating systems, required plugins, special hardware or peripherals, or software installation?

Video Example 5.5 Instructional Game Software in Algebra

In this video, a principal describes how game software can help teach complex algebra skills. As he describes why students liked a particular game, listen for some of the criteria he used to select it.

- **Physical dexterity**—Teachers should ensure that students will be motivated rather than frustrated by the activities. Unless the object of the game is to learn physical dexterity (e.g., for students with physical challenges or in physical education disciplines), the game's focus should be on learning content-area knowledge so all students should be able to manage the level of physical dexterity.

Benefits of Instructional Games

A study across 23 countries by Biagi and Loi (2013) found a significant positive relationship between high use of gaming activities and content-area test scores in languages (11 countries), mathematics (15 countries), and science (13 countries) among 15-year-olds. Yet, definitive and conclusive empirical evidence that instructional game software contributes to educational effectiveness does not yet exist because of the relatively recent emergence of this field and the need for more rigorous research studies. In a literature review of 34 studies examining the use of serious games in education, Tsekleves et al. (2016) identified prominent benefits for learning from the inclusion of

- Problem-based learning (PBL)
- Collaborative learning
- Realistic and immersive environments

They also identified learning benefits from the game design mechanics, including:

- Motivation and competition
- Interactivity and feedback
- Achievement and rewards
- Playfulness

Successful uses of games have been reported in many content areas, including mathematics (Bai, Pan, Hirumi, & Kebritchi, 2012) and language learning, history, and physical education (Young et al., 2012). Merchant et al. (2014) found that student learning in

Figure 5.12 *Crystals of Kaydor* Game
SOURCE: Crystals of Kaydor Image. © Learning Games Network- LGN. http://www.gameslearningsociety.org

games was retained beyond a short time frame. Yet, some studies do not reveal a learning advantage from games; one experimental study found no evidence that students' learning of biology in a biotechnology-based computer game exceeded that of students learning the content in a nongame instructional approach (Sadler, Romine, Menon, Ferdig, & Annetta, 2015). Even when studies show no difference in learning impact, the instruction through games could be more appealing to students by centering on their desire to compete and play. Some educators and observers feel strongly that video games hold special promises for improving classroom teaching strategies and making learning more engaging and motivational (Ash, 2011a; Corbett, 2010), especially in serious games that tend to be problem-based and collaborative or have abilities to foster what Herold (2013) calls "noncognitive skills" or abilities such as empathy, attention, and tenacity. For example, in Crystals of Kaydor, the game's goal is to foster empathy and ability to pay attention while students must successfully interact with aliens (Figure 5.12).

Challenges Related to Instructional Games

Some teachers believe that any time they can sneak in learning under the guise of a game, it is altogether a good thing. However, games have been criticized for several reasons:

- **Focus on learning versus having fun**—Although students obviously find many computer games exciting and stimulating, it is sometimes difficult to pinpoint their educational value. Some schools forbid any use of games because they believe that games convince students that they are escaping from learning, thus drawing attention away from the intrinsic value and motivation of learning. Some critics believe that winning the game becomes a student's primary focus and that the instructional purpose is lost in the pursuit of this goal. These perceived problems tend to be promulgated by adopting games that are branded as educational but do not support learning, which some refer to as "edutainment" (Van Eck, 2009). The issue of fun versus learning was one of three prominent barriers to the use of serious games in education based on a literature review by Tsekleves et al. (2016).
- **Correspondence between game goals and learning objectives**—The goals of some games do not correspond with learning objectives. This can create a problem in which the player can work to win the game regardless of the learning challenge (Tsekleves et al., 2016; Young et al., 2012).

- **Transfer of learning**—Some teachers observe students having difficulty transferring their learning from games to later nongame situations. Tsekleves et al. (2016) identified this lack of short- or long-term transfer as a top barrier to the use of serious games and explain that lack of transfer can occur if the game lacks correspondence between rules and learning objectives, if the game does not involve the player in cognitive processes similar to those that she or he will apply in a nongame situation that requires application of the knowledge, or if the game pedagogy does not support gradual learning from simple to complex concepts.
- **Alignment with the curriculum and teaching practices**—Research suggests that the learning objectives of serious games are not presented in ways that assist in meeting school curricular goals and pedagogy (Young et al., 2012). This was another top barrier in Tsekleves et al.'s (2016) literature review. Tsekleves et al. suggest that educators choose to use serious games when they align well with curricular goals and can be used in concert with face-to-face pedagogy and other learning resources, such as books or laboratory activities. Many serious games can conflict with directed instructional approaches, so teachers and schools could need to modify curriculum and instruction to teach with serious games.
- **Classroom barriers**—In a review of research on video games for instruction, Rice (2007) found barriers to widespread classroom implementation. These included negative teacher perceptions toward video games, lack of adequate computing hardware required to run advanced video games, short class periods that hindered long-term engagement in complex games, real-world affordances, and alignment with state standards. More recent research found that although most teachers would like to use games for learning, substantial barriers include cost, access to technology resources, schools' emphasis on standardized test results (Millstone, 2012), and difficulty gaining approval for use of games as instructional materials (Sansing, 2014).

Integration Strategies and Guidelines for Instructional Games

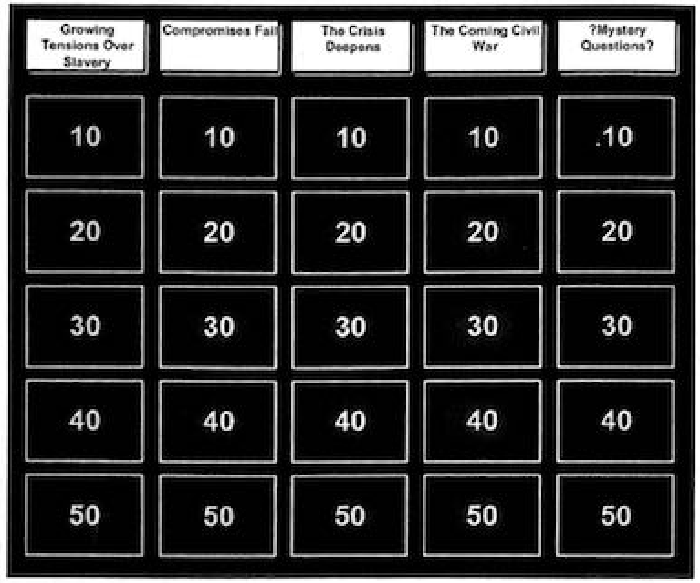
Instructional games can serve several classroom needs. We offer the following strategies and guidelines.

STRATEGIES FOR USING INSTRUCTIONAL GAMES Strategies for integration of instructional games include the following:

- **To encourage lifelong playing**—Sansing (2014) encourages teachers to reintroduce playful learning to children's learning lives in the classroom by using games that reflect meaningful challenges in content areas or topics that are difficult to teach but to avoid games that spotlight trivial tasks.
- **To encourage problem- or inquiry-based learning**—Many serious games use problem-based learning as their pedagogical framework. Learners, the players, become characters who must find solutions to content-connected challenges. These games support constructivist-based instruction.
- **To take the place of worksheets and exercises**—As with drill and practice software, teachers can use games to help students acquire automatic recall of prerequisite skills. These games support directed instruction. *Jeopardy!*-style games facilitate this strategy when teachers generate content review items for a *Jeopardy!* game (see Figure 5.13).
- **To teach "noncognitive skills"**—Some newer games are designed especially to teach skills such as attention and perseverance, which are useful across content areas.
- **To teach cooperative group work skills**—Like simulations, many instructional games serve as the basis for or introduction to group work. In addition, some

Figure 5.13 *Jeopardy Review* Game

Source: Jeopardy Review Generator reprinted by permission of SuperTeacher Tools. Courtesy of Jason Kries. Copyright © Jason Kries. www.superteachertools.com



games can be played collaboratively over the Internet (e.g., via an Internet-enabled game console or in web-based games). A game's competitive qualities can present opportunities for competition among groups.

GUIDELINES FOR USING INSTRUCTIONAL GAMES Consider the following guidelines when you choose to integrate a game into your teaching.

- **Align and integrate serious games with curriculum**—Tsekleves et al. (2016) advise that serious games not be used as an isolated add-on but that they be chosen for their connection to curriculum and be carefully incorporated into broader instructional activities occurring in the classroom. The Technology Integration Example 5.4 exemplifies using an online game blended with other in-class learning activities.

Technology Integration

Example 5.4

TITLE: Do I Have a Right?

CONTENT AREA/TOPIC: Civics—The Bill of Rights

GRADE LEVELS: 8–10

ISTE STANDARDS•S: Standard 1—Empowered Learner

CCSS: ELA-LITERACY.RH.6-8.3 ELA-LITERACY.RH.9-10.6, ELA-LITERACY.RH.11-12.5

NCSS: Theme 6-Power, Authority, and Governance

DESCRIPTION: Using a packet of materials on the Bill of Rights, the teacher reads a scenario in which the world has been destroyed and a "Pamphlet of Protections" must be created to define the rights people will have. Students identify their "top-ten" rights from a checklist, and the teacher polls the class to see which were selected. The teacher compares this task to the challenge that the framers of the Constitution faced and reviews each of the Bill of Rights the students created. After review and discussion, students apply what they learned with "Do I Have a Right?" online game software. They become lawyers who must decide whether potential clients "have a right." The more clients they serve and the more cases they win, the faster the law firm grows.

SOURCE: Based on ideas from lesson plan at the iCivics free lesson plans website.

- **Identify and assess your own learning objectives**—Not all games have sufficient assessment tools to approximate a learner's content learning. Thus, Tsekleves et al. (2016) suggest that after careful review of games and choosing one to adopt, teachers develop learning objectives and assessments external to the game for learners who will use the game for learning, which corresponds with Step 4 of the TTIPP model.
- **Involve all students**—Make sure that all students—girls and boys, English language learners, students with varying achievement levels or with disabilities—are participating and that all students have a meaningful role in game playing. The single most common use of games is to reward good work, but this approach often does not involve all students, limits games as a behaviorist tool to accomplish other tasks, and undermines the power of games to be teaching software. Therefore, we do not recommend using games as rewards.
- **Assign individual game playing**—A meta-analysis (Merchant et al., 2014) found that students who play games learn better when they play individually versus collaboratively.
- **Emphasize the content-area skills**—Before students begin playing, make sure that they know the relationship between game rules and content-area (e.g., math) objectives. Students should recognize the knowledge and skills that they will be developing in the game to use (transfer) in other curricular work.

Check Your Understanding 5.5

Characteristics of Problem-Solving Software

Although many instructional software programs often include problem-based learning and problem-solving skills, **problem-solving software** is designed especially for this purpose. Such software can focus on fostering component skills in general problem solving and provide opportunities to practice solving various kinds of content-area problems. According to Mayes (1992), problem solving is cognitive processing directed at achieving a goal when the solution is not obvious. One way to think about problem solving is through three of its most important components:

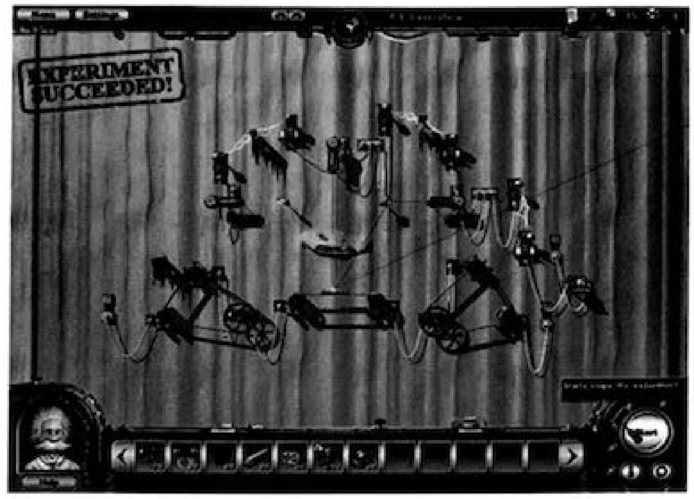
- Recognition of a goal (an opportunity for solving a problem)
- A process (a sequence of physical activities or operations)
- Mental activity (cognitive operations to pursue a solution)

Although most problem-solving literature focuses on skills related to mathematical problems, research on the topic covers a wide variety of desired component behaviors. The literature mentions varied subskills such as metacognition, observing, recalling information, sequencing, analyzing, finding and organizing information, inferring, predicting outcomes, making analogies, and formulating ideas. Although there are many opinions about the proper role of instructional software in fostering these abilities, there seem to be two main approaches to integrating these problem solving skills in software:

- **Content-area problem-solving skills**—Some problem-solving software focuses on teaching content-area problem solving, primarily in mathematics and science. Seo and Bryant (2012) report using a program called Math Explorer to help students learn a systematic approach to solving mathematics word problems. Other software

Figure 5.14 *Crazy Machines* for Problem Solving

Source: Crazy Machines 2: The Wacky Contraptions Game © FAKT. Published by Viva Media. Reprinted by permission. http://www.viva-media.com



are what might be called problem-solving "environments." These complex, multifaceted packages offer a variety of tools that allow students to create solutions to complex problems presented by a scenario. One of these is Alien Rescue (Liu, Rosenblum, Horton, & Kang, 2014), which helps students solve problems in science environments. Other programs designed to practice solving specific kinds of math or science problems include Crazy Machines (see Figure 5.14) in which learners recognize math situations in word problems and use graphic organizers to understand and plan solutions.

- **Content-free problem-solving skills**—Some educators feel that general problem-solving ability can be taught directly by specific instruction and practice in its component strategies and subskills (e.g., recalling facts, breaking a problem into a sequence of steps, predicting outcomes). Others suggest placing students in problem-solving environments and, with some coaching and guidance, letting them develop their own heuristics for attacking and solving problems. In Memory Challenge (see Figure 5.15), learners engage in practice exercises to improve visual memory skills required for reading and math activities.

Figure 5.15 *Memory Challenge* for Problem Solving

Source: Memory Challenge. Reprinted by permission of the Critical Thinking Co. http://www.criticalthinking.com

The purposes of the two approaches overlap somewhat, but the first is directed more toward motivating students to attack problems and recognizing problem solving as an integral part of everyday life whereas the second aims to help students practice component skills in specific kinds of problem solving.

Selecting Appropriate Problem-Solving Software

The qualities to look for in problem-solving software depend on its purpose. In general, problem formats should be interesting and challenging, and software should have a clear link to developing a specific problem-solving ability. Software documentation should state clearly which specific skills students will learn and how the software fosters them.

Benefits of Problem-Solving Software

Research and practice indicate that problem-solving software can help students in at least three different areas:

- **Visualization in mathematics problem solving**—Research into mathematical problem-solving skills tends to support the hypothesis that software programs that rely on graphical displays, such as Geometer's Sketchpad, help students visualize abstract concepts and, thus, better understand how to solve problems that call for those concepts. For example, Salden, Koedinger, Renkl, Alevne, and McLaren (2010) used the program Cognitive Tutor that facilitates student problem solving with scaffolding, including identifying problem goals and receiving feedback and hints.
- **Interest and motivation**—Students are more likely to practice solving problems in activities that they find interesting and motivating. Some educators believe that students will become more active, spontaneous problem solvers if they experience success in their initial problem-solving efforts. For example, in reporting research results for problem-based environments, Pedersen (2003) found that these environments can be highly motivational although Samsonov, Pedersen, and Hill (2006) found that they were more motivational for higher achieving students.
- **Inert knowledge**—Content-area problem-solving environments, such as Thinkport's Flossville Town Park and Windjammer Environmental Center, can make knowledge and skills more meaningful to students because they illustrate how and where information applies to actual problems. Students learn both the knowledge and its application at the same time. Also, they gain opportunities to discover concepts themselves, which they frequently find more motivating than simply being told concepts.

Challenges Related to Problem-Solving Software

Problem-solving software packages are among the most popular of all software functions; however, the following issues are still of concern to educators:

- **Multiple, imprecise problem-solving labels**—Software packages use many terms to describe problem solving, and their exact meanings are not always clear. Terms that appear in software catalogs as synonyms for problem solving include *thinking skills, critical thinking, higher level thinking, higher-order cognitive outcomes, reasoning, use of logic,* and *decision making*. Because of this diversity in language, teachers must identify the skills that a software package addresses by looking at its activities. For example, assessing a software package that claims to teach inference skills would involve seeing how it defines *inference* by examining the tasks it presents, which can range from determining the next number in a sequence to using visual clues to predict a pattern.

- **Software claims versus research-based effectiveness**—It would be difficult to find a software catalog that did not claim that its products foster problem solving, yet few publishers of such software packages have data to support their claims. When students play a game that requires skills related to problem solving, they do not necessarily learn them. They could enjoy the game thoroughly—and even be successful at it—without learning any of the intended skills. Teachers might have to use problem-solving software themselves to confirm that it achieves the desired and intended results.
- **Lack of skill transfer**—Although some educators feel that general problem-solving skills, such as inference and pattern recognition, will transfer to content-area skills, scant evidence supports this view. In general, research tends to show that skill in one kind of problem solving will transfer primarily to similar kinds of problems that use the same solution strategies. Researchers have identified nothing as "general thinking skills" except in relation to intelligence quotient (IQ) variables.

For example, the Federal Trade Commission (FTC) investigated two "brain-training" software producers for making inappropriate claims regarding their products' abilities to improve memory, focus, and school work among other benefits (Sparks, 2016) and levied settlement judgments against them. Many hoped such brain-training software products would yield improved cognitive abilities, but research indicates that they improve short-term working memory but do not yield sustained or transferable cognitive abilities (Max Planck, 2014; Melby-Lervåg, Redick, & Hulme, 2016). However, a recent randomized controlled study with adolescents diagnosed with attention-deficit/hyperactivity disorder (ADHD) who used the Cogmed RoboMemo working memory software had increased psychomotor speed and reading and mathematics. Eight months later, the improved reading scores were maintained (Egeland, Aarlien, & Saunes, 2013). Researchers and the FTC's investigations emphasize the need to examine whether products' promises for transferability are based on verifiable, independent research.

Integration Strategies Using Problem-Solving Software

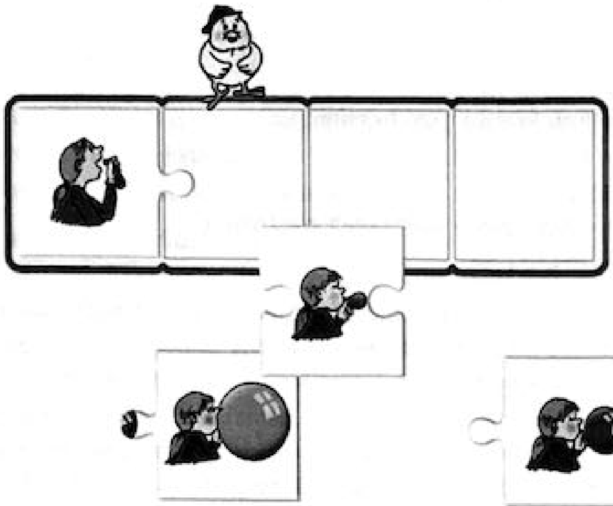
Problem-solving software can serve several classroom needs. This section describes integration strategies to meet each of these needs and offers guidelines and practical tips on how to integrate these strategies in the classroom.

STRATEGIES FOR USING PROBLEM-SOLVING SOFTWARE Strategies for integrating problem-solving software include the following:

- **To teach component skills in problem-solving strategies**—Many problem-solving packages provide good, hands-on experience using one or more of the skills required to use a problem-solving approach. These include understanding the problem, identifying and following a logical sequence, identifying relevant information to solve problems, remembering relevant information, not jumping to conclusions too quickly, and evaluating the process and outcomes. For example, in Sequences (see Figure 5.16), learners use text-free practice in sequencing by placing events in a correct order.
- **To provide support in solving problems**—Most problem-solving software packages are specifically designed to scaffold students as they practice solving complex problems. For example, Geometer's Sketchpad helps students draw objects and investigate their mathematical properties.
- **To encourage group problem solving**—Some software provides environments that lend themselves to solving problems in small groups. For example, software at the Thinkport site provides opportunities for collaborative problem solving.

Figure 5.16 *Sequences* for Problem Solving

Source: Sequences is a Sherston Software product. Reprinted by permission. http://www.sherstonamerica.com



- **To provide practice in solving problems**—Some packages provide opportunities to practice applying problem solving in ways to make it more likely that the skills will transfer to real-life situations.

GUIDELINES FOR USING PROBLEM-SOLVING SOFTWARE The following steps can help to integrate problem-solving software for teaching:

- **Identify focal problem-solving skills or general capabilities**—Ensure that the software will help you build or foster skills in (1) solving one or more kinds of content-area problems (e.g., building algebra equations), (2) using a scientific approach to problem solving (i.e., identifying the problem, posing hypotheses, planning a systematic approach), and (3) identifying the components of problem solving, such as following a sequence of steps or recalling facts.

Video Example 5.6 Using Geometer's Sketchpad for Inductive Reasoning

Watch this video and identify how Geometer's Sketchpad is used for problem-solving functions.

Technology Integration

Example 5.5

TITLE: Wait for a Date: Calculating Probability with Geometer's Sketchpad

CONTENT AREA/TOPIC: Mathematics—Precalculus

GRADE LEVELS: 8–10

ISTE STANDARDS•S: Standard 1—Empowered Learner; Standard 5—Computational Thinker

CCSS: MATH.CONTENT.HSS.CP.A.1, CCSS.MATH.PRACTICE.MP5

DESCRIPTION: The teacher presents students with this scenario: "You and a friend arrange for a lunch date next week between 12:00 and 1:00 p.m. However, neither of you remembers the exact meeting time. Each of you arrives at a random time between 12:00 and 1:00 p.m. and waits exactly 10 minutes, then leaves if the other person has not arrived. Under these circumstances, what is the probability that you two will meet?" Students use a premade Sketchpad model (available at the Geometer's Sketchpad site) to gather sample data and, by viewing data as points in a plane, they uncover a geometric pattern that allows them to compute a precise probability. Sketchpad also supports activities called "black box tasks" for students with more sophisticated knowledge of the software. In these activities, students use the software to re-create a given figure or deduce underlying properties that two or more objects have in common.

SOURCE: Based on lesson plan idea at the Geometer's Sketchpad website.

- **Ensure that software fits in the teaching sequence**—For directed instruction, you can use the software to introduce a skill and gain attention or to provide practice activity after demonstrating problem solving, or both. For constructivist instruction, you might need to allow students sufficient time to explore and interact with the software but provide some structure in the form of directions, goals, a work schedule, and organized times for sharing and discussing results.
- **Build in transfer activities**—Make students aware of the skills they are using in the software program; one approach is to have students talk and reflect about the methods they use in the software and its relationship with other learning activities. The teacher might need to point out the relationship between software activities and other kinds of problem solving.

The Technology Integration Example 5.5 exemplifies a problem-based lesson with Geometer's Sketchpad that aligns with constructivist instructional approach.

✓ **Check Your Understanding 5.6**

Characteristics of Personalized Learning Systems

A **personalized learning system (PLS)** is a multifunction computer-based management program that (1) assesses individual student learning needs using complex algorithms and collections of data across many students and (2) provides a customized instructional experience matched to each student's needs. PLS evolved from the **integrated learning system (ILS)**, a product introduced in the early 1970s that provided assessment, instruction, and reports on student progress. Recent software developments have combined adaptive testing, content, and databases of instructional strategies to enable systems that can personalize a given student's path through a topic based on his or her performance. The accountability emphasis within education created a demand for products such as PLS that can help teachers assess needs and assign instructional solutions quickly and efficiently. Many software companies now claim that their products support personalized learning; some claims are simply part of firms' marketing strategies (Molnar, 2016a). How PLS programs work varies widely, but six characteristics are currently common to them:

- **Student centered**—The student is the center of the PLS and should have a learning profile that helps both the student and teacher contribute to learning and instructional decisions. Students should have some level of choice in their learning pathway.
- **Adaptive assessment**—Adaptive assessment strategies are the heart of the PLS approach. Students use digital devices (e.g., computers, tablets) to take tests in a given topic, and teachers get results that enable them to decide what to do next to support learning or enable the software's next instructional step for the learner.
- **Curriculum matched to Common Core State Standards or state standards**—Each report that the PLS generates ties to performance on CCSS or state standards. Software uses various ways, such as color coding to show strengths and to reveal students' mastery of standards.
- **Competency or mastery based**—Many PLS programs are designed for students to show proficiency, mastery, or competency and then move to more advanced topics. This approach supports learners working in different timeframes with the necessary amount of time needed to master the content.
- **Multiple learning media**—School leaders prefer PLS products because they can vary according to the type of instruction offered, multimedia materials, instructional modes, and feedback (Data Dive, 2016). Many PLSs integrate all the functions described in this chapter: drill and practice, tutorials, simulations, games and gamification, and problem-based learning in order to meet the multifunction goal of personalizing learning.
- **Data reports on individual and group progress**—PLSs products can give summary reports on progress according to standards for any designated student, which helps teachers know which student has or has not mastered standards and to make decisions on the next instructional steps for the topic. Data-based information about student progress should be available to students and their parents.

Examples of PLSs include Amplify, Knewton, Lexia Learning Core5, Renaissance Learning, Edgenuity, Read 180, and Scientific Learning Reading Assistant.

Selecting Appropriate PLSs

One way to ensure the appropriate use of PLSs is to have a careful, well-planned initial review and selection process that involves both teachers and school administrators. Because PLSs provide a new way of thinking about assessment and curriculum, the selection process should consider the ease of implementation, the amount of company support and professional development offered, and the degree to which the PLS approach supports district and state priorities. According to Molnar (2016a), the following aspects of the PLS product should be considered:

Video Example 5.7 What is Personalized Learning?

This video shows how a PLS can assess a student's learning needs and provide a customized instructional experience for the student.

https://youtu.be/6oLNLCO0vfI

- **Student agency**—Determine whether students have some level of control over setting learning goals and whether learning and activity feedback is understandable and contributes to forward progress.
- **Content depth and quantity**—Examine the content and lessons. Is the content aligned with content standards that you follow? Will the product meet the needs of both beginning learners and advanced learners in the content area? Is there sufficient instructional content to support learning across an academic year?
- **Meaningful data**—PLS products track and mine a significant amount of data, but consider carefully whether the data are meaningful. Data dashboards for students and teachers must contain easily understandable information that can be directly transformed into actions within the system. For example, a teacher might see alerts about a student's progress and then be able to choose instructional interventions available within the software to assign to the student.
- **Aligned and validated assessments**—PLSs function heavily on their embedded assessments because they should gauge the needs of the learner and move him or her to meaningful content. But consider carefully how these assessments align with the standards and the standardized tests your students will ultimately take. These products assessments might not have been externally validated.
- **Evidence of impact**—As with other instructional products, ask whether independent research studies have examined the PLSs' impact on student achievement and other student outcomes such as motivation and engagement.
- **Integration approach**—Consider whether the PLS has been designed for particular instructional approaches, such as directed or constructivist approaches, and whether they match the expected or preferred instruction in your school and classrooms.

Application Exercise 5.2 Key Functions of Instructional Software

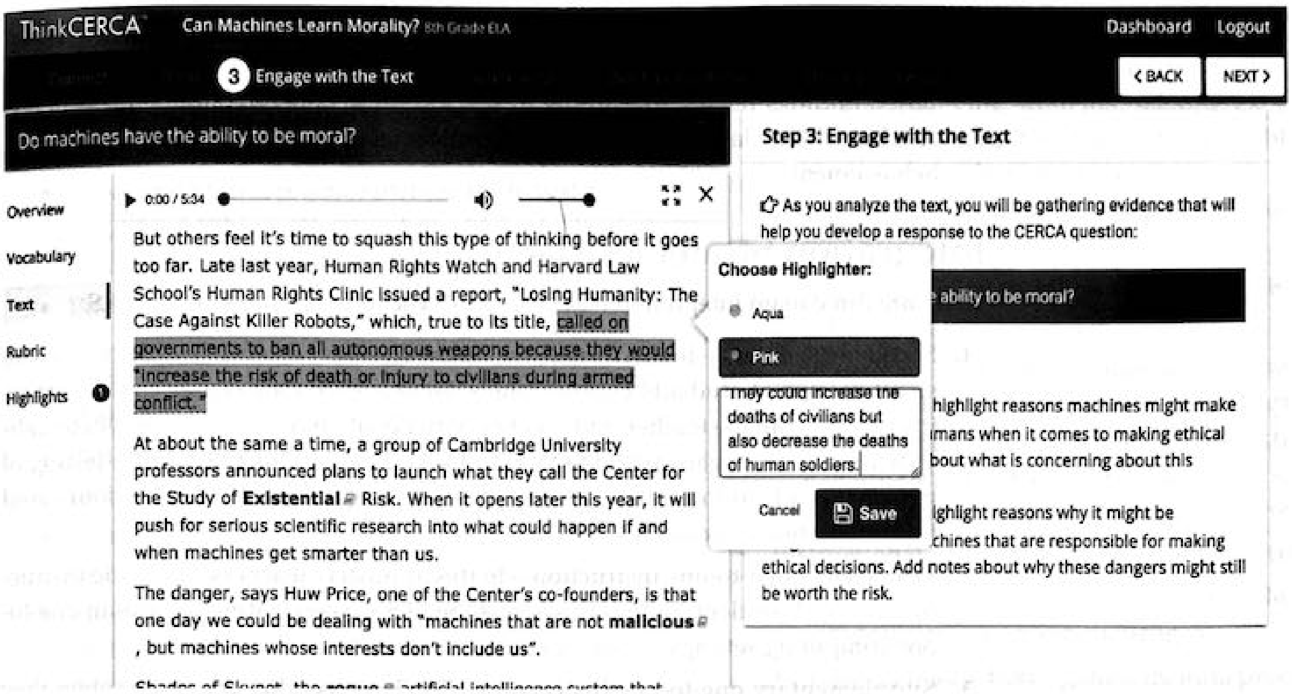
Benefits of PLSs

PLSs are becoming popular in school districts because of their ability to use a wealth of available data on student performance to produce personal learning solutions related to standards and their ability to provide achievement information for students, teachers, classes, and schools. This approach to diagnosing and selecting materials targeted to learning needs frees teacher time for guiding student learning. A well-designed PLS can provide the following benefits:

- Accurate assessments based on formative, summative, and longitudinal test data across groups
- Progress linked to curriculum and standards
- Personalized instructional prescriptions matched to student needs
- Summary progress data that help meet teacher/district accountability requirements

Some large-scale studies using controlled experimental methods show no significant differences in achievement between classrooms using these systems and those using traditional materials and methods (U.S. Department of Education, Institute of Education Sciences 2010; Wijekumar, Hitchcock, Turner, Lei, & Peck, 2009). But promising classroom-based research trials of PLSs are being conducted in Chicago by LEAP Innovations (2016). In its first-year study, the company identified promising edtech products that personalized literacy learning for students in K–8 grades, matched them to schools' and students' needs in each, and evaluated their impact across a year. The

Figure 5.17 *ThinkCERCA*



pilot results showed statistically significant growth in literacy learning among the students using the products as compared to a control group. The researchers argue these results indicate the products can contribute to reducing the achievement gap for students living in poverty and students in racial minorities. Two of the products that led to achievement gains were Lexia Reading Core5 and ThinkCERCA. Figure 5.17 shows a student working within the ThinkCERCA software. The student is engaged with the text in an eighth grade lesson, "Can machines learn morality?"

Arnett (2016) suggests that technological developments such as PLSs are key contributors to enhancing teachers' work, not threatening it. He describes how technological innovations can (1) assist less-effective teachers when the product contains high-quality content and instruction, (2) assist all teachers in reaching students' wide-ranging learning needs, and (3) support coverage of nonacademic skills that contribute to overall success.

Challenges Related to PLSs

There are proponents and critics of PLS. Proponents say that technology has enabled personalized instruction to be delivered to every student at an affordable cost, and critics say that teachers already match a student's strengths and weaknesses to appropriate strategies and materials (Fletcher, 2013). For those implementing a PLS into their classroom or school, several challenges exist:

- **Obtaining support from school community**—Teachers need support from their principals and community, including students and parents, for any PLS adoption to succeed. Teachers must be able to provide extensive information on what is working or not working, and principals need to hear and respond to needs.
- **Recognizing that PLS is part of a broad strategy, not a panacea**—LEAP Innovations (2016) suggests that schools situate the PLS as one part of a broader set of strategies working toward personalized learning. The researchers identified teaching and learning practices within which the PLS activity is situated as being crucially important. The researchers recommend practices that allow the learner to lead the focus, direction, and pace of learning.

- **Obtaining extractable data**—Data amassed within the PLSs is not always extractable and available to teachers or to schools in usable ways.
- **Identifying research-based impact**—Although many products have been carefully designed with rich content that aligns with standards and honors the research-based findings related to learning in specific subject areas, continued research is needed within classrooms to gauge the outcomes and impact of PLSs on learning achievement.

Integration Strategies Using PLSs

There are three main integration ways in which teachers can implement a PLS.

1. **Stations or centers**—In this format, the teacher creates centers of instruction where small groups of students cycle through using a PLS. One center could involve instruction with the teacher, and another center could have a computer where individual students engage with the PLS. In LEAP Innovations (2016) pilot testing of personalized learning software for English Language Arts, most classrooms used this integration approach.
2. **Whole class one-to-one instruction**—In this approach, teachers can guide instruction while all students in the class access the PLS via computing devices in one-to-one computing settings.
3. **Supplementary one-to-one instruction**—Teachers use this approach when they create time outside the core instruction for students to choose their own activities (including the use of PLS) and for students identified as needing additional work in areas that the PLS program can provide, sometimes with assistance from a teacher or an aide.

LEAP Innovations (2016) found that whole one-to-one and supplementary one-to-one instruction with PLSs were more effective in improving student achievement than the station or centers integration approach. However, the company noted that the supplementary one-to-one approach involved more instructional time for learners. Because this research was small in scale, we recommend the use of all integration strategies as viable options.

Check Your Understanding 5.7

Chapter 5 Summary

The following is a summary of the main points covered in this chapter.

1. **Introduction to instructional software**
   - Instructional software (a.k.a. computer-assisted instruction or courseware) is a computer program or app designed specifically to deliver instruction or assist with the delivery of instruction on a topic through pre-programmed curricular materials that are instructionally sequenced.
   - Functions provided by instructional software include drill and practice, tutorial, simulation, games or gamification, problem solving, and personalized learning.
   - Websites have emerged to help educators select well-designed instructional software products aligned to standards. These include edshelf, EdSurge Product List, and Learning List.
   - Evaluation criteria to determine the instructional strength of instructional software include examining the content, instruction, integration model, special needs features, research on learner involvement, learner engagement, technical and implementation

aspects, design and aesthetics, and evidence of effectiveness.

2. **Drill and practice software functions**
   - These provide exercises in which students work examples, usually one at a time, and receive feedback on the correctness of their responses.
   - Types of drill and practice include flash card activities, chart fill-in activities, branching drills, and extensive feedback activities.
   - Effective drill and practice includes control over the presentation rate, answer judging, appropriate feedback for correct and incorrect answers, and characteristics tailored to young learners.
   - Benefits include immediate feedback, increased motivation, and saving teacher time.
   - Challenges include instructional misuse, criticism by constructivists, and inequity in use.
   - Integration strategies for drill and practice software are to supplement or replace worksheets and homework exercises and to prepare for tests.
3. **Tutorial software functions**
   - These programs provide an entire instructional sequence on a topic similar to a teacher's classroom instruction.
   - Types of tutorials include linear and branching.
   - Effective tutorials include extensive interactivity, thorough user control, appropriate pedagogy and content, adequate answer-judging and feedback capabilities, appropriate graphics and/or video, and adequate recordkeeping.
   - Benefits include all the same benefits of drill and practice as well as self-paced instruction.
   - Challenges include use of directed instruction, the lack of well-designed products, and the limit to one instructional approach.
   - Integration strategies for tutorial software are to provide self-paced reviews of instruction, provide alternative learning strategies, and give instruction when teachers are unavailable.
4. **Simulation software functions**
   - These programs provide computerized models of a real or theoretical system designed to teach how the system works.
   - Types of simulations include products that teach about something, including physical and iterative simulations, and products that teach how to do something, including procedural and situational simulations.
   - Simulations are most effective when they have adequate documentation or scaffolding and are used in conjunction with hands-on activities.
   - Benefits include compressing time, slowing processes, involving students, making experimentation safe, making the impossible possible, saving money and other resources, allowing repetition with variations, and allowing observation of complex processes.
   - Challenges include criticism of virtual lab software, accuracy of models, and misuse.
   - Integration strategies for simulation software are to use in place of or as supplements to lab experiments, role-playing, or field trips; to introduce and/or clarify a new topic; to foster exploration and process learning; and to encourage cooperation and group work.
5. **Instructional game and gamification functions**
   - These are programs that add game rules and/or competition to learning challenges.
   - Types of instructional games include products that have an artificial game environment, game mechanics that govern rules of game-play, content-based conflict, contest or challenge often situated in a narrative story, and feedback and assessment that marks progress.
   - Effective instructional games include creation by teams of developers and content and teaching experts; curricular value; a pedagogical framework; assessment capabilities; social, societal, and cultural considerations; playfulness and motivation; and technical and physical ease of use.
   - Benefits include inclusion of problem-based learning (PBL), collaborative learning, realistic and immersive environments, motivation and competition, interactivity and feedback, achievement and rewards, and playfulness.
   - Challenges include learning versus having fun, lack of correspondence between game rules and learning objectives, transfer of learning, alignment with curriculum and teaching practices, and classroom barriers.
   - Integration strategies for instructional game software are to encourage lifelong playing and problem- or inquiry-based learning, to use in place of worksheets and exercises, to teach "noncognitive skills," and to teach cooperative group working skills.
6. **Problem-solving functions**
   - These are designed especially for the purpose of teaching component skills in problem solving or to practice solving various kinds of content-area problems.

- Types of problem-solving programs include those that teach content-area problem-solving skills and those that teach content-free problem-solving skills.
- Effective problem-solving programs are interesting and challenging and have a clear link to developing a specific problem-solving ability.
- Benefits include abilities to promote visualization in mathematics problem solving, improve interest and motivation, and prevent inert knowledge.
- Challenges revolve around multiple, imprecise problem-solving labels, software claims versus research-based effectiveness, and lack of skill transfer.
- Integration strategies for simulation software are to teach component skills in problem-solving strategies, to provide support in solving problems, to encourage group problem solving, and to provide practice in solving problems.

7. **Personalized learning systems (PLSs)**
    - These systems assess individual student learning needs by using complex algorithms and collections of data across students and provide a customized instructional experience matched to each student.
    - Characteristics of PLSs include student centeredness, adaptive assessment, curriculum matched to standards, competency or mastery base, multiple learning media, and data reports on individual and group progress.
    - Effective PLSs include student agency, content depth and quantity, meaningful data, aligned and validated assessments, evidence of impact, and an integration approach aligned with the teacher's instruction.
    - Benefits include accurate assessments based on formative, summative, and longitudinal data, progress linked to curriculum and standards, personalized instructional prescriptions matched to student needs, and data for meeting teacher and district accountability requirements.
    - Challenges for PLSs include the lack of buy-in from the school community, consideration as a panacea, data that are not extractable from the system, and no research-based impact evidence.
    - Integration strategies include using PLSs in stations or centers, in whole-class one-to-one instruction, or supplementary one-to-one instruction.

Technology Integration Workshop

1. Apply What You Learned

In this chapter, you learned about instructional software resources. Now apply your understanding of these concepts by completing the following activities:

- Reread Ms. Igwe's lesson *Math by Design* at the beginning of this chapter. Pay close attention to Step 3 of her TTIPP in which she identifies the technological possibilities for her problem of practice: engaging students in geometry skills by identifying geometry concepts in everyday life. Using your knowledge about instructional software functions introduced in this chapter (drill and practice, tutorials, simulations, games and gamification, problem solving, and personalization), generate at least one or more new technological possibilities for targeting Ms. Igwe's problem of practice.
- Review how Ms. Igwe RATified the lesson in Step 5 of her TTIPP as represented in Figure 5.2. Use the RAT Matrix to analyze the role(s) and the relative advantage that your new technological possibilities (identified in the last step) would play in the lesson. You must reflect on the roles that your identified technological possibilities play as replacement, amplification, and/or transformation of instruction, student learning, and/or curriculum. Do you feel your proposed technology would provide relative advantage?

2. Technology Integration Lesson Planning: Evaluating Lesson Plans

Complete the following exercise using the sample Technology Integration Examples 5.1–5.5, any lesson plan you find on the web, or one provided by your instructor.

a. Locate lesson ideas—Identify three lesson plans that focus on any of the instructional software functions you learned about in this chapter, for example:

- Drill and practice
- Tutorial
- Simulation
- Game or gamification
- Problem based
- Personalized learning

b. Evaluate the lessons—Use the Technology Lesson Plan Evaluation Checklist and the RAT Matrix to evaluate each of the lessons you found. Based on the evaluation and your RATification of the lessons, would you adopt these lessons in the future? Why or why not?

3. Technology Integration Lesson Planning: Creating Lesson Plans with the TTIPP Model

Review how to implement the TTIPP Model (see Figure 2.6) for technology integration planning and use Ms. Igwe's lesson *Math by Design* in this chapter as a model. Create your own technology-supported lesson that uses instructional software by completing the following activities:

a. Describe Phase 1—Analysis of Learning and Teaching Assets and Needs:

- What is the problem of practice or main content topic in your lesson?
- What are the technology resources that your students, their families, you, and your school could bring as assets to the lesson?
- What are the technological possibilities for helping to solve or help the identified problem of practice? Identify the technology(ies) you will integrate into the lesson and ensure you have skills and resources you need to carry it out.

b. Describe Phase 2—Design of the Integration Framework:

- What are the objectives of the lesson plan?
- How will you assess your students' accomplishments of the objectives?
- What integration strategies are used in this lesson plan?
- What is the relative advantage of using the technology(ies) in this lesson?
- How will you prepare the learning environment?

c. Describe Phase 3—Post-Instruction Analysis and Revisions:

- What strategies and/or instruments would you use to evaluate the success of this lesson in your classroom to determine any needed revision?
- Create descriptors for your new lesson (e.g., grade level, content and topic areas, technologies used, ISTE standards, 21st-century learning standards).
- Save your lesson plan with all its descriptors and TTIPP Model notes and share with your peers, teacher, and others.

When you use your new lesson with students, be sure to assess it using the Technology Impact Checklist.