Sokoban Sample Lesson Plan
Day 3 – 50 minutes
Scalable Game Design Summer Institute 2011

Note: Red links take you to portions of the tutorial.
Blue links provide background information.

1. Learning Objectives:
   In this unit, students will create a simple version of a Sokoban game while learning AgentSheets software program. Students will apply design process to identify objects “agents” and interactions “operations”. Throughout this unit, students will be introduced to basic computational thinking, including basic object interaction, stacks, creating object instances, rule based programming, and message sending.

   In this lesson, students will be programming the sokoban to follow the crate when it moves so he doesn’t get left behind! This will complete the basic version of the Sokoban game.

2. Standards:
   ISTE (International Society for Technology in Education) NETS (National Educational Technology Standards)
   • # 1a apply existing knowledge to generate new products
   • #4b plan and manage activities to develop a solution or complete a project.
   • #4d use multiple processes and diverse perspectives to explore alternative solutions.
   • #6c troubleshoot systems and applications.

   ISTE NETS are referred to by CDE Performance Standards for Teachers #7- Technology
   Please check with your district’s technology department to see if there are additional standards at the district or school level.

3. Anticipatory Set / Formative Assessment of progress: 10 minutes
   This assessment can be an individual check by the teacher of each student’s work or can be done as a peer evaluation - students working in pairs. Alternately, each student could evaluate his or her own program.

   Hit Run and see if everything works correctly. Check:

   • Does the sokoban move all directions?
   • Is sokoban unable to move through walls?
   • Does the sokoban push the crates in all directions? (Note: It is OK if sokoban is still being left behind –we will fix that today)
   • Are the crates stopped by walls?

   If answer to any of these is no, go back to the related section and see what might have done wrong. Otherwise, if everything works correctly GOOD JOB!
4. **Guided Practice / Monitoring: 20 minutes**

Demonstrate how to open AgentSheets program and have students open saved versions of Sokoban from last lesson. Check for understanding.

Remind students that to make sokoban push the crates, we must complete four separate programming steps:
- First, we must program sokoban to send push messages to the crate to move in each of the four directions.
- Second, the crate must understand and react to the push messages to actually move when sokoban pushes them.
- Third, the crate must send a message to sokoban to move with it, and
- Fourth, sokoban must understand and react to the move messages to move behind the crate.

We will be completing the third and fourth steps today.

**Step 3:**
At this point, our sokoban does not follow the crate after he has pushed it. The crate moves but sokoban stays put. To fix this problem, show students how to send one of four move messages to sokoban.

*Move Messages to the Sokoban*

**Step 4:**
Just like when the crate didn’t know how to react to the push messages until we created the methods and programmed the corresponding behaviors, sokoban will not know how to react to the move messages being sent by the crate. If students run the program now, the error messages given will demonstrate this. So we must create the four methods in sokoban’s behavior.

*Programming Sokoban to React to Messages from Crate*

Ask students to test the sokoban’s movement by playing the game (see *Play Test: Test whether your Sokoban now moves when you push the crate* in the tutorial).

Assist students in doing a final play test of programs to check for correct programming.

**Finishing Up**

Allow students time to answer the following question:

Now let's test our program one final time:

*Does everything work like you expect it to?*

Great Job! You have just completed a basic Sokoban game! Feel free to go back through and make any improvements to your game or experiment with any ideas/agents/behaviors you may have. You can even add another level or a way to win. Awesome work!
Remind students to save the worksheet periodically and check with the reset button to verify that it worked.

5. **Final Assessment: 15 minutes**
   To allow for assessment of the student’s final project, students will upload games to a central location. Grade student work based on Rubric discussed at the beginning of the unit.

6. **Closure: 5 minutes**
   Restate the scope of the project. Provide information for students about accessing programs from remote locations (at home, the library, etc.) so they can share their program with family and friends.

7. **Extension/Remediation** – At this point, there is no way to win this game. As an extension, students can set up guidelines for scoring and winning. Perhaps the object is to push the crate to a target spot with the fewest number of moves, or the player must push several crates to corresponding targets before going to another level. Students may want to develop multiple levels of increasing difficulty. Students can play classmate’s Sokoban games and/or those published by students from other middle schools and/or the University of Colorado- see *Scalable Game Design Arcade.*