Sokoban Sample Lesson Plan
Day 2 – 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 Frogger 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 refining the sokoban’s movements so he cannot walk through walls. Students will also be creating the crate agent and learning how to use the computational thinking pattern Push.

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 / Modeling: 5 minutes
Student work showcase: Select one of the student’s worksheets from the last lesson and project it on an overhead screen. Demonstrate what can be done so far on her/his worksheet (sokoban can move in all 4 directions). Compare this to a completed version of Sokoban so students can see where they are headed. Inform students that today they will be making a crate agent and programming the sokoban to push the crate.

4. Teaching: 5 minutes
Input – Project Overview and AgentSheets

Review components of AgentSheets:
• Gallery- where agents are
• Worksheet – where game is created
• Behavior – how to tell each agent what to do
And the computational thinking pattern in Sokoban:
- **Push**: Sokoban pushes the crates to the goals

Remind students about the nouns (the agents) and the verbs (the operations) of Frogger. We will creating the next most prominent noun today – the crate, and corresponding behaviors.

5. **Guided Practice / Monitoring: 35 minutes**

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

Explain that currently the sokoban can walk through walls and we need to modify his behavior to prevent this. **Not_Moving_Through_Walls**

Remind students how to create a new agent and have students make a crate agent. Also remind students that agents can be edited at any time, so they should not spend a lot of time on the artwork right now. **Creating_the_Crate**

Have students place at least one crate agent on worksheet using pencil tool. Remind students to **save worksheet** and ask students to test the sokoban’s interaction with the crate by playing the game (see **Play Test: How does your Sokoban interact with the new Crate Agent** in the tutorial). Check for understanding.

Introduce the computational thinking pattern **Push**.

To make sokoban push the crates, we must complete three 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 first two steps today and steps three and four next lesson.

**Step 1:**
Ask students to duplicate the rules under the sokoban behavior. They will modify these to see a crate instead of a floor and they will change the move to make then statements. This will send one of four messages to the crate. Demonstrate this first, then have students work independently to complete. **Programming_Sokoban_to_push_the_Crates**
Ask students to test run the game. Students should get one of four error messages because we have not programmed the crate to respond to the messages. See Play Test: Testing the Sokoban Pushing Crates in tutorial. Check for understanding.

Step 2:
Now it is time to instruct students how to program the crate’s reaction to the push messages sent. Because there are 4 messages being sent, students must create and program 4 separate methods in the crate behavior. Reacting to Push Messages

First, just ask students to make the methods and play test (see Play Test: Testing Crate Receiving Messages in tutorial). This way the students can learn how to use error messages to help them debug their programs. Check understanding of each student.

Next, show students how to program each of the four methods in the crate behavior. Programming Crates

Finish today’s lesson by asking students to test if the crate moves as expected when being pushed. See Play Test: Test whether your Crate moves correctly when pushed in tutorial. Check for understanding.

6. Closure: 5 minutes
Restate the scope of the project. In the next lesson, we will be programming the sokoban to follow the crate when it moves so he doesn’t get left behind!

7. Extension/Remediation – students can edit their agents at any time. Encourage students to spend a short time on the initial creation and edit later as desired. Optional Activity: Students can go to the Scalable Game Design Arcade to see other versions of Sokoban for ideas of how they would like to complete their games.