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

1. **Learning Objectives:**
   In this unit, students will create a simple but complete version of Pac-Man game while expanding knowledge of the AgentSheets software program. Students will apply basic and advanced design process to identify objects “agents” and interactions “operations”. Throughout this unit, students will be introduced to **computational thinking** patterns and skills, including basic object interaction, creating object instances, rule based programming, and message sending. Over the course of the unit, the difficulty of the game with ghosts with random movement will be contrasted to the difficulty of the game with ghosts with artificial intelligence.

   In this lesson, Pac-Man gets deflated! We will be programming the collision between Pac-Man and the ghosts. Our goal is to have a playable game by the end of the class period today.

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, which states, “The teacher will have demonstrated the ability to instruct students in basic technology skills. He/She will: … instruct students in basic technology skills by imbedding them in their standards-based, content instruction (7.5.3)”

   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 (Pac-Man can move in all 4 directions eating the pellets as he goes and the depiction changes to match the direction, ghosts move randomly, but nothing happens when they run into each other). Compare this to a completed version of Pac-Man so
students can see where they are headed. Inform students that today they will be fixing this glitch – now there will be a consequence when Pac-Man meets up with a ghost.

4. **Teaching: 5 minutes**
   
   **Input** – Overview of project and AgentSheets
   
   If needed, remind students of the components of AgentSheets:
   
   - Gallery - where agents are
   - Worksheet – where game is created
   - Behavior – how to tell each agent what to do

   Discuss relevant computational thinking patterns: Today we will be using Collision.
   
   - **Collision**: Pac-Man collides with ghosts.
   - Artificial Intelligence using **Collaborative Diffusion**
   - the **Hill Climbing** algorithm

   Remind students about the nouns (the agents) and the verbs (the operations) of Pac-Man. We will be programming one of the most important verbs (behaviors) of the game today – Pac-Man meets ghost.

5. **Guided Practice / Monitoring: 25 minutes**

   Demonstrate how to open AgentSheets program and have students open saved versions of Pac-Man from last lesson. Check understanding of each student.

   Ask students to import or create a “deflated” Pac-Man agent.

   Remind students how to modify behavior of existing agents. The Collision of ghosts and Pac-Man has two parts:

   1. **Ghost Behavior** - when ghost sees Pac-Man he sends message to Pac-Man to “deflate” himself
      
      **Ghosts tell Pac-Man to deflate**

   2. **Pac-Man Behavior** – add method in Pac-Man agent to change itself into deflated Pac-Man icon and make a sound, then disappear, could display “You’ve been Deflated!” or some other message to indicate game over.
      
      **Pac-Man deflate**

   ** If you ask students to test play the game between parts 1 and 2 they will get the error message, “I am just a Pac-Man, I don’t know how to react to the message ‘deflate’”. This might be good for students to try so they have a place to begin when they are in the de-bugging stage.
Students should save the worksheet periodically and check with the reset button to verify that it worked.

**IMPORTANT: Saving the Worksheet**

6. **Assessment of progress: 10 minutes**
   This assessment can be an individual check by the teacher of each students work or can be done as a peer evaluation - students working in pairs. Alternately each student could evaluate his or her own program.

   ![Play Test: Finished game?](link)

   Click on “Run” and see if everything works correctly. Check:

   - Does Pac-Man move all directions?
     - Do the walls prevent Pac-Man and Ghosts from moving over them?
     - Does the depiction change based on which direction Pac-Man is going?
   - Does Pac-Man “eat” the pellets, making them disappear with a sound?
   - Does Pac-Man deflate when he gets caught by a ghost??

   If the answer to one of these questions is no, turn on **conversational_programming** and go back to the related section to see what might have gone wrong.

   Otherwise, if everything works as it should, GOOD JOB!

   Make sure students have ample time to try out the game with ghosts with random movement. As we begin to change the ghosts’ behavior to use artificial intelligence, we would like students to discover how much more difficult the game becomes.

7. **Closure: 5 minutes**
   Restate the scope of the project. Tomorrow Pac-Man starts emitting an odor – we hope a pleasant one! – as we start programming the artificial intelligence of the ghosts.

8. **Extension/Remediation** – students can edit the depictions of agents at any time. Encourage students to spend a short time on the initial creation and edit later as desired.

   Students can program **automatic regeneration** of Pac-Man after he gets deflated so that the player doesn’t have to hit reset each time.

   Optional activity: Have students go to the **Scalable Game Design Arcade** to play other Pac-Man games to generate ideas for how they would like to design their games.