DANCE PARTY—LEARNING FACILITATOR’S GUIDE

Introduction
The children will be considering a career as a software engineer (or software developer). In Step 1.1 of the activity, they should click on the Software Engineering link to see a video on this career path.

Software developers are the creative minds behind computer programs. Some develop the applications that allow people to do specific tasks on a computer or another device. Others develop the underlying systems that run the devices or that control networks. [http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm](http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm)

More information on careers in software engineering/developers can be found at:

As software engineers, club members will be using a programming language called Scratch to develop a program a dance routine. The Scratch website is [https://scratch.mit.edu/](https://scratch.mit.edu/)

Safety note
There are no particular safety concerns in this activity. Please note that there are places where the children will be able to customize what the sprites (dancers) can “say.” Be sure to monitor that the words are appropriate.

Preparation
1. Materials – none

2. This activity cannot be done using the Tablets. It must be done using computers with Adobe flash installed.

3. Timing – If a club is allocating 75 minutes for this activity, the approximate breakdown of the time should be:
   - 20 minutes – computer introduction and practice with Scratch
   - 35 minutes – design a dance sequence using the Scratch program
   - 20 minutes – computer follow-up, review of other dances, reflection, and revisions

   Note – much more time or several sessions could be utilized, if desired.

4. Be sure to go through the computer experience and try the Scratch program before you do the activity with a group!
The Design Challenge Notes
The children should sign on to WISEngineering as individuals and should do the online programming by themselves, if enough computers are available.

1. The specifications to be met are:
   - Have one dancer do at least two dance moves
   - Play two different instruments as part of the dance move
   - Have your dancer say something by pressing a key
   Possibly the biggest challenge is to coordinate the two dancers’ routines to produce an orchestrated performance.

2. The Scratch program is accessed by clicking on the Scratch button on the right side of the menu bar. You must be in a project “run.” It does not work in the preview mode.

3. In Step 1.4 (Develop Knowledge), the children will be guided through the steps and procedures to create a dance with Scratch. Students will have to toggle between the WISEngineering tab and the Scratch tab while going through the tutorial.

4. In Step 1.5, participants plan their dance design based on the techniques learned in Step 1.4.

5. Key steps to create a dance:
   - A backdrop is picked by clicking on the backdrops library icon and selecting from the choices presented.
   - Similarly the sprites are picked from the sprites library. Participants will need to select two sprites.
>Leave the tabs set on SCRIPTS

To program a particular sprite, first click on that sprite.
>**MOTION** (dance moves) are created by clicking on the Motion scripts and moving blocks to the programming area on the right. The numbers in the block can be changed. Negative (-) signs can be used to indicate the opposite direction for steps or turns.
>**CONTROL** commands allow one to repeat the dance moves (the number of repeats can be changed) and also place delays (“wait”) in the routine. The amount of time for each delay can be changed.
>**EVENT** commands include pressing the green flag to start the dance(s). This is needed when two dancers are going to perform as it will signal both to start. Also in EVENT, one can set up an action when the space bar or other key is pressed. This can allow the sprite to say something.
>
>**LOOKS**, the dancers can be programmed to say something. BE SURE TO MONITOR THE WORDS USED!
A program for one sprite may look like this. All words and numbers in white can be changed:

6. In Step 1.6, participants go to the Scratch site and develop their dance. They may use other commands from the Scratch program not introduced in Step 1.4, if they want to.

The programs should be uploaded to the Design Wall when completed:
- First save the dance under FILE (a name will have to be assigned)
- Then upload to the Design Wall, under FILE

(Think about creating a naming system so club members can identify the Dance(s) from others that they are going to review.

Reflection
As with all the WISEngineering activities, the reflection part is crucial to the children’s experience. In addition to thinking about their own program (Step 1.7), they will be evaluating another team’s program (Step 1.8). In both steps, they will be judging whether the program meets the design specifications:
- Have one dancer do at least two dance moves
- Play two different instruments as part of the dance move
- Have your dancer say something by pressing a key

Facilitators should assign which club member will look at and evaluate which other club member’s dance. Encourage the use of constructive comments.

Throughout the WGG experience, children should post notes, comments, pictures, and videos to the Design Journal and Design Wall. Items uploaded to
the Design Wall are shared and club members may comment on others' uploads. This may happen during the activity with the computer or after the activity, if children have been given access to WISEngineering outside the club using a phone, personal tablet, or computer etc.

Final notes:
If time allows or if additional sessions can be scheduled, three important aspects of engineering can be addressed:
1. Sharing of ideas and designs. Club members can meet as a group and share their designs and discuss their work and what was helpful and what was not.
2. Engineering is iterative. It would be great if groups could go back and “try again” modify their design, use new ideas – ultimately try to improve on their work.
3. Becoming and being a software engineer/developer.