

SCRATCH DANCE PARTY

LET'S MAKE THIS A PARTY!

In this lesson, you will add more dancers to your dance project to make it more fun and interesting!



START HERE

- Pair up with the person to your left.
- Show each other your dance party projects.
- Pick out the best part of your partner's dance and tell them why you like it.
- Share how you coded your dance with your partner.



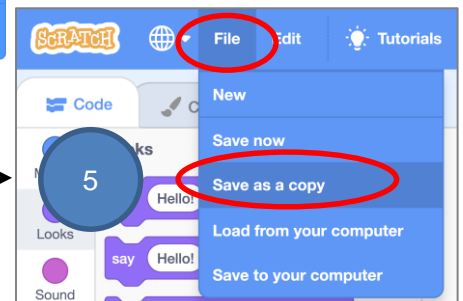
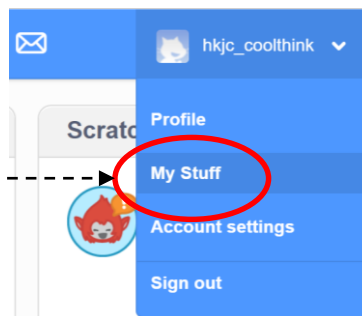
- Sign into your account at scratch.mit.edu

- Click on My Stuff.

- Open your Dance Party project.

- Save as a copy.

- Name it **Dance Party 2**.



SCRATCH

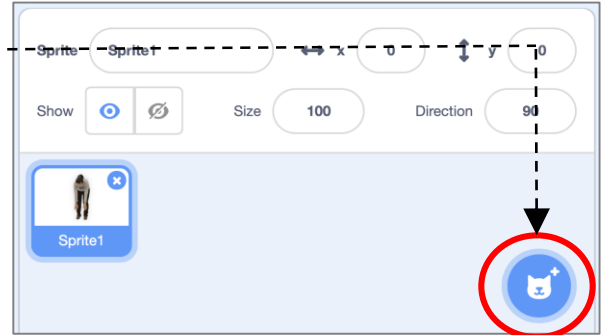
DANCE PARTY

LET'S ALL DANCE!

CT Tips
Parallelism: Multiple sprites can dance together in parallel.

Add another sprite to your dance party.

7



See if you can make your sprites dance together.

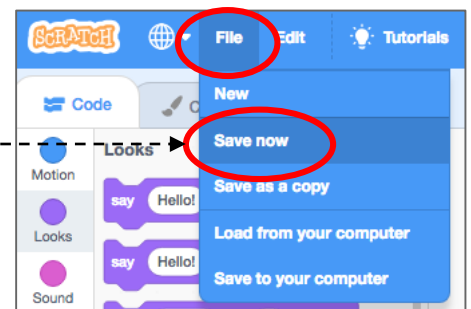
8

Or take turns!

When you finish, you will add your project to your teacher's Studio

- Save your project by clicking **"Save now"** under the File menu.
- Click the orange **"Share"** button.
- Go to your teacher's Studio (they will give you a URL).
- Click the **"Add projects"** button.
- Select your project from the bottom of the page and add it to the Studio.

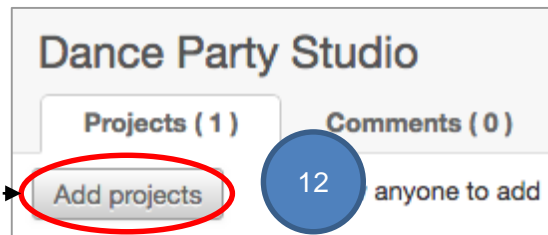
9



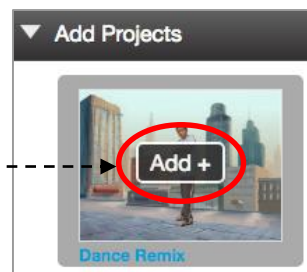
10



11



12



13

SCRATCH DANCE PARTY

COMPUTATIONAL THINKING CONCEPTS

The following is the computational thinking concept learnt in Lesson 2.

L1U8.2 Dancing Cat

1. Parallelism:



```
when green flag clicked
  forever loop
    wait .5 seconds
    next costume
```



```
when green flag clicked
  forever loop
    move 50 steps
    change x by 20
    wait 1 seconds
    change x by -20
    wait 1 seconds
    move -50 steps
    wait .5 seconds
```

SCRATCH

DANCE PARTY

COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking practices used in this unit.

L1U8.2 Dancing Cat

1. Reusing and remixing:

- a) Watch classmates' projects and remix dance ideas

2. Being incremental and iterative:

- a) Make a sprite dance
- b) Add a second sprite and have them dance together

3. Testing and debugging:

- a) Test that the sprite dances as expected using code blocks
- b) Test that the **two** sprites dance together as expected using code blocks

4. Algorithmic thinking:

- a) Use code blocks and sequences correctly to get the sprites to dance the way you want them to