Level 2 Christmas Capers

These projects are for use outside the UK only. More information is available on our website at http://www.codeclub.org.uk/. This coursework is developed in the open on GitHub, https://github.com/CodeClub/ come and join us!

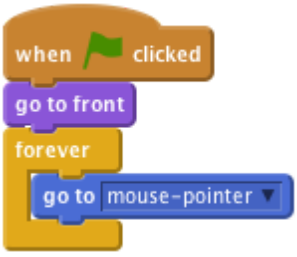
Introduction

In this project we'll create a game with scrolling backgrounds, scoring and a festive game over screen.
A disaster in a toy factory has sent presents flying into the sky, help Rudolf to save Christmas by catching the presents!

STEP 1: Make Rudolph fly

Activity Checklist

1. Start a new Scratch project. Delete the cat by right-clicking it and selecting Delete
2. Replace the background with **SkyBackground.png**.
3. Add the Rudolph sprite to the project (use the **resources/Rudolph.png** file)
4. Make Rudolph follow the mouse by using the following script:

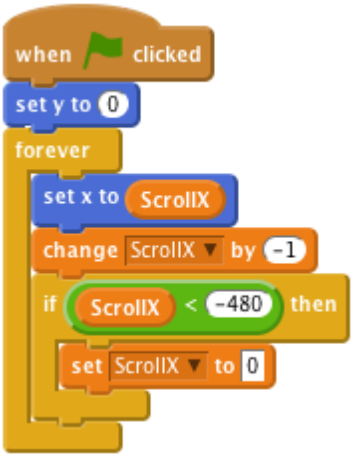


Test Your Project

Click the green flag and move the mouse, does Rudolph follow the mouse?

Save your project

1. To make the game more interesting we will add some moving snowy hills to make it look like Rudolf is flying. Add the Snow sprite to the project (use the **SnowHills.png** file).
2. Rename the sprite to **Snow1**.
3. Create a new variable by clicking the *Data* tab and then **make a variable**. Call it `ScrollX` and make it for all sprites, then uncheck the box next to it to remove it from the stage. This will be used to control how the hills move.
4. Add the following script to make the hills move:



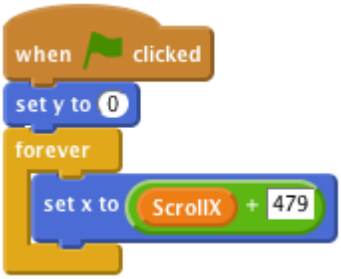
Test Your Project

Click the green flag, do the hills move? What happens as the hills move to the side of the screen?

Save your project

1. Let's fix the issue with the snowy hills flickering when they reach the right of the screen. Add more hills to the stage use the **new sprite from file** button to add the Snow sprite to the project again (use the **SnowHills.png** file).
2. Rename the sprite to **Snow2**.
3. Add the following script to the Snow2 sprite to allow the 2nd set of hills to follow

closely behind the first:



Test Your Project

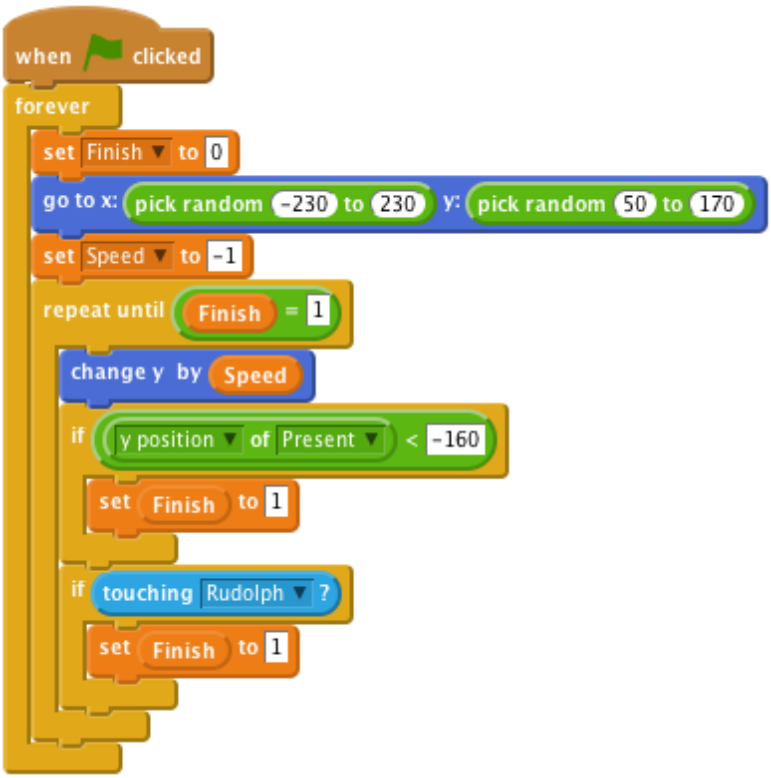
Click the green flag, do the hills move? Has the issue with the flickering trees been fixed?

Save your project

STEP 2: Falling Presents

Activity Checklist

- [ ] 1. We now need to add in the presents for Rudolph to collect. Add the **Present** sprite to the project (use the **Present.png** file).
- [ ] 2. **Create a new variable** by clicking the `Data` tab and then **make a variable**. Call it `Finish` and make it for this sprite only, then uncheck the box next to it to remove it from the stage. This will be used to control when the present should be removed from the game.
- [ ] 3. **Create another variable** and call it `Speed` and make it for this sprite only, then uncheck the box next to it to remove it from the stage. This will be used to control the speed that the present falls down the screen.
- [ ] 4. Add the following script to the **Present** sprite to allow it to fall from the sky. Note that we will use `pick random` to make the present appear in a different place each time.
- [ ] 5. By using the `touching [ Rudolph ]` block we can make the present disappear when touched, we can use this later to keep a score.

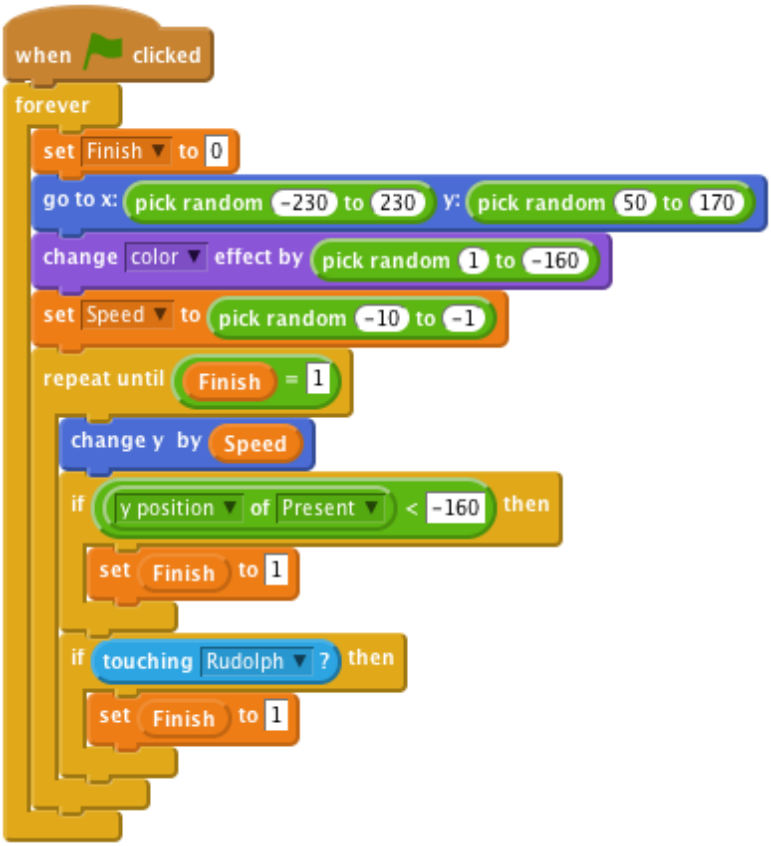


Test Your Project

Click the green flag, do the presents fall from the sky? Do they disappear when Rudolph touches them or they hit the ground?

Save your project

1. Let's make the game more interesting by changing the colour of the presents each time they fall. Do this by using the `change colour` block.
2. Change the speed of each present by replacing `set Speed to -1` with the `pick random` block, try different values such as **-10** to **-1**. Your script should now look like this.



Test Your Project

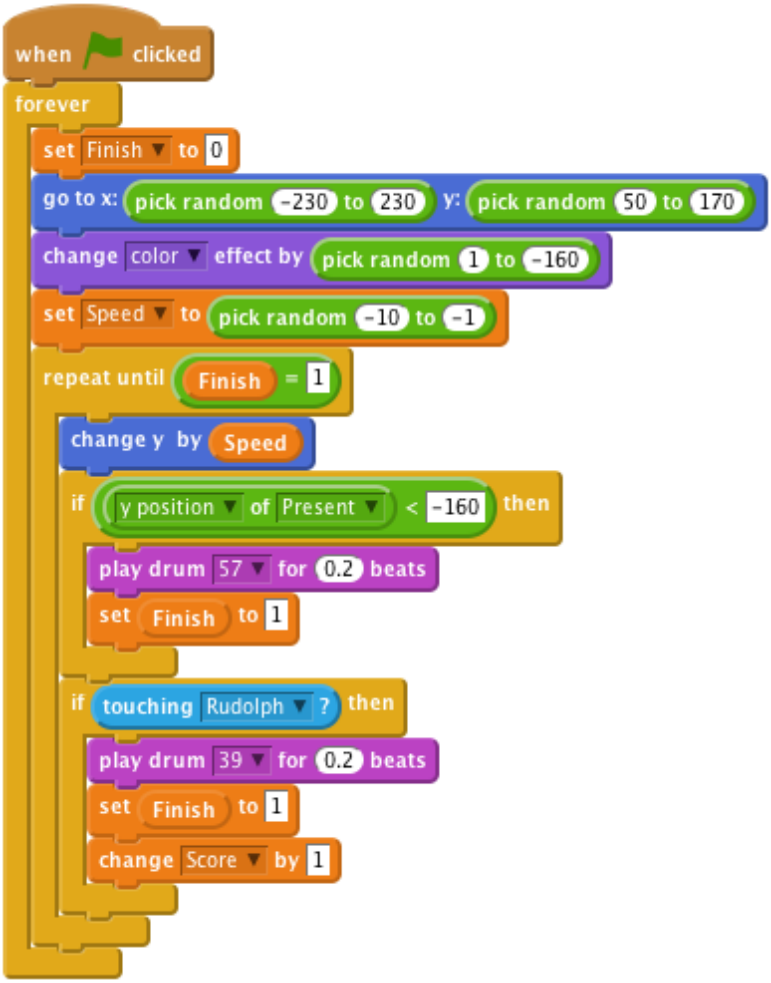
Click the green flag, do the presents fall at different speeds and colours?

Save your project

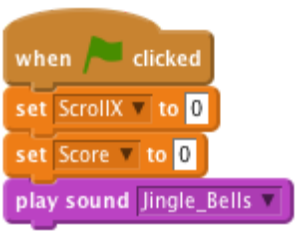
STEP 3: Scoring and Sound Effects

Activity Checklist

- [ ] 1. **Let's change our script to keep track of a score within the game.** We can then use this later to work out when the game over message should appear.
- [ ] 2. Create a new variable. Call it `Score` and make it for all sprites. Leave this variable ticked so it appears on the screen.
- [ ] 3. Change the script behind the **Present** sprite to look like this. Note we have both added sound effects with the `play drum` command and also `change [ score ] by 1` when Rudolph touches the present.



- [ ] 1. Let's add some music to the game, import the sound file **Jingle_Bells.mp3** to the **Stage**.



1. Add the following script to the **Stage**, this will `set score to 0` when the game is started. It will also play Jingle Bells while the game is being played.

Note, if at first the music sounds 'choppy' save your project, close Scratch and then open your project again.

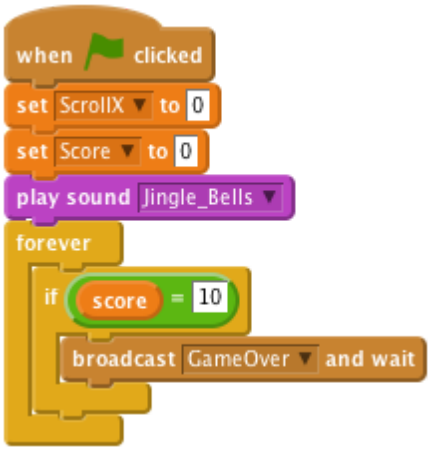
Test Your Project

Click the green flag, does the score change when Rudolph touches a present?

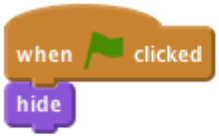
Save your project

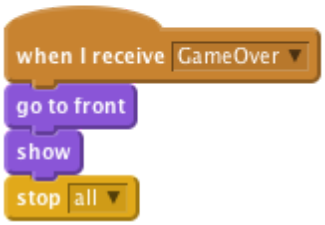
STEP 4: Game over

1. **Let's change our script to keep track of a score within the game.** We can then use this later to work out when the game over message should appear.
2. Change the script on the **Stage** so when the `Score` reaches **10** we will `broadcast` a **GameOver** message.



1. We now need to add in our GameOver message. Add the **GameOver** sprite to the project (use the **GameOver.png** file).
2. **Add the following script to the GameOver sprite.** This will `hide` the picture when the game starts and `show` it when the GameOver message is received.





Test Your Project

Click the green flag, does the score change when Rudolph touches a present?

Save your project

Challenge: Make the game harder

- [ ] Can you make the presents wobble on their way down the screen?
- [ ] Can you add more than one present to the game at the same time?
- [ ] Change the game over message to appear after 20 presents are collected.
- [ ] Can you reduce the score by 1 when a present hits the ground?

Save your project

Well done you've finished, now you can enjoy the game.
Have a very Merry Christmas!