Project 3 - Interactive thank you card

Now for something new. Project 3 introduces **clipart** (turtle shapes and backgrounds), **animation**, **text boxes**, **multiple pages** and **page navigation**, **buttons** and **sounds**.

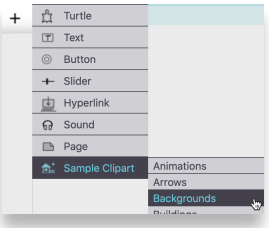
You have certainly bought or made a paper birthday card. This project shows you how to make a digital *Thank You* card to send to frontline or essential workers. Of course, the same skills can be used to make any other type of card: for a birthday or Mother's Day...Ready?

Start a new project. If there isn't a turtle on the page, choose **Turtle** in the + menu.

As a good habit, start by **naming your project**. And remember to save often.

Create a background scene

Basically, you need to put a shape or costume on the turtle. Click on the + menu, choose **Sample clipart**, then choose **Backgrounds** in the sub-menu.



The left pane will now become a **Clipart Pane** and will contain a set of backgrounds you can use.

In the Clipart Pane, each box is numbered. You can see # 1 in the image on the left.

Type this in the Command Centre:

```
setshape 1
```

USE ANY NUMBER IN A BOX THAT HAS CLIPART

If you wish, use **setsize** to make it bigger or smaller. In the Command Centre, type:

```
setsize 55
```

THE DEFAULT SIZE OF THE TURTLE IS 40

This looks good, but it is still just a turtle wearing a costume. You can drag your turtle / background around (try it). Use the command **home** if you want to put the turtle in the centre of the page. Now, type this to **stamp** the turtle on the page:

```
stamp
```

THE TURTLE COSTUME IS "PRINTED" ON THE BLANK PAGE

It may seem that nothing really happened, but the turtle's shape (the background clipart) is now **stamped** on the page. You are looking at a **huge turtle** standing on **an image of itself**. Try to move it around: the image that moves is the turtle, the one that doesn't is the background:

To eliminate any confusion between an active turtle and stamped background, set the turtle's shape back to its original shape:

```
setshape 0
```

THE DEFAULT NUMBER OF THE TURTLE IS 0

Now you have both an active turtle and a nice background (as shown above). If you want to erase the background, and perhaps other lines or drawings, use:

```
cg
```

OR THE LONG FORM, CLEARGRAPHICS

A QUICK WAY TO GIVE A SHAPE TO A TURTLE: CLICK & CLICK

- Click on a clipart you like. The mouse pointer turns into a hand, holding the clipart.
- Now click on the turtle to give it that shape.
- Then, use stamp to make it a background image.

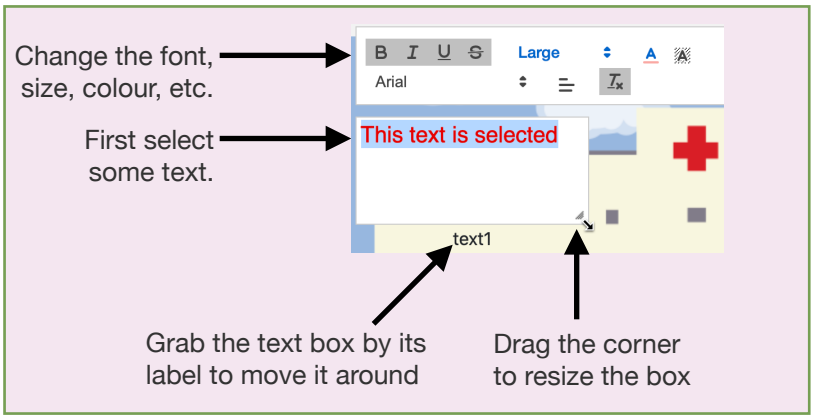
Create a text box and write your message

Cards you buy in a store have words already in them and then usually you add some words of your own. Let's do that here.

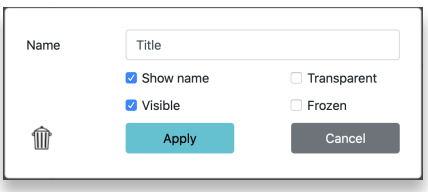
Click on the + menu then choose **Text**.

A text box appears on the page. **Move** it anywhere by dragging the Name (**Text1**) and **resize** it by using the arrow in the bottom right corner.

Type in the title of your "Thank You" card, select the text and **format the text** using the options provided (the picture below shows you how to move and resize the text box).



Right-click on the text box to open its dialog box. Change its name to **Title**. The name must be a **single word**, no space anywhere (not even at then end - remember that). Then click **Apply**.



About the several options in the Text box dialog box

If ☐ **Show name** is unchecked, or if ☑ **Transparent** is checked, you can't move the text box around.

If ☑ **Frozen** is checked, you can't resize or move the text box, but you can still type in it. It is frozen in the place you put it.

If ☐ **Visible** is unchecked, oops, it's not deleted, but you can't see it anymore! There are two things you can do about this:

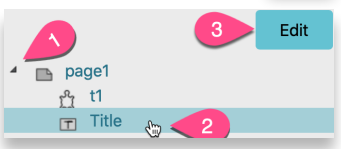
- Type this in the Command Centre (don't forget the comma):

```
title, showtext
```

`TITLE,` MEANS "I AM TALKING TO YOU"

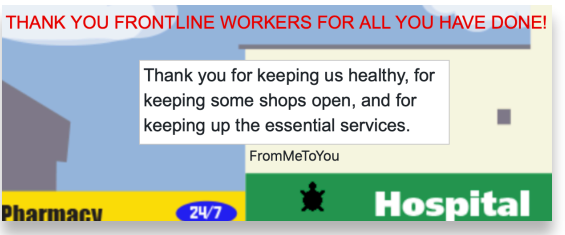
or…

- Click on the **Project tree** icon to see absolutely everything your project contains.
  1. Click on the **triangle** besides **page1** to reveal its contents.
  2. Click on the **name** of the text box.
  3. Click on **Edit** to open its dialog box and check the box ☑ **Visible**.



Make adjustments to the text so it looks nice on the page. This example shows a **transparent** text box. You will not see its label.

Add a new text box. Open its dialog box and name it **FromMeToYou** (one word, no space). Type your *Thank You* message in that text box, and format it to your liking. As an example, we wrote some words for you.



Let's hear it!

Just for fun, type this in the Command Centre:

```
show frommetoyou
```

```
Thank you for keeping us healthy, for keeping…
```

You see, the **name of the text box** (`frommetoyou`) is also a command that reports the **contents of the box**.

Here is a nice way of using this feature. Type this in the Command Centre:

```
say 'hello'
```
THE COMPUTER READS ALOUD HELLO

```
say frommetoyou
```
THE COMPUTER READS THE CONTENTS OF THE BOX

You should **hear** the computer reading the message. If you don't, check the volume on your computer.

The command `frommetoyou` reports the contents of the text box as a big long word. So instead of just reading `'hello'`, Lynx reads the big long word.

You can use different voices for reading text. Type this in the Command Centre to get a list of all the voices available on your computer (may be different or not available at all on your computer):

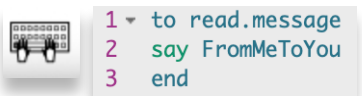
```
show voices
Alex Alice Alva Amelie Anna Carmit Chantal…
```

Your list will be different. Try other voices using **sayas** instead of **say**:

```
sayas frommetoyou 'anna'
```

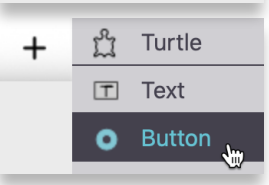
Keep in mind that this *Thank you* card is something to share. People other than you will **NOT** have a Command Centre to run this instruction. So let's create a **procedure**, and a **button** to do that.

First, create this **procedure** in the Procedures Pane:



```
to read.message
say FromMeToYou
end
```

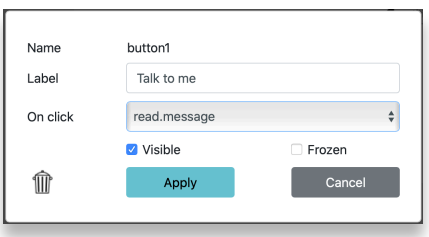
Then, click on the + menu and choose **Button**.



A button, named **nothing** by default, appears in the centre of the page. A button is just a visible object that runs code when clicked.

Right-click on the button to open its dialog box.

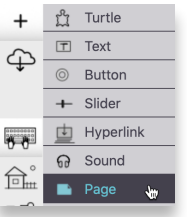
In the dialog box that appears, type **Talk to me** as the label (it can be anything as it is plain English), and choose **read.message** in the **On Click** menu. Click **Apply**.



Now click on the button. Did you hear your message?

Drag the button by its centre to relocate it to the lower part of the page. Drag the bottom-right **corner** to resize it.

Add a page



Just like cards bought in a store, your digital card should have more than just a cover page. Let's add a second (inside) page to make this card more interesting. Click on the + menu then choose **Page**.

BLANK! Don't worry, your work is safe. This is a **new blank page**. Look above the Procedures Pane. It says "**page2**" (**a**). Click on the **arrows** (**b**) to turn pages, but come back to **Page2** to continue making the card.

GOOD TO KNOW

A page name (such as `page1` or `page2`) is also a **command** that goes to that page. That's why a page name must be a single word, no space, unlike commands like `forward` or `right` which require a space before the number.

You can rename or delete a page from the Project tree. Click on the page name and click on **Edit** to open its dialog box.

Buttons to turn pages

There are a few ways to go from page to page.

1. The **arrows** above the **Procedures Pane** (as described above).
2. Typing the **name of the page** in the Command Centre.
3. Writing a **procedure** and using it in a **button** or in a **clickable turtle**.

Let's use the **button** method. Create this procedure:

```
to go.page1
page1
end
```

Then create a new button, open its dialog box, type **Go to page 1** (or anything you like; as explained earlier, this is plain English not code) as the label, and choose `go.page1` (the procedure) in the **On Click** menu.

Again, resize and relocate the button in the corner of the page (usually, you place a button that goes “back” on the left side of the page).

Test your button. It should bring you to **Page1**. Now that you’re good at it, while you are on **Page1**, create a new button that goes to **Page2** (this requires a new procedure and a new button, place it on the right side).

Your first animation: a moving ambulance

New turtle, new background

Make sure that you are now on **Page2** (blank). There was no turtle to start with on **Page2**. Click on the + menu, choose **Turtle**.

Give it the shape of a city street scene:

```
setshape 2
```
Use the number of a background clipart that you like

Now use the same steps as you did before to create a background:

- Use the command **`home`** to centre the turtle on the page.
- Use the command **`setsize number`** to change the size of the turtle, if you want.
- Use the command **`stamp`** to stamp it on the page.
- Use the instruction **`setshape 0`** to return the turtle to its original shape.

More Clipart

You will need new clipart for the animation. Click on the + menu, choose **Sample clipart**, then choose **Nature, Other** in the sub-menu. This adds more clipart to your collection - you will have to scroll down to see them all; there is an ambulance towards the end.

Get the Ambulance to move

Drag the turtle (still has the default turtle shape) towards the bottom left side of the page, on the street. A turtle will always move in the direction its head is pointing. Make it point to the right - type this in the Command Centre:

```
setheading 90
```
This means “face East” - 90 degrees on a compass

Project 3 - Interactive thank you card

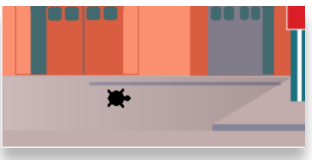
It is a good idea to “prepare” the turtle while it has a turtle shape - it is the only shape that shows its current heading.

When the turtle is “all good”, you can give it the shape you want.

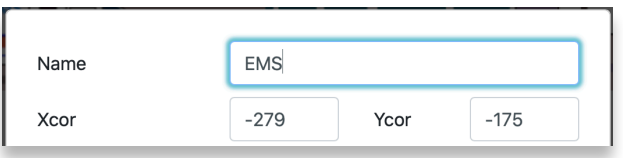
Now that the turtle is “ready to roll”, give it the shape of the ambulance. Type this in the Command Centre:

```
setshape 34
```

USE YOUR NUMBER WHICH MAY BE DIFFERENT



There will be more than one turtle on that page. Giving them good names is a great idea. Right-click on the turtle (ambulance) and type **EMS** (Emergency Medical Services) as its name:



Notice its `xcor` and `ycor` position also (-279 and -175 in this example, your values will be different). These are the X and Y coordinates of the turtle.

Now that the turtle has a name, you may as well use it when you talk to it. Type this in the Command Centre (don’t forget the comma):

```
ems,
repeat 100 [forward 3 wait 1]
```

THIS MEANS: TURTLE NAMED **EMS,** I AM TALKING TO YOU

Cool, you have just added animation! Bring the turtle back to its starting position: type this in the Command Centre (use the `xcor` and `ycor` you saw in your turtle’s dialog box above - you can round the values if you wish):

```
setpos [-280 -175]
```

SETPOS MEANS SET POSITION

Put that in a procedure

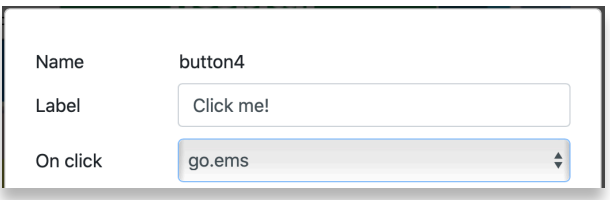
If you want to launch the action from a clickable turtle or a button, you first have to make it into a procedure. Create one like this in the Procedures Pane.

```
13 to go.ems
14  ems,
15  setshape 34
16  setheading 90
17  setpos [-280 -175]
18  repeat 100 [forward 3 wait 1]
19  end
```

Run the procedure from a button

Remember, when the card is shared with someone, it will not have a Command Centre to run its features. Create a new button, relocate and resize it like you did before.

Right click on the button to open its dialog box, and choose the procedure (`go.ems`) in the **On Click** menu:



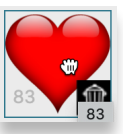
Great! Close the dialog box and **click on the button** to launch the action. It should go to the starting position and ride for a few seconds.

A clickable turtle: add a beating heart

You can easily do this by having a turtle grow and shrink. Stay on **Page2** and Click on the + menu, choose **Turtle**.

Right-click on the turtle to open its dialog box, name the turtle **Heart**.

Open the Clipart Pane and look for a **heart** shape near the end. Click on it, and your mouse pointer turns into a **hand**.



Now click on the turtle to give it that clipart as a shape. This is the same as running the instruction `setshape 38` (or whatever your number is).

Woah! This is a huge heart! Run an instruction such as `setsize 10`, or a number, higher or lower, that looks good on your page.

And for the animation, type this in the Command Centre:

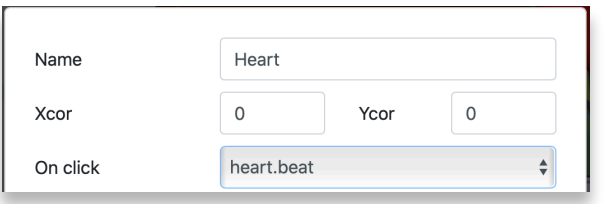
```
heart, forever [setsize 20 wait 4 setsize 10 wait 4]
```

This tells your turtle (named **heart**) to run the instructions inside the square brackets *forever.* Basically, keep changing the size of the heart and wait a bit between each change. Click on **Stopall** in the Command Centre if you get dizzy!

If you want to launch the action just by clicking on the **heart**, you first have to make it into a procedure. Create one like this in the Procedures Pane. Choose different values for **setsize** if you like.

```
21 to heart.beat
22   heart,
23   forever [setsize 20 wait 4
24            setsize 10 wait 4]
25   end
```

Right-click on the Heart and select this procedure as the **On Click** instruction by using the Up and Down arrows in the **On Click** field:



GOOD TO KNOW

Notice that you can click on the heart to start the animation, and click on it again to stop it.

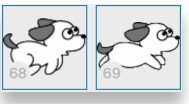
Suggestion: It may be useful to add a Text box on the page with a simple message like: *Click on the Heart.*

Run Spot Run!

So far, we have things you could never have on a printed card. Let's add more! While on **Page2,** click on the + menu, choose **Turtle**.

Right-click on the turtle to open its dialog box, name the turtle **Dog**.

Click on the + menu, choose **Sample clipart**, then choose **Animations** in the sub-menu. Open the Clipart Pane and look for **dog** shapes. The dog shapes are pointing to the right. So set the heading of the new turtle-dog:




```
dog, setheading 90
```

Here's a new trick. If you give **more than one shape** to the turtle, it will **switch shapes** each time it moves (`forward`, `back`). Type this in the Command Centre:

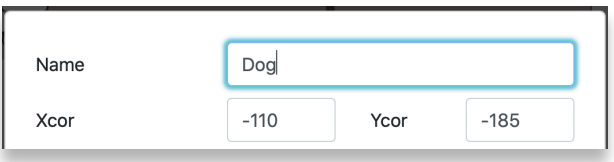
```
setshape [68 69]
```

USE YOUR OWN NUMBERS WITHIN THE **[ ]**

The turtle is now wearing two shapes. Check this out: type this in the Command Centre:

```
forward 5
forward 5
forward 5
```

Drag the **dog** somewhere on the street, perhaps in front of the ambulance. Now open its dialog box to find out its position on the page. In this example, the position is `-110 -185`



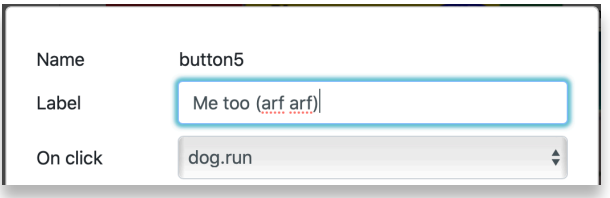
Now create this procedure to prepare the dog and launch the animation:

```
to dog.run
dog,
setshape [68 69]                    DOG, I'M TALKING TO YOU
setheading 90                       FACE EAST AND RUN LEFT TO RIGHT
setpos [-110 -185]                  THE POSITION YOU SAW IN THE DIALOG BOX
repeat 100 [forward 4 wait 1]       CHOOSE NUMBERS YOU LIKE
end
```

Run the procedure from a button

Create a new button, relocate and resize it like you did before.

Right click on the button to open its dialog box, and choose the action procedure (`dog.run`) in the **On Click** menu:



Great! Click **Apply** to close the dialog box, and click on the button to try it. Wonderful!

Somewhere over the Flashing Rainbow

Here is another type of animation. No movement or shape switching. The turtle-rainbow will simply appear and disappear (hide, wait, show, wait).

Stay on **Page2** and Click on the + menu, choose **Turtle**.

Right-click on the turtle to open its dialog box, name the turtle **Rainbow**.

Open the Clipart Pane and look for a **rainbow** shape. Click on it, and your mouse pointer turns into a **hand**.

Now click on the turtle to give it that clipart as a shape. This is the same as running the instruction `setshape 29` (or whatever your number is).

Use a `setsize` instruction if you want to change its size. The default size is 40, so anything above 40 will be larger, below 40 is smaller.

For the animation, type this in the Command Centre:

```
rainbow, repeat 10 [ht wait 4 st wait 4]
```

`Ht` and `st` mean **H**ide**T**urtle and **S**how**T**urtle. These instructions mean: do this 10 times: hide the turtle-rainbow, wait 4/10 of a second and show the turtle-rainbow, wait 4/10 of a second. `Wait 10` means pause 1 second.

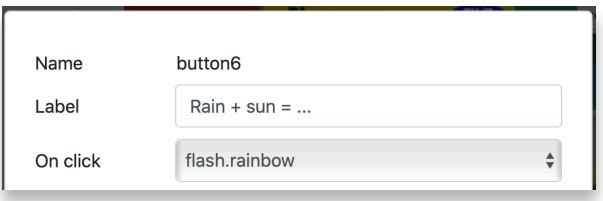
In the Procedure pane, make a procedure like this:

```
35 to flash.rainbow
36   rainbow,
37 repeat 10 [ht wait 4
38                st wait 4]
39 end
```

Run the procedure from a button

Create a new button, relocate and resize it like you did before.

Right click on the button to open its dialog box. Type a label for the button and choose the action procedure (`flash.rainbow`) in the **On Click** menu:



Finally, click on this button to try the animation. It will flash 10 times.

Now make your words flash!

Now here's not one, but two magic tricks. Create this procedure in the Procedures Pane:

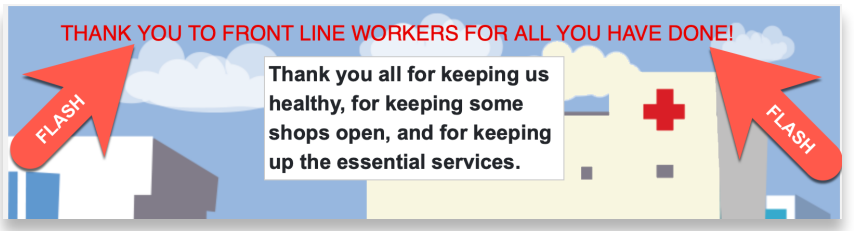
```
43 to startup
44  page1
45  title,
46  repeat 10 [hidetext wait 5
                showtext wait 5]
47  end
```

It is VERY IMPORTANT that you name this procedure `startup`, with this exact spelling. `Startup` is a special word. When someone opens your project, this procedure is executed automatically.

What does it do?

- `page1` is the name of the first page, and it is a command that brings up that page. It is important to do so when the project opens.
- **title** is the name of the first text box on that page. Add a comma (`title,`) to say "**title, I'm talking to you**".
- `hidetext` means hide the Text box and `showtext` means show the Text box. The `repeat` instruction does this 10 times: make the text box invisible, then a delay of 5/10 of a second. them make the text box visible 10 times and add a delay of 5/10 of a second.

To test this, save your project ( ) and go to **My Projects** ( ) where all your projects are saved. Click on your **Thank you** project to see it in Play mode. You will be seeing Page1, and the **title** should flash.



Let's make some noise: Add sound

The final touch: add a cheering sound for the opening of the project, when the title flashes.

First, import a sound to your computer

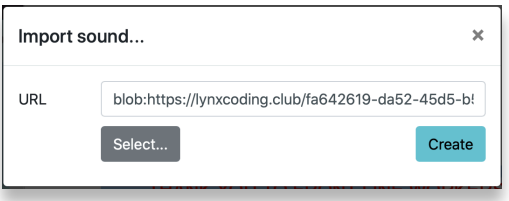
You can import WAV and MP3 sound files. If you don't have a cheering sound on your computer, you may go to this web site to find one:

`bbcsfx.acropolis.org.uk`

Type **cheers** in the **Search** box and click on **Duration** to see the shortest sounds at the top of the list. In this example, we are using a short 10 second applause and cheers sound. Try the sound on the Web page, and if you like it, download it to your computer (probably in the **Downloads** folder).

Bring the sound into your project

Back to your project. On **page1**, click on the + menu and choose **Sound**. This brings up the Import sound dialog box:



Click on **Select** and locate the sound (WAV or MP3) file that you have downloaded to your computer. When you have it, the URL field will show the link. Now click on **Create**.

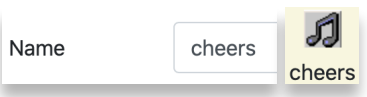
This creates a sound icon named **sound1** on your page. Click on it to hear the sound.

Play the sound

The sound has a name (sound1) and that name is also a command to play the sound. Type this in the Command Centre:

```
sound1
```

Great! Give it a better name now. Right-click on the sound icon to open its dialog box, and name it cheers (one word, no space).



Include the sound in the Startup procedure

Finally, include it in the `startup` procedure, so it plays when the project opens:

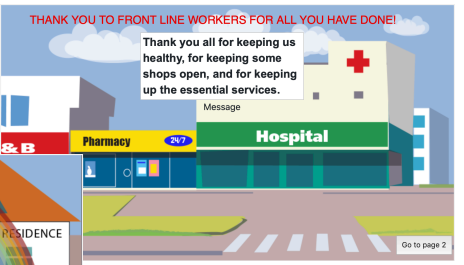
```
to startup
page1
launch [cheers]
title,
repeat 10 [hidetext wait 5
           showtext wait 5]
end
```

Why `launch`? `Launch` will start (launch) the sound and let it play on its own, and while the sound is playing, Lynx will immediately jump to the next instruction and flash the **title** text box. The cheering and the flashing will occur at the same time.

Final test and debugging

You're a coder now… And good coders sometimes have… good bugs in their code! Time to test all the features of your project. Check for things that don't work as expected, and check for error messages in the Command Centre. Test:

- the button Talk to me
- the button that goes from page1 to page2
- the button that goes from page2 to page1
- the flashing heart
- the ambulance moving
- the dog running
- flashing rainbow & words



If everything is fine, you may right-click on the **sound** icon and uncheck the **Visible** box to hide it. It will still play the sound!

Sharing the card

Nice work! Time to share it with some people, or with the entire Lynx community. *Appendix F* has all the instructions for sharing projects.

All the procedures of this project

Just in case you need to check something, here are all the procedures in the project. Yours can be different, of course!

```
to read.message
say FromMeToYou
end

to go.page1
page1
end
```

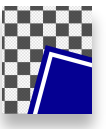
```
to go.page2
page2
end

to go.ems
ems,
setshape 34
setheading 90
setpos [-280 -175]
repeat 100 [forward 3 wait 1]
end

to heart.beat
heart,
forever [setsize 20 wait 4
         setsize 10 wait 4]
end

to dog.run
dog,
setshape [68 69]
setheading 90
setpos [-110 -185]
repeat 100 [forward 4 wait 1]
end

to flash.rainbow
rainbow,
repeat 10 [ht wait 4
           st wait 4]
end

to startup
page1
launch [cheers]
title,
repeat 10 [hidetext wait 5
           showtext wait 5]
end
```

More advanced ideas

Add your own clipart

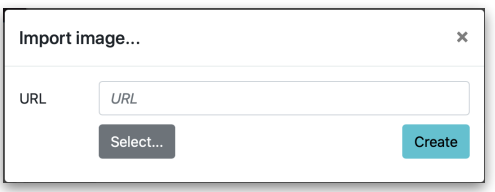
Add your own clipart to one of the scenes. It could be the logo of your local hospital or pharmacy, a wheelchair symbol, or anything related to your *Thank You* card.

Look in *Appendix D* for clipart resources on the Web, and special instructions regarding clipart with transparent surrounding. In short:



- Find a PNG image with a transparent surrounding. Often seen on a grey or a checkered background.
- Download the image to your computer or your work space in the cloud. Chromebook users, see Appendix E, FAQ.

- Open the Clipart Pane, select an empty spot and click anywhere on it to get this icon:  . Click on it.
- This opens the Import image dialog box. Click on SELECT to find your downloaded image. Once the URL field is full, click on CREATE to add the new clipart to your Clipart Pane.



Create a new turtle and use the new clipart as the turtle shape, and set the size of the turtle, not too big, not too small. You can always adjust this detail later.

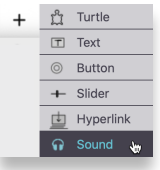
```
setshape 4
setsize 10
```

USE YOUR OWN CLIPART NUMBER
TRY DIFFERENT NUMBERS

Automatic page turning

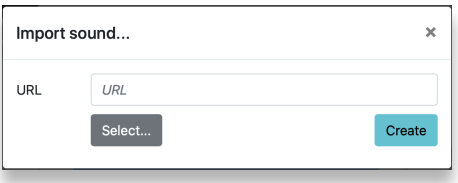
Something else you can do: look at your `startup` procedure. You can, after the Text Box flashing, add an instruction that goes automatically to `page2`. Remember, a page name is also an instruction that goes to that page.

Record your voice

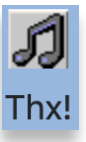


If you know how to record your voice on your computer, you can record a **SHORT** message and save it as a **WAV** or **MP3** file on your computer. *Appendix D* suggests some voice recording tools. Then…

Choose **Sound** in the + menu. This opens the Import sound dialog box.



Click on **Select** to navigate to the sound file you have recorded and saved, then click on **Create** to create the sound icon on the page.



Click on the icon to hear your recording.

If you wish to rename the sound icon, right-click on the icon to open its dialog box, and type a new name. Remember: use a one-word name, no space.

You can relocate the icon on the page and let the user click on it to hear your message, or you can:

- Create a procedure that plays the sound, and run this procedure from a button

```
to hear.me
thx!
end
```

USE THE NAME OF YOUR SOUND ICON

- Include this command (`thx!`) inside an existing procedure.
- Include this command in the `startup` procedure, so it happens automatically.

Curriculum Links for Ontario

- **C3.1** - Solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves conditional statements and other control structures.
- **C3.2** - Read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes.

Project 4 - Secret codes (i*l*o*v*e*)

Let's do something totally different now. No turtles or graphics (unless you want to add some), but lots of fancy text manipulation. Secret codes have been used for ages, even Caesar, two thousand years ago, had his own "recipe" for encrypting messages.

You should know that you can talk to and give commands to the **cursor** (the flashing bar inside the text box): move the cursor around and have it insert, delete, or change the text as it moves. Let's try this!

The cypher (encoding) in this project is pretty simple, nothing like spies and governments use nowadays, but it is a fun starting point. The process, for you, will resemble this:

- Create a text box for your message.
- Think (pseudo-code) about a way to "scramble" your text.
- Create a procedure that manipulates the text, in a way that matches your pseudo-code.
- Create a procedure that does the exact opposite, to decode the scrambled text.