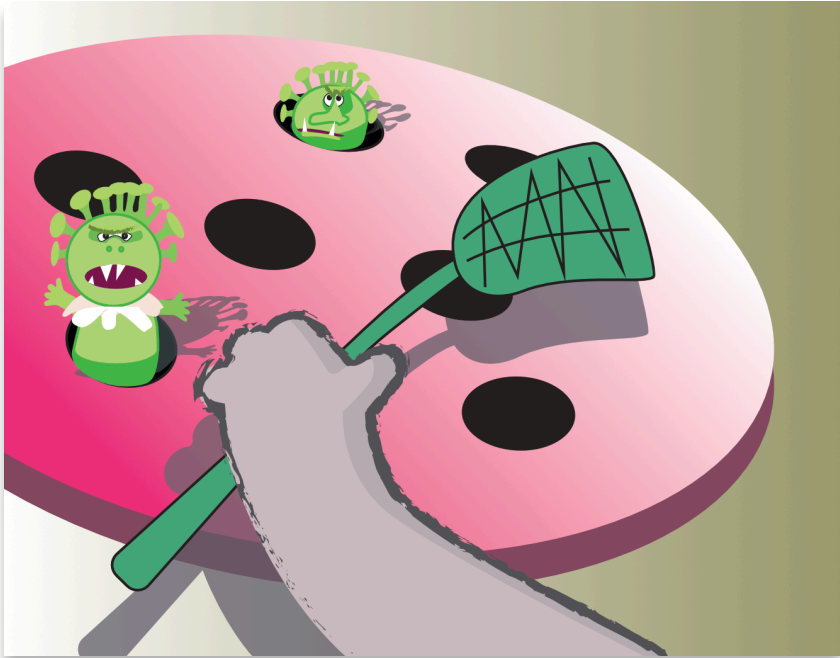
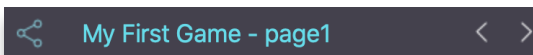


Project 5 - Kill the virus



Many years ago there was a game called Whack-a-mole. Moles appear **randomly** and very **briefly** on a board or in an arcade game machine and you have to hit them before they go back underground. Let's modify it for the reality of COVID-19. Let's program a video game and call it *Kill the Virus*. Our game will involve **animation**, **score keeping**, and **controls** (to make the game harder).

Start a new project, and give it a new name now. Don't forget to save often!



A plain background

First, select a nice background for the game. It can be as simple as a coloured background (use the **setbg** command, check the colour chart in *Appendix C*). Or, find a clipart image of a protective mask and import it into your Clipart Pane.

If you chose to use a **setbg** background, you probably still have your turtle on the screen, waiting for instructions. Perfect.

If you chose to use a stamped image background, don't forget to **stamp** the large image, and set the turtle back to its original turtle shape using **setshape 0**.

Find a virus clipart

Time to get a clipart for the virus popping up. Look in the **Sample Clipart / Nature, Other** for something you could use. Otherwise, look in *Appendix D* for clipart resources on the Web, and special instructions regarding clipart with transparent surrounding. In a few words:

- Find a PNG image with transparent surrounding. Often seen on a grey or a checkered background.
- Download the image to your computer.
- Open the Clipart Pane, click on a free spot, then on the **+** in that free spot.
- This opens the Import image dialog box. Navigate to the image that you have just downloaded. You now have your clipart!



Change the turtle to the virus

Use the virus clipart to change the turtle's shape, and set the size of the turtle-virus, not too big, not too small. You can always adjust this detail later if the game is too easy or too difficult.

```
setshape 2  
setsize 10
```

USE YOUR OWN CLIPART NUMBER
PLAY WITH DIFFERENT NUMBERS

Then, create a procedure that the virus will execute when you "hit" it.

```
to hit  
; grow big for a split second, then back to normal  
virus,  
setsize 15  
wait 3  
setsize 10  
end
```

YOU ARE ABOUT TO RENAME THE TURTLE FROM T1 TO VIRUS

FYI: WAIT 10 IS 1 SECOND

Later on, you will add to this procedure, to keep the score of the good hits. But for the moment, right-click on the virus to open its dialog box:

Project 5 - Kill the virus

Plenty of things to do and learn here! Let's start with just three things:

a) First, choose the **hit** procedure in the **On click** menu:

The dialog box contains the following fields and options:

- Name: VIRUS (with callout 'b')
- Xcor: 52, Ycor: 52 (with callout 'c')
- On click: hit (with callout 'a')
- On touch: -
- On message: -
- On colour: -
- Visible: Visible
- Frozen: Frozen
- Buttons: Apply, Cancel
- Trash icon

b) At the top of the dialog box, you see the name of the turtle (**t1**). When you add turtles to the page, the turtles are named **t1**, **t2**, **t3** and so on. You can change the name, but always use a single word, no space. Change the name to **VIRUS** in order to match the name in the **hit** procedure you just made.

c) Then you see its current **x** and **y** coordinates positions. More about these in a moment.

Click on **Apply** to save your changes, and give it a try: Click on the virus, it should grow and shrink, as instructed in the **hit** procedure.

Play procedure

Time to pseudo-code. What does it mean to “play” this game. Think about it before continuing to read.

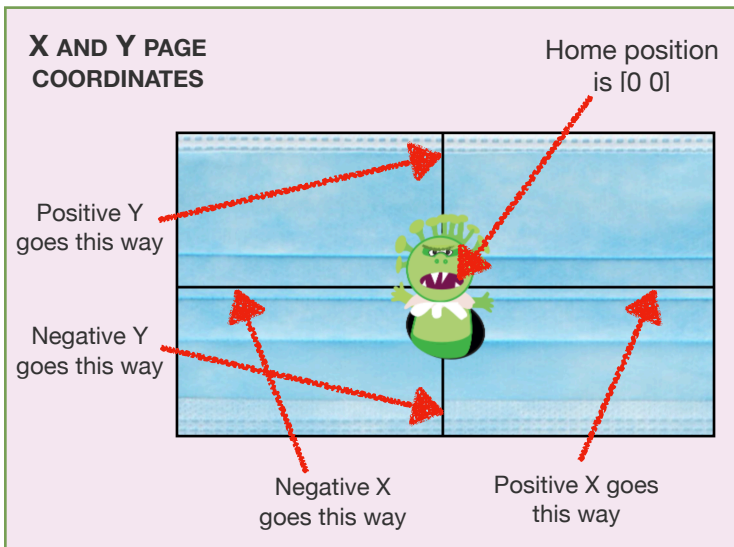
- Hide the virus.
- While hidden, move the virus to a random place.
- Make the virus appear for a very brief moment.
- If the user manages to click on the virus while it is visible, run the **On click hit** procedure.

The key action here is “move the virus to a random place”. Let’s see how you can make this happen. Type this in the Command Centre:

```
home
show pos
0 0
setx -200
setx 200
setx 0
sety 200
sety -200
sety 0
```

```
THE VIRUS GOES TO THE CENTRE OF THE PAGE
SHOW ME YOUR POSITION
THE CENTRE COORDINATES [X Y] ARE [0 0]
MOVE TO THE X COORDINATE -200
MOVE TO THE X COORDINATE 200
MOVE TO THE X COORDINATE 0
DO THE SAME FOR THE Y COORDINATE
```

What have we learned here...



If you are wondering how much right-left, top-down you can go, type this in the Command Centre:

```
show projectsize
800 450
```

This is the size of a **standard** project (see footnote ¹ below). You can select a different size when you start a new project. In this example, the **maximum X** and **Y** positions are -400 and +400 horizontally (for **X**), -225 and +225 vertically (for **Y**).

¹ To create a project of a different size, choose New project in the **Down from the cloud** menu (arrow down from the cloud). Choose a size in the dialog box (standard, large, small, custom).

Project 5 - Kill the virus

These are the limits for the **page**, for this **project size**. We will make the **play area** just a bit smaller: -300 to 300 horizontally (**X**), and -200 to 200 vertically (**Y**).

MOVE THE VIRUS TO A RANDOM POSITION

This game is based on the fact that the virus will move to a random position on the screen before appearing briefly. Here's what to do:

Try this in the Command Centre:

```
setx 300           THE VIRUS GOES EXACTLY TO THE X COORDINATE 300
setx random 300   A RANDOM X COORDINATE, LESS THAN 300
setx random 300   ANOTHER RANDOM X COORDINATE, LESS THAN 300
```

Random 300 will always give you a positive number (between 0 and 299), and using that for **setx**, the virus will always end up on the right-hand side of the page. Too easy to hit, right! What you really need is a random number between say... **-300** and **300**. To get this result, take **-300**, and add a **random number between 0 and 601**. The result will always be between **-300** and **300**. Just imagine: $(-300 + 0)$ $(-300 + 1)$ $(-300 + 2)$ up to $(-300 + 601)$. Type this in the Command Centre and try it a few times:

```
setx -300 + random 601   RANDOM 601 REPORTS A NUMBER
                           BETWEEN 0 AND 600 BUT WE
                           STARTED ON THE LEFT EDGE AT -300
```

With that in mind, create this **move.show** procedure in the Procedures Pane:

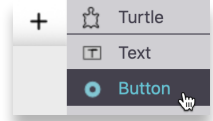
```
to move.show
; hide, move, show for a short while, hide again
; play area is 600 x 400
virus,                               OR WHATEVER NAME YOU USED FOR T1
ht
setx -300 + random 601
sety -200 + random 401               SAME LOGIC MUST APPLY TO THE Y AXIS
st
wait 8
ht
end
```

And a **play** procedure, for the game to keep going:

```
to play  
  forever [move.show]  
end
```

Create a button and set the **On Click** action to the procedure **play** as follows:

Click on the **+** menu and choose **Button**.



Right-click to open its dialog box, type a label, and set its instruction to **play**.

A screenshot of the Scratch 'Button' dialog box. The 'Name' field contains 'button1'. The 'Label' field contains 'PLAY'. The 'On click' dropdown menu is set to 'play'.

Make sure you put the button in a far corner, away from the main play area.



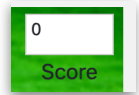
Test the button: the virus should appear in random places. Try to hit it when it shows itself. When hit, the virus should grow and shrink.

Let's record the number of hits

Time to keep score! You can record the score in a text box.

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

A text box appears on the page. Type **0** in it. Make it quite small and move it to a corner of the page. Then, right-click on it to open its dialog box, and rename it **score**.

A screenshot of the Scratch 'Text' dialog box. The 'Name' field contains 'Score'. There are checkboxes for 'Show name' (checked), 'Visible' (checked), 'Transparent' (unchecked), and 'Frozen' (unchecked). There are 'Apply' and 'Cancel' buttons.

Try this in the Command Centre:

```
show score SHOW ME THE CONTENTS OF THE TEXT BOX NAMED SCORE
```

```
0 CURRENT CONTENT IS 0
```

```
setscore score + 1 SET THE CONTENT TO "CURRENT VALUE" + 1
```

```
show score SHOW AGAIN PLEASE
```

```
1 CURRENT CONTENT IS NOW 1
```

Project 5 - Kill the virus

See, `score` reports the contents of the text box with that name, and `SETscore` sets the contents of the text box with that name. That's how you increase the score.



When exactly do you need to increase the score? In which procedure?

Right! When you manage to hit the virus. Add this to the hit procedure:

```
to hit
; grow big for a split second, then back to normal
setscore score + 1
virus,
setsize 15
wait 3
setsize 10
end
```

Score keeping requires two new actions for your game to work. At the beginning of the game, you must reset the score to 0. And each time the score increases, you have to check if the player is a winner (when the top score or maximum hits permitted are reached).

A Reset procedure and button

A reset procedure is a simple thing. Create this procedure:

```
to reset
setscore 0
virus, ht
end
```

Create a button to reset the game, place it near the other button.

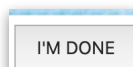


Try the **RESET** button, then the **PLAY** button, and after a few hits, click on **PLAY** button to stop the game and try **RESET** again.

Now YOU have access to the **Stopall** button in the Command Centre, but if you **share** the project with friends, there won't be a Command Centre. Create this `stop.now` procedure and a button that goes with it. Next add a Button and choose `stop.now` in the **On Click** field.



```
37 to stop.now
38 stopall
39 end
```



Announce the champion!

Time to reward the champion. You remember the procedure where you increase the score? Add a conditional instruction: if the score reaches 10, announce the winner and stop the game.

Edit the procedure `hit`:

```
to hit
; grow big for a split second, then back to normal
setscore score + 1
if score = 10 [winner]
virus,
setsize 15
wait 3
setsize 10
end
```

And now make a `winner` procedure. It uses the primitive `announce`, which causes a message to appear on the screen.

```
to winner
announce [Congrats! You killed the virus 10 times]
stopall
end
```

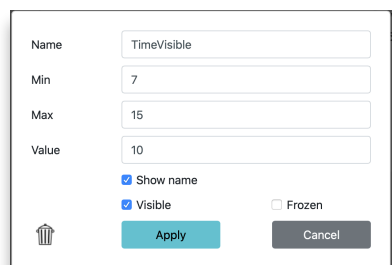
Don't click on **RESET**, type **8** in the text box (just cheating to complete the game faster), and click on **PLAY**. After two hits, you get to **10** and you should get a big message and the game should stop.

Add different levels of difficulty

Many games start “easy” and increase the level of difficulty as you go. What would make this game easier or more difficult?

The brief moment when the virus is visible could be longer in the beginning, and shorter later on.

Click on the **+** menu, choose **Slider**. Right-click on the slider and name it **TimeVisible** (one word, no space). Set the minimum at 7, and the maximum at 15 and the current value at 10. You can change these values later if you are not happy with them. Place the slider near the top or bottom edge of the page.



The image shows a configuration dialog box for a slider control. It has the following fields and options:

- Name:** TimeVisible
- Min:** 7
- Max:** 15
- Value:** 10
- Show name
- Visible
- Frozen
- Apply** button
- Cancel** button

Project 5 - Kill the virus

You can use the slider just like you did for the **Score** text box. The slider name reports its value. The word set, followed by the name of the slider, can be used to set its value. Try these instructions in the Command Centre:

```
show timevisible          SHOW ME THE CURRENT VALUE OF THE SLIDER
10                        THIS ANSWER DEPENDS ON YOUR SLIDER
settimevisible 15        SET THE SLIDER'S VALUE TO 15
settimevisible timevisible - 1  LOWER THE VALUE BY 1
```



When should you decrease the value of the slider to reduce the time the virus is visible and make the game more difficult?

Yes, when the player manages to hit the virus. Edit the **hit** procedure again:

```
to hit
  setscore score + 1
  if score = 10 [winner]
  if timevisible > 7 [settimevisible timevisible - 1]
  virus,
  setsize 15
  wait 3
  setsize 10
end
```

WHY THE CONDITION “IF TIMEVISIBLE > 7” ?

The slider cannot go below 7, the lower limit that you have set. If you try to go below, Lynx will display an error message. So when you hit the virus, the slider will go lower ONLY if it is 8 or above.

Finally, you should also reset the slider's value when you reset the game. Edit the **reset** procedure:

```
to reset
  setscore 0
  settimevisible 15
  virus, ht
end
```



Where is the right place to use this value? In your procedures, where do you manage how long the virus is visible?

Yes, it is in the `move.show` procedure, where it says “show, wait a bit, hide”. Instead of using a fixed number for `wait` use the value of the slider:

```
to move.show
virus,
ht
setx -300 + random 601
sety -200 + random 401
st
wait timevisible
ht
end
```

Testing and sharing

Time to test everything. Test it in the **Lynx editor** first, so you can make corrections, then test it in **Play mode** (on the page **My Projects**) to see the game as *your friends will see it*.

- Click on **RESET**. Is the text box reset to 0? Is the slider reset to 15?
- Click on **PLAY**. Is the virus moving and showing? Is it easy at the beginning? Is the slider decreasing when you hit the virus?
- Can you stop the game by using the button **I'M DONE**?
- Is the game playable (not too difficult)? What values can you change to make it easier or harder?

Did everything work as you expected? Ready to share this game with your friends? See *Appendix F* about sharing.



Is the game too easy? Too difficult? Would you know how to fix that? What about setting the `timevisible` to a larger number at the start (`reset`) so the game is easier for very young children?

Project 5 - Kill the virus

All the procedures of this project

Just in case you need to check something, here are all the procedures in the project, along with some comments in grey. Your comments can be different, of course!

to hit

```
; grow big for a split second, then back to normal
setscore score + 1
if score = 10 [winner]
if timevisible > 7 [settimevisible timevisible - 1]
virus,
setsize 15
wait 3
setsize 10
end
```

to move.show

```
; hide, move to a random position
; show for a short while hide again
; play area is 600 x 400
virus,
ht
setx -300 + random 601
sety -200 + random 401
st
wait timevisible
; timevisible is the value of the slider
ht
end
```

to play

```
forever [move.show]
end
```

to reset

```
; text box and slider
setscore 0
setTimeVisible 15
virus, ht
end
```

to stop.now

```
; need a stopall button to halt the game before its
normal ending
stopall
end
```

```
to winner
```

```
; 10 hits and it's game over... for the virus!  
announce [Congrats! You killed the virus 10 times!]  
stopall  
end
```

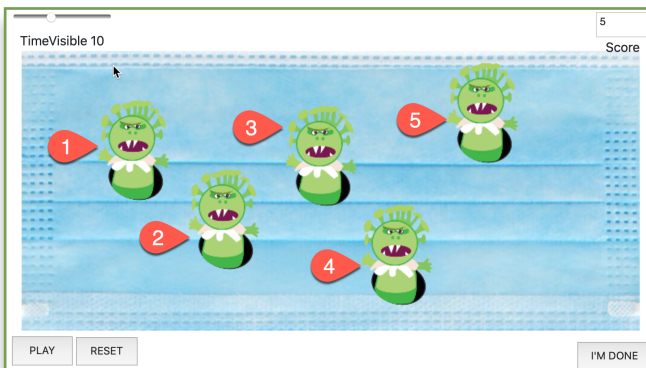
More advanced ideas

There is a totally different way to code this game. You can have five *fixed* viruses, in five different locations, *never* moving around the page. Your **play** procedure would not do **forever [move.show]**, but **forever [pick.show]**. In this example, we will keep the original turtle names (t1, t2...). **Pick.show** would be something like:

```
to pick.show  
ask pick [t1 t2 t3 t4 t5] [st wait timevisible ht]  
end
```

The trick here is **ask**. Ask needs two inputs: which turtle is being *asked*, and what it is being asked to do. You need to use square brackets **[]** around each input.

- The first input will be a random turtle-virus
pick [t1 t2 t3 t4 t5].
- The second input (**[st wait timevisible ht]**) is the “short appearance” of the turtle, same as the **move.show** procedure.



In this image, there are five turtles, named **t1** to **t5**. They will appear briefly one at a time, randomly picked by Lynx.

Project 5 - Kill the virus

You can prepare a “master” turtle-virus and then clone it 4 times. All the properties of of the master turtle-virus will transfer to the new turtle-viruses. Run this instruction 4 times:

```
clone 't1'
```

```
THE NAME OF THE TURTLE YOU WANT TO CLONE
```

Curriculum Links for Ontario

C3.1 - Solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves events influenced by a defined count and/or sub-program and other control structures

C3.2 - Read and alter existing code, including code that involves events influenced by a defined count and/or sub-program and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code