Game Programming with Scratch

For this exercise, you will start with one of the example projects that come with Scratch. In the search bar, search for projects called **FishChomp**. Select the project called "**3 FishChomp"** by **SampleProjectsTeam**. Run this project to see how it works and look at the scripts that make it work:

> *The big fish sprite follows the mouse and eats the little fish when they come in contact with his teeth. When a little fish is eaten, it disappears for a while and then reappears, as though it is a new fish. The three little fish sprites all look and behave the same way – the sprites are duplicates of one another. They move around the screen in a random, wiggly swimming motion. There is a sprite that shows the instructions on the screen.*

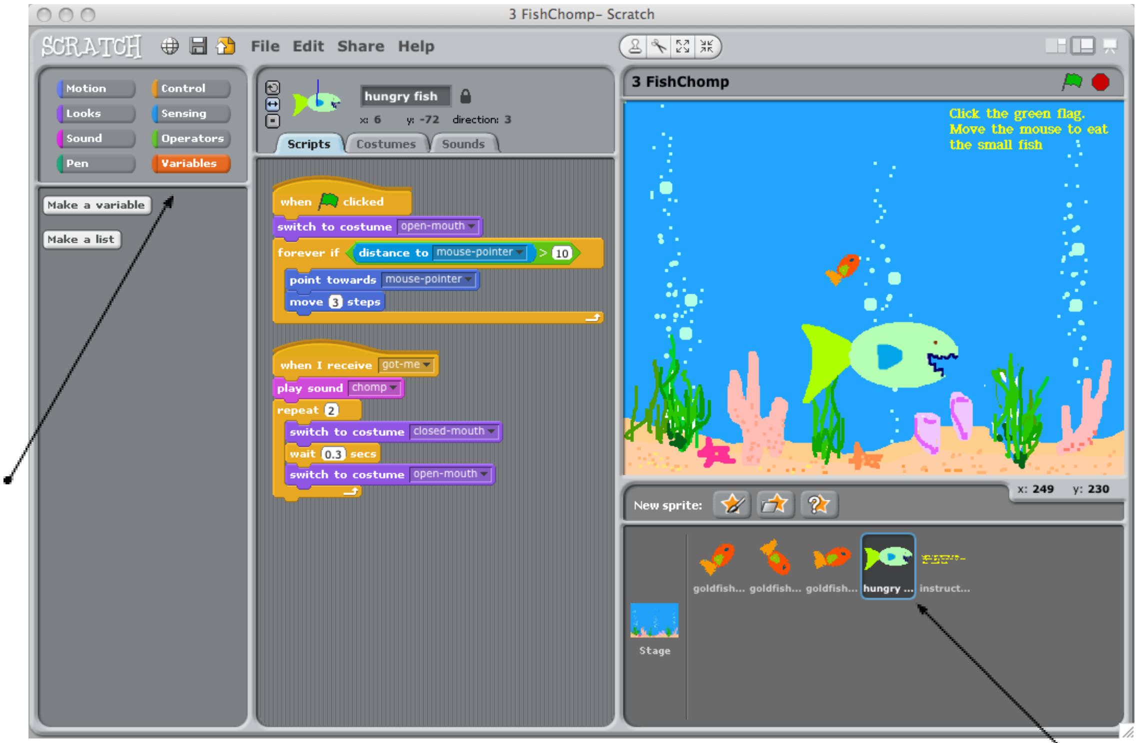
This is a very nice project, but it is missing some enjoyable components that most games have: a scoring mechanism that allows the player to win or lose, and increasing challenges. In this exercise, you will add these features, and maybe some more. To start, click the green REMIX button at the top of the FishChomp project, which will start a new project for you, based on this one.

Keeping Score

Imagine that the big fish starts the game healthy, but he has to eat regularly in order to remain healthy. So the scoring will be based on health points. The first step to setting this up is to create a *variable* called **Health**:

1. Click on the big fish in the sprite gallery at the bottom right of the Scratch screen to prepare to add scripts to this sprite.

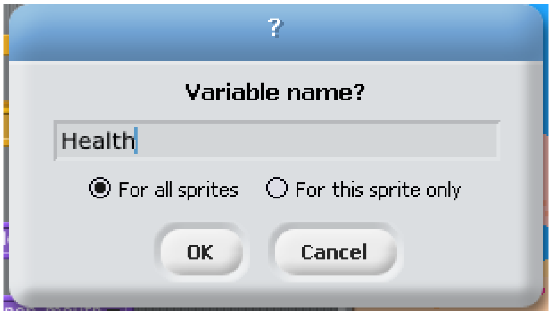
2. Click on the **Variables** button on the left side of the screen to bring up the **Variables** blocks.



```
when [flag] clicked
switch to costume [open-mouth]
forever if <(distance to [mouse-pointer]) > (10)>
    point towards [mouse-pointer]
    move (3) steps

when I receive [got-me]
play sound [chomp]
repeat (2)
    switch to costume [closed-mouth]
    wait (0.3) secs
    switch to costume [open-mouth]
```

3. Click on the **Make a variable** button and create a variable *for all sprites* called **Health**. This will cause a whole set of new blocks to appear.



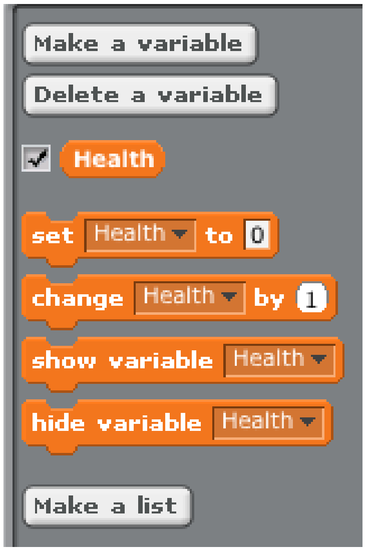
?

Variable name?

Health

● For all sprites ○ For this sprite only

OK Cancel



Make a variable

Delete a variable

☑ Health

set [Health] to (0)

change [Health] by (1)

show variable [Health]

hide variable [Health]

Make a list

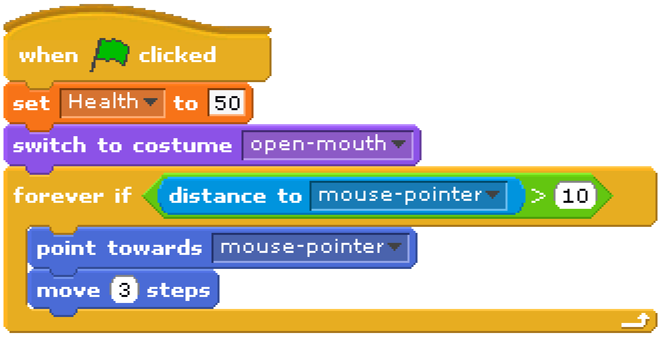
4. When the Health block has a check mark next to it, it shows on the stage. If you double-click on the variable on the stage, its display style will change. Choose one of the 3

styles (the slider bar is not appropriate for a game because it lets the player change the score). Now the player will be able to see the big fish's health status during the game.

Changing the Score and Winning/Losing the Game

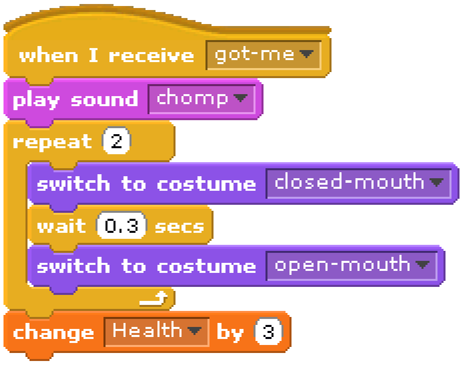
Start the big fish out with 50 health points. Have the fish earn 3 health points every time he catches and eats a little fish. Have him lose 3 health points every 3 seconds. It's easy to see that the player should lose the game when the health points reach 0, but choosing how to define a win is a *design decision* that you can make yourself. You might let the player play as long as the fish has health points, or have the player strive for a certain amount of point (maybe 70), or have them stay alive for a certain amount of time (maybe 2 minutes). You won't set a winning condition right now.

1. Add a block to the big fish's initial (flag) script to set the health points to 50 when the game starts:



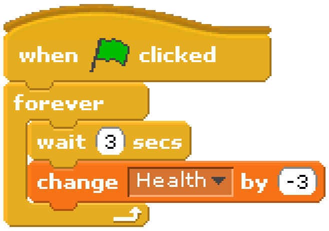
```
when 🏁 clicked
set Health ▾ to 50
switch to costume open-mouth ▾
forever if < (distance to mouse-pointer ▾) > 10 >
    point towards mouse-pointer ▾
    move 3 steps
```

2. In order to have the big fish's health improve when he eats a little fish, modify his *got-me* script by adding a block to increase the health points by 3, as shown below:



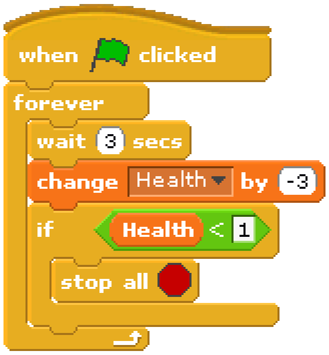
```
when I receive got-me
play sound chomp
repeat 2
    switch to costume closed-mouth
    wait 0.3 secs
    switch to costume open-mouth
change Health by 3
```

3. In order to have the big fish's health get worse when he doesn't eat, add a new script to the big fish that subtracts from the health points every 3 seconds, as shown below. This script keeps time *forever*, or as long as the game is running.



```
when clicked
forever
    wait 3 secs
    change Health by -3
```

4. To have the game end when the big fish runs out of health points, extend the script above to check whether the health points have reached 0, and if they have, to end the program by stopping all scripts. There are many possible improvements to this game ending, such as displaying a message to the player, or having the big fish show that he has died somehow, before stopping the game. Add the basic losing ending now. Get the *less-than* block from the *Operators* set of blocks, and drop the **Health** variable block in on one side and type the number ***1*** on the other side of the less-than sign.

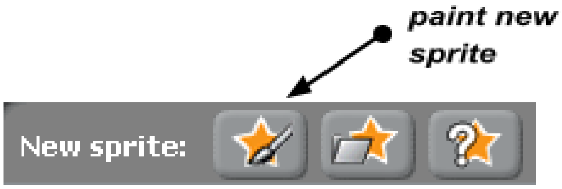


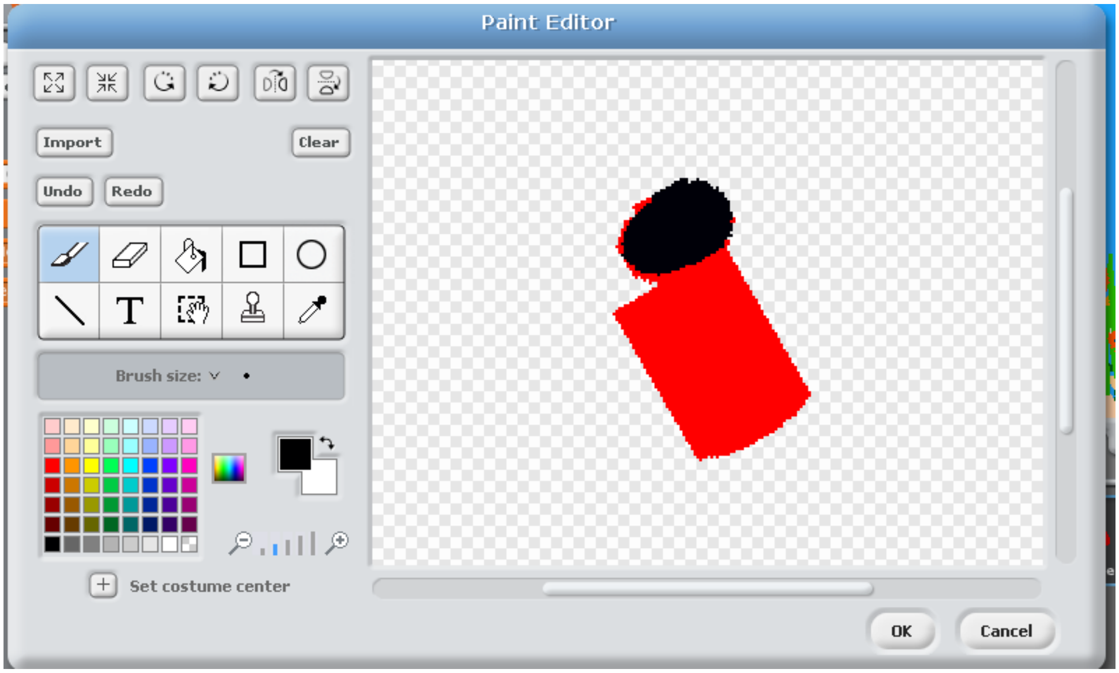
```
when clicked
forever
    wait 3 secs
    change Health by -3
    if Health < 1
        stop all
```

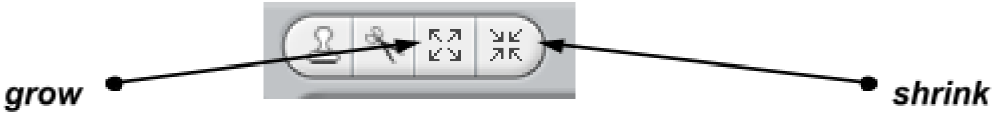
Adding a Challenge

A fish's life has many more challenges than just keeping fed, especially in today's world. You can enhance your game by adding more challenges as the player keeps the fish alive longer. Possible challenges might be avoiding fishermen, bigger fish, or pollution. We'll add the challenge of avoiding pollution. To do this, we'll create a new sprite with multiple costumes that look like various kinds of pollution, such as tin cans and plastic six-pack holder rings. This sprite will periodically fall to the bottom of the ocean, and the big fish will have to avoid it. If the big fish touches the pollution, he will lose health points.

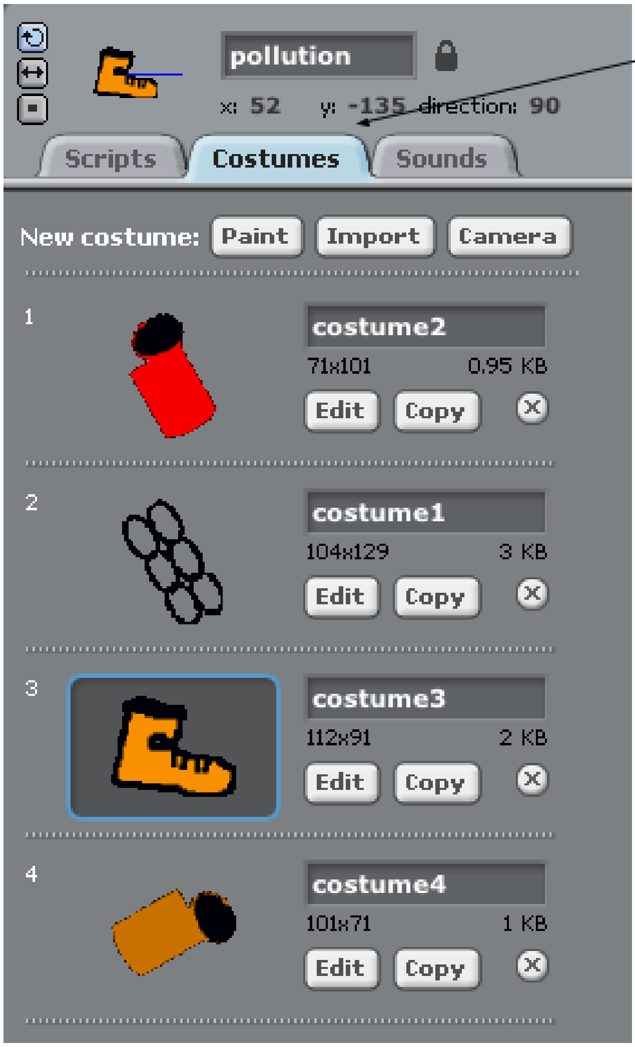
1. Create a new sprite using the *paint-new-sprite* button. In the paint window, draw a simple tin can, using the *rectangle* and *ellipse* tools to help draw the shape. Use the *rotation* buttons at top left of the paint window to orient the can slightly off-center. Once you have saved the sprite's costume, the sprite will appear on the stage and you can use the *grow* and *shrink* buttons (at the top of the Scratch window over the stage) to adjust its size to have it fit well in the scene.







2. Now add more costumes to the pollution sprite. Go to the ***Costumes*** tab for the sprite and paint a new costume that looks like a plastic six-pack holder. Copy the tin can costume and then edit it to change its color and orientation, so that it appears to be a new piece of trash. Along with six-pack holders, plastic fishing line and plastic grocery bags are some of the most common and problematic types of marine pollution. Consider adding costumes for these items, or maybe the classic old boot, as shown below:



pollution

x: 52 y: -135 direction: 90

costume2 71x101 0.95 KB

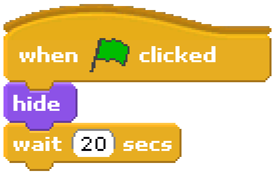
costume1 104x129 3 KB

costume3 112x91 2 KB

costume4 101x71 1 KB

Costumes tab

3. New challenges usually appear after a player has proved herself by reaching some threshold. Have the pollution challenge begin after the player has survived for a certain amount of time, say 20 seconds. Until then, the pollution sprite should be hidden. Go to the Scripts tab and add an initial script to the pollution sprite to hide it (from the *Looks* set of blocks) and then wait 20 seconds, as shown below:



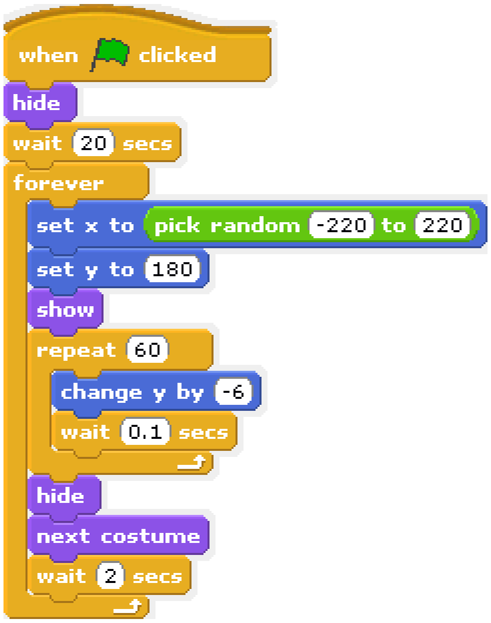
```
when clicked
hide
wait 20 secs
```

4. Now you will make the pollution drift down from the top of the stage to the ocean floor, threatening the big fish. It will do this over and over, every few seconds. It should start in a

random location each time, and it should change its costume each time, so that the one sprite appears to be many different pieces of trash. Add to your first script as shown below.

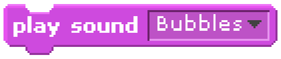
These blocks have the sprite first move to a random location (***x*** between -220 and 220) across the top (***y*** = 180) of the stage, then become visible. The random block is available from the *Operators* set. The sprite then falls little by little to the bottom of the ocean. Its ***y*** position is slowly subtracted from (by 6 each time) until it reaches -180, the bottom of the stage (6 x 60 = 360). The wait of 0.1 second slows down the sprite's fall. A larger value would make the sprite fall slower, and a smaller value would make it fall faster. Choose a value that creates the level of challenge that you want.

Once the sprite reaches the bottom, it disappears again and waits for 2 seconds, before reappearing again, wearing a different costume, and in another location. Again, choosing a different wait time between each fall will affect the game's difficulty.



```
when ⚑ clicked
hide
wait 20 secs
forever
    set x to pick random -220 to 220
    set y to 180
    show
    repeat 60
        change y by -6
        wait 0.1 secs
    hide
    next costume
    wait 2 secs
```

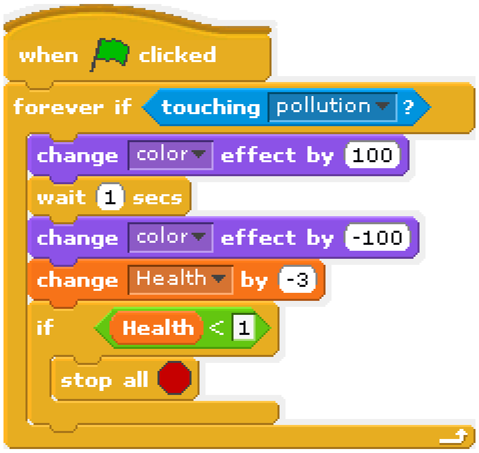
5. You might want to add an alert to the player that a piece of pollution is falling in the area by having the sprite make a sound each time it appears. To do this, go to the *Sounds* tab next to the *costumes* tab, and import one of Scratch's pre-defined sounds. Try the *Bubbles* sound, which is located in the *Effects* directory under the *Sounds* directory. Once the sound is imported, you may add a block to play the sound just before the ***show*** block in the script you created in the step above. The ***play-sound*** block comes from the *Sounds* set of blocks.



```
play sound Bubbles ▾
```

6. The final step to adding the pollution challenge is to have the big fish lose health points when he comes in contact with pollution. To do this, you will add a new script to the big fish that continuously monitors if the big fish is touching the pollution. In order to get the *touching-pollution* block shown below from the Sensing set, you must have named your sprite **pollution**. If not, just select its actual name from the list.

   Whenever the big fish touches the pollution, **3** health points will be subtracted. The player may also get a visible and/or audible alert. In the script shown below, the fish temporarily changes color, which gives the player a visible alert. Changing the color by 100 gives the fish a noticeable pinkish hue. The **1** second spent in the color change also gives the big fish time to get away from the pollution, because if the fish stays still, more points will be subtracted as long as he is still touching the trash. It is good to check if the game is over every time points are subtracted.



```
when ⚑ clicked
forever if touching pollution ?
    change color effect by 100
    wait 1 secs
    change color effect by -100
    change Health by -3
    if Health < 1
        stop all
```

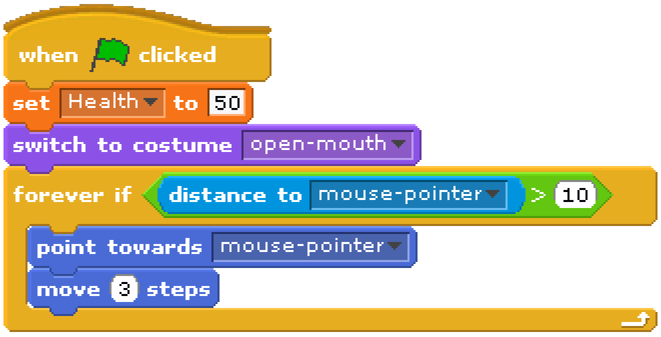
Taking the Game Further

Now you have created a simple, but complete game, where the player must have his fish eat to live. Once the player has succeeded in keeping the fish alive for some time, a new challenge appears: avoiding pollution. This game might be extended and improved in many ways; some examples include:

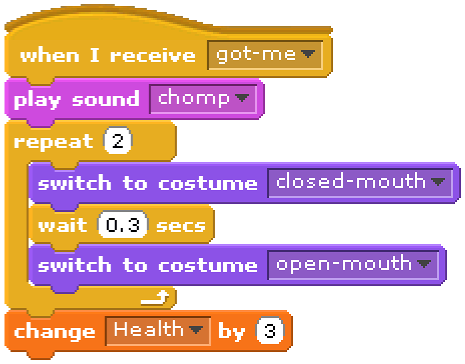
- Making the pollution challenge more difficult as the player keeps the fish alive longer by adding more pollution sprites (falling in different parts of the stage at the same time)
- Having the pollution fall faster or more often as time progresses
- Adding new challenges, such as fishermen or bigger predators (sharks), to the game
- More sound and graphics effects could be added to make the game more attractive

- The lose and win states could be made more interesting
- Moving the help information from always being on the stage to a separate screen that explains the whole game at the beginning

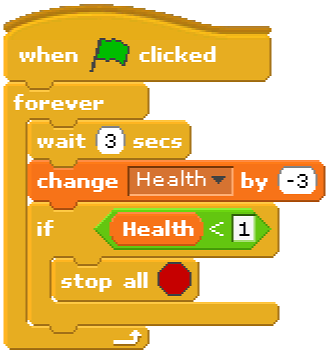
The final scripts for the big fish sprite



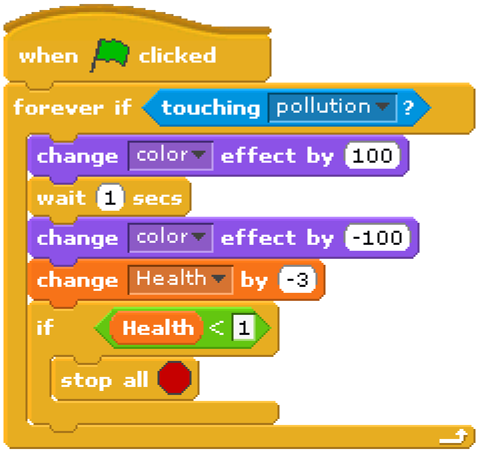
```
when ⚑ clicked
set Health ▾ to 50
switch to costume open-mouth ▾
forever if < distance to mouse-pointer ▾ > 10 >
    point towards mouse-pointer ▾
    move 3 steps
```



```
when I receive got-me ▾
play sound chomp ▾
repeat 2
    switch to costume closed-mouth ▾
    wait 0.3 secs
    switch to costume open-mouth ▾
change Health ▾ by 3
```

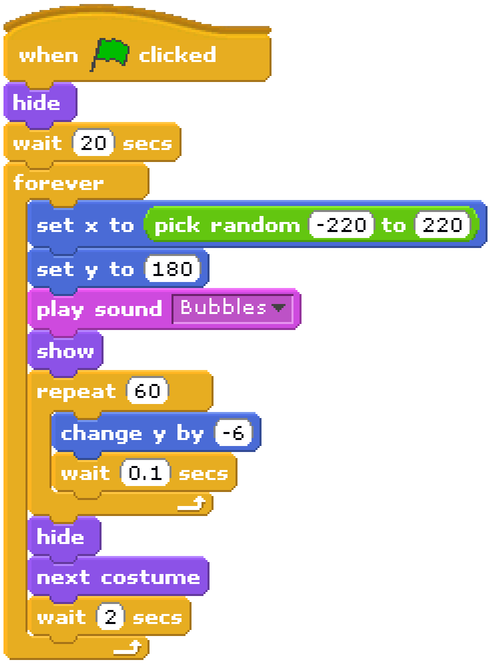


```
when ⚑ clicked
forever
    wait 3 secs
    change Health ▾ by -3
    if < Health < 1 >
        stop all 🛑
```



```
when ⚑ clicked
forever if < touching pollution ▾ ? >
    change color ▾ effect by 100
    wait 1 secs
    change color ▾ effect by -100
    change Health ▾ by -3
    if < Health < 1 >
        stop all 🛑
```

The final script for the pollution sprite



```
when ⚑ clicked
hide
wait (20) secs
forever
    set x to (pick random (-220) to (220))
    set y to (180)
    play sound [Bubbles ▼]
    show
    repeat (60)
        change y by (-6)
        wait (0.1) secs
    end
    hide
    next costume
    wait (2) secs
end
```