1. Go to **shadertoy.com/new**
2. Alter the code in the code window to look like this:

```
void mainImage( out vec4 color, in vec2 U ){
// "U" are the coordinates of the pixel that will be "color"
   color = vec4(1,0,0,1);
}
```

   Color = vec4 (red, green, blue, transparency);

   Before you hit run can you predict what will happen?
3. Now alter the last line to look like this:

```
   color = vec4( U.x/iResolution.x );
```

   Before you hit run can you predict what will happen?
4. What about this?

```
   color = vec4( U/iResolution.xy, 0, 1 );
```

5. Now add a texture to your program by clicking the box below the code window labeled "iChannel0". Click the tab called "textures" and pick out an image to use.
6. Update the code to draw the image:

```
   color = texture(iChannel0, U/iResolution.xy );
```

7. Click the "**+**" icon at the top left of the code window. Click on "Buffer A". Then, copy paste the code from the "Image" tab to replace the code in "Buffer A".
8. Click on "iChannel0" in "Buffer A" and select "Buffer A". Do the same thing in "Image". When you run you should see a blue or black screen.
9. Now in "Buffer A" add a line of code that lets you paint with the mouse:

```
void mainImage( out vec4 color, in vec2 U )
{
   color = texture(iChannel0,U/iResolution.xy);
   if (iMouse.z>0. && length(iMouse.xy-U) < 10.) color.x = 1.;
}
```

   Did it work? Can you paint red with your mouse?
10. And one last step before we get to the really fun stuff. We have to calculate the local average. That means we add up the pixels that are up, down, left, and right and then divide by 4.

```
   vec4 average = (
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);
```

Differential Equations are not scary! They're your friend :)

Here is the first one we're going to learn :

Diffusion equation :

$$\frac{d}{dt}\varphi = \frac{d^2}{dx^2}\varphi$$

I agree, that makes no sense! But it has a lot of meaning. That means that the change in our field in time is equal to the difference between the field and the local average. We can write that! So in "Buffer A" update your code to diffuse the red paint!

```
void mainImage( out vec4 color, in vec2 U )
{
   vec4 average = (
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);

   color.x += average.x - color.x;

   if (iMouse.z>0. && length(iMouse.xy-U) < 10.) color.x = 1.;
}
```

This is how heat diffuses and this is how particles of fart make it to your nose. This is how many many things in nature occur!

Does the red paint decay over space? It should!

Wave Equation

$$\frac{d^2}{dt^2}\varphi = \frac{d^2}{dx^2}\varphi$$

Can you identify the difference between this equation and the diffusion equation?

The d^2/dt^2 means that we not only have to track the value of the field, but also how fast it's changing! This is the concept of momentum! So what this equation is saying is that the acceleration of the field is equal to the difference between the field and the local average.

This corresponds to using another variable - one for how much red there is, and one for how fast that red is changing.

```
void mainImage( out vec4 color, in vec2 U )
{
   vec4 average = (
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);

   color.y += average.x - color.x;
   color.x += color.y;

   if (iMouse.z>0. && length(iMouse.xy-U) < 10.) color.x = 1.;
}
```

Does it make waves? Make sure you use the "x (red)" and "y (green)" channels correctly.

Quantum Wave Equation

$$\frac{d^2}{dt^2}\varphi = \frac{d^2}{dx^2}\varphi - \varphi$$

What is the difference between the quantum wave equation and the regular wave equation?

This time, the wave interacts with itself!

```
void mainImage( out vec4 color, in vec2 U )
{
   vec4 average = (
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);

   color.y += average.x - color.x - color.x;
   color.x += color.y;

   if (iMouse.z>0. && length(iMouse.xy-U) < 10.) color.x = 1.;
}
```

Does it make crazy wavy patterns? It should

Mass Wave Equation

$$\frac{d^2}{dt^2}\varphi = \frac{d^2}{dx^2}\varphi - \frac{\varphi}{|\varphi|}$$

What is the difference between the mass wave equation and the quantum wave equation?

This time, the wave interacts with the direction of the field!

```
void mainImage( out vec4 color, in vec2 U )
{
   vec4 average = (
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);

   color.y += average.x - color.x - sign(color.x);
   color.x += color.y;

   if (iMouse.z>0. && length(iMouse.xy-U) < 10.) color.x = 1.;
}
```

You will also want to adjust the “Image” tab, otherwise the values will be too high and you won’t see what’s going on:

```
void mainImage( out vec4 color, in vec2 U )
{
   color = texture(iChannel0,U/iResolution.xy);
   color = vec4(.1*log(1.+color.x*color.x));
}
```

Does it look like a bunch of stuff flying around?

Electromagnetic Wave Equation

$$\frac{d^2}{dt^2}P = \frac{d^2}{dx^2}P - \frac{P}{\sqrt{P\cdot P+N\cdot N}}$$
$$\frac{d^2}{dt^2}N = \frac{d^2}{dx^2}N - \frac{N}{\sqrt{P\cdot P+N\cdot N}}$$

What is the difference between the electromagnetic wave equation and the mass wave equation?
We have to track two systems now. P is for positive and N is for negative → positive particles interact with both positive and negative particles! This time, the wave interacts with the direction of both the positive and negative fields!

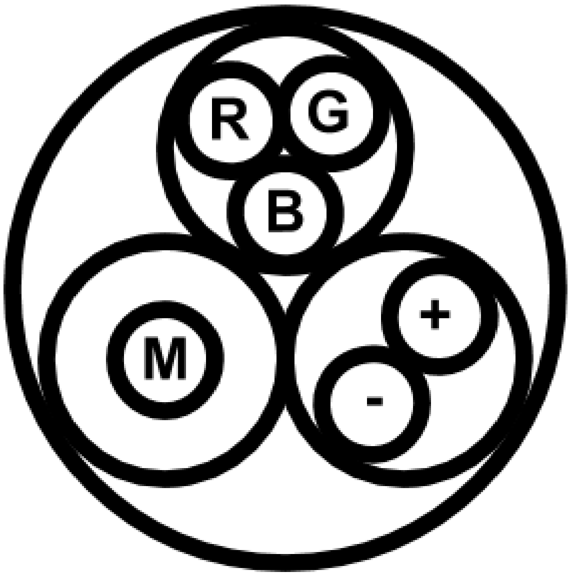
```
void mainImage( out vec4 color, in vec2 U )
{
   vec4 average = (
       texture(iChannel0,(U+vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U+vec2(0,1))/iResolution.xy)+
       texture(iChannel0,(U-vec2(1,0))/iResolution.xy)+
       texture(iChannel0,(U-vec2(0,1))/iResolution.xy))/4.;
   color = texture(iChannel0,U/iResolution.xy);

   color.y += average.x - color.x;
   color.w += average.z - color.z;
   if (length(color.xz) > 0.) color.yw -= color.xz/length(color.xz);
   color.x += color.y;
   color.z += color.w;

   if (iMouse.z>0. && length(iMouse.xy-U) < 10.)
        color.xz = vec2(cos(iTime),sin(iTime));
}
```

You will also want to adjust the “Image” tab, otherwise the values will be too high and you won’t see what’s going on:

```
void mainImage( out vec4 color, in vec2 U ){
   vec4 a = texture(iChannel0,U/iResolution.xy);
   color.x = (.1*log(1.+a.x*a.x));
   color.y = 0.;
   color.z = (.1*log(1.+a.z*a.z));
}
```

Does it look like the red (positive) and blue (negative) dance?

Unity Wave Equation



$$\frac{d^2}{dt^2}\Psi = \frac{d^2}{dx^2}\Psi + K\left(\frac{\Psi}{N|SuperGroup|} - \frac{n\Psi}{N|SubGroup|}\right)$$

$$\frac{d^2}{dt^2}M = \frac{d^2}{dx^2}M + K\left(\frac{M}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{1M}{6\sqrt{M\cdot M}}\right)$$

$$\frac{d^2}{dt^2}P = \frac{d^2}{dx^2}P + K\left(\frac{P}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{2P}{6\sqrt{P\cdot P+N\cdot N}}\right)$$
$$\frac{d^2}{dt^2}N = \frac{d^2}{dx^2}N + K\left(\frac{N}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{2N}{6\sqrt{P\cdot P+N\cdot N}}\right)$$

$$\frac{d^2}{dt^2}R = \frac{d^2}{dx^2}R + K\left(\frac{R}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{3R}{6\sqrt{R\cdot R+G\cdot G+B\cdot B}}\right)$$
$$\frac{d^2}{dt^2}G = \frac{d^2}{dx^2}G + K\left(\frac{G}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{3G}{6\sqrt{R\cdot R+G\cdot G+B\cdot B}}\right)$$
$$\frac{d^2}{dt^2}B = \frac{d^2}{dx^2}B + K\left(\frac{B}{6\sqrt{M\cdot M+P\cdot P+N\cdot N+R\cdot R+G\cdot G+B\cdot B}} - \frac{3B}{6\sqrt{R\cdot R+G\cdot G+B\cdot B}}\right)$$

Demonstration :
https://www.shadertoy.com/view/wllSRr