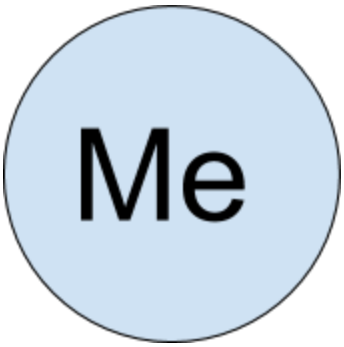
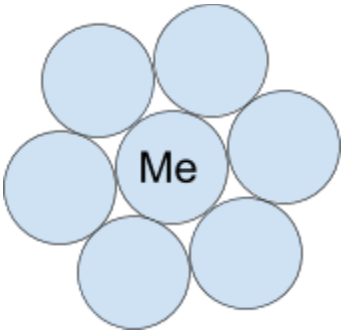
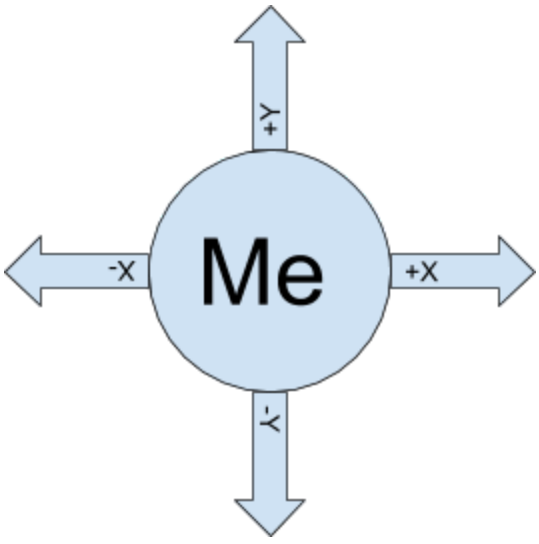
Hello, I’m Me.



I live in a neighborhood.



We are FREE



I am Energy in a bubble.

Energy

My Energy can be **ordered** and move Me.

X + **Y** (+ *Z*)

Or **disordered** and evaporate.

B

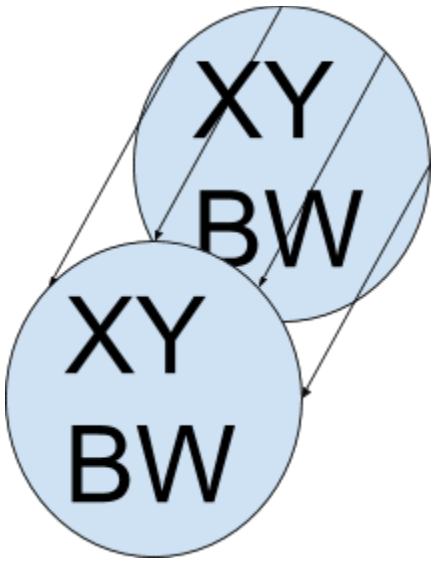
Or it can be **internal** and follow Me.

W

I have **1 2 3 4** rules.

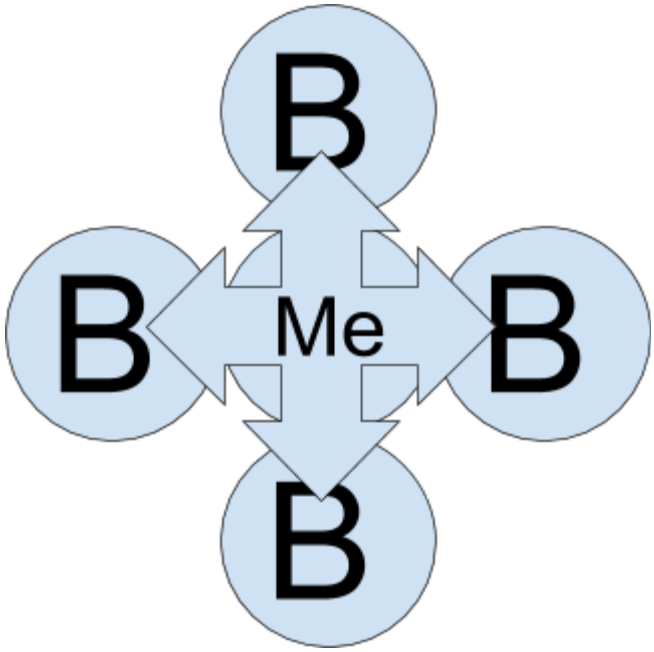
1

All My Energy moves with Me.



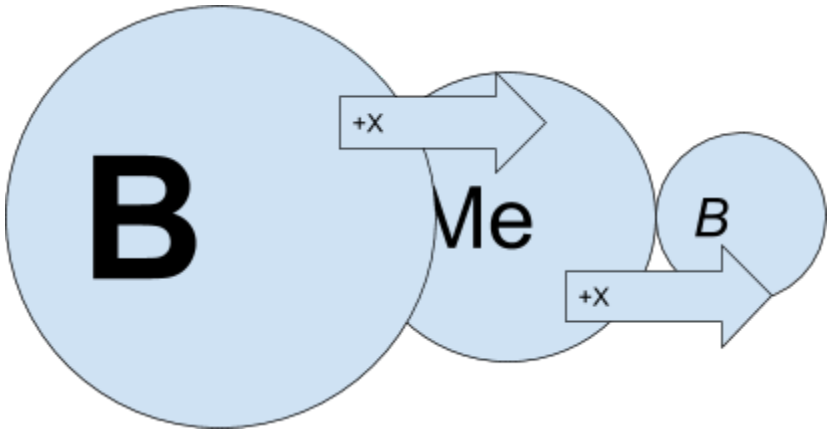
2

Disorder is always exchanging completely.
It is extremely volatile.



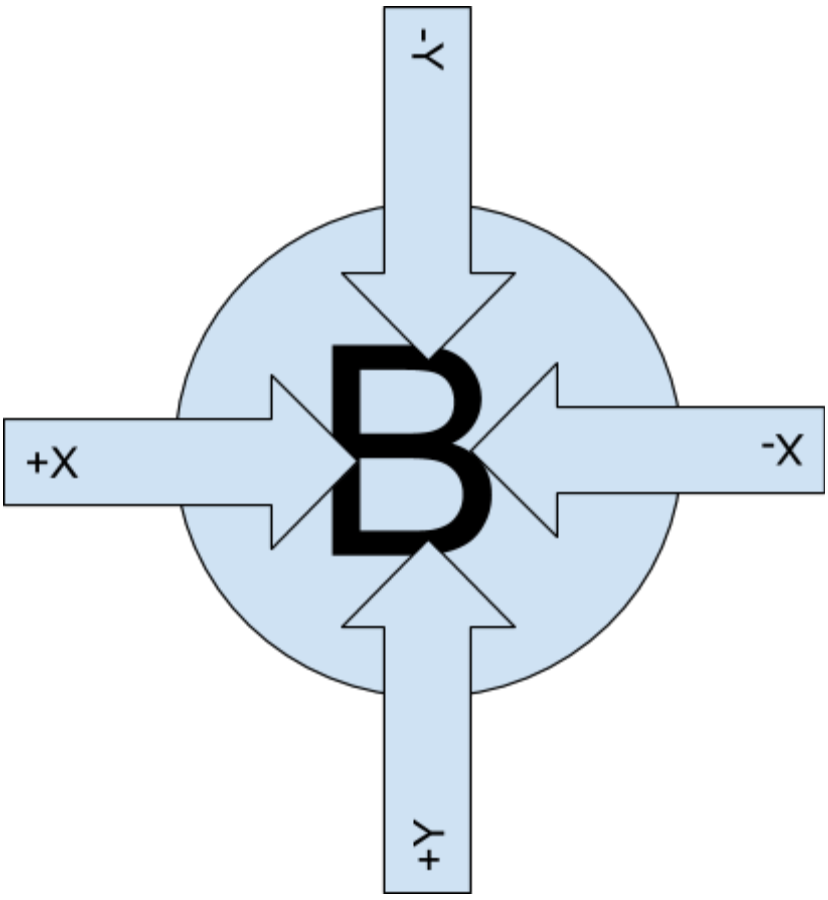
3

The ***order*** of the disorder around Me gives Me ***order***.



4

The ***disorder*** of the order around Me gives Me ***disorder***.



Go to Shadertoy.com/new

And prepare a field simulation :

You should be able to draw white on the screen using the **W** channel. This will be the internal Energy.

`Energy.xy` = **X+Y** : Ordered Energy
`Energy.b` = **B** : Disordered Energy
`Energy.w` = **W** : Internal Energy

Apply My first rule.

1 All My Energy follows Me.

If I look to the past to find My Energy,
I should find My Energy not where it is,
But where it was.
$MeNow_{xy} = Me_{xy} + \mathbf{XY}$
$MeThen_{xy} = MeNow_{xy} - \mathbf{XY}Then$

Where I was is where I am now **minus** how fast I was going.

I don't know any of that for sure, but I can guess :

```
vec4 Field (vec2 position) {
    vec2 velocityGuess = LOOKUP (position).xy;
    vec2 positionGuess = position - velocityGuess;
    return LOOKUP (positionGuess);
}
```

(There are better ways to do this)

Apply My second rule

2 Free Energy diffuses at the speed of information.

All of My disordered Energy **B** comes from My neighborhood.

Exchange disorder symmetrically in all directions :

```
Energy.b = (pX.b + pY.b + nX.b + nY.b)/4.0;
```

Apply My third rule

3 The change in volatile Energy **B** across Me will push Me in that direction.

d/dt **X** = -d/d**x B**
d/dt **Y** = -d/d**y B**

Sum the force from each neighbor :

```
vec2 Force;
Force.x = nX.b - pX.b;
Force.y = nY.b - pY.b;
```

Add the average force from all the neighbors to the ordered Energy :

```
Energy.xy += Force/4.0;
```

Apply My fourth rule

4 The **disorder** in the order around Me enters Me as **disorder**.

d/dt **B** = -(d/dx **X** + d/dy **Y**)

B changes by the amount the neighborhood converges on Me.

Add the average convergence to My disordered Energy.

```
Energy.b += (nX.x - pX.x + nY.y - pY.y)/4.;
```

Last Steps!

Maybe the internal Energy **W** is mass.
And maybe there's some gravity.

d/dt **Y** = -g**W**

Add the force of gravity to the **Y** Energy

```
Energy.y -= Energy.w/200.0;
```

Boundary Conditions

Let's put our fluid in a box and force the velocity at the edges to be zero.

```
if (Me.x<1. || Me.y<1. || iResolution.x-Me.x<1. || iResolution.y-Me.y<1.)
{
    Energy.xy *= 0.0;
}
```

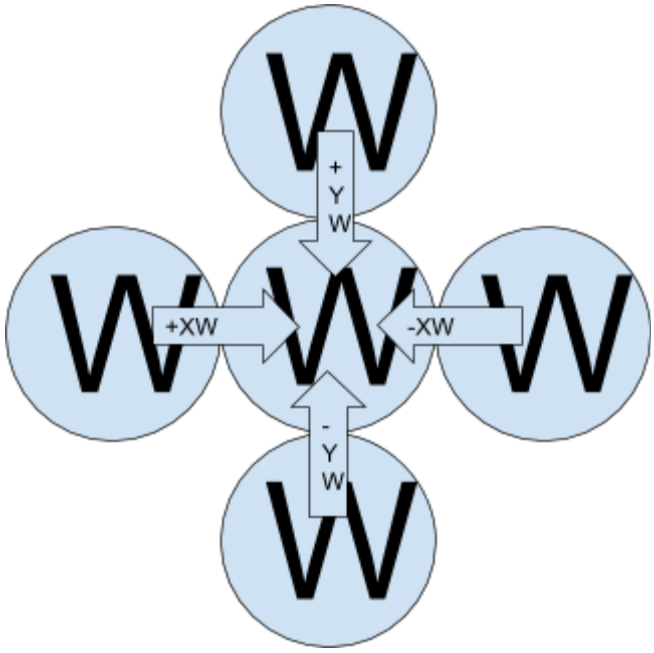
At this point your fluid should work!

If your code doesn't work, maybe try consulting this link with the correct code:
Shadertoy.com/view/WtsSz2

Mass Conservation

My **rules** will resist changes in Energy density, but sometimes I do exchange internal Energy **W** with My neighbors and it is important to account for that.

d/dt **W** = -(d/dx (**XW**) + d/dy (**YW**))



Add the average exchange of mass with each neighbor :

```
Energy.w += (nX.x*nX.w-pX.x*pX.w+nY.y*nY.w-pY.y*pY.w)/4.;
```