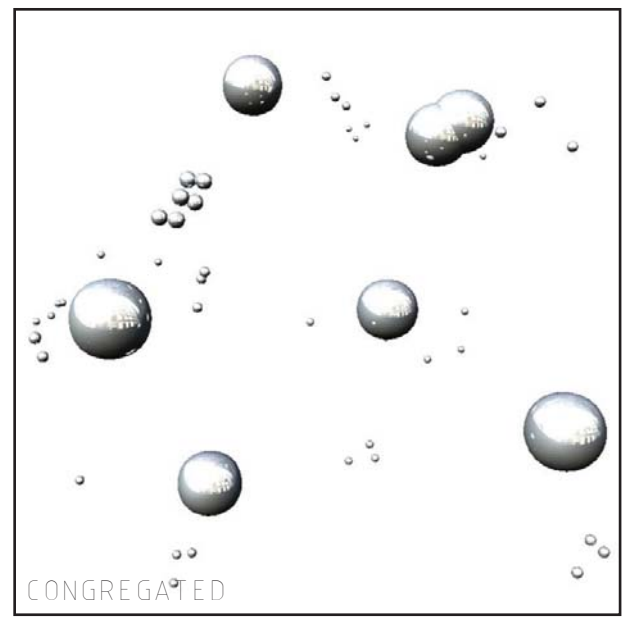
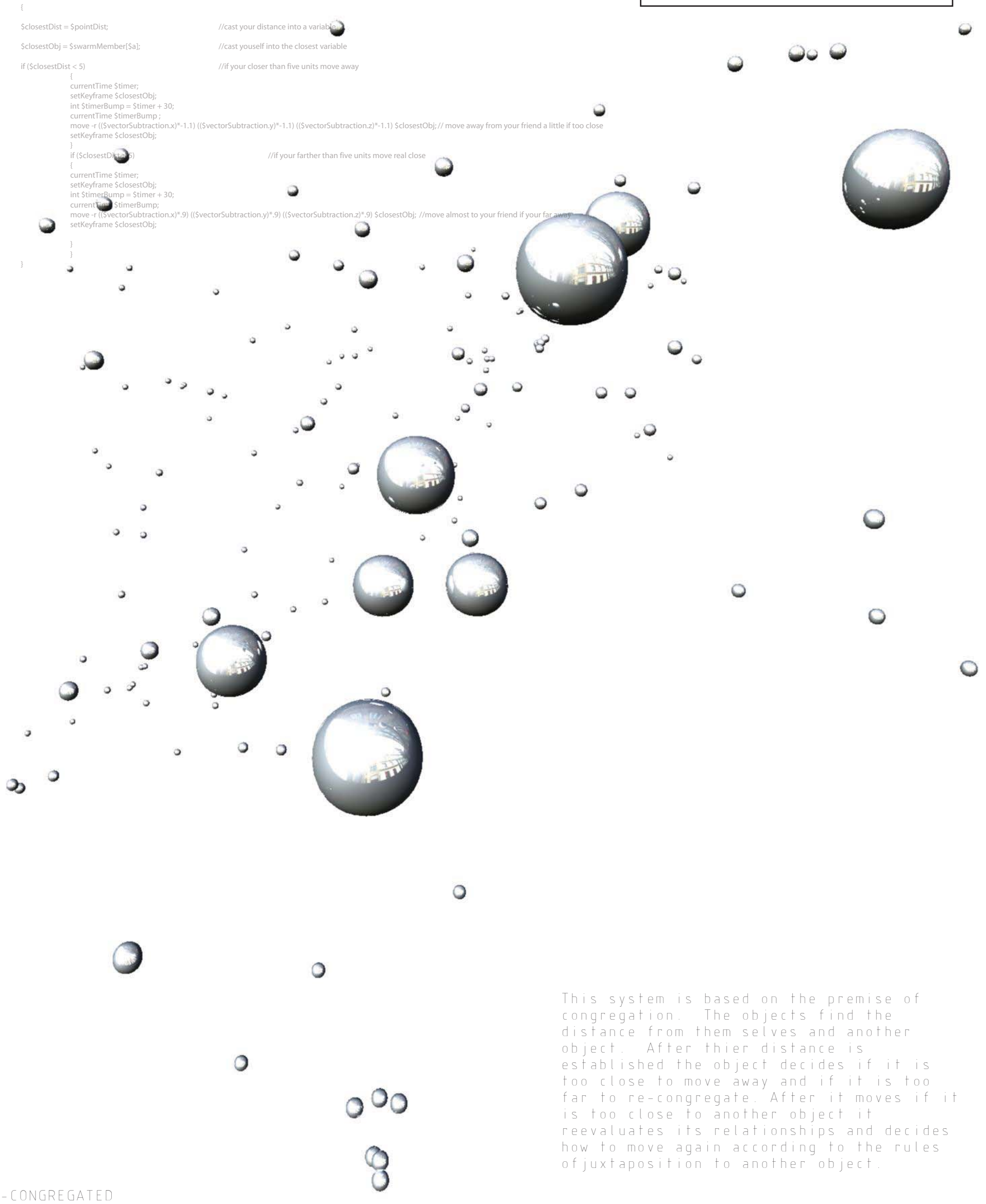
```
//Author:       David_Wolfe
//Version:      v4.0
//working for all versions
//info:         trying to swarm_add color


float $closestDist = 9999999;                       \\declaring variables outside the loop for access
string $closestObj  ;                               \\declaring variables outside the loop for access
vector $vectorSubtraction ;                         \\declaring variables outside the loop for access
$swarmMember[]  = `ls -sl`;                         \\making the swarm an array
int $arraySize = size($swarmMember);                \\sizing the array for number size
for ($i=0; $i <$arraySize; $i++)                    \\opening the for loop

{

int $timer = $i*4;                                          //counter for keyframing
float $selCenter[] = `exactWorldBoundingBox $swarmMember[$i]`;                          //find center of original swarm member
vector $selVector = <<(($selCenter [0] + $selCenter [3])*.5), (($selCenter [1] + $selCenter   [4])*.5), (($selCenter [2] + $selCenter [5])*.5)>>;// puts orignal swarm member into vector
$closestDist = 9999999;                             //referance for closest dist to start with

    for ($a=0; $a < $arraySize; $a++) //find proxity of friends

      {
      float $testCenter[] = `exactWorldBoundingBox $swarmMember[$a] `;
      vector $testVector = <<(($testCenter[0] + $testCenter[3])*.5), (($testCenter[1] + $testCenter[4])*.5), (($testCenter[2] + $testCenter[5])*.5)>>; //find position and casts into a vector

        $vectorSubtraction = $selVector - $testVector;      //finds the difference in the vectors

        float $pointDist = mag($vectorSubtraction);                 //finds the magnitude
              if ($pointDist < $closestDist && $pointDist != 0)    //if you find a closer friend but not yourself

              {

              $closestDist = $pointDist;                            //cast your distance into a variable

              $closestObj = $swarmMember[$a];                       //cast youself into the closest variable

              if ($closestDist < 5)                                 //if your closer than five units move away
                          {
                          currentTime $timer;
                          setKeyframe $closestObj;
                          int $timerBump = $timer + 30;
                          currentTime $timerBump ;
                          move -r (($vectorSubtraction.x)*-1.1) (($vectorSubtraction.y)*-1.1) (($vectorSubtraction.z)*-1.1) $closestObj; // move away from your friend a little if too close
                          setKeyframe $closestObj;
                          }
                          if ($closestDist > 6)                                 //if your farther than five units move real close
                          {
                          currentTime $timer;
                          setKeyframe $closestObj;
                          int $timerBump = $timer + 30;
                          currentTime $timerBump;
                          move -r (($vectorSubtraction.x)*.9) (($vectorSubtraction.y)*.9) (($vectorSubtraction.z)*.9) $closestObj;  //move almost to your friend if your far away
                          setKeyframe $closestObj;

                          }
                          }
              }
}
```





This system is based on the premise of congregation. The objects find the distance from them selves and another object. After thier distance is established the object decides if it is too close to move away and if it is too far to re-congregate. After it moves if it is too close to another object it reevaluates its relationships and decides how to move again according to the rules ofjuxtaposition to another object.