

Boids

Grupo 01

Horacio Gomez ($L : 50825$) Juan Pablo Orsay ($L : 49373$)

Segundo Semestre 2018

72.25 Simulación de Sistemas
Instituto Tecnológico de Buenos Aires



Índice

Fundamentos

Algoritmo

Reglas básicas

Reglas extendidas

Implementación

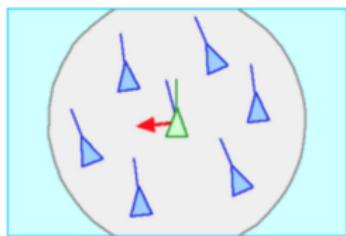
Código

Resultados

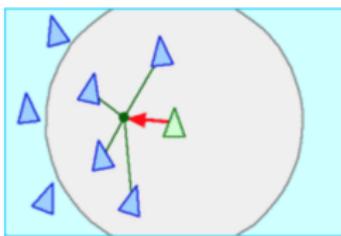
Conclusiones



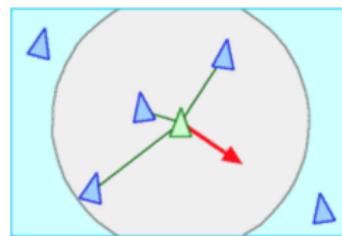
Boids: Reglas



Alineamiento



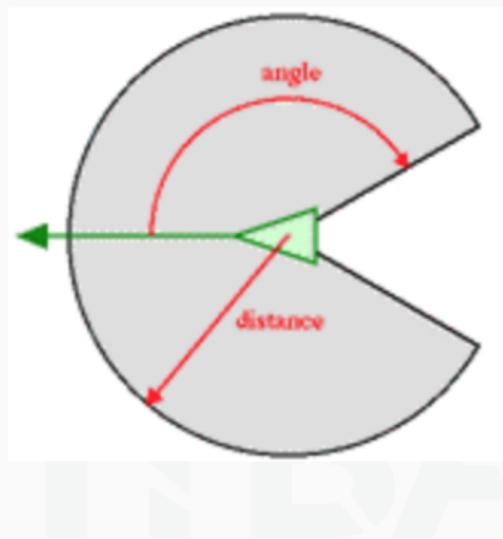
Cohesión



Separación

Boids: Vecinos

- Debe estar a menos de *distance*
- Debe estar posicionado a menos de *angle* del vector velocidad.



Boids: Algoritmo

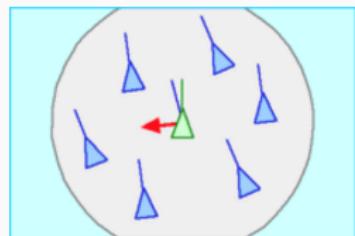
Algoritmo 1 Boids

```
1: procedure Step(rules, factors, boids, dT)
2:   for boid  $\leftarrow$  boids do
3:     v = boid.velocity
4:     for rule  $\leftarrow$  rules do
5:       v += factori * rule(boid, boids)
6:     end for
7:     if magnitude(v) > MAX_SPEED then
8:       v = v/magnitude(v) * MAX_SPEED
9:     end if
10:    boid.velocity = v
11:    boid.position += boid.velocity * dT
12:  end for
13: end procedure
```

Regla: Alineamiento

Algoritmo 2 Alineamiento

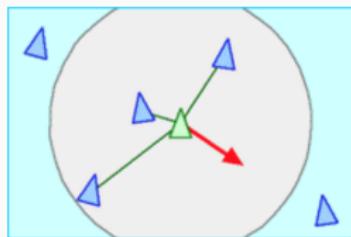
```
1: procedure Alineamiento(boid, boids)
2:    $v_{out} = \{0, 0, 0\}$ 
3:   for neighbour  $\leftarrow$  boids do
4:      $v_{out} += \text{neighbour.velocity}$ 
5:   end for
6:    $v_{out} = v_{out}/\text{length(neighbours)}$   $\triangleright$  Velocidad promedio de vecinos
7:    $v_{out} = v_{out} - \text{boid.velocity}$ 
8:   return  $v_{out}$ 
9: end procedure
```



Regla: Separación

Algoritmo 4 Separación

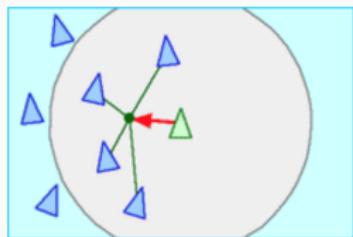
```
1: procedure Separación(boid, boids)
2:    $v_{out} = \{0, 0, 0\}$ 
3:   for neighbour  $\leftarrow$  boids do
4:     if distance(neighbour, boid) > unsafeDistance then
5:       continue
6:     end if
7:      $v_{out} -= boid.position - neighbour.position$ 
8:   end for
9:    $v_{out} = v_{out}/length(neighbours)$      $\triangleright$  Posición promedio de vecinos
10:   $v_{out} = v_{out} - boid.position$ 
11:  return  $v_{out}$ 
12: end procedure
```



Regla: Cohesión

Algoritmo 3 Cohesión

```
1: procedure Cohesión(boid, boids)
2:   vout = {0, 0, 0}
3:   for neighbour  $\leftarrow$  boids do
4:     vout += neighbour.position
5:   end for
6:   vout = vout/length(neighbours)     $\triangleright$  Posición promedio de vecinos
7:   vout = vout - boid.position
8:   return vout
9: end procedure
```



Algoritmo 5 TendencyTo

```
1: procedure TendencyTo(boid, neighbours, interactWith)
2:    $v_{out} = \{0, 0, 0\}$ 
3:   for neighbour  $\leftarrow$  neighbours do
4:     if neighbour.type != interactWith then
5:       continue
6:     end if
7:      $v_{out} += (\text{neighbour.position} - \text{boid.position})/10$ 
8:   end for
9: end procedure
```



Algoritmo 6 Boundary

```
1: procedure Boundary(boid, universe)
2:    $x_{min} \leftarrow \text{universe.metadata.boundaries.minx}$ 
3:    $x_{max} \leftarrow \text{universe.metadata.boundaries.maxx}$ 
4:    $y_{min} \leftarrow \text{universe.metadata.boundaries.miny}$ 
5:    $y_{max} \leftarrow \text{universe.metadata.boundaries.maxy}$ 
6:    $z_{min} \leftarrow \text{universe.metadata.boundaries.minz}$ 
7:    $z_{max} \leftarrow \text{universe.metadata.boundaries.maxz}$ 
8:   speed  $\leftarrow \text{CONST}$ 
9:    $v \leftarrow \{0, 0, 0\}$ 
10:  if boid.x <  $x_{min}$  then
11:     $v_x = \text{speed}$ 
12:  end if
13:  if boid.x >  $x_{max}$  then
14:     $v_x = -\text{speed}$ 
15:  end if
16:  if boid.y <  $y_{min}$  then
17:     $v_y = \text{speed}$ 
18:  end if
19:  if boid.y >  $y_{max}$  then
20:     $v_y = -\text{speed}$ 
21:  end if
22:  if boid.z <  $z_{min}$  then
23:     $v_z = \text{speed}$ 
24:  end if
25:  if boid.z >  $z_{max}$  then
26:     $v_z = -\text{speed}$ 
27:  end if
28:  return v
29: end procedure
```

Índice

Fundamentos

Algoritmo

Reglas básicas

Reglas extendidas

Implementación

Código

Resultados

Conclusiones



Implementación

Lenguaje

Kotlin

Visualización

Ovito

Avance de simulación y animación

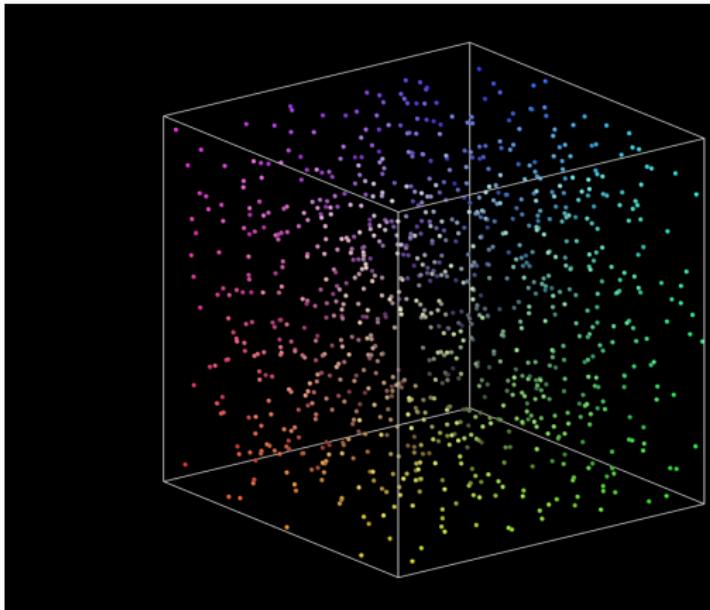
$$\Delta t = \frac{1}{60} \text{ (s)}$$



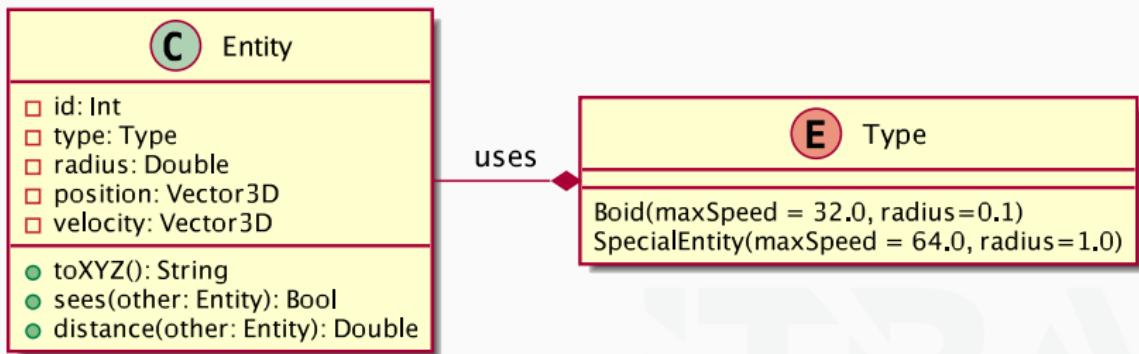
Visualización: Color

Color depende de la posición:

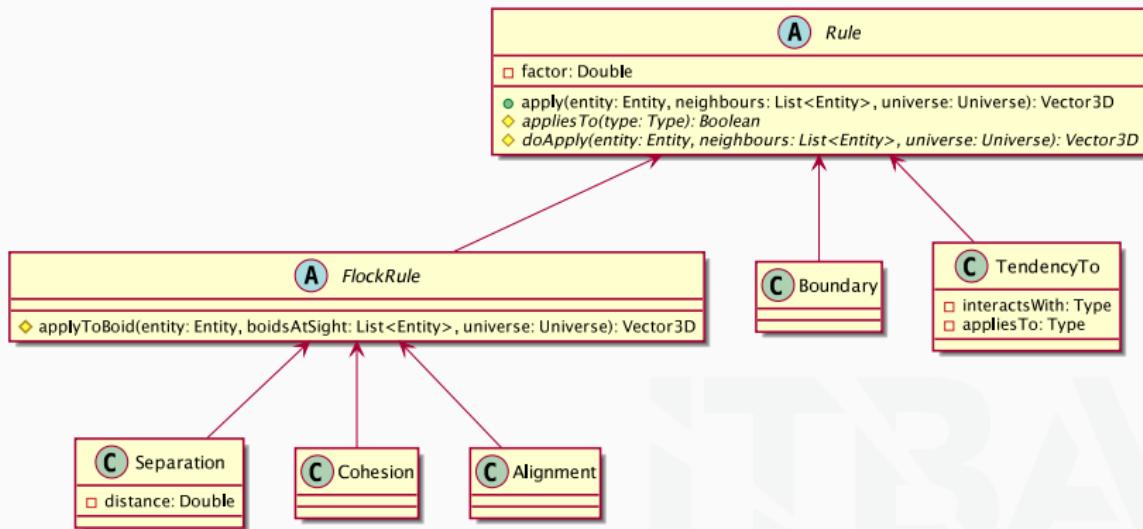
- $\text{R} = 0,2 + (position.\text{x}/width) * 0,8$
- $\text{G} = 0,2 + (position.\text{y}/height) * 0,8$
- $\text{B} = 0,2 + (position.\text{z}/depth) * 0,8$



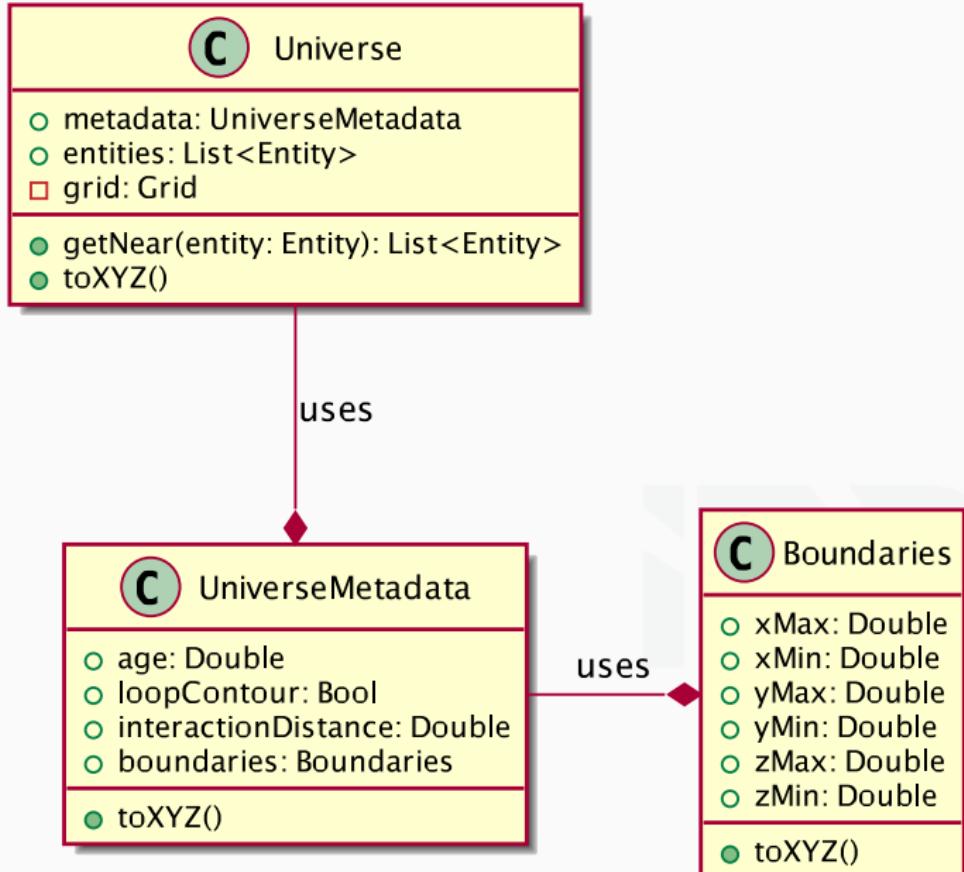
Entidades

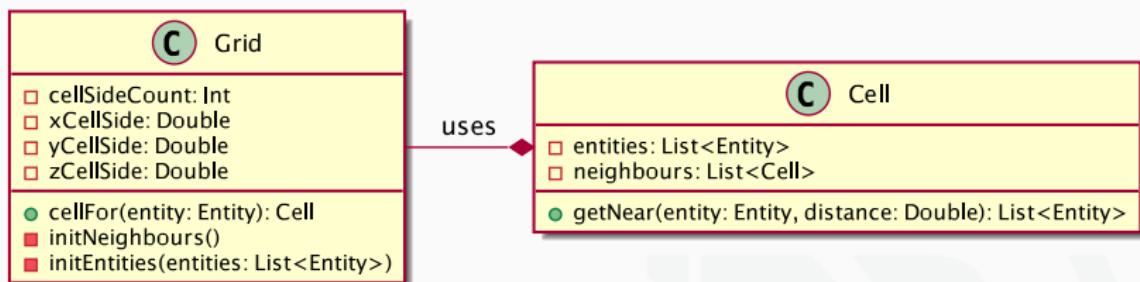


Reglas



Universo







Simulation

- ❑ universe: Universe
 - ❑ rules: List<Rule>
 - ❑ dT: Double
-
- step(): Universe
 - clampVelocity(velocity: Vector3D, maxSpeed: Double): Vector3D
 - loopPosition(position: Vector3D): Vector3D
 - loopValue(value: Double, max: Double): Double

Índice

Fundamentos

Algoritmo

Reglas básicas

Reglas extendidas

Implementación

Código

Resultados

Conclusiones



Parámetros de simulación

Tamaño del universo

32m x 32m x 32m

Condiciones de contorno

habilitadas

Cantidad de boids

1024

Cantidad de especiales

0

Distancia de interacción

2m

Parámetros de simulación

Radio: boids

0.1m

Radio: especiales

1.0m

Velocidad máxima: boids

32 m/s

Velocidad máxima: especiales

32 m/s

Reglas: Factores

Alineamiento

0.25

Cohesión

0.25

Separación

0.25

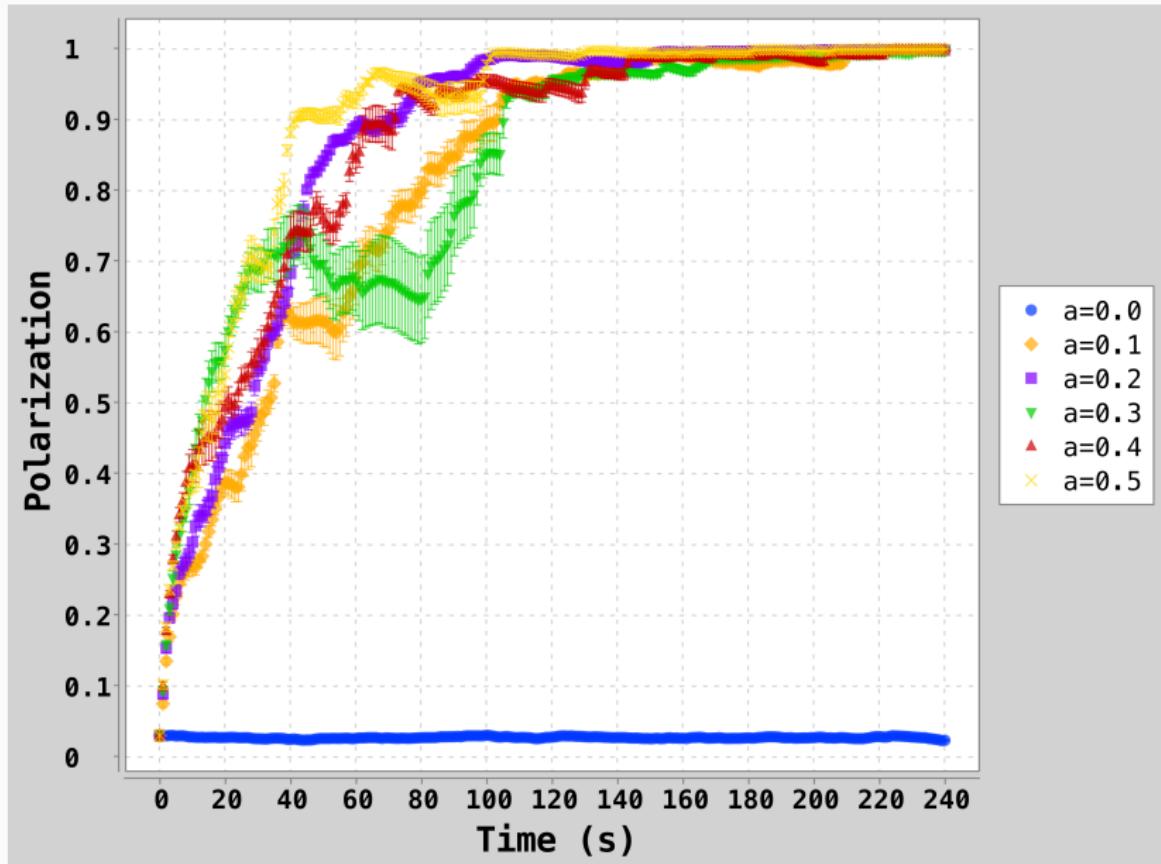
Tendencia: Boid a Especial

-0.25

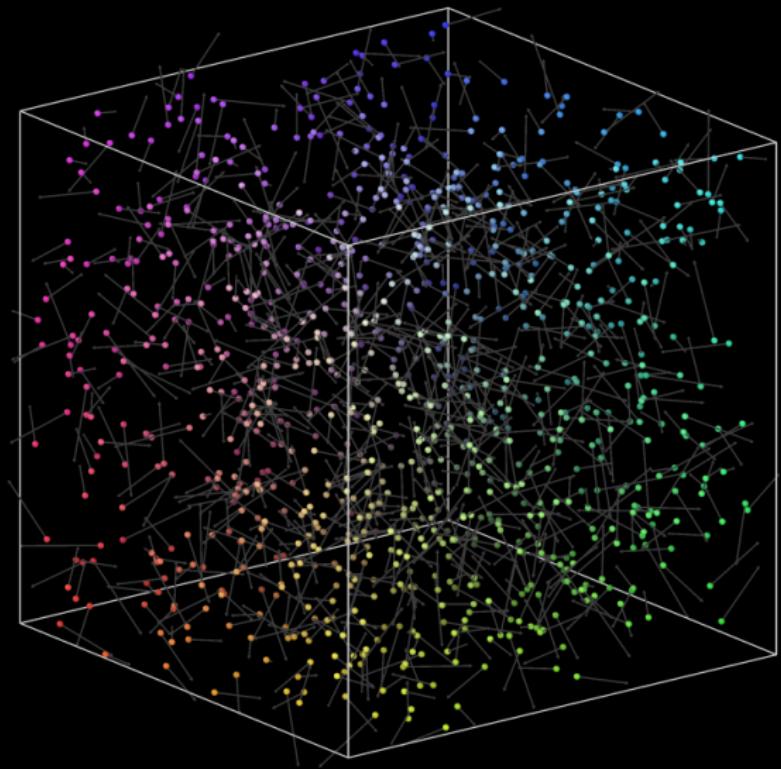
Tendencia: Especial a Boid

0.25

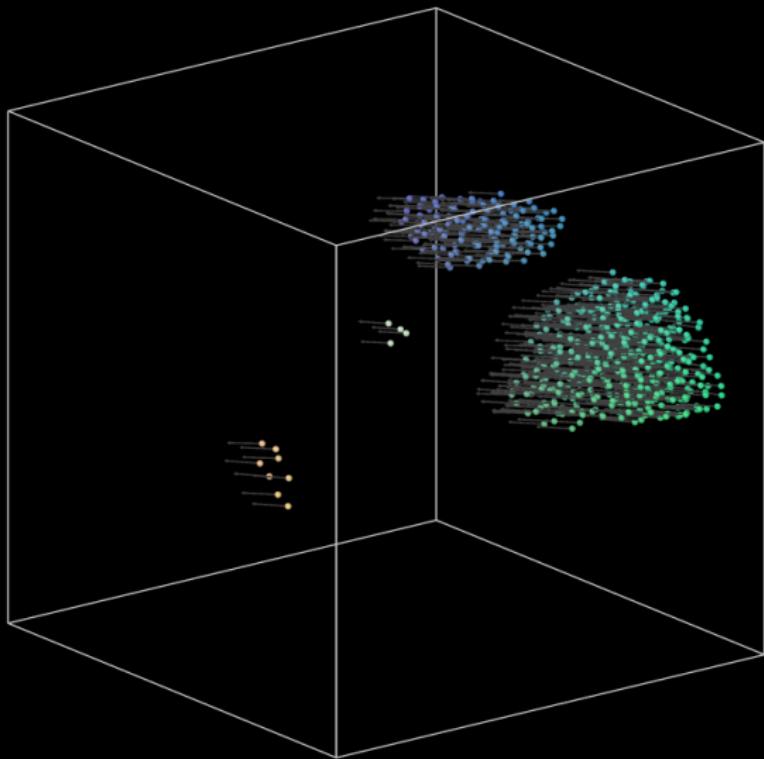
Polarización: Alineamiento



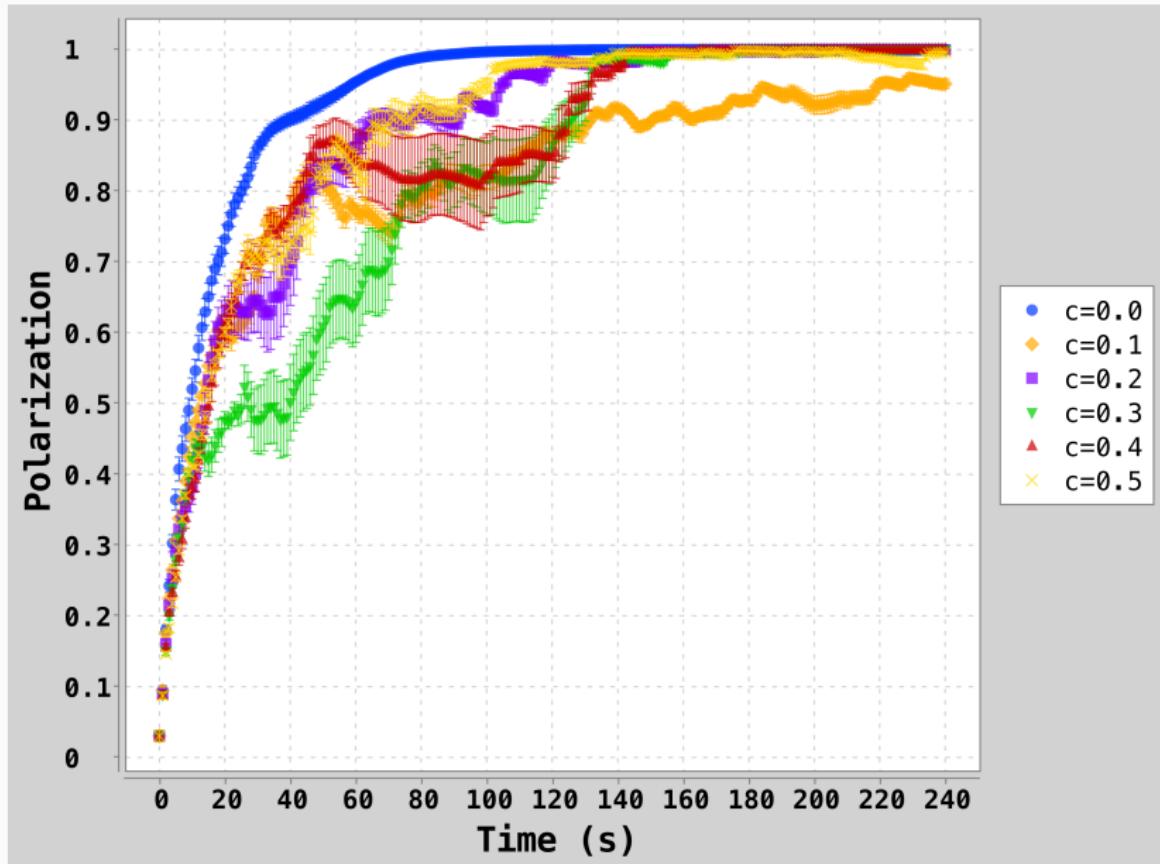
Alineamiento: $a = 0.0$



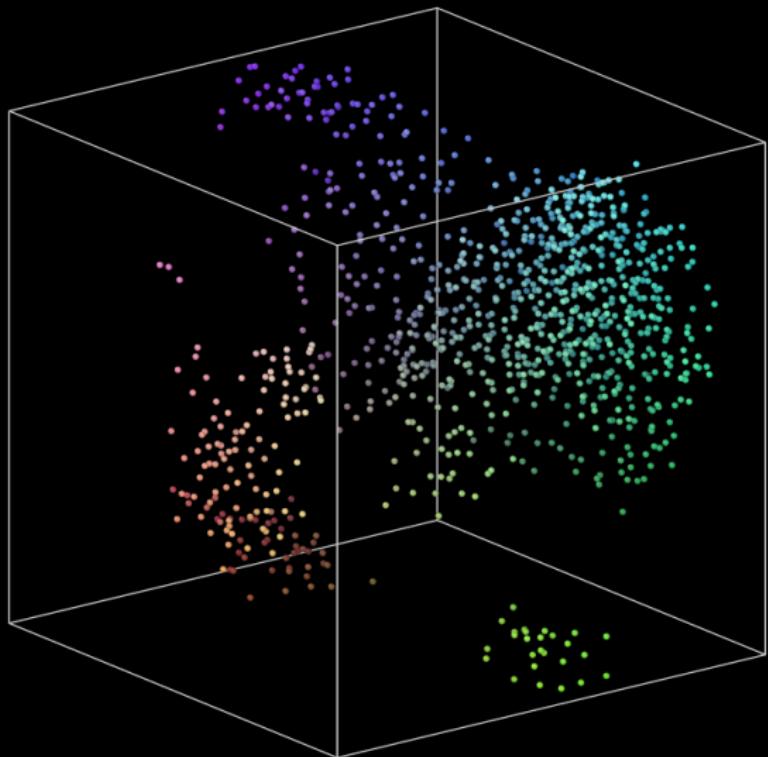
Alineamiento: $a = 0.5$



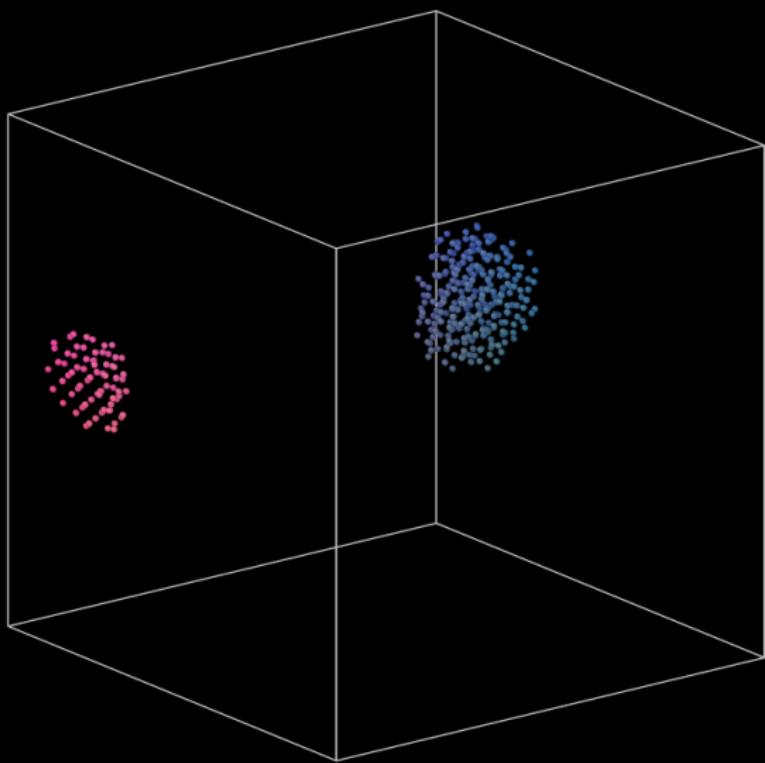
Polarización: Cohesión



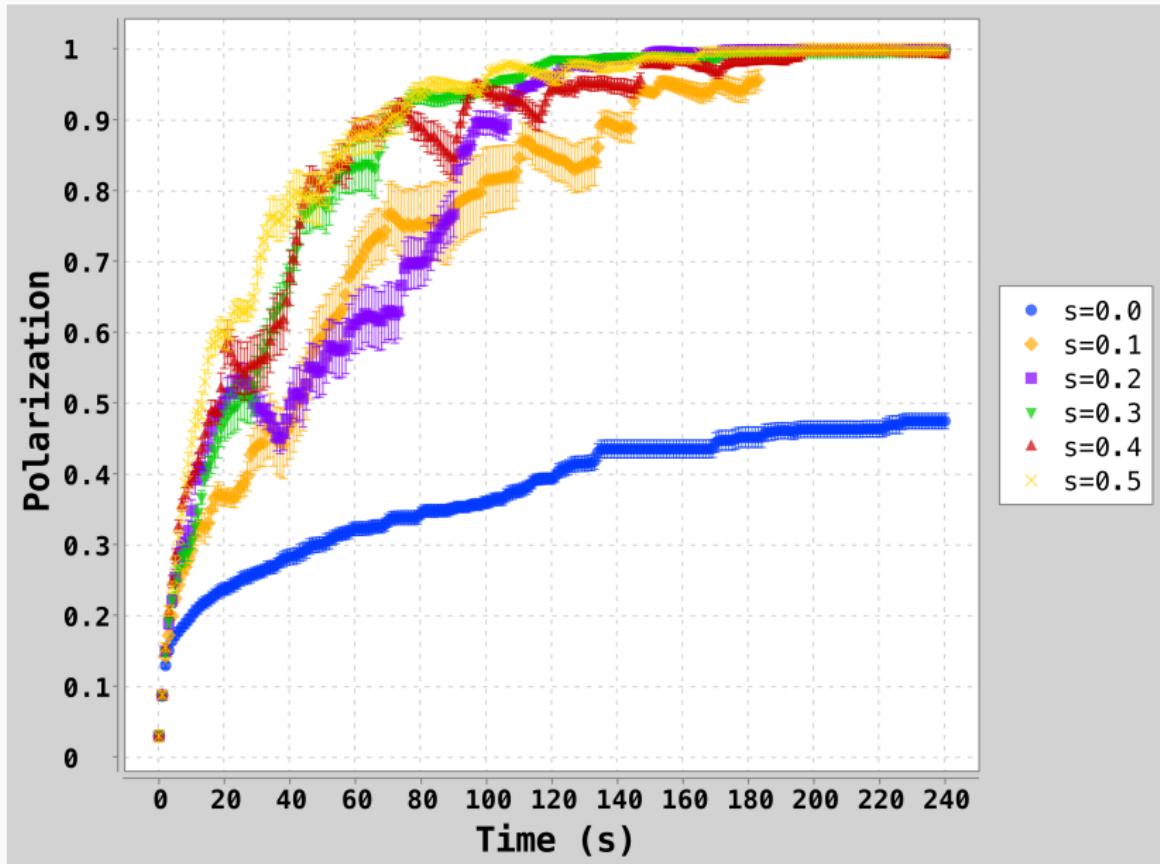
Cohesión: $c = 0.0$



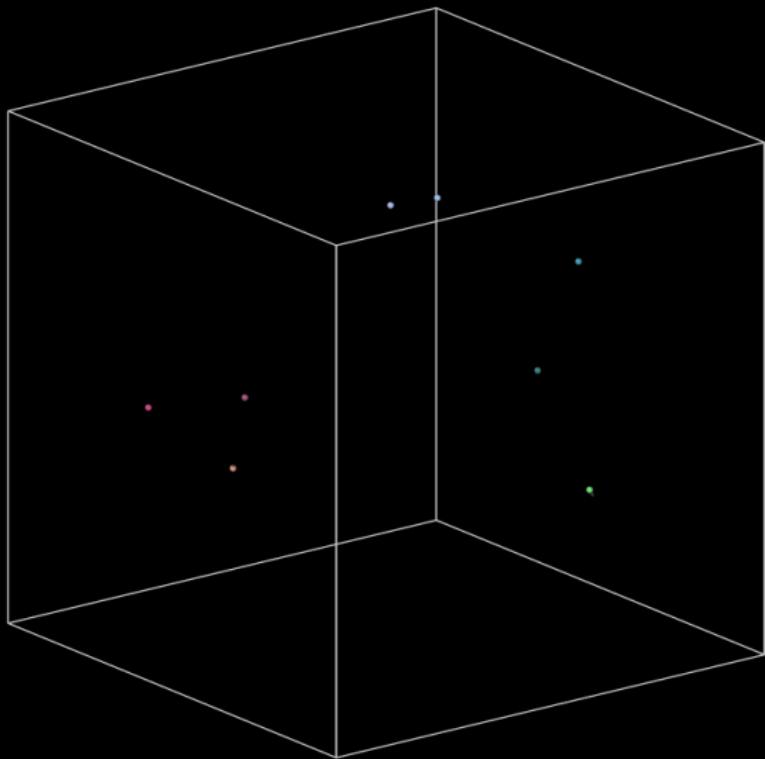
Cohesión: $c = 0.5$



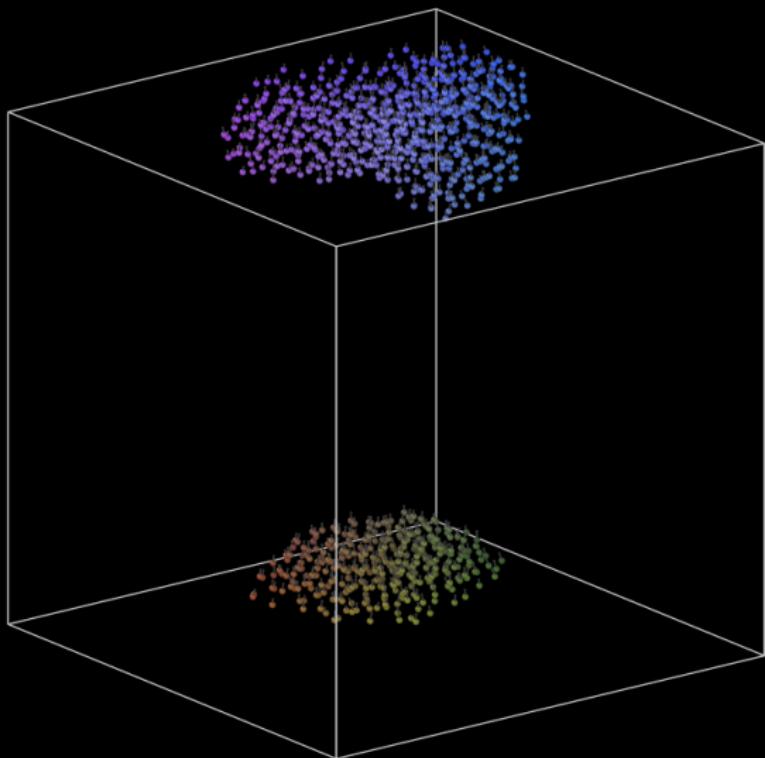
Polarización: Separación



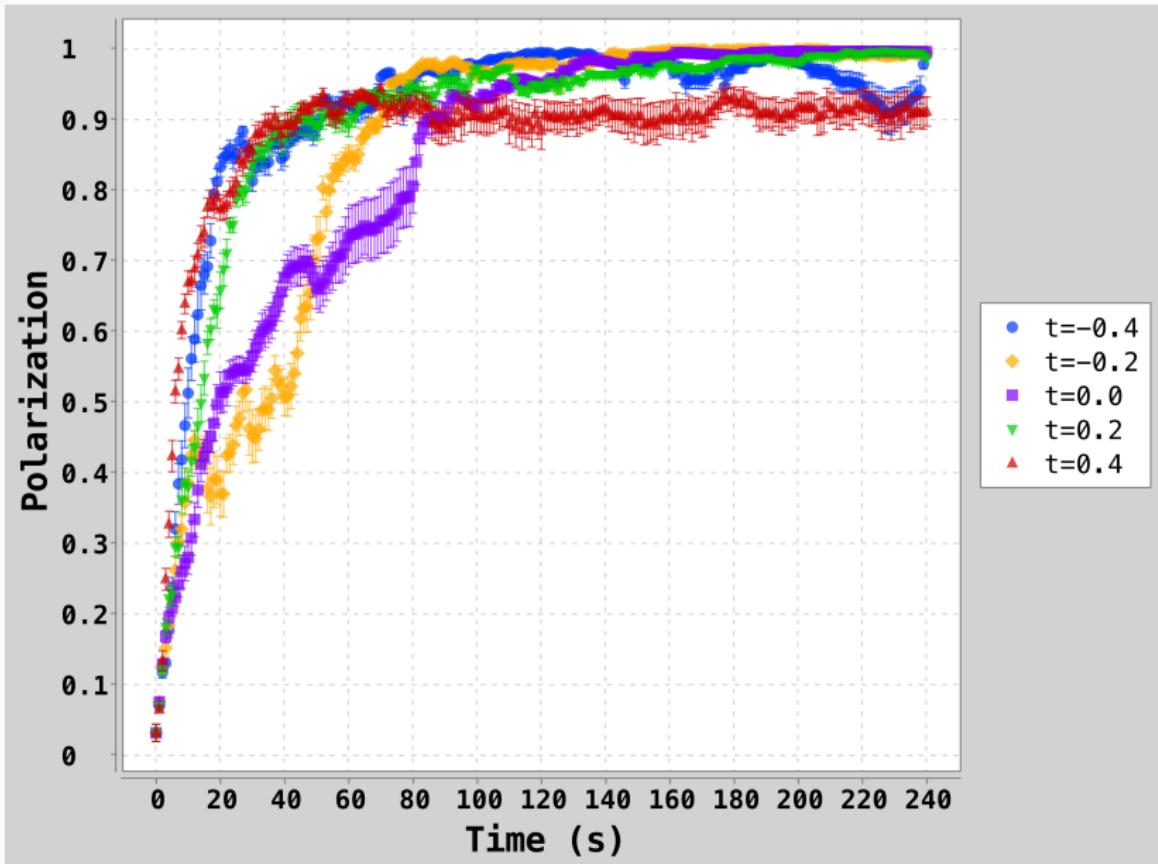
Separación: $s = 0.0$



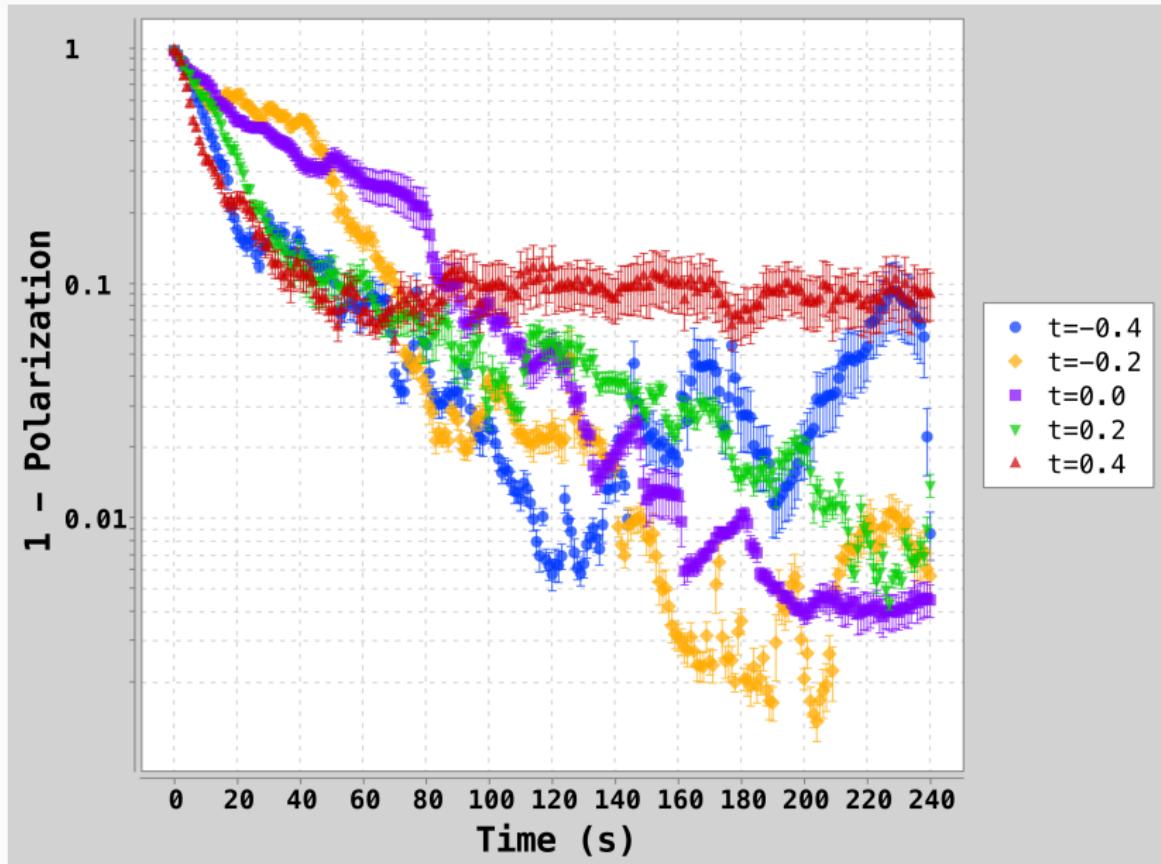
Separación: $s = 0.5$



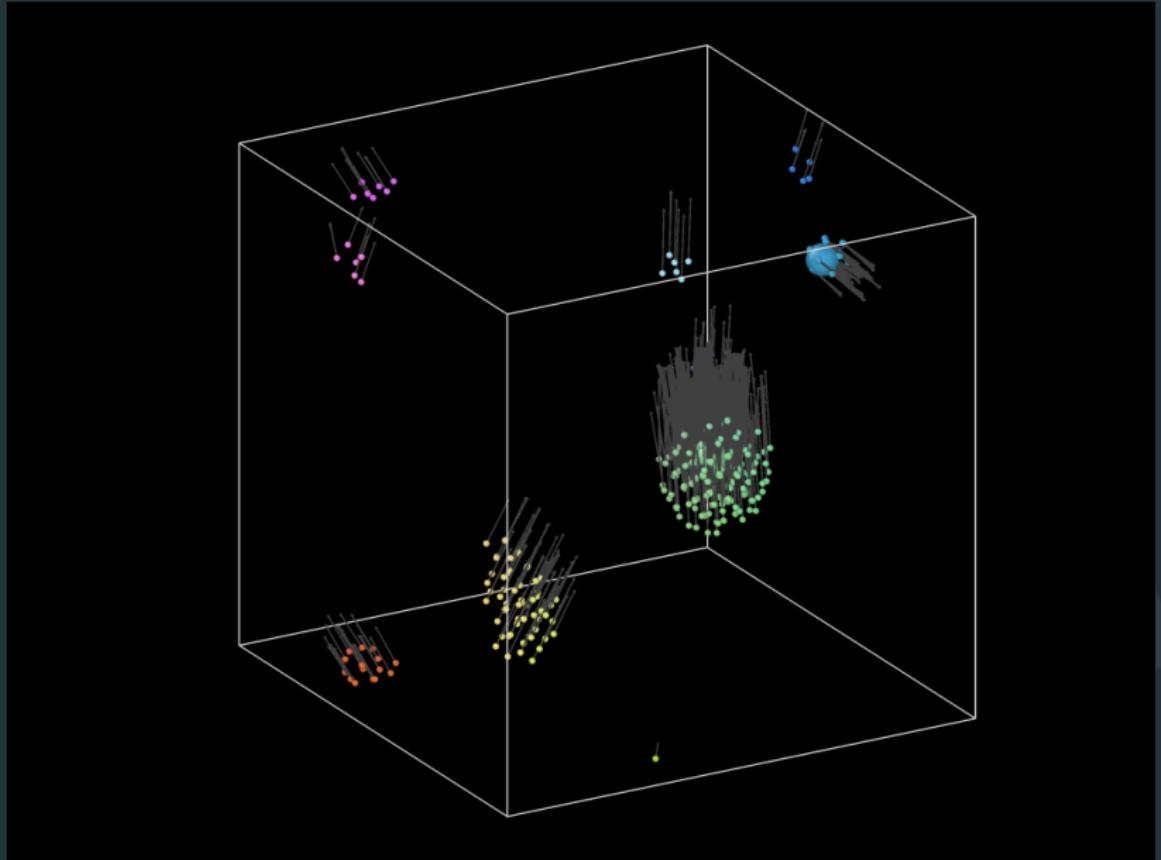
Polarización: Tendencia a



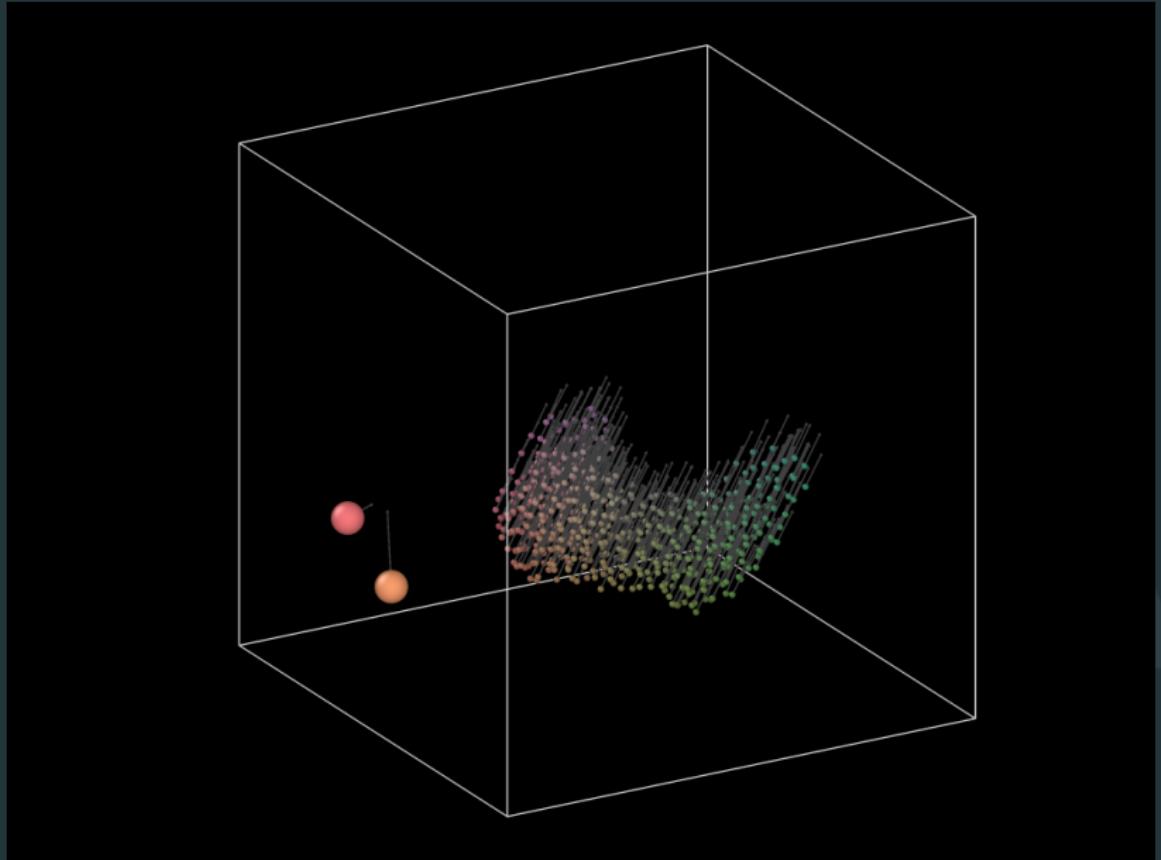
1 - Polarización: Tendencia a



Tendencia a: $t = 0.4$ (2 boids especiales)



Tendencia a: $t = -0.4$ (2 boids especiales)



Índice

Fundamentos

Algoritmo

Reglas básicas

Reglas extendidas

Implementación

Código

Resultados

Conclusiones



Conclusiones

- La regla más importante para polarización: **Alineamiento**.
- Desactivar la regla de **Cohesión** acelera la polarización.
- Desactivar la regla de **Separación** hace que los boids se unan puntualmente.
- Una alta **Separación** genera bandadas grandes con velocidad media mayor a sin **Cohesión**.
- Regla de **Tendencia a** cambia en gran medida el comportamiento de los *boids*.