



A scene from the movie Frozen. In the foreground, Elsa and Anna are shown from the waist up, standing in a snowy landscape. Elsa, on the left, has white hair and is wearing a light blue dress with a white fur-trimmed hood. Anna, on the right, has brown hair styled in braids and is wearing a pink dress with a green patterned cape. They are looking towards the camera. A large, jagged ice castle rises behind them against a blue sky with falling snowflakes. A blue rectangular border surrounds the lower-left portion of the image, containing the word "Animación".

Animación



Teaching at Stanford

CS106A

Programming
Methodologies

CURRENT

CS106B

Programming
Abstractions

LAST: FALL 2016

CS109

Probability for Computer
Scientists

LAST: FALL 2018

CS221

Intro to Artificial
Intelligence

LAST: SUM 2013

3 días aqui es igual a
2 semanas en Stanford

Como aprendi a programar

¿Cuándo adivinarías
que aprendí a programar?

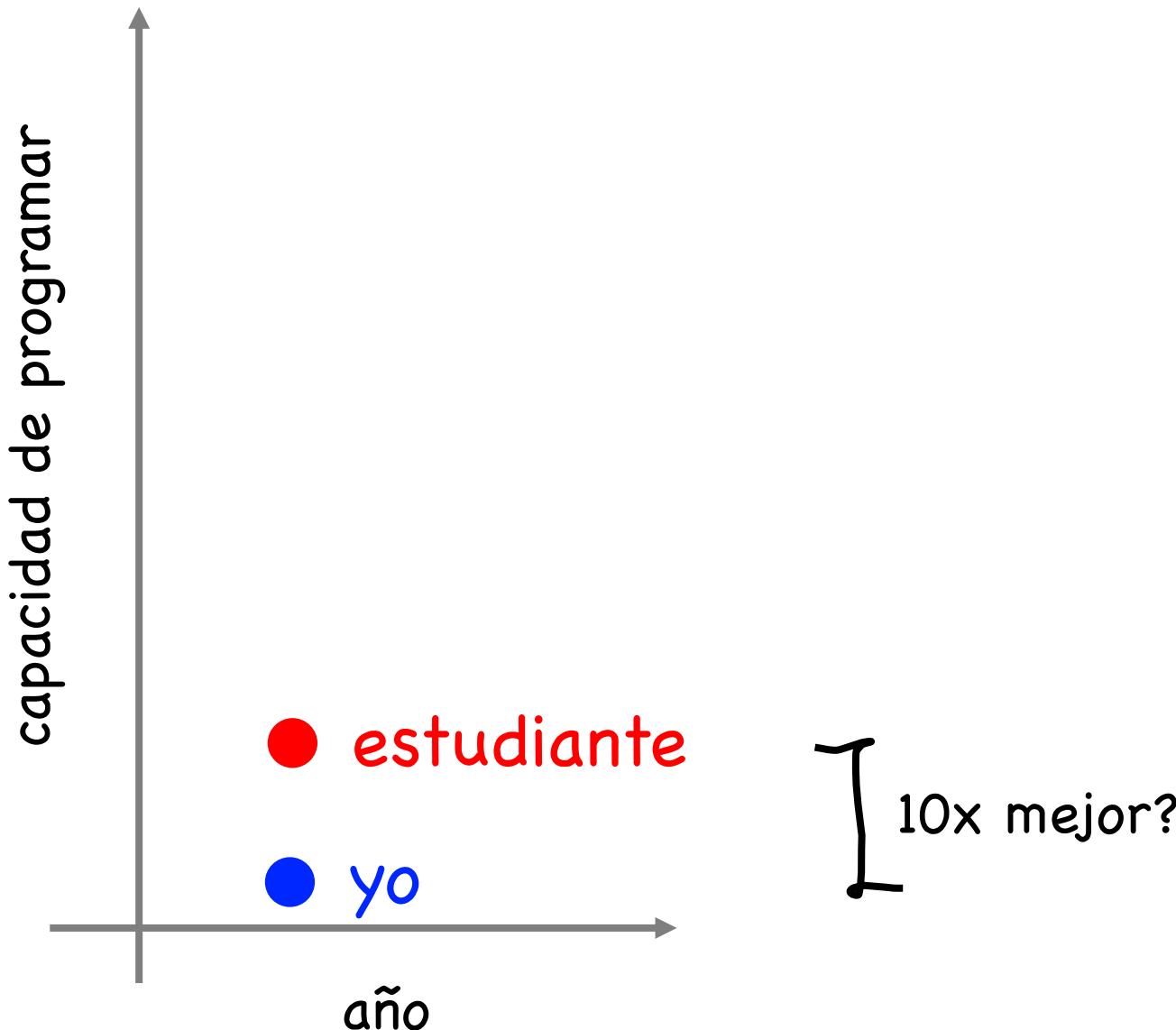
Cuando era joven?





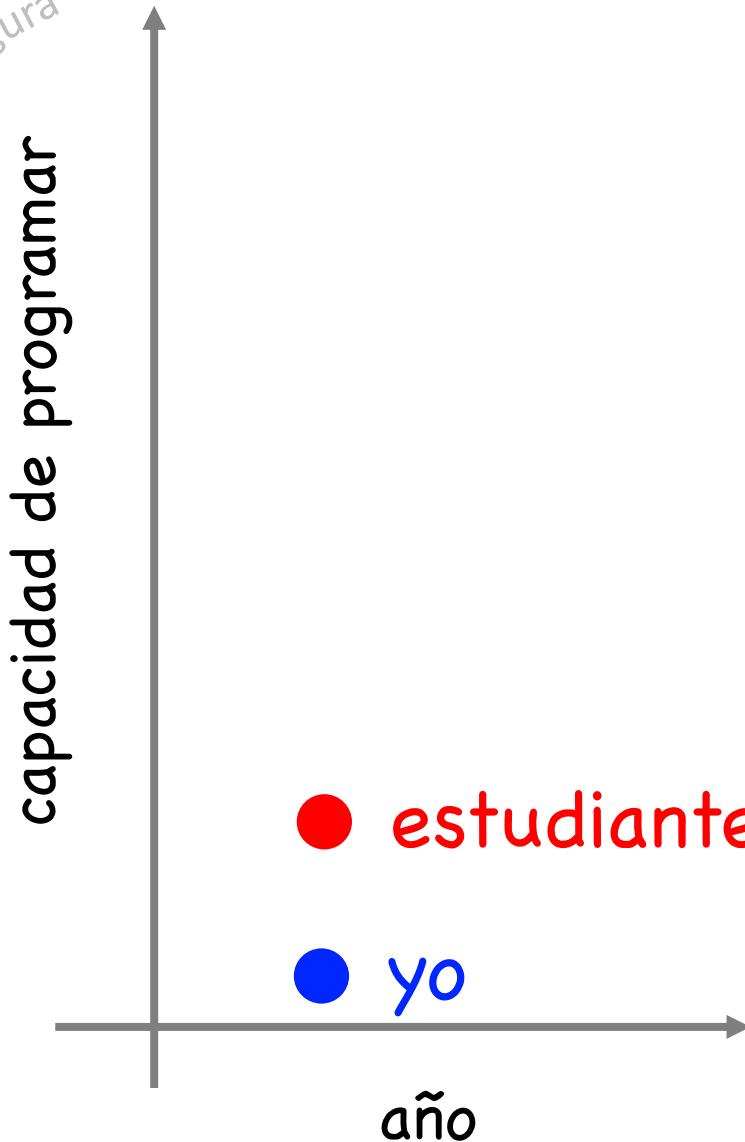
Creando cosas con solo
su mente y una computadora.

Al comienzo de mi viaje



Al comienzo de mi viaje

La figura



Ahora sé que el
comienzo es duro
para todos.

Estás
aprendiendo una
nueva forma de
pensar.

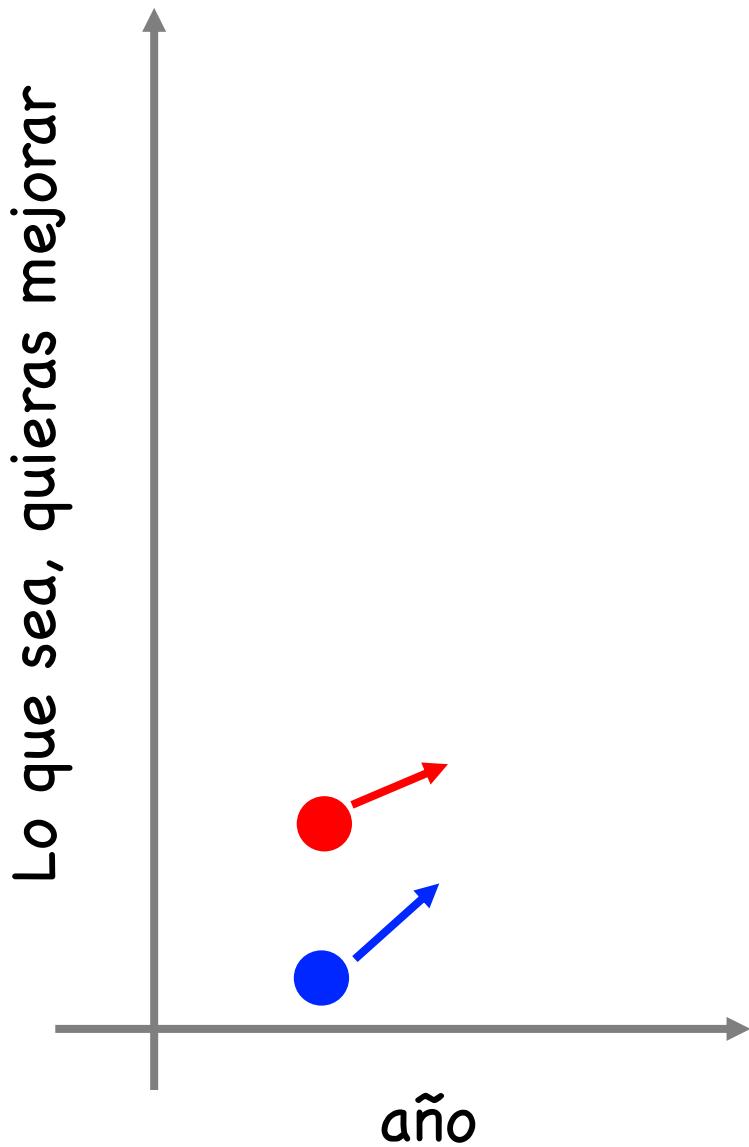
10x mejor?



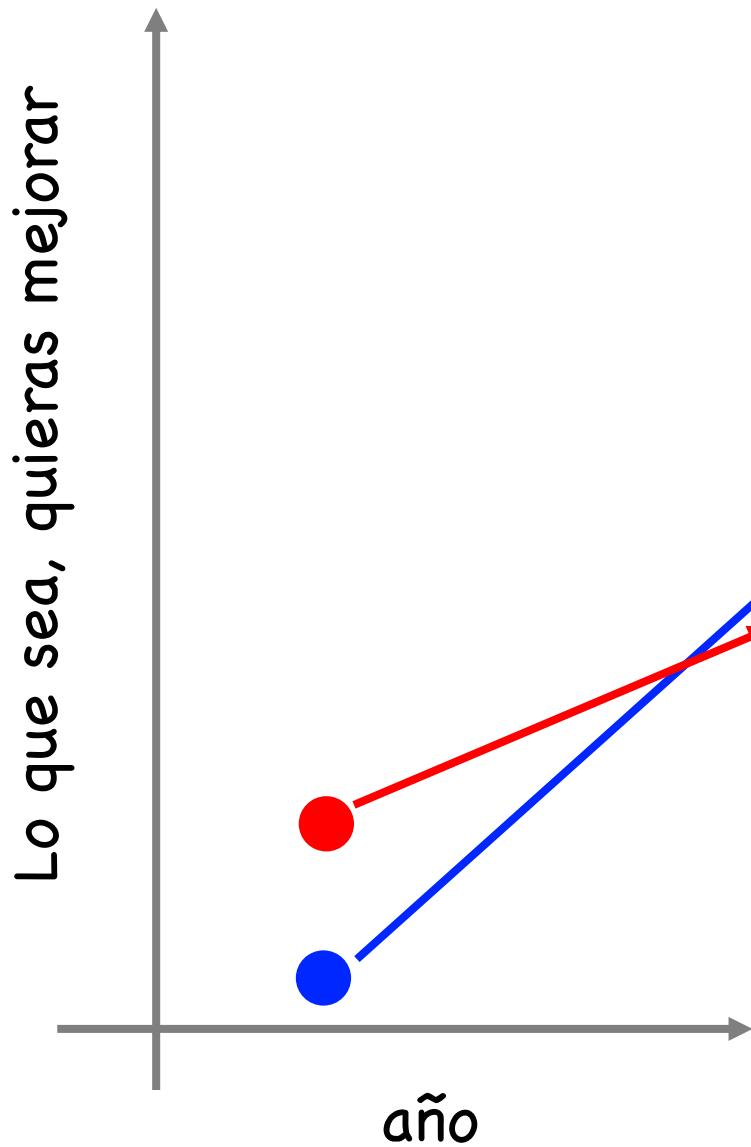
Imagínate a ti mismo



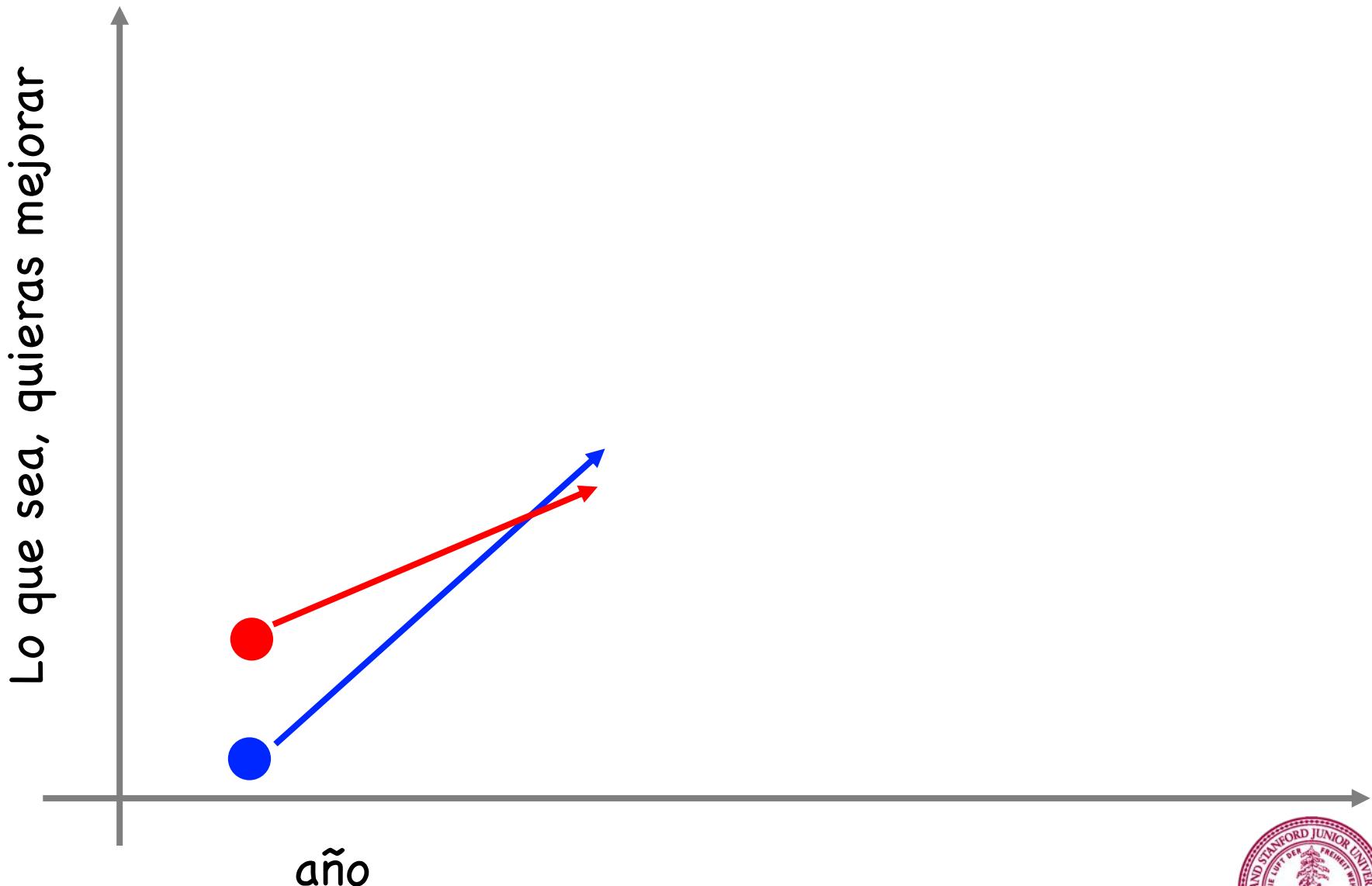
¿Quién preferirías ser?



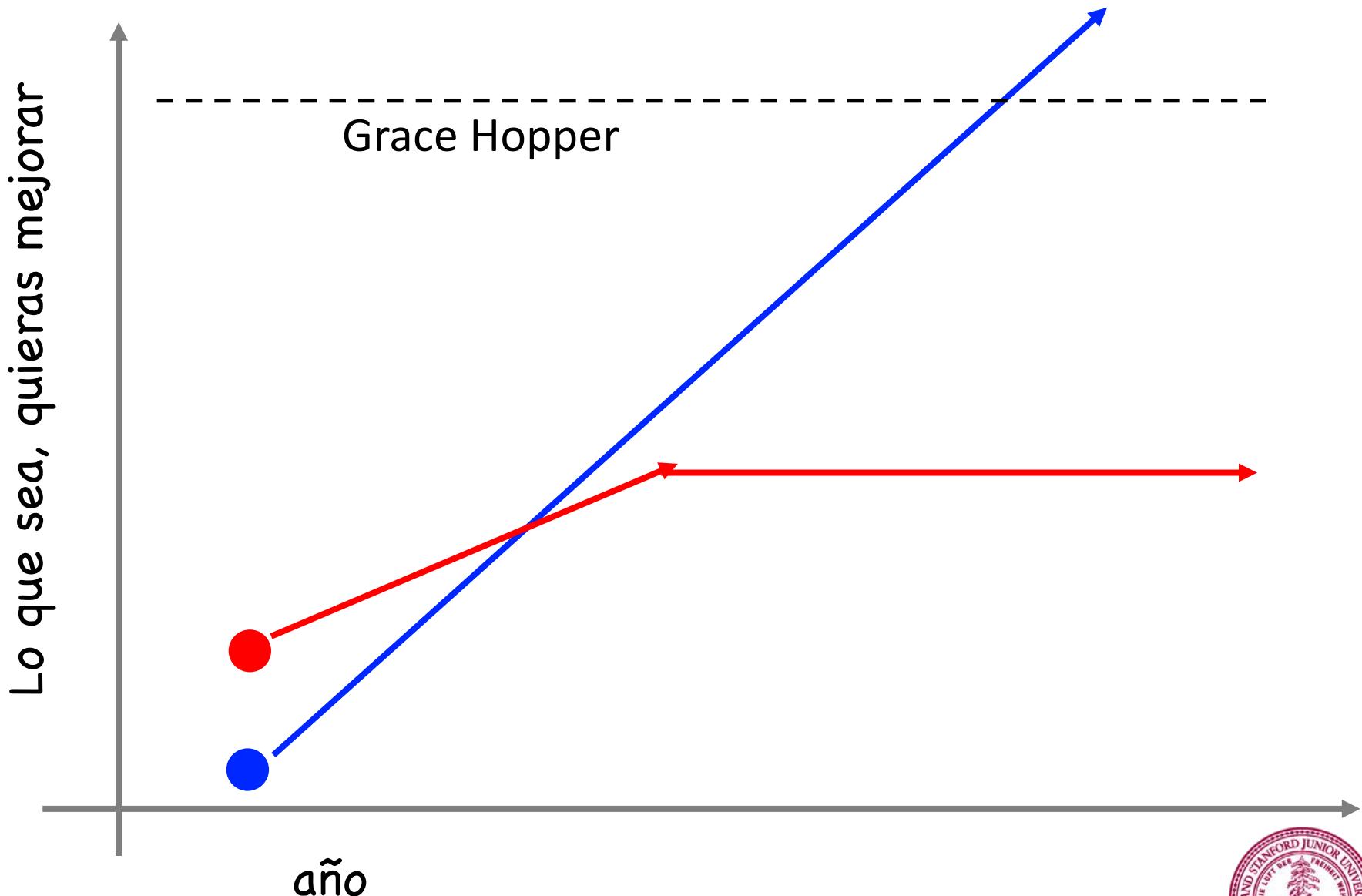
¿Quién preferirías ser?



¿Quién preferirías ser?

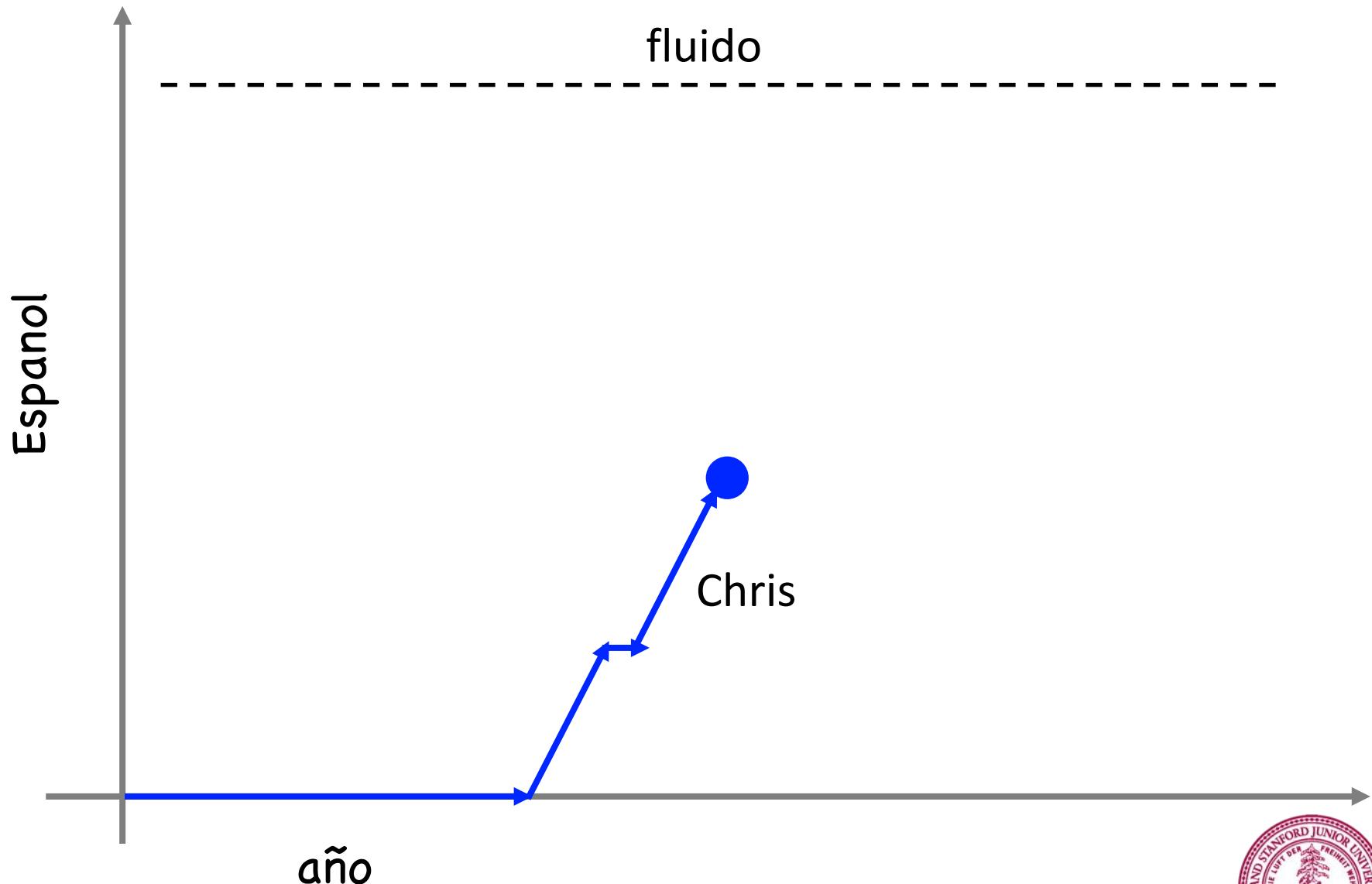


Imagínate a ti mismo

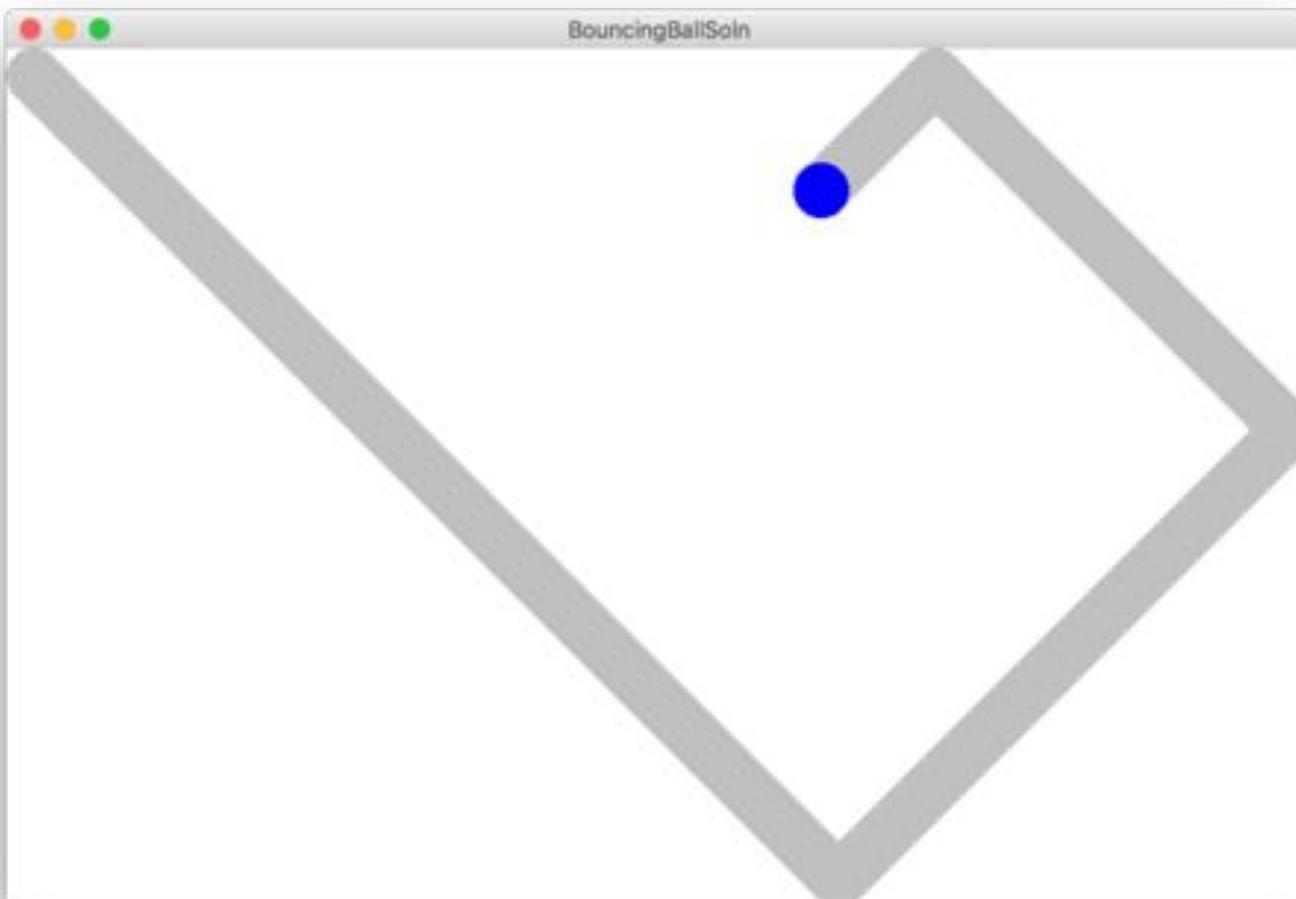


Si quieres hacer algo
grandioso, importa cuánto
aprendas cada día, no
cuánto sabes cuando tienes
17 años.

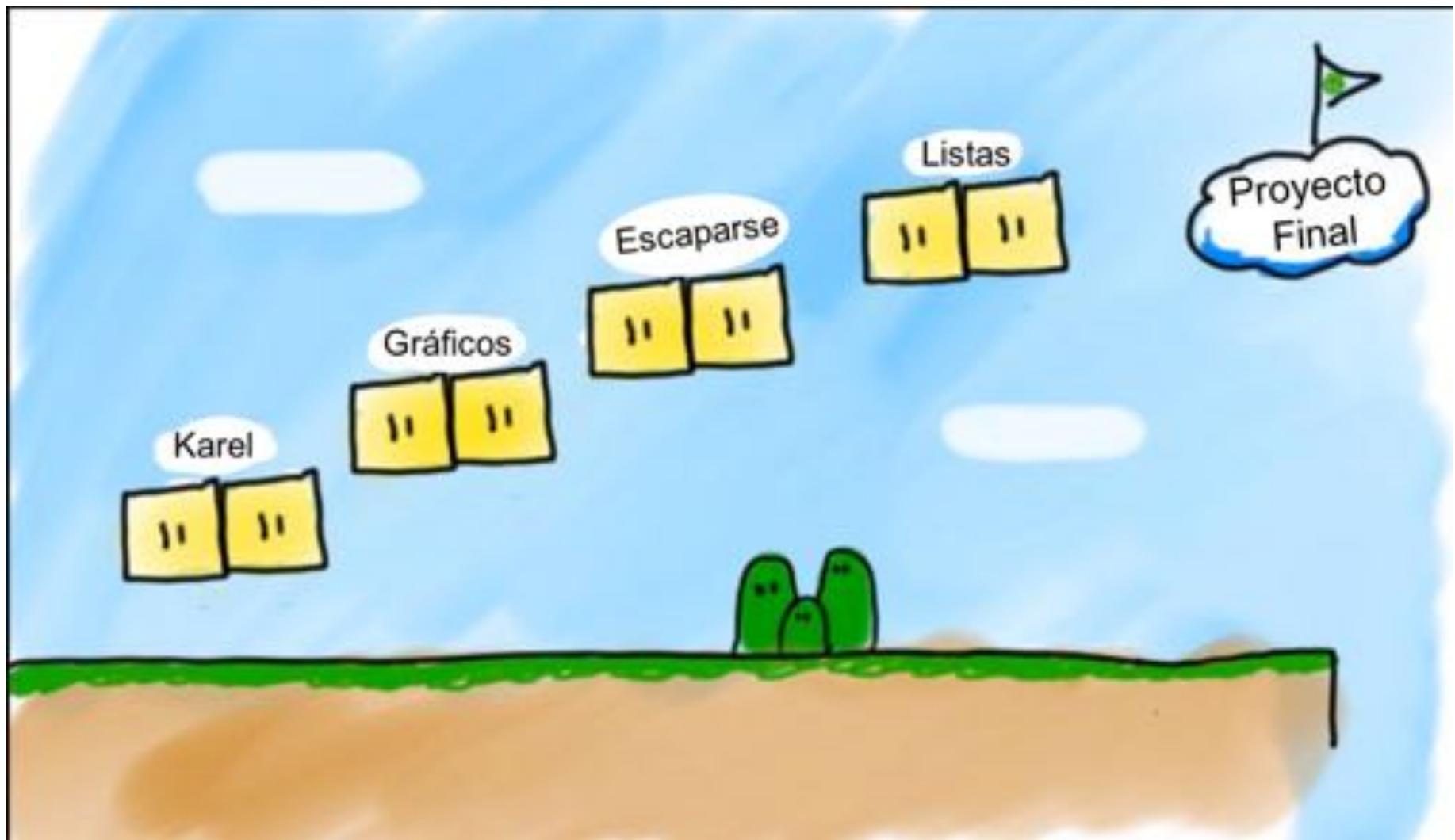
El español de Chris



Para empezar



Nuestra historia hasta ahora ...



Learning Goals

1. Write animated programs
2. Center an object



© Disney Pixar

Piech, CS106A, Stanford University



revisión

A Variable love story

By Chris

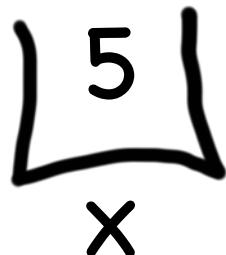
Piech, CS106A, Stanford University



Había una vez ...

x estaba buscando amor ...

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



x estaba buscando amor ...

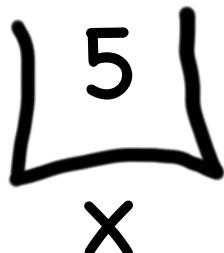
```
int x = 5;
```

```
if(buscandoAmor()) {
```

```
    int y = 5;
```

```
}
```

```
imprimir(x + y);
```



x estaba buscando amor ...

x definitivamente
estaba buscando amor

```
int x = 5;
```

```
if(buscandoAmor()) {
```

```
    int y = 5;
```

```
}
```

```
    imprimir(x + y);
```

5
x



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y

Hola, soy y

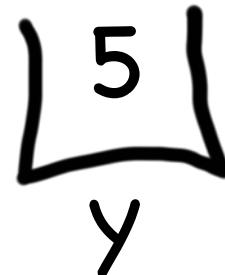
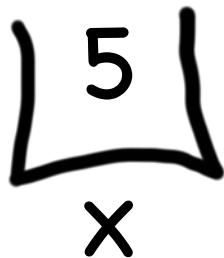


“Wow!”

x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

Wow



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y

Tenemos mucho
en comun



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

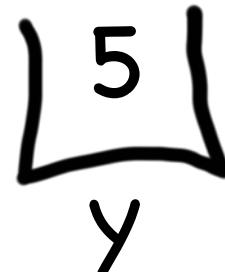
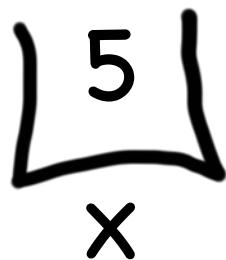
5
y

¡Ambos tenemos
valor 5!



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



Tal vez algún día
podamos ...



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y

imprimir juntos?



Se llevan bien

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y



Era una pareja hermosa ...

Pero entonces la tragedia golpeó.

La tragedia golpeó

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

5
x

5
y



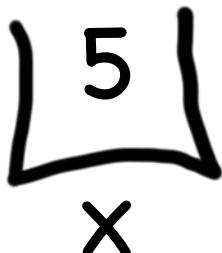
La tragedia golpeó

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



La tragedia golpeó

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

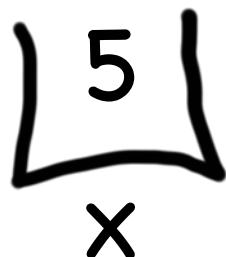
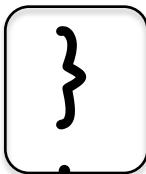


Noooooooooooooo!

Lo ves...

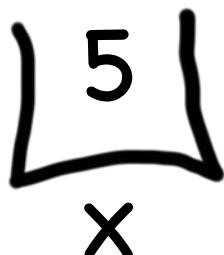
Cuando un programa sale de un bloque de código ...

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



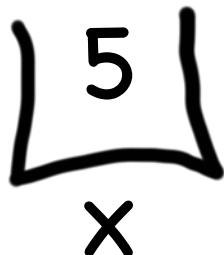
...donde una variable fue declarada ...

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



¡Se borra de la memoria!

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



y fue declarado dentro del if

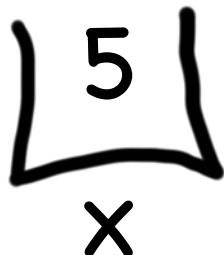
```
int x = 5;
```

```
if(buscandoAmor()) {
```

```
    int y = 5;
```

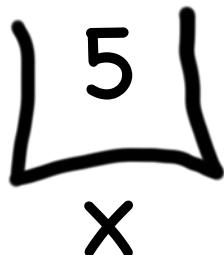
```
}
```

```
imprimir(x + y);
```



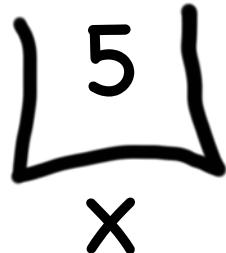
Se borra de la memoria aquí

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```



Y no existe aquí.

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
}  
imprimir(x + y);
```

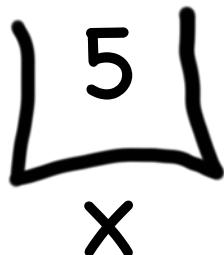


Y no existe aquí.

```
int  
if(b  
{  
}  
Error. Undefined  
variable y.
```

```
}
```

```
imprimir(x + y);
```



El fin

¿O es eso?

Variables tienen una duración de vida

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... algún otro código  
}
```



Variables tienen una duración de vida

```
public void run( ) {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... algún otro código  
}
```



Entrar en existencia cuando se declara

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... otro código  
}
```



Viene a la vida aquí

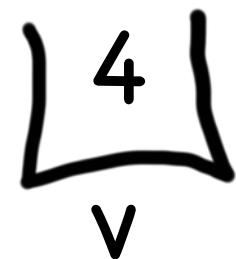
8
v



Viven hasta el final del bloque de código.

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... otro código  
}
```

Este es el bloque de código más interno en el que estaba declarado....



Variables tienen una duración de vida

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4; ←  
        ... algún código  
    }  
    ... otro código  
}
```

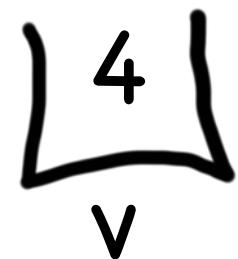
Todavía vivo aquí ...

4
v



Viven hasta el final del bloque de código.

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... otro código  
}
```



Muere aquí (al final de su bloque de código)



Viven hasta el final del bloque de código.

```
public void run() {  
    double v = 8;  
    if (condición) {  
        v = 4;  
        ... algún código  
    }  
    ... algún otro código  
}
```



← Muere aquí (al final de su bloque de código)



A Variable Love story

Capítulo 2

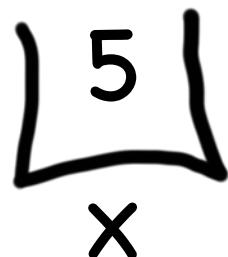
Piech, CS106A, Stanford University



El programador reparó su error.

x estaba buscando amor ...

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
    imprimir(x + y);  
}
```



x estaba buscando amor ...

x definitivamente
estaba buscando amor

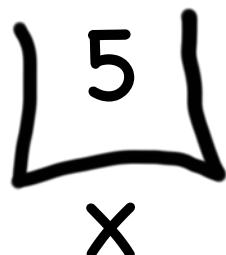
```
int x = 5;
```

```
if(buscandoAmor()) {
```

```
    int y = 5;
```

```
    imprimir(x + y);
```

```
}
```



x conoció a y

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
    imprimir(x + y);  
}
```



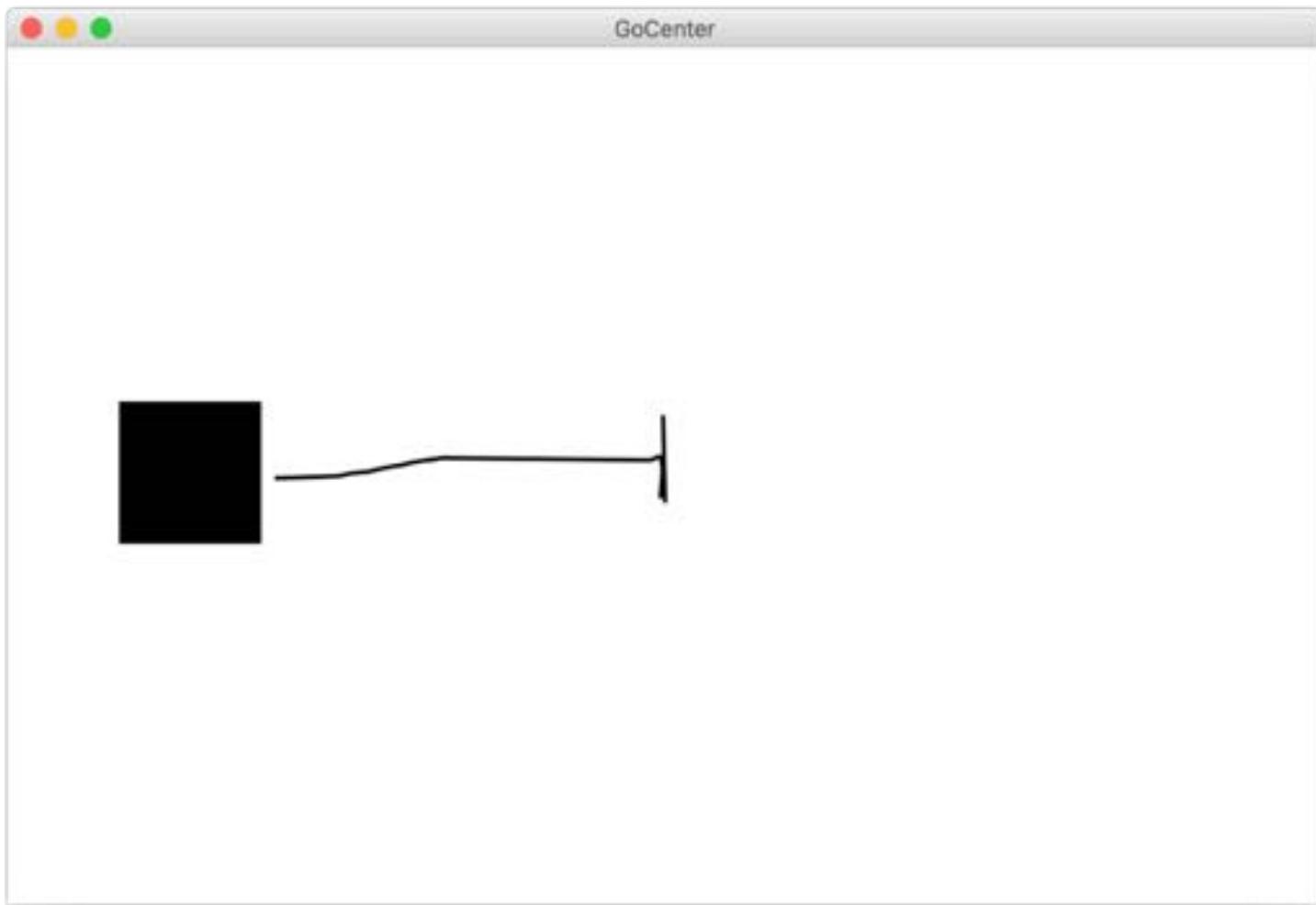
Ya que ambos estaban en “scope” ...

```
int x = 5;  
if(buscandoAmor()) {  
    int y = 5;  
    imprimir(x + y);  
}
```



La historia tuvo un final feliz!

Mover SRect



Piech, CS106A, Stanford University



Ciclo de animación

```
private void run() {  
    // prepara  
  
    while(true) {  
        // actualice el mundo  
  
        // pausa  
        pausa(RETARASAR);  
    }  
}
```



Ciclo de animación

```
private void run() {  
    // prepara  
  
    while(true) {  
        // actualice el mundo  
  
        // pausa  
        pausa(RETASAR);  
    }  
}
```

Haz todas las variables que necesites.
Aregar los objetos gráficos a la pantalla.



Ciclo de animación

```
private void run() {  
    // prepara  
    while(true) {  
        // actualice el mundo  
  
        // pausa  
        pausa(RETARASAR);  
    }  
}
```

El ciclo de animación es una repetición de latidos.



Ciclo de animación

```
private void run() {  
    // prepara  
  
    while(true) {  
        // actualice el mundo  
        // pausa  
        pausa(RETRASAR);  
    }  
}
```

Cada latido,
actualiza el
mundo hacia
adelante un
paso.



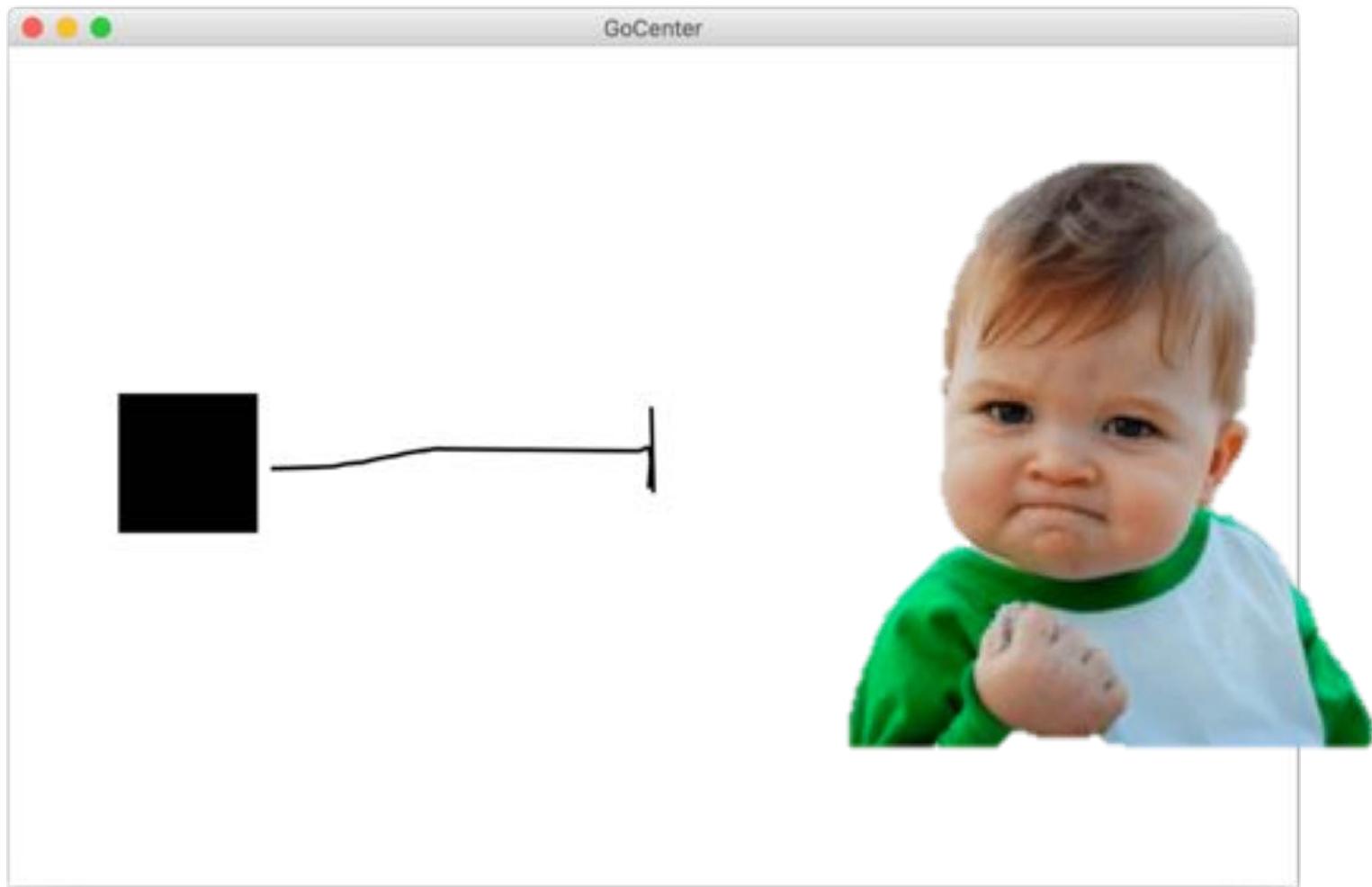
Ciclo de animación

```
private void run() {  
    // prepara  
  
    while(true) {  
        // actualice el mundo  
  
        // pausa  
        pausa(RETRASAR);  
    }  
}
```

Si no haces una pausa, los humanos no podrán verlo.



Mover SRect



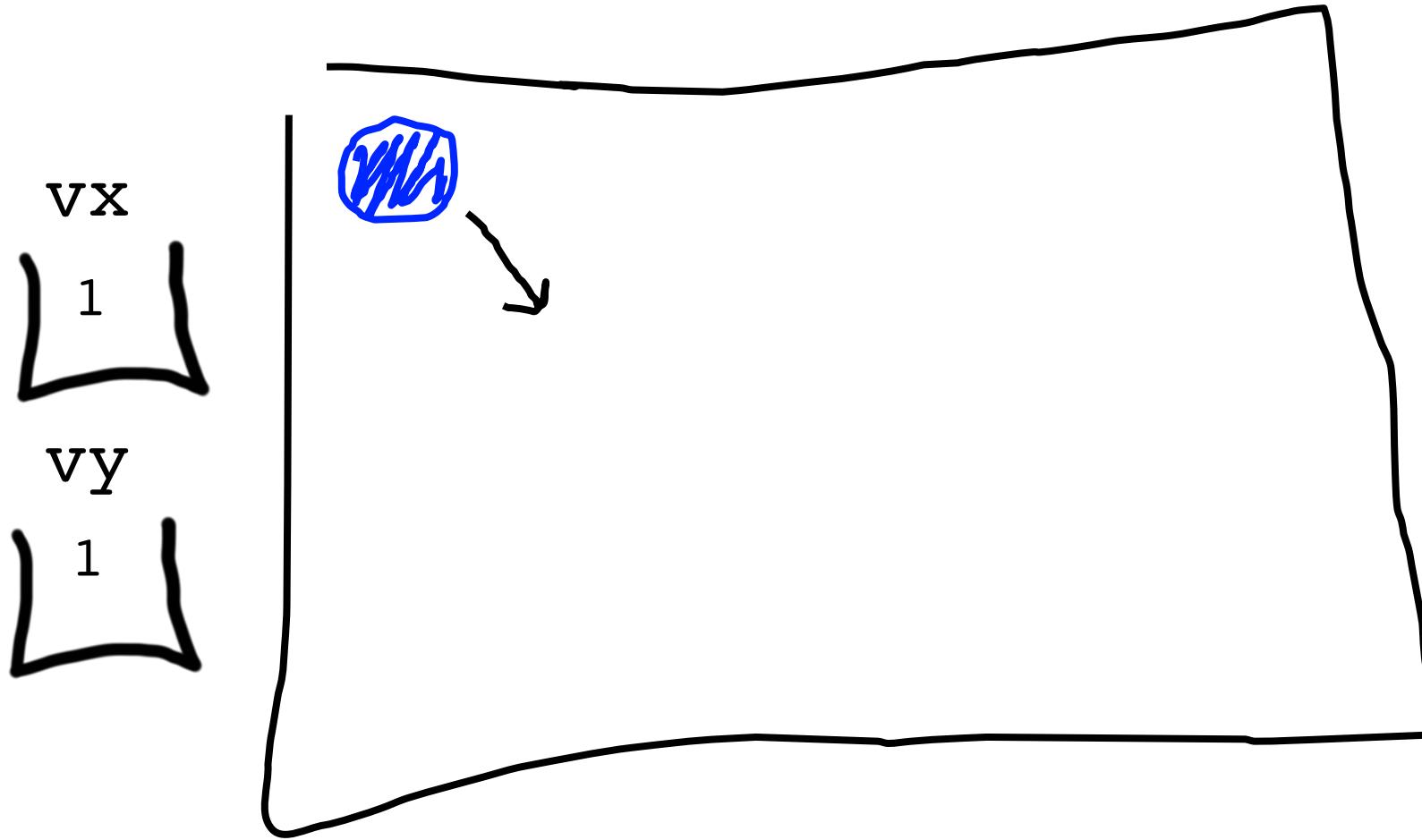
Mover SRect

```
private void run() {  
    // prepara  
    SRect r = new Srect(0, 250, 100, 100);  
    s.cambiarFilled(true);  
    agregar(s);  
  
    while(true) {  
        // actualice el mundo  
        r.moverse(1, 0);  
  
        // pausa  
        pausa(RETRASAR);  
    }  
}
```



Pelota que rebota

Primero latido

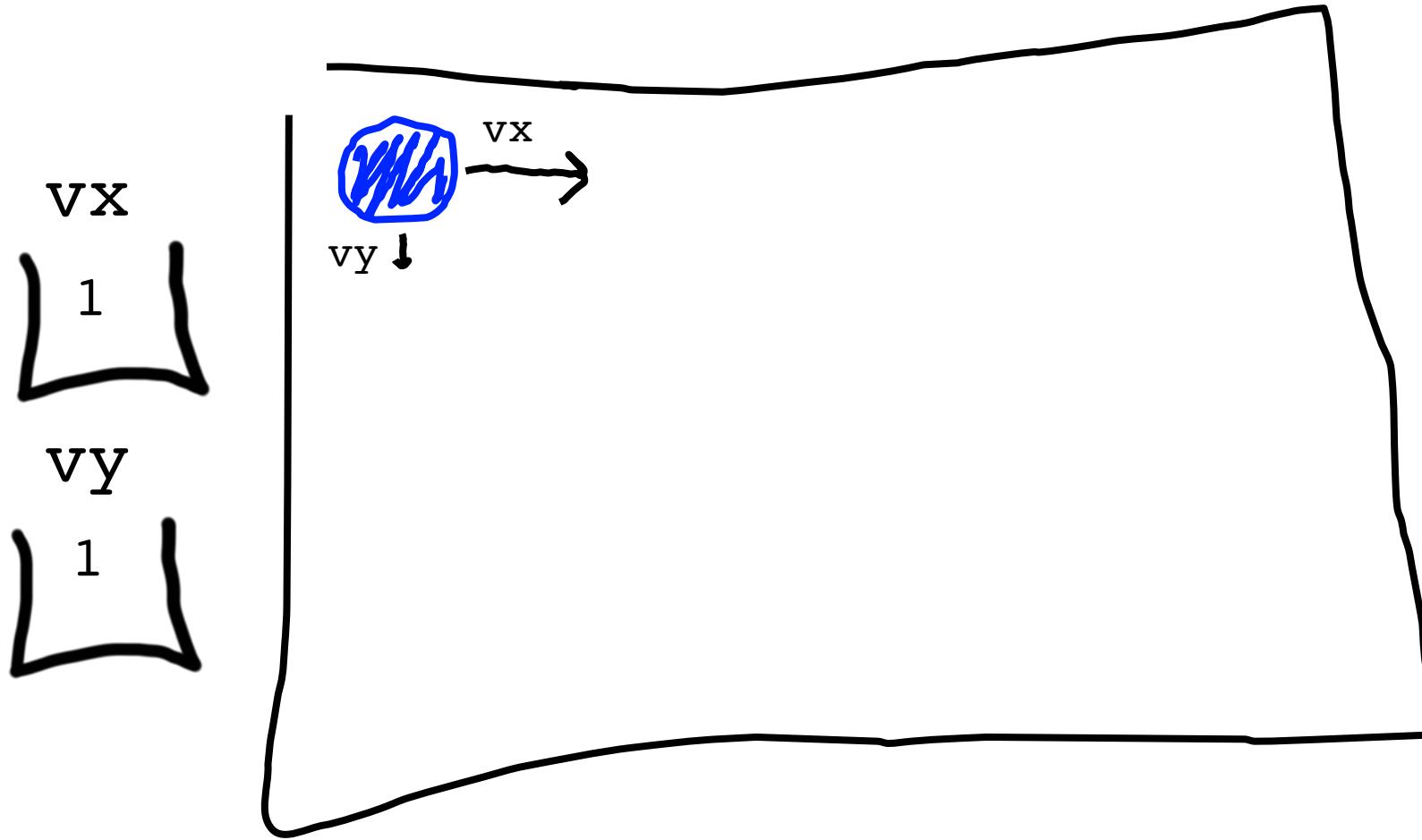


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

First heartbeat

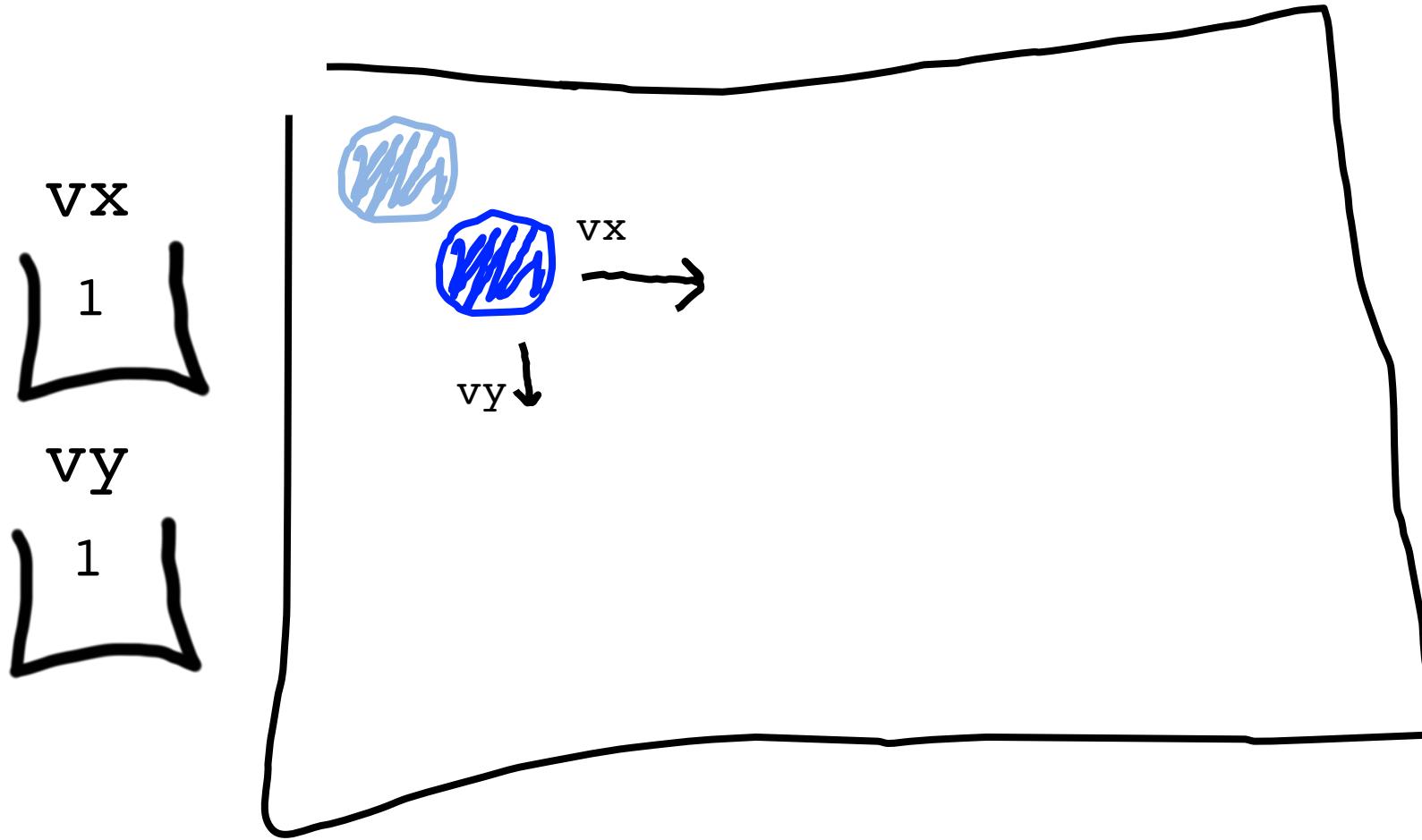


El SOval **moverse** El método toma
un cambio en x y un cambio en y



Pelota que rebota

Second heartbeat

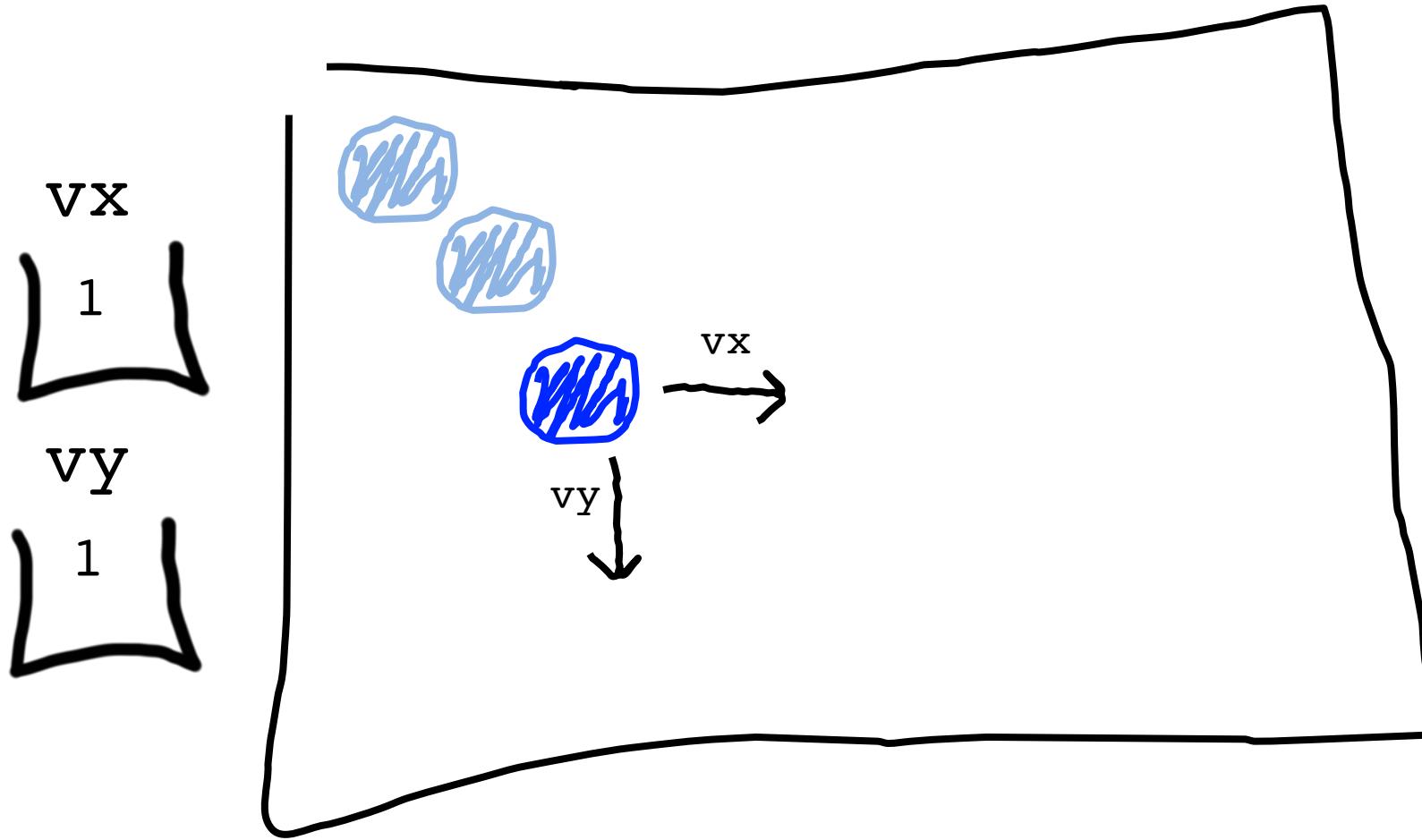


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

Third heartbeat

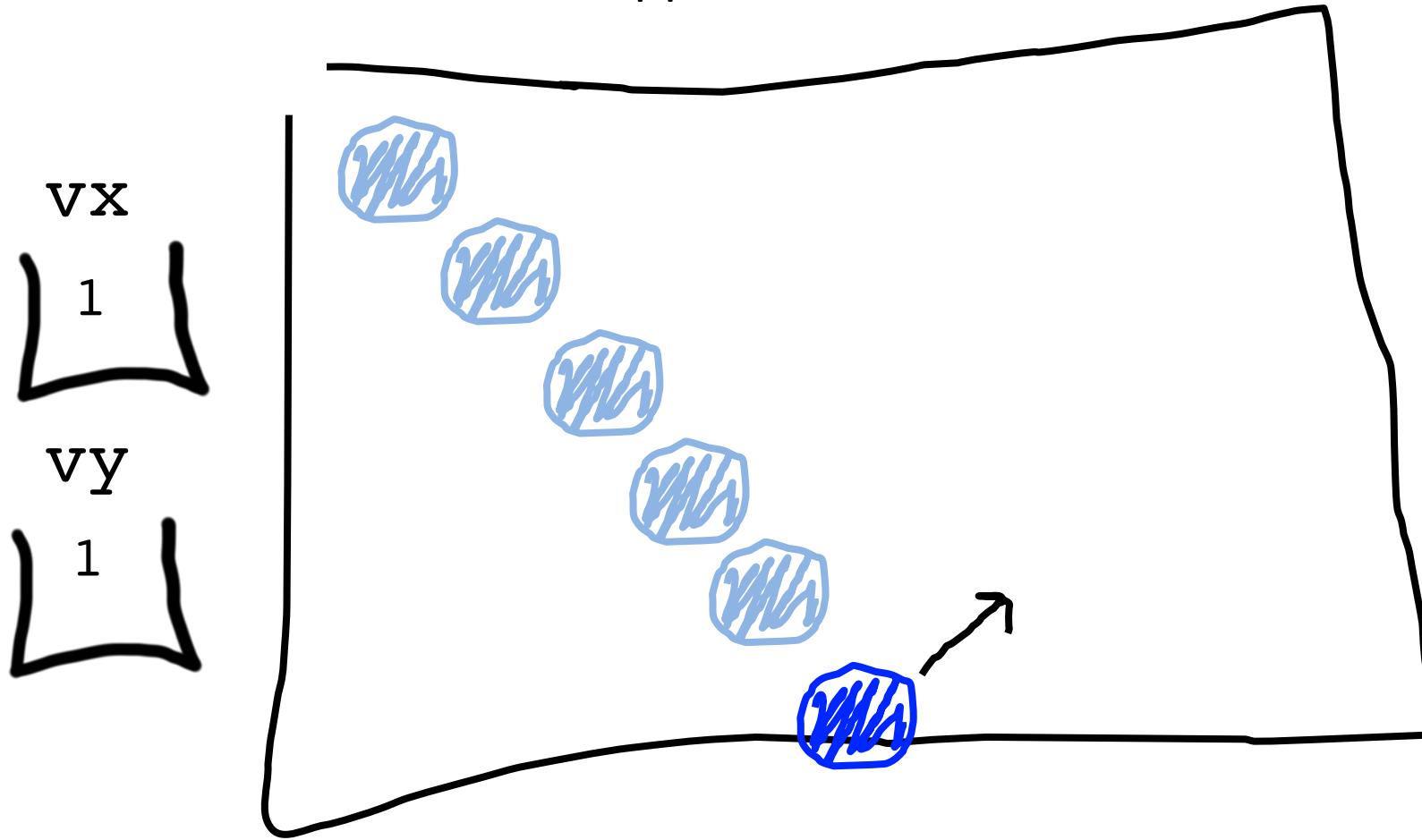


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

What happens when we hit a wall?

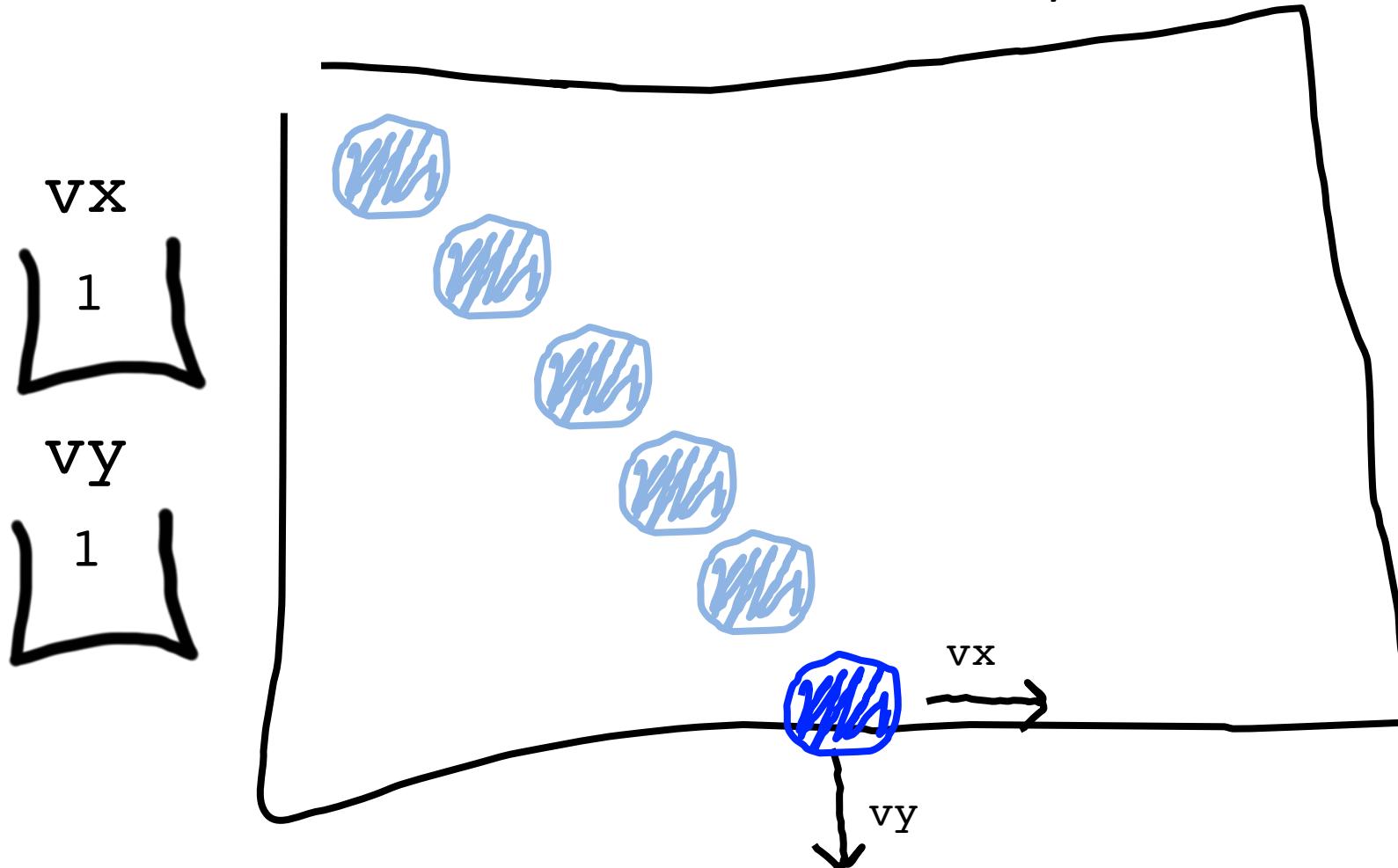


Velocidad: cuántos píxeles
la pelota mueve cada latido



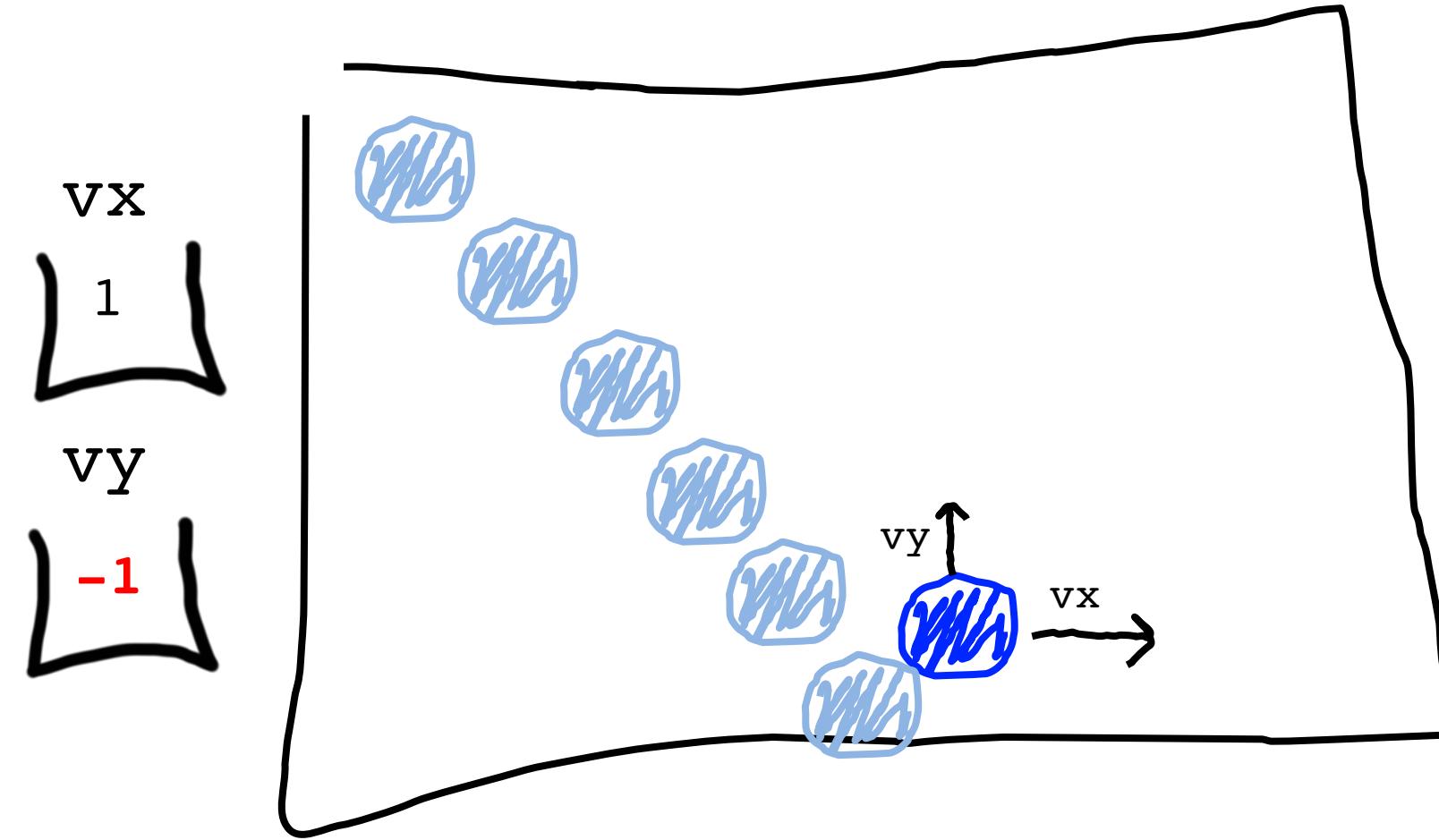
Pelota que rebota

We have this velocity



Pelota que rebota

Seventh Heartbeat

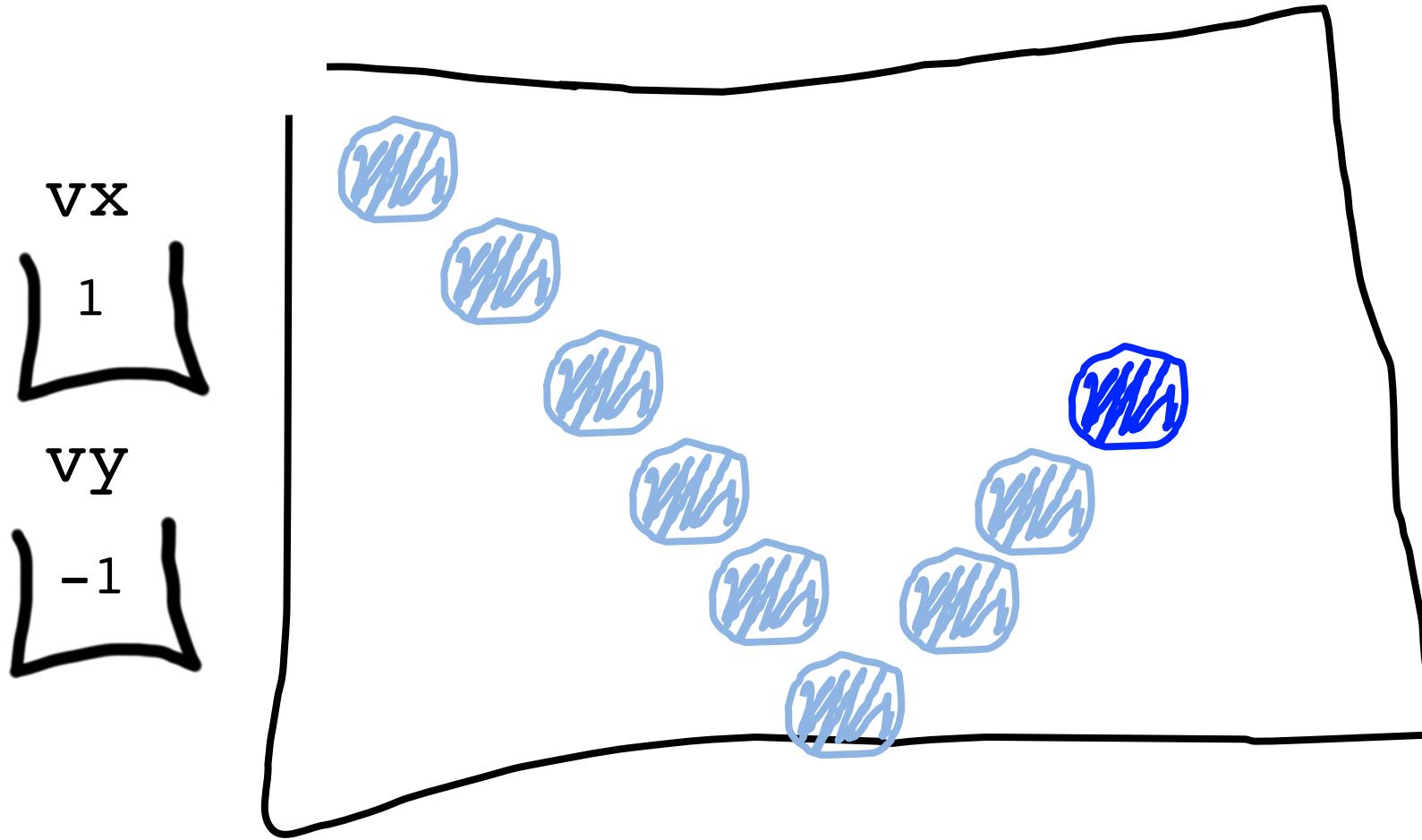


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

Seventh Heartbeat

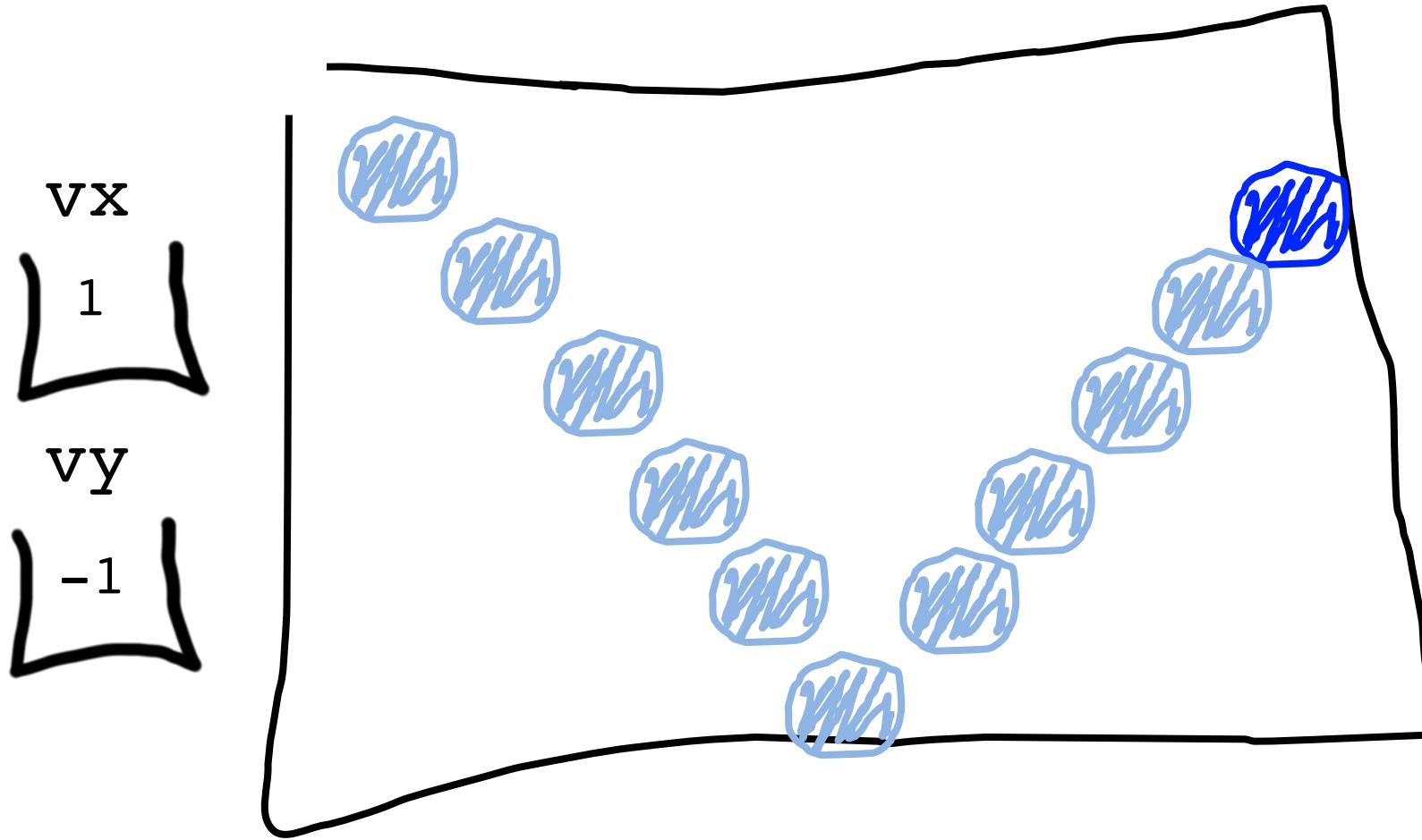


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

Seventh Heartbeat

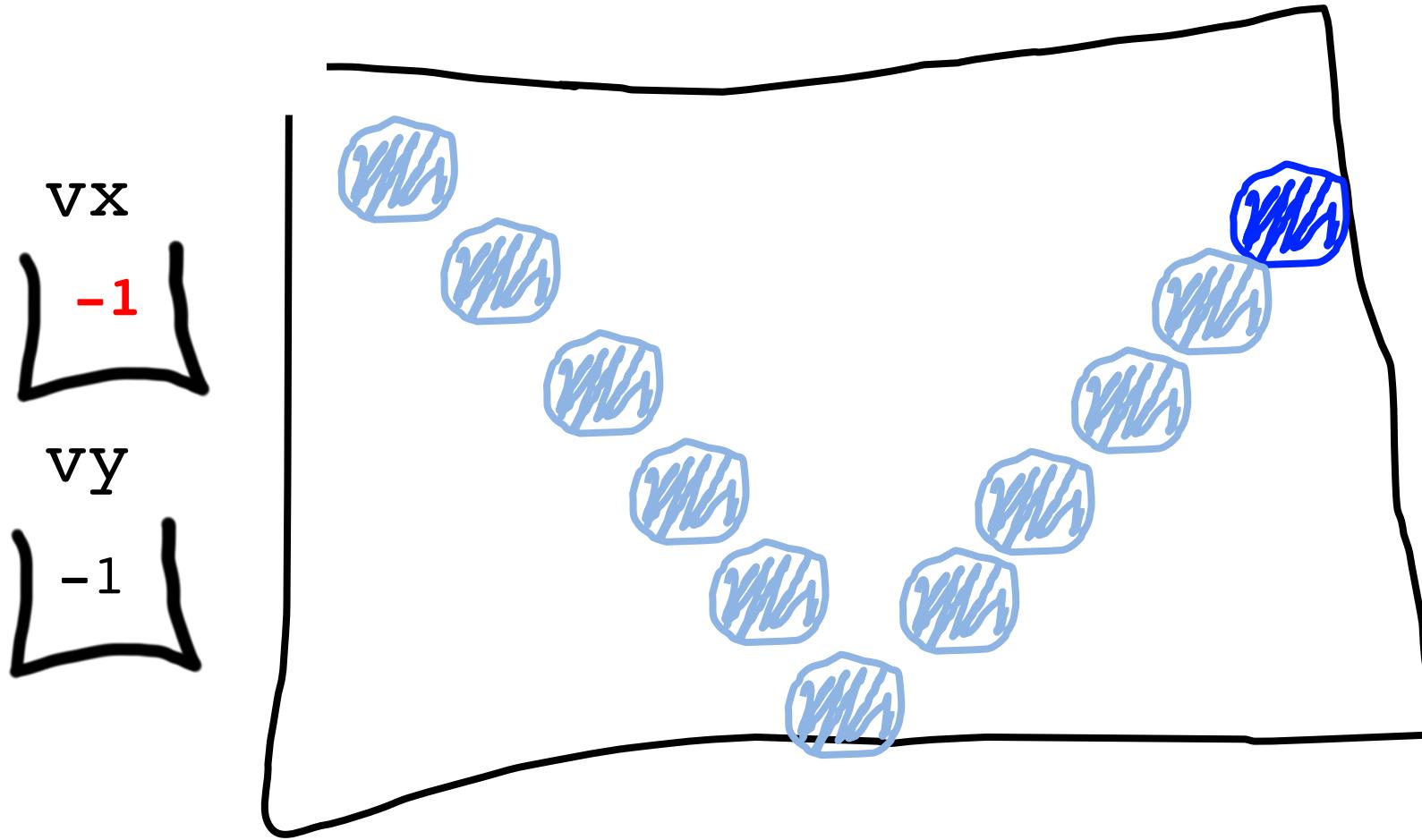


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

Seventh Heartbeat

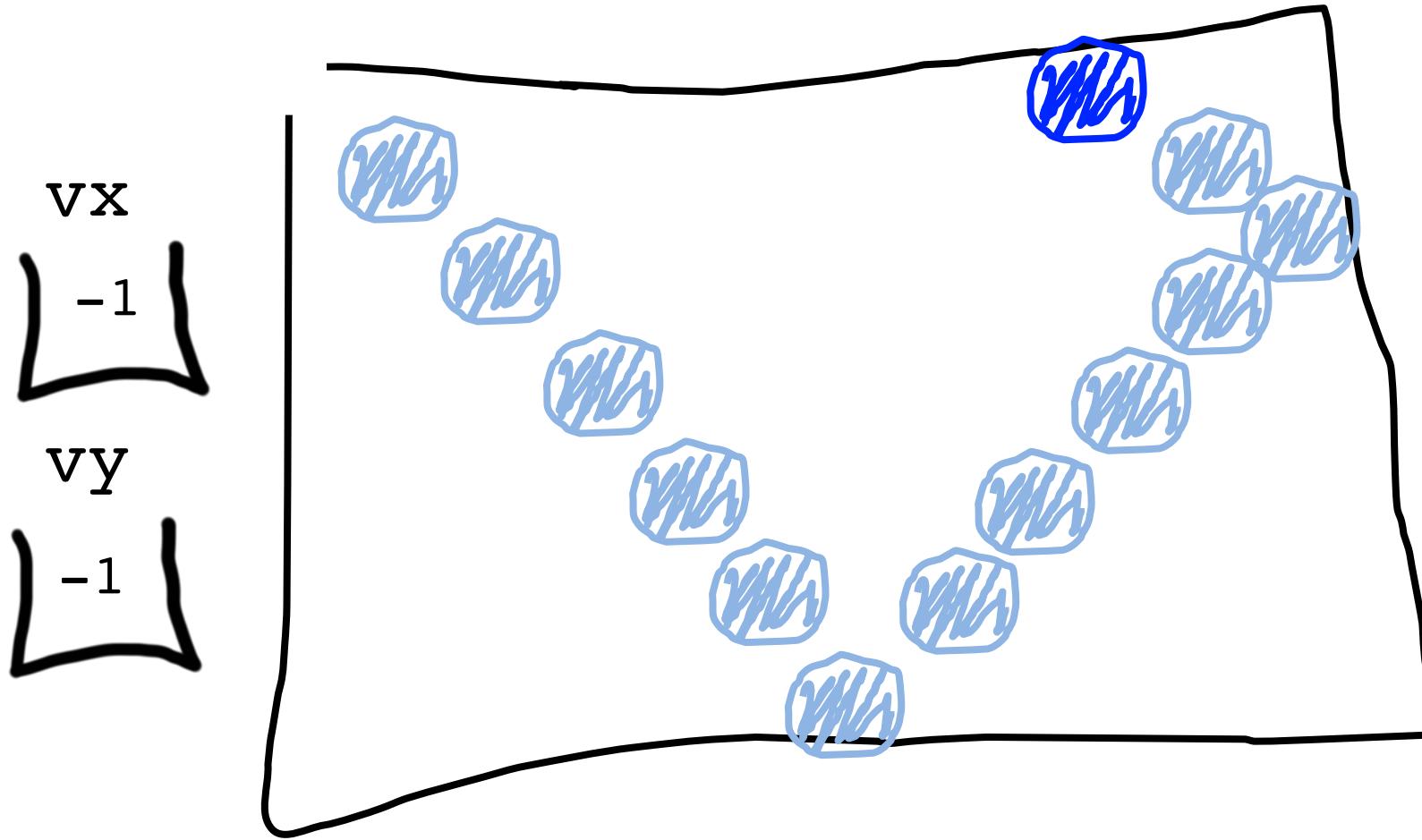


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

Seventh Heartbeat

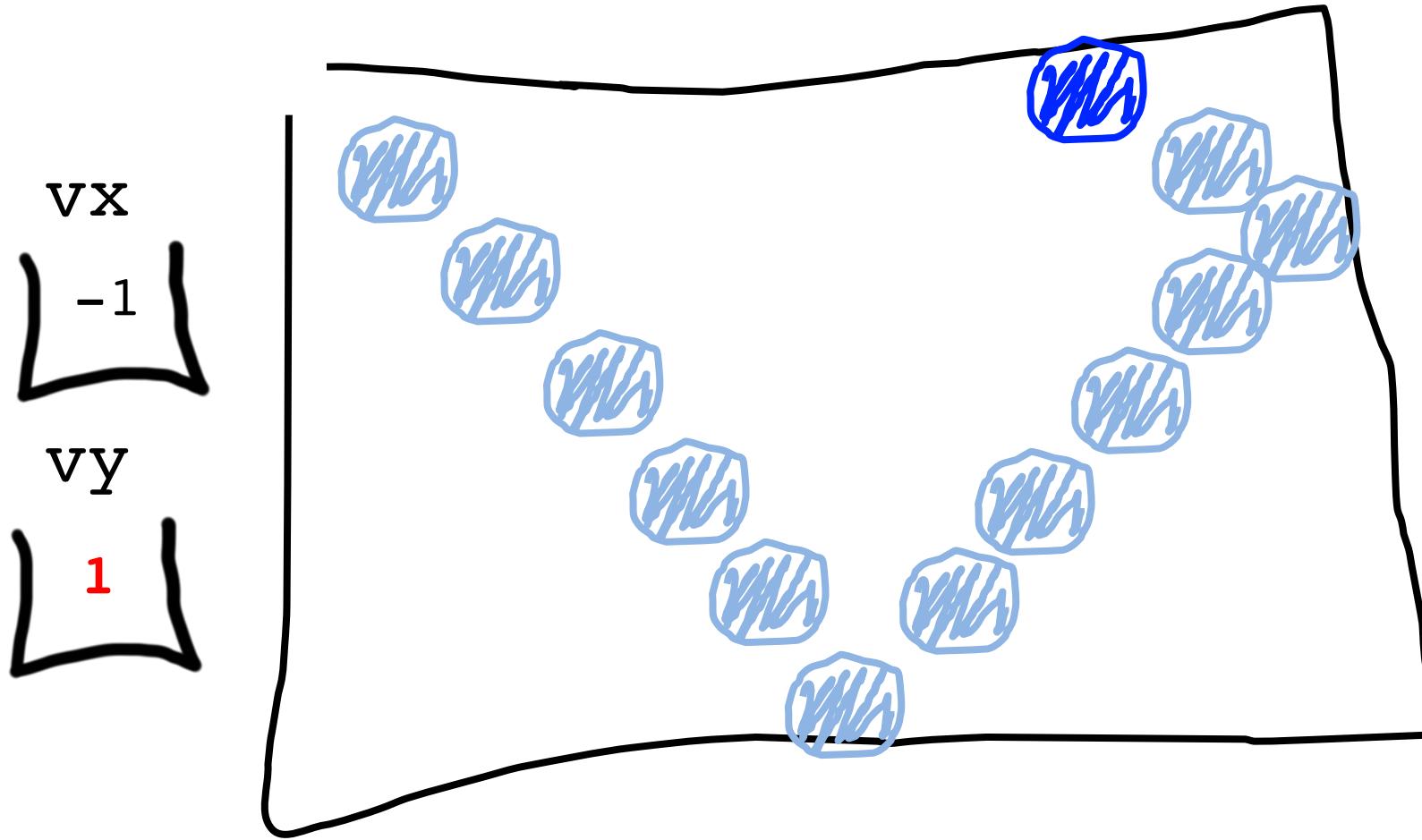


Velocidad: cuántos píxeles
la pelota mueve cada latido



Pelota que rebota

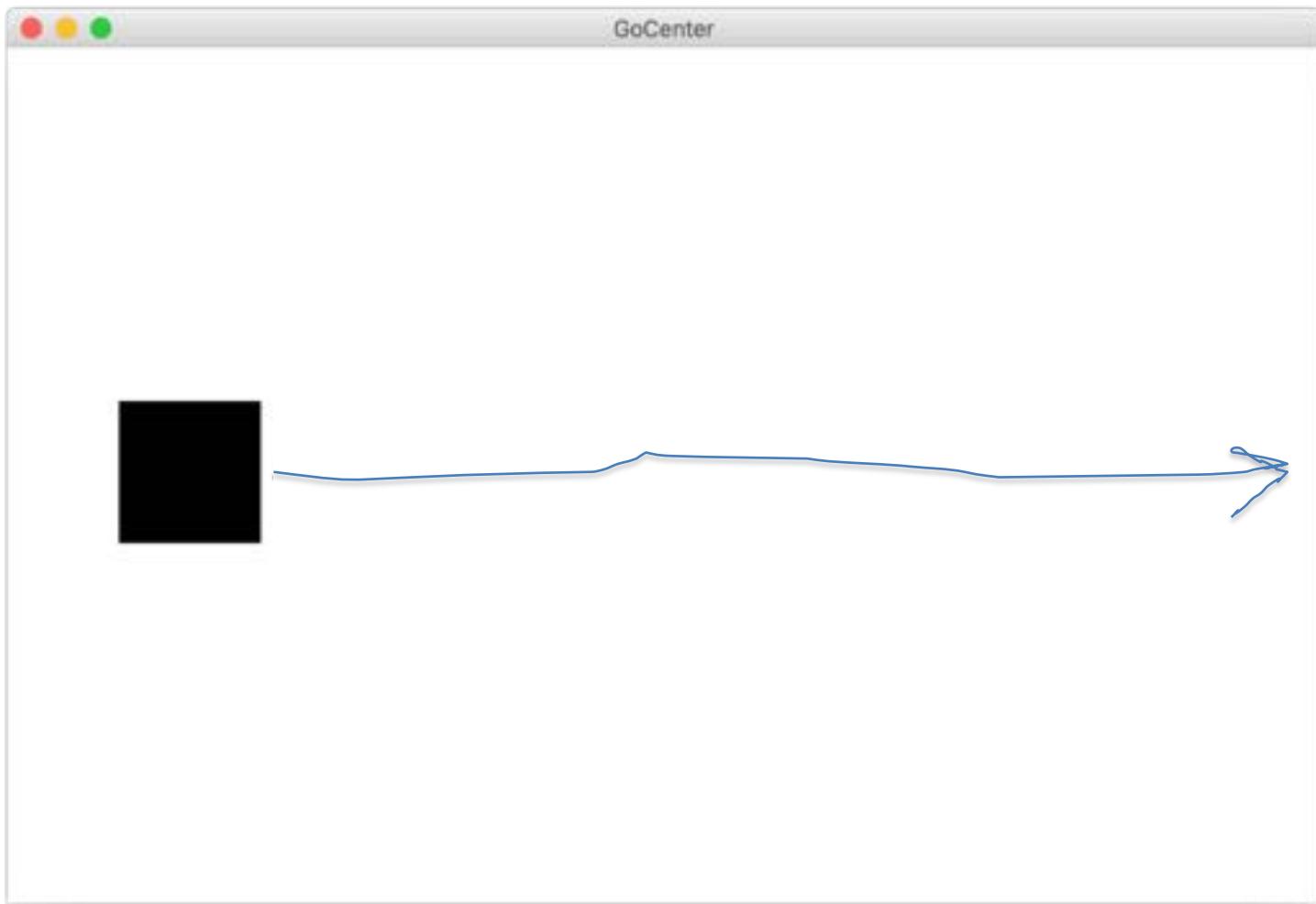
Seventh Heartbeat



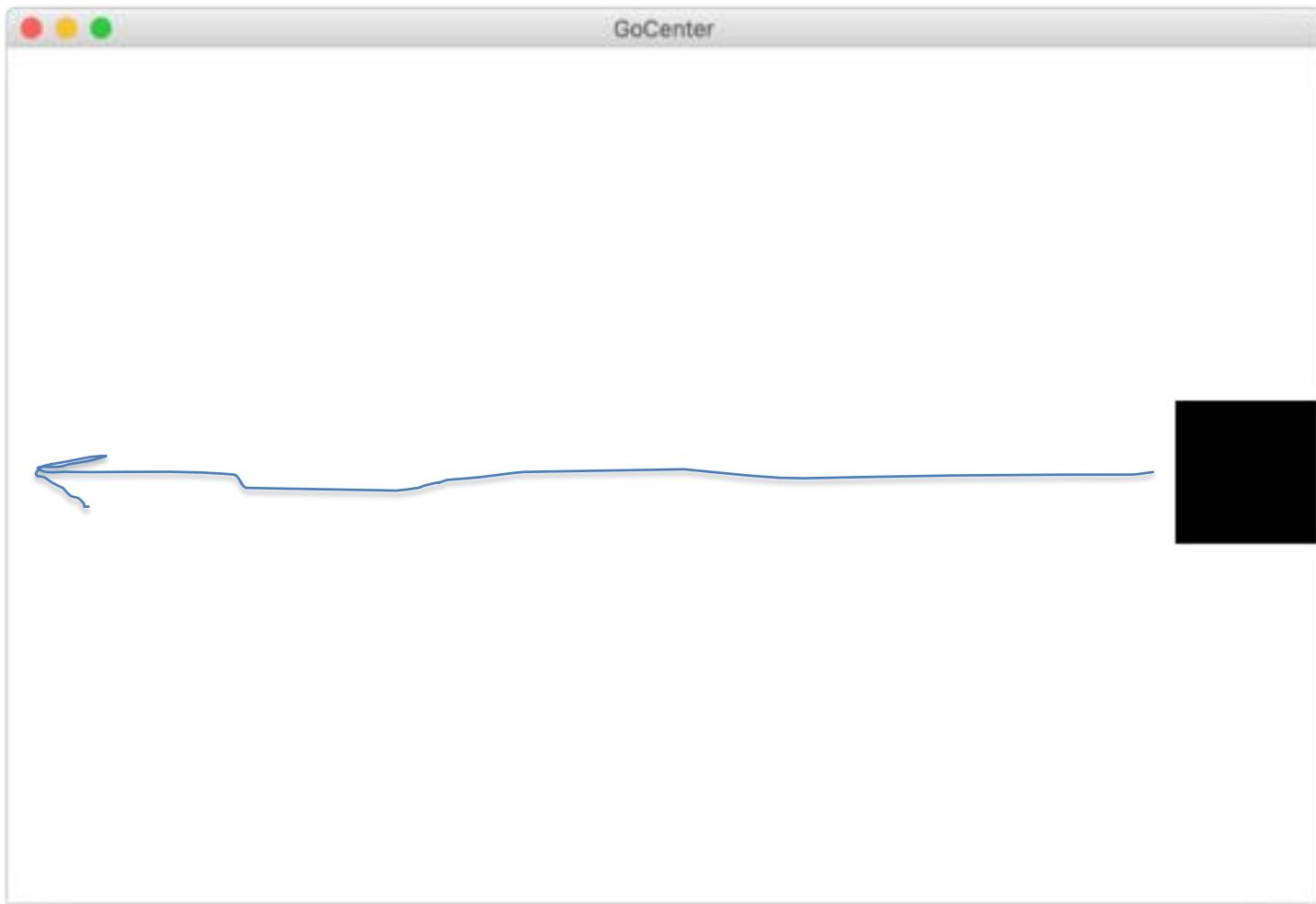
Velocidad: cuántos píxeles
la pelota mueve cada latido



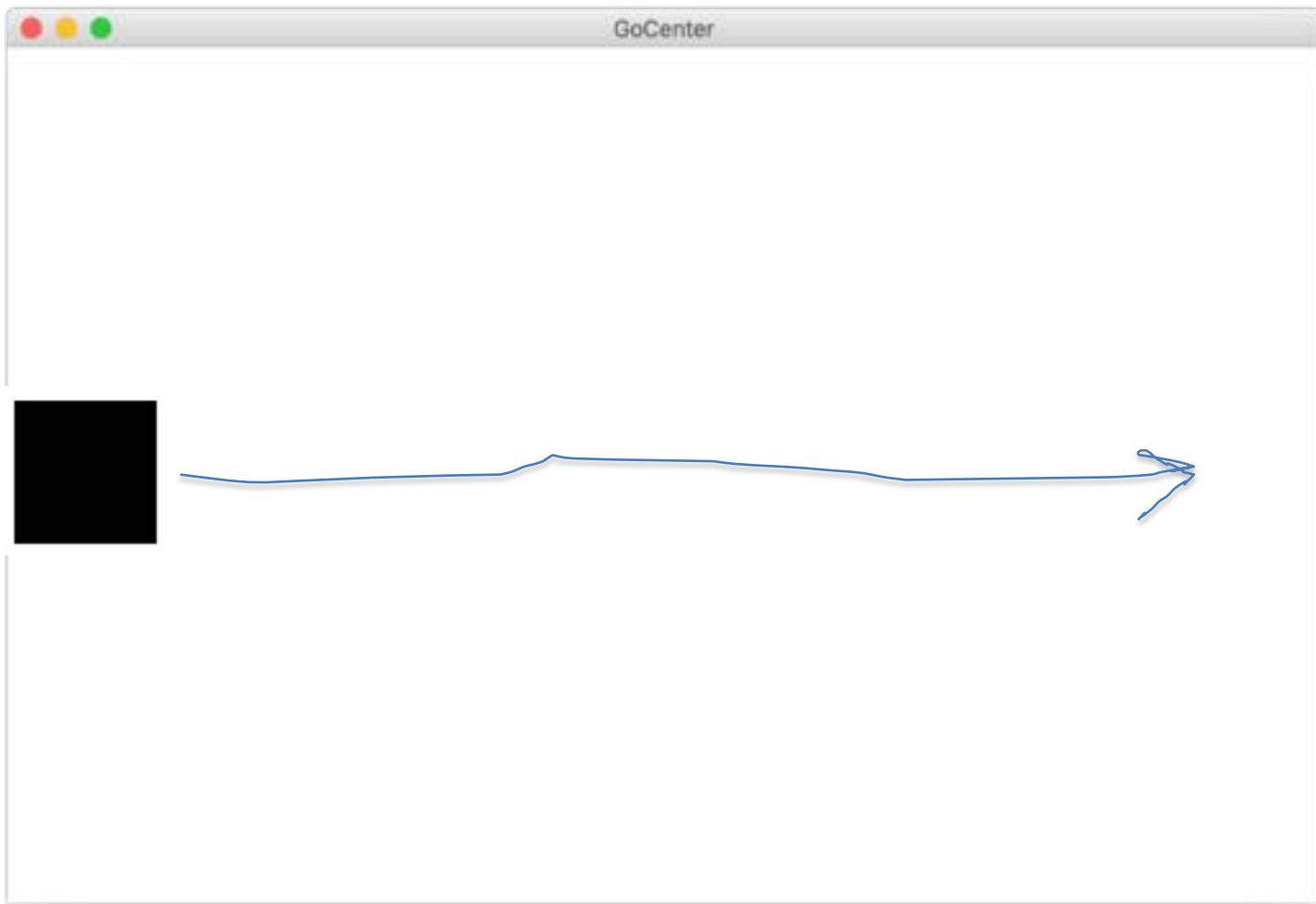
SRect que rebota



SRect que rebota



SRect que rebota



¿Preguntas?