

Muovere una figura
Note molto semplici per **Processing**

LM1, NTA, ABATO, a.a. 25-26

mario chiari

21 novembre 2025

1 Movimento e Velocità.

- a) Muovere una figura orizzontalmente, da Sx verso Dx.
Considera innanzitutto incremento +1 della x .
Cosa succede se incremento x di un valore maggiore, +3, +9, ...?
Come ottengo movimento da Dx a Sx?
Cosa succede se **decremento** il valore di x ?

```
float x = 35;
//float vx = 1;

void setup() {
  size(1400, 760);
  smooth(5);
  fill(255, 204);
  frameRate(25); // change frameRate to check speed effect
}

void draw() {
  background(255);

  line(0, height/2, width, height/2);
  stroke(0);
  strokeWeight(3);
  fill(255,0,0);
  ellipse(width*0.75, height/4, 25, 25); // target
  ellipse(x, height/4, 25, 25); // moving ball
  strokeWeight(1);
  line(x, height/4, width*0.75, height/4);
  // ball moves with constant speed to target
  x = x+1;
  // x += vx; // better code
}
```

- b) Movimento verticale, su e giù.
Modificare valori similamente al punto precedente.

```
float y = 15;
float vy = 1;

void setup() {
  size(1400, 760);
  smooth(5);
  fill(255, 204);
```

```
    frameRate(25); // change frameRate to check speed effect
}

void draw() {
    background(255);

    line(0, height/2, width, height/2);
    stroke(0);
    strokeWeight(3);
    fill(255,0,0);
    ellipse(width*0.5,560,25,25); // target
    ellipse(width*0.5,y,25,25); // moving ball
    strokeWeight(1);
    line(width*0.5, y, width*0.5,560);
    // ball moves with constant speed to target
    y += vy;
}
```

- c) Movimento obliquo, mediante incremento di ambedue le componenti, orizzontale e verticale.

Cosa succede se incrementi le due componenti diversamente, il valore di vx diverso da vy? (Spiega effetti che ottieni).

```
float x = 35;
float vx = 1;
float y = 15;
float vy = 1; // try different values from vx

void setup() {
    size(1400, 760);
    smooth(5);
    fill(255, 204);
    frameRate(25); // change frameRate to check speed effect
}

void draw() {
    background(255);

    line(0, height/2, width, height/2);
    stroke(0);
    strokeWeight(3);
    fill(255,0,0);
    ellipse(35,15, 5,5); // starting position
    ellipse(x,y,25,25); // moving ball
    strokeWeight(1);
```

```
line(35, 15, x, y);  
x += vx;  
y += vy;
```

- d) Nel caso precedente, la figura **non** si muove verso un target.
Il problema è: dato un target, come calcolo gli incrementi vx e vy tali che la figura si muova, dal punto di partenza, in linea retta, verso il punto target?

Prima soluzione, dato la coordinata orizzontale x , trova la corrispondente coordinata verticale y , secondo l'equazione lineare della (unica) linea che unisce i due punti.

```
// given two points, draw a line by its linear equation.  
// you need to calculate slope and y-intercept  
float a1,b1,a2,b2;  
float x,y, m, b;  
  
void setup() {  
    size(1200,760);  
    a1=20; // start point, x component  
    b1=50; // start point, y component  
    a2=1100; // target point, x component  
    b2=670; // target point, y component  
    x=a1; // moving point, x component, start from starting  
        point  
    y=b1; // moving point, y component, start from starting  
        point  
}  
  
void draw() {  
    background(255);  
    fill(0,0,255);  
    ellipse(a1, b1, 15, 15); // start point  
    ellipse(a2, b2, 15, 15); // target point  
    // line(a1,b1,a2,b2);  
    m = (b2 - b1) / (a2 - a1); // slope  
    b = b2-(m*a2); // y-intercept  
  
    x += vx; // next x-axis component  
    y = m * x + b; // corresponding y-axis component,  
        // thanks to linear equation of the line  
    fill(255,0,0);  
    ellipse(x, y, 15, 15); // moving circle;  
    line(a1, b1, x, y); // draw the line  
}
```

- e) **Seconda soluzione** al problema formulato al punto precedente d): calcola l'angolo della linea tra i due punti, e considera le relazioni trigonometriche.
-

```
float x = 35;
float y = 25;
float targetX = 1400*0.75;
float targetY = 760*0.75;
float angle;

void setup() {
    size(1400, 760);
    smooth(5);
    fill(255, 204);
    frameRate(25); // change frameRate to check speed effect
    float dx = targetX - x;
    float dy = targetY - y;
    // arctang(slope) = angolo di pendenza rispetto a x-axis
    angle = atan2(dy, dx);
}

void draw() {
    background(255);
    stroke(0);
    strokeWeight(1);
    line(0, height/2, width, height/2);
    fill(255,255,255);
    ellipse(35,25,5,5); // start point
    fill(255,0,0);
    ellipse(targetX,targetY,25,25); // target ball
    ellipse(x,y,25,25); // moving ball

    line(35, 25, targetX,targetY); // start/target line
    line(x, y, targetX,targetY); // moving ball/target line

    noFill();
    strokeWeight(3);
    x = x+cos(angle); // move to next location, x-component
    y = y+sin(angle); // move to next location, y-component
}
```

- f) Ricorda, la posizione è calcolata ad ogni ciclo di *draw()*, e quindi posso permettere di spostare interattivamente il target, e la figura continuerà a muoversi verso il target.
-

```
float x = 135;
```

```
float y = 125;

void setup() {
  size(1400, 760);
  smooth(5);
  fill(255, 204);
  frameRate(25); // change frameRate to check speed effect
}

void draw() {
  background(255);
  stroke(0);
  strokeWeight(3);
  line(0, height/2, width, height/2);

  float targetX = mouseX; // interactive
  float targetY = mouseY; // interactive
  fill(255,0,0);
  strokeWeight(1);
  ellipse(35,25,12,12); // starting point
  line(35, 25, targetX,targetY); // start/target line
  textSize(16);
  text("Start", 55,25);
  text("Target", targetX+20,targetY);
  ellipse(targetX,targetY,25,25); // target
  ellipse(x,y,145,135); // moving ball

  strokeWeight(1);
  ellipse(35,25,12,12); // starting point

  line(x, y, targetX,targetY); // moving ball/target line
  line(x-50,y,width-150,y);
  float dx = targetX - x;
  float dy = targetY - y;
  float angle = atan2(dy, dx);

  noFill();
  strokeWeight(3);
  if(angle>0){arc(x, y, 120, 120, 0, angle,
    OPEN);}else{arc(x, y, 120, 120, angle, 0, OPEN);}
  text(degrees(angle), x+65, y+25); // to debug
  x = x+cos(angle); // move to next location, x-component
  y = y+sin(angle); // move to next location, y-component
}
```

g) **Accelerazione.**

L'accelerazione è un incremento di velocità.

Un effetto di accelerazione si realizza incrementando la quantità di spostamento (= velocità) a ogni ciclo di *draw()*.

Movimento orizzontale accelerato.

```
float x = 35;
float vx = 1;

void setup() {
    size(1400, 760);
    smooth(5);
    fill(255, 204);
    frameRate(25);
}

void draw() {
    background(255);
    line(0, height/2, width, height/2);
    stroke(0);
    strokeWeight(3);
    fill(255,0,0);
    ellipse(x, height/4, 25, 25); // moving ball
    strokeWeight(1);
    line(x, height/4, width*0.75, height/4);
    // velocity increases
    vx += 1; // try with vx +=0.5, +=1, +=vx
    // ball moves with accelerated speed
    x += vx;
    print("vx: "+vx+"\n"); // to debug
}
```

h) **Accelerazione.**

Movimento accelerato obliquo, ovvero in ambo le componenti (usando trig).

```
float x = 35;
float y = 25;
float targetX;
float targetY;
float vx = 0; // note: you need to start with value 0, why?
float vy = 0; // note: you need to start with value 0, why?
float acc = 0.1; // increase to speed up

void setup() {
    size(1400, 760);
```

```
    smooth(5);
    fill(255, 204);
    targetX = width*0.85;
    targetY = height*0.85;
    frameRate(15); // change frameRate to check speed effect
}

void draw() {
    background(255); // comment out to check successive
                     // locations
    stroke(0);
    strokeWeight(3);
    line(0, height/2, width, height/2);
    fill(255,0,0);
    ellipse(35,25,12,12); // starting point
    textSize(16);
    text("Start", 55,25);
    text("Target", targetX+20,targetY);
    ellipse(targetX,targetY,25,25); // target
    strokeWeight(1);
    line(35, 25, targetX,targetY); // start/target line

    float dx = targetX - x;
    float dy = targetY - y;
    float angle = atan2(dy, dx);
    acc +=0.1;
    float ax = cos(angle)*acc; // we use it as the next
                             // increase of velocity
    float ay = sin(angle)*acc;
    vx +=ax; // increase of speed
    vy +=ay; // increase of speed
    print("vx: "+vx+", vy: "+vy+"\n"); // to check speeds
    x+=vx;
    y+=vy;

    if(x+vx<targetX){
        circle(x,y,15); // moving ball
        strokeWeight(3);
        line(x, y, targetX, targetY); // moving ball/target
        line
    }else{
        fill(0,0,255);
        circle(min(x+vx, targetX), min(y+vy, targetY),15);
        // ball on target
        noLoop();
    }
}
```

```
}
```

i) Rallentamento (**Easing**).

La figura rallenta, avvicinandosi al target.

```
float x = 35;
float y = 760*0.25;
float targetX = 1400*0.75;
float targetY = 760*0.25;
float easing = 0.02; // why doesn't it work with =1?
float vx = 0;

void setup() {
    size(1400, 760);
    fill(255, 204);
    frameRate(25);
}

void draw() {
    background(255);
    stroke(0);
    strokeWeight(6);
    line(0, height/2, width, height/2);
    fill(255,0,0);
    ellipse(35,height/4,12,12); // starting point
    ellipse(targetX, targetY, 25, 25); // target
    ellipse(x, y, 25, 25); // moving ball
    // ball moves to target,
    // slowing down as it get closer to target (easing)
    float dx = targetX - x;
    vx = dx * easing; // dx is decreasing
    if(dx>1){
        strokeWeight(1);
        line(x, height/4, width*0.75, height/4);
        x = x+vx;
    }
}
```

j) Disegna una circonferenza (mediante funzioni trig fondamentali). (Esercizio, ottieni effetti di accelerazione, rallentamento, come sopra ma per il movimento circolare).

```
float x;
float y;
float speed = 0.015; // angle increase
```

```
float a=45;

void setup() {
  size(400,400);
}
void draw() {
  background(255); // check uncommented
  fill(255,0,0);
  x = width/2 + cos( a )*120; // next point on the circle
  y = height/2 + sin( a )*120; // next point on the circle
  line(width/2,height/2,x,y);
  ellipse(x,y,5,5);

  a+=speed; // angle increase
  // speed+=0.001; // uncomment to speed up
}
```

k) Disegna una spirale, cosa si deve incrementare? Perché?

```
float angle = 0.015;
float a=-90;
float s=0;

void setup() {
  size(900,900);
  background(255);
}

void draw() {
  // background(0);
  fill(0);
  float px = width/2 + cos( a )*(60+s );
  float py = height/2 + sin( a )*(60+s );
  ellipse(px,py,1,1);
  a+=angle;
  s+=0.05;
}
```
