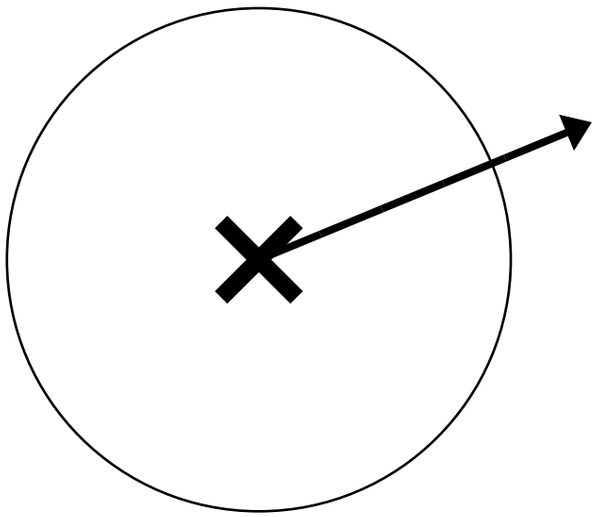
The Agent

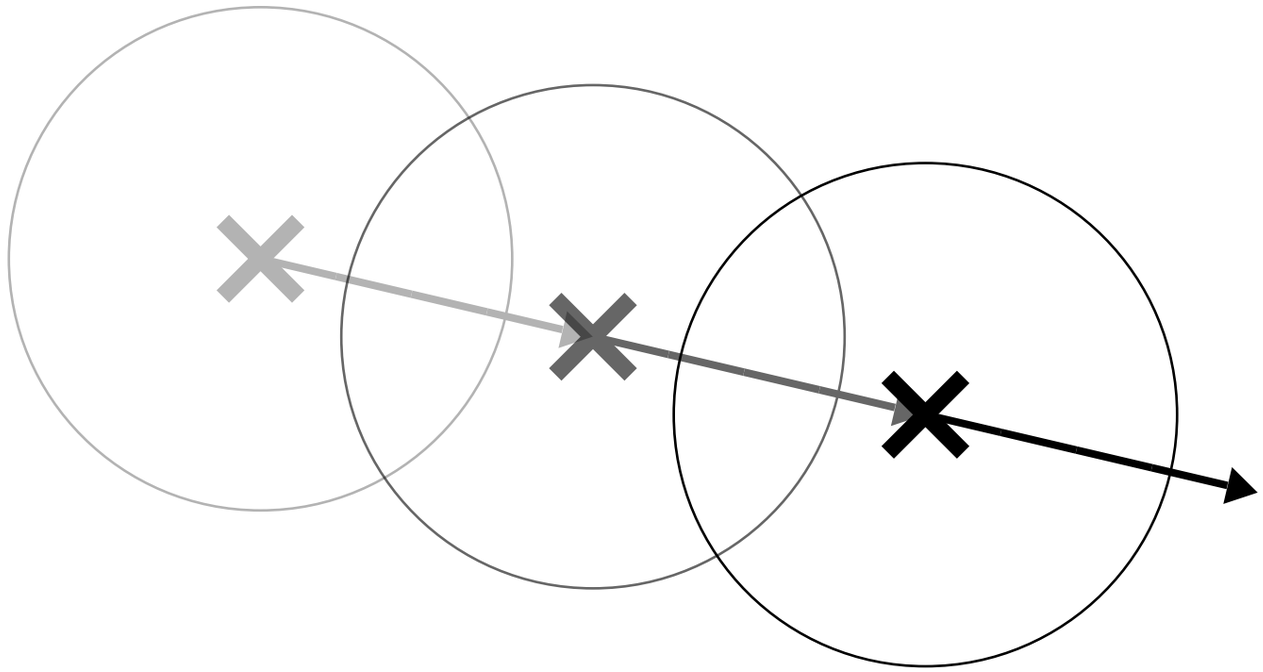
step01_asimpleagent



```
Agent myAgent;

void setup() {
  size(320, 240);
  smooth();
  noFill();
  ellipseMode(CENTER);

  myAgent = new Agent();
  myAgent.position.set(width/2, height/2);
  myAgent.velocity.set(14, 18);
  myAgent.radius = 15;
}

void draw() {
  background(255);
  myAgent.draw();
}
```

```
class Agent {

  Vector2f position = new Vector2f();
  Vector2f velocity = new Vector2f();
  float radius = 0;

  void draw() {
    stroke(0);
    ellipse(position.x, position.y, radius, radius);
    stroke(255, 0, 0);
    line(position.x, position.y,
         position.x + velocity.x, position.y + velocity.y);
  }
}
```

step02_movinginadirection

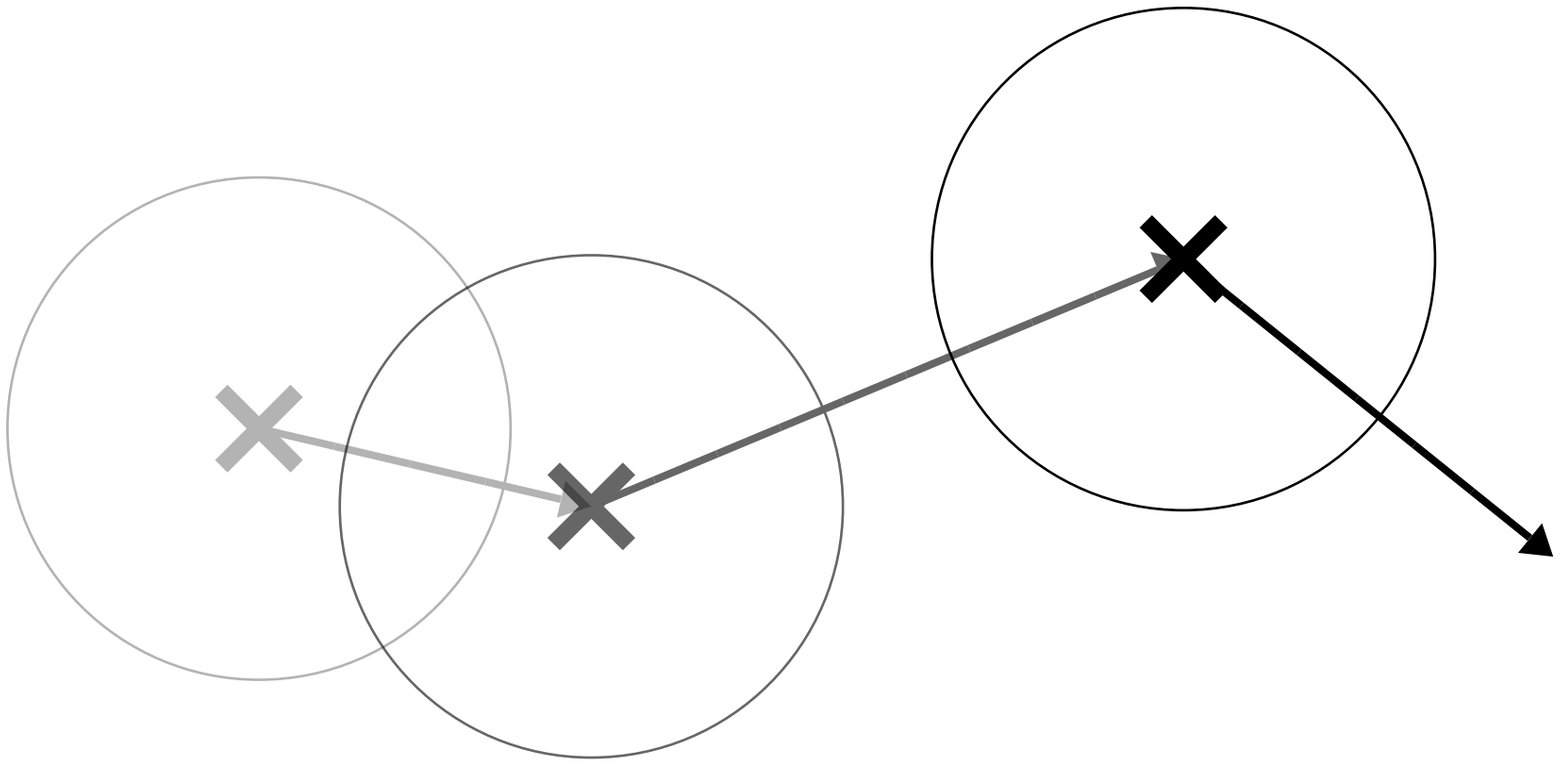


```
class Agent {

  Vector2f position = new Vector2f();
  Vector2f velocity = new Vector2f();
  float radius = 0;

  void draw() {
    stroke(0);
    ellipse(position.x, position.y, radius, radius);
    stroke(255, 0, 0);
    line(position.x, position.y,
         position.x + velocity.x, position.y + velocity.y);
  }
}
```

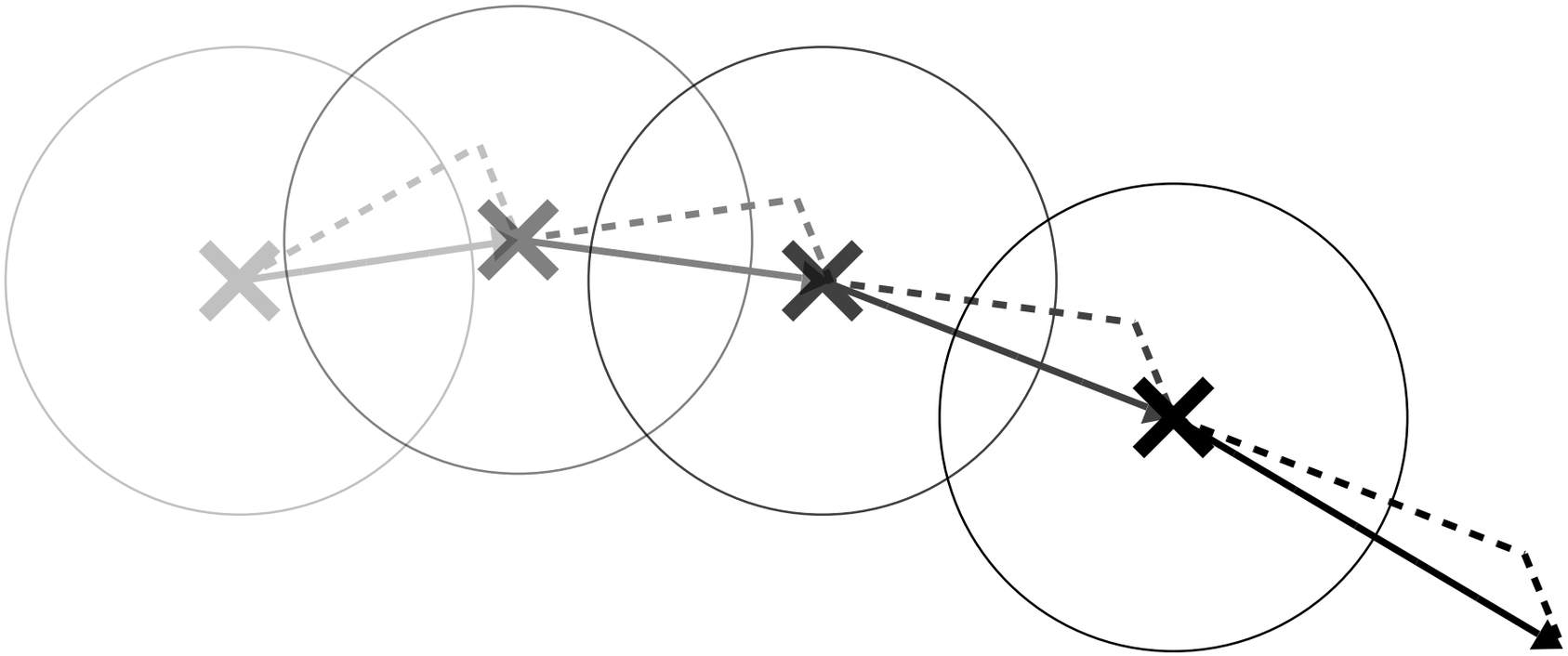
```
class Agent {

  Vector2f position = new Vector2f();
  Vector2f velocity = new Vector2f();
  float radius = 0;

  void draw() {
    stroke(0);
    ellipse(position.x, position.y, radius, radius);
    stroke(255, 0, 0);
    line(position.x, position.y,
         position.x + velocity.x, position.y + velocity.y);
  }

  void loop() {
    position.add(velocity);
  }
}
```

```
/* SIMPLE */

  void loop() {
    position.add(velocity);
  }

/* TIME BASED */

  void loop(float theDeltaTime) {

    float myAccelerationSpeed = acceleration.length();

    if (myAccelerationSpeed > maxacceleration) {
      acceleration.normalize();
      acceleration.multiply(maxacceleration);
    }

    Vector2f myTimerAcceleration = new Vector2f();
    myTimerAcceleration.set(acceleration);
    myTimerAcceleration.multiply(theDeltaTime);

    velocity.add(myTimerAcceleration);

    float mySpeed = velocity.length();
    if (mySpeed > maxspeed) {
      velocity.normalize();
      velocity.multiply(maxspeed);
    }

    Vector2f myTimerVelocity = new Vector2f();
    myTimerVelocity.set(velocity);
    myTimerVelocity.multiply(theDeltaTime);

    position.add(myTimerVelocity);
  }
```

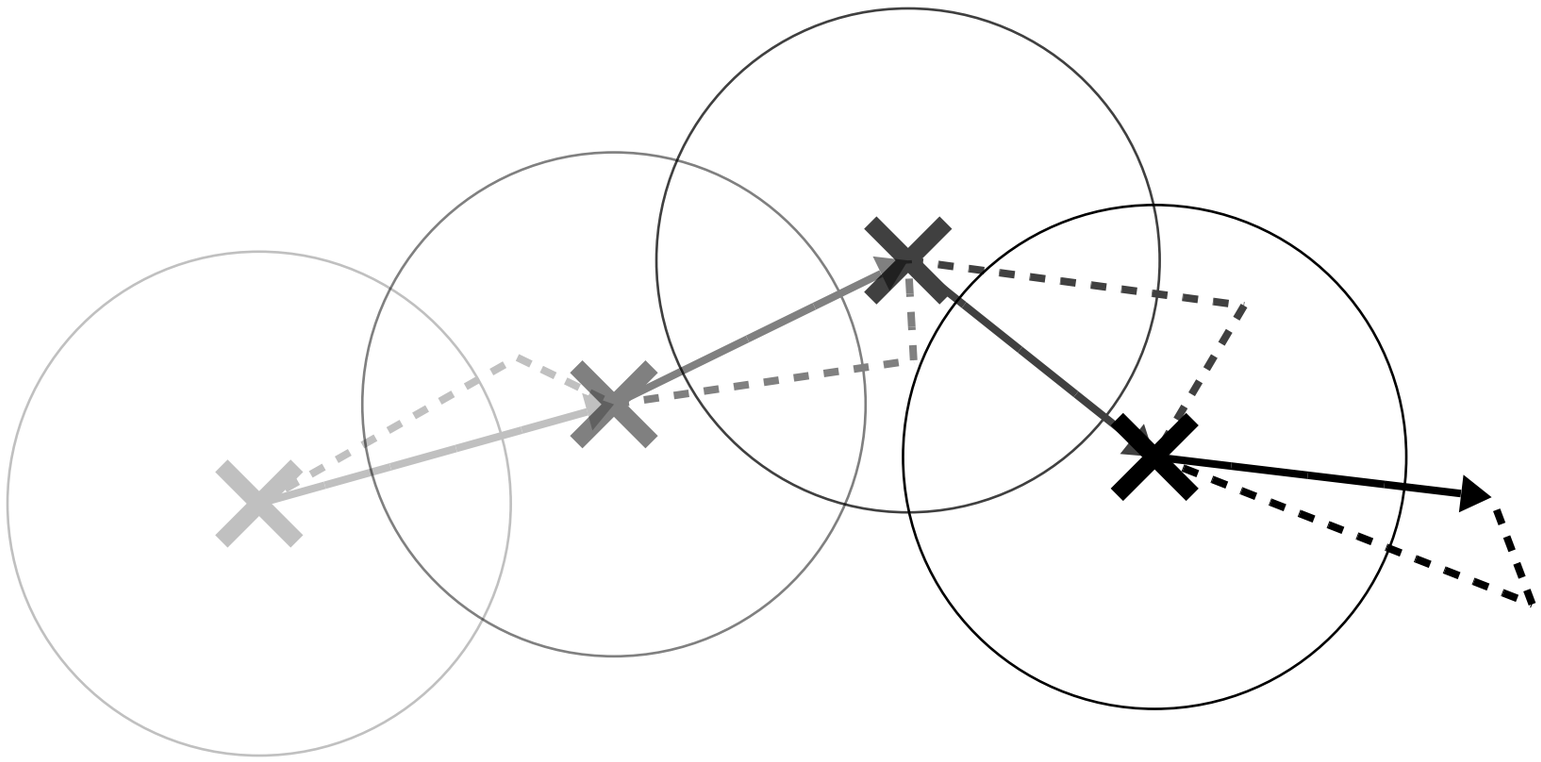
step03_movingindifferentdirections



step04_movingsmoothlyindifferentdirections



step05_movingatconstantspeed



step06_turningatconstantspeed
step07_introducingtime