games

learning objectives

- writing games!

- understanding the built-in 2D game engine
- events in games
- using art (pictures, sounds)

language recap

- **local variables**: store and reuse a value

```
var x := 5
x->post to wall
```

- **if statement**: make decisions!

```
if x = 5 then ...
else ...
```

- **for loop**: repeat code multiple times

```
for 0 <= i < n do
    ...
```

global variables

- **a global variable holds a value in memory and "on disk";**
- **a global variable can be read or written too;**
- **A global variable is available everywhere**
  - definition

```
data->x := 0
◳x := 0
```

  - reading

```
◳x->post to wall
```

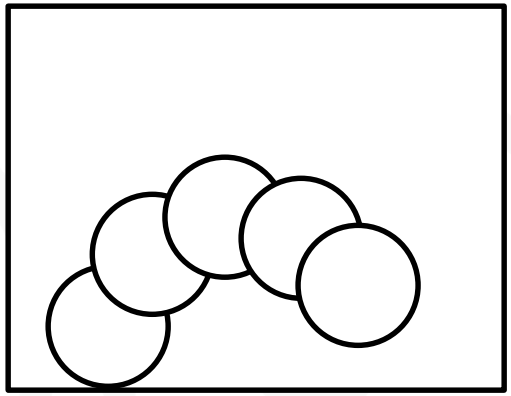
  - update with new value

```
◳x := 5
```

◳ is the shorthand for 'data'

fruit ninja lite

- one sprite shoots up from the bottom of the screen
- user needs to tap the sprite before sprite falls back on the bottom
- when tapped, the sprite shoots up again from the bottom
- 1 point per tap successful
- game over if 3 missed sprites

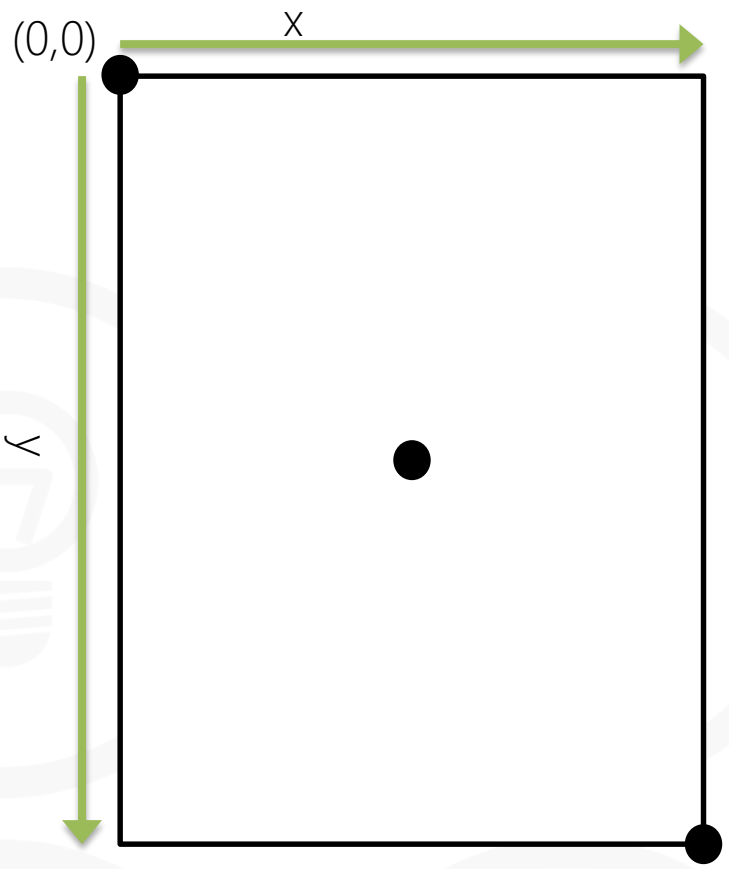


board

- 2D sprite-based game engine
- physics: sprites have speed, gravity, friction
- game loop: timer event to ‘step’ the world
- touch events on sprites

coordinates and units

- origin (0, 0) is top left corner
- landscape or portrait
- **y coordinates positive going down**



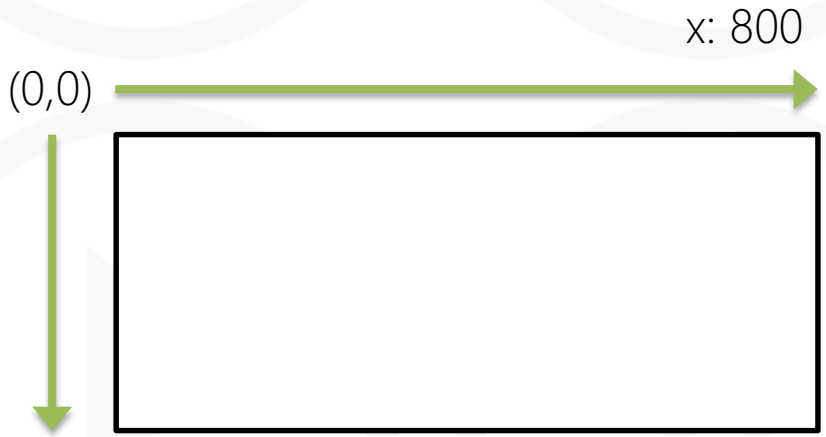
1. basic game template

global variable

creates a 2D game engine

```
var board: Board
action main
    board := media->create landscape board(800,480)
    board->post to wall
```

display on screen

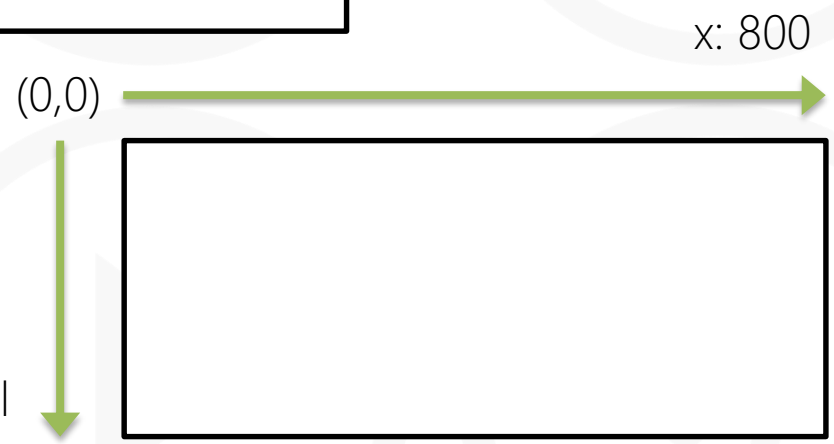


1.5 setting up physics

```
var board: Board
action main
    board := media->create landscape board(800,480)
    board->post to wall
    board->set gravity(0, 700)
event gameloop
    board->evolve
    board->update on wall
```

gravity falling down

applies physics every 50 ms



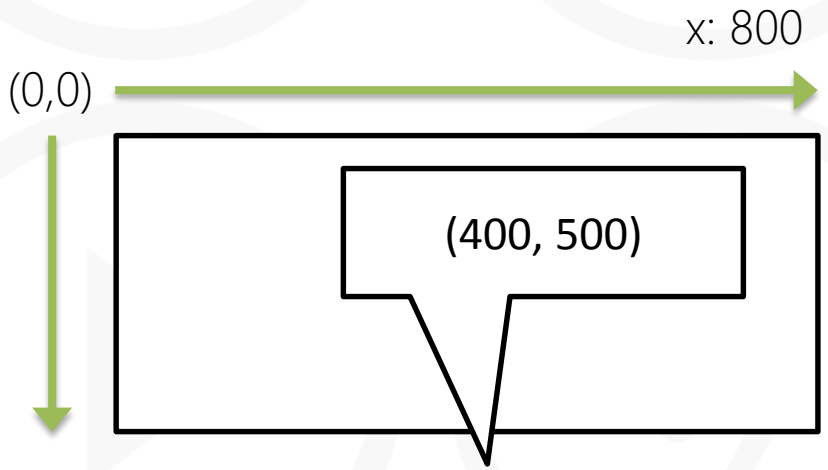
2. creating a sprite

- create an ellipse in the board
- store the sprite in a local variable
- promote the sprite variable to data

3. sprite shooting from the bottom

- position the sprite just above bottom of the board
- set speed such that the sprite goes up the screen

```
action shoot
    fruit->set pos(400, 500)
    fruit->set speed(0, -800)
action main
    ...
    code->shoot
```



tap sprite event

- **`tap sprite`** is an event raised when the user taps a sprite. Attaches to a global variable of kind **`Sprite`**.

```
event tap sprite: fruit(
        sprite : Sprite,
        x : Number,
        y : Number)
```

name of the global variable

(x,y) is the coordinate of the tap

evelop

4. tap and shoot

- When sprite is tapped, shoot the sprite from the bottom again

```
event tap sprite: fruit(...)
    code->shoot
```

5. score

- on tap, increase the score by 1

```
action main
    ...
    ◱score := 0
event tap sprite: sprite(...)
    ...
    ◱score := ◱score + 1
```

6. when the sprite falls through the bottom…

- the sprite shoots back

```
action gameloop
    ...
    if ◱fruit->y > 480
        and ◱fruit->speed y > 0
    then
        code->shoot
```

beyond the bottom

going down

7. when the sprite falls through the bottom for the 3rd time…

- the user has 3 chances and them game over

```
action main
    ...
    ◳life := 3
action gameloop
    ...
    if ◳fruit->y > 800
       and ◳fruit->speed y > 0
    then
       ◳life := ◳life - 1
       if ◳life < 1
       then
          ◳score->post to wall
          time->stop
```

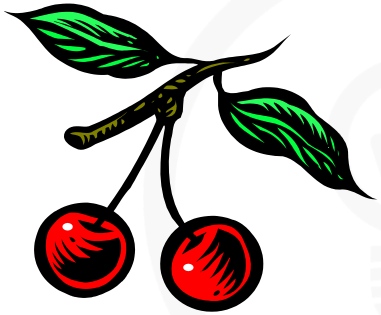
one life lost

no more life, stop the game

customizations

art - pictures

- using pictures in your games



background

- set screen background color or image

```
wall->set background(color->blue)
```

or...

```
wall->set backround picture(art->background)
```

picture sprite

- use **`create picture`** instead of **`create ellipse`**

```
action main
    ◳fruit := ◳board->create ellipse(100,100)
    ◳fruit := ◳board->create picture(art->fruit)
```

- resize the picture (if needed)

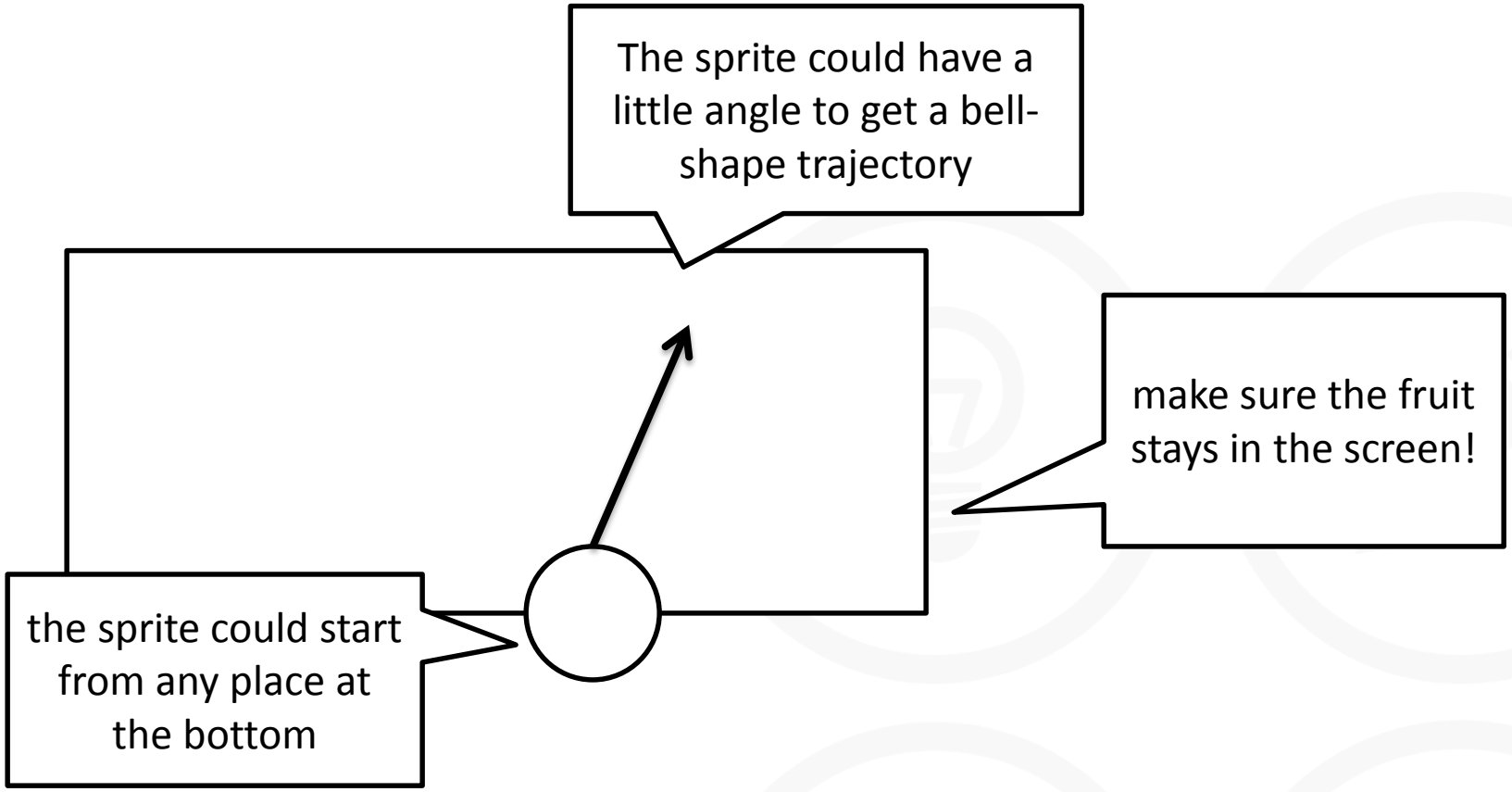
```
    var pic := art->picture->clone
    pic->resize(100,-1)
    ◳fruit := ◳board->create picture(pic)
```

art - sounds

- same as picture for sounds
  - *still under development in web version*
- When sprite is tapped, play the splash sound

```
event tap sprite: sprite(...)
    code->shoot
    art->splash->play
```

randomness fruit trajectories



randomness fruit trajectories

- `math->random(5)` returns a number between 0 and 5 not included, i.e. 0,1,2,3 or 4.

tweak these values!

```
action shoot
    var x := 120 + math->random(540)
    fruit->set pos(x, 500)
    var speedx := -50 + math->random(100)
    fruit->set speed(speedx, -800)
```

leaderboards

- use `bazaar->post leaderboard score(score)` to publish a score online
- use `bazaar->post leaderboard to wall` to display the leaderboard

- leaderboards only work on published scripts