gosu

https://www.libgosu.org

https://github.com/gosu/gosu

- silnik 2D
- bazuje na OpenGL
- wysokopoziomowy
- ~100 metod w 9 klasach
- uproszczona składnia, szybki rozwój

Instalacja

```
$ gem install gosu
```

```
int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR lpCmdLine,
                   int nCmdShow)
{

    HWND hWnd;
    WNDCLASSEX wc;

    ZeroMemory(&wc, sizeof(WNDCLASSEX));
…
}

CreateWindowEx(NULL,
                          L"WindowClass1",
                          L"Our First Windowed Program",
                          WS_OVERLAPPEDWINDOW,
                          300,
                          300,
                          500,
                          400,
                          NULL,
                          NULL,
                          hInstance,
                          NULL);


    ShowWindow(hWnd, nCmdShow);
…
```

```ruby
require 'gosu'
Gosu::Window.new(640, 480, false).show
```

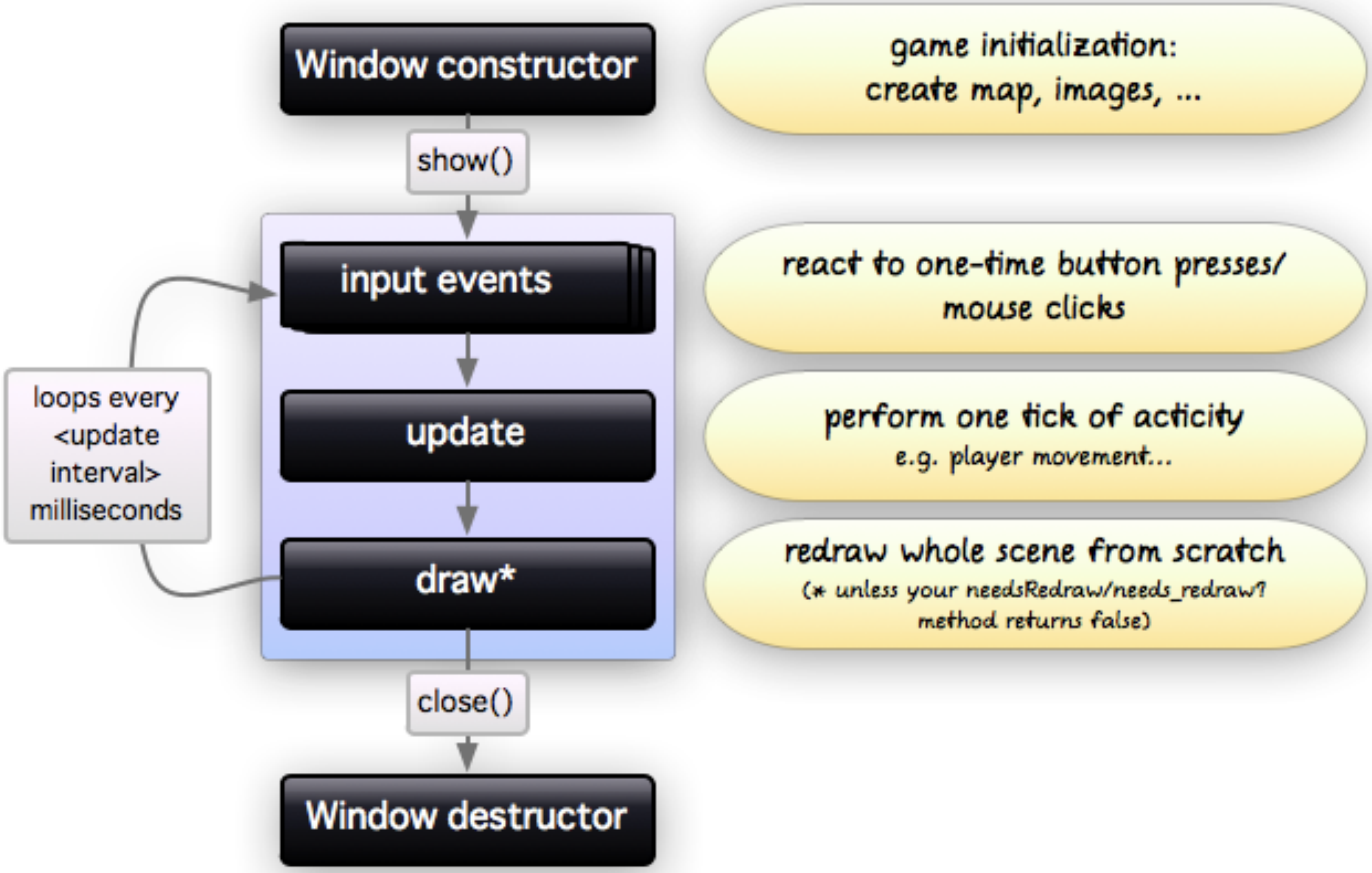
Definicja okna

```
class GameWindow < Gosu::Window

  def initialize
    super(640, 480)
    self.caption = "Gosu Tutorial Game"
  end

  def update

  end

  def draw

  end

end

window = GameWindow.new
window.show
```

konstruktor

główna pętla - logika

rendering



Window constructor

show()

input events

update

draw*

close()

Window destructor

loops every <update interval> milliseconds

game initialization: create map, images, ...

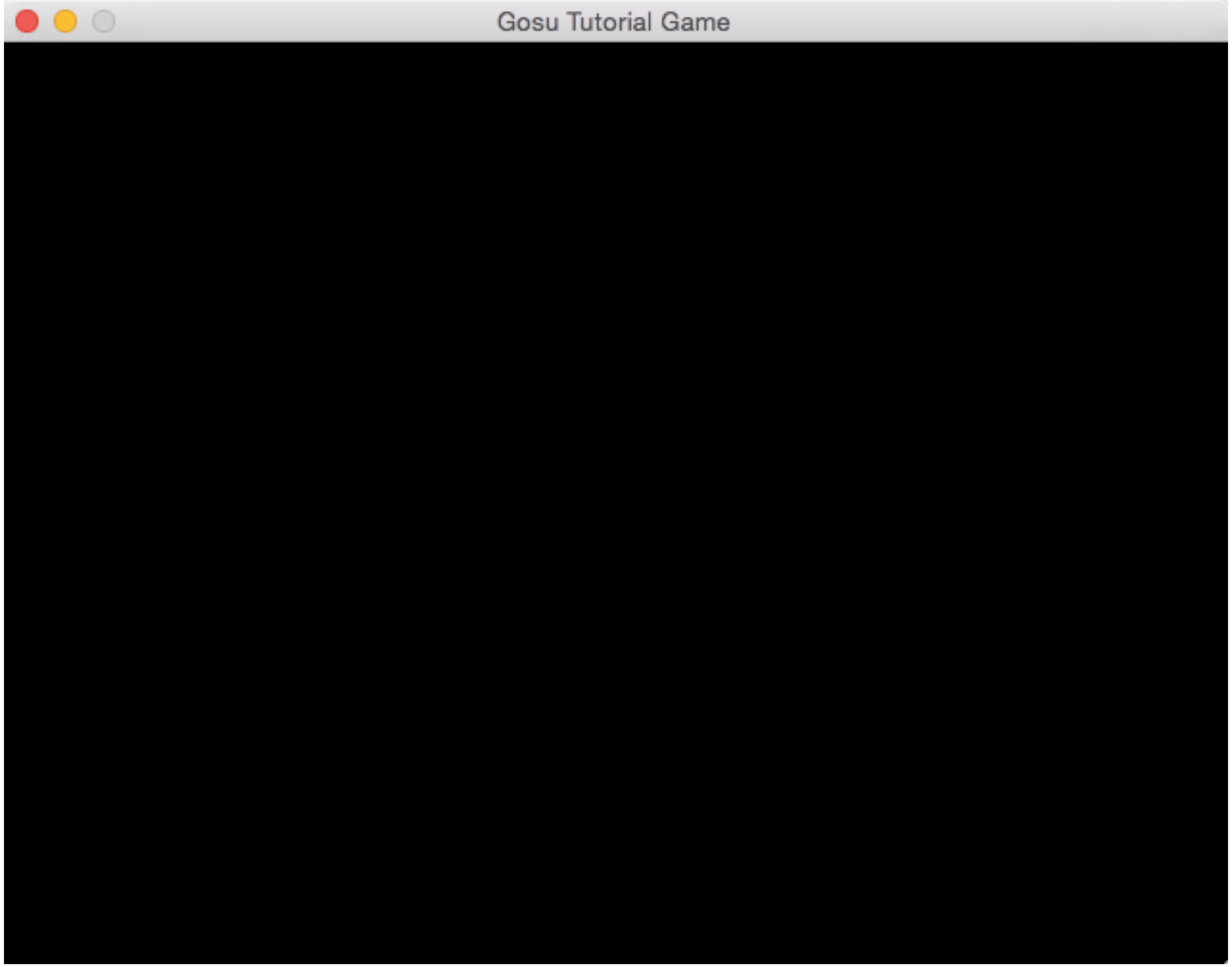
react to one-time button presses/ mouse clicks

perform one tick of acticity

e.g. player movement...

redraw whole scene from scratch

(* unless your needsRedraw/needs_redraw? method returns false)

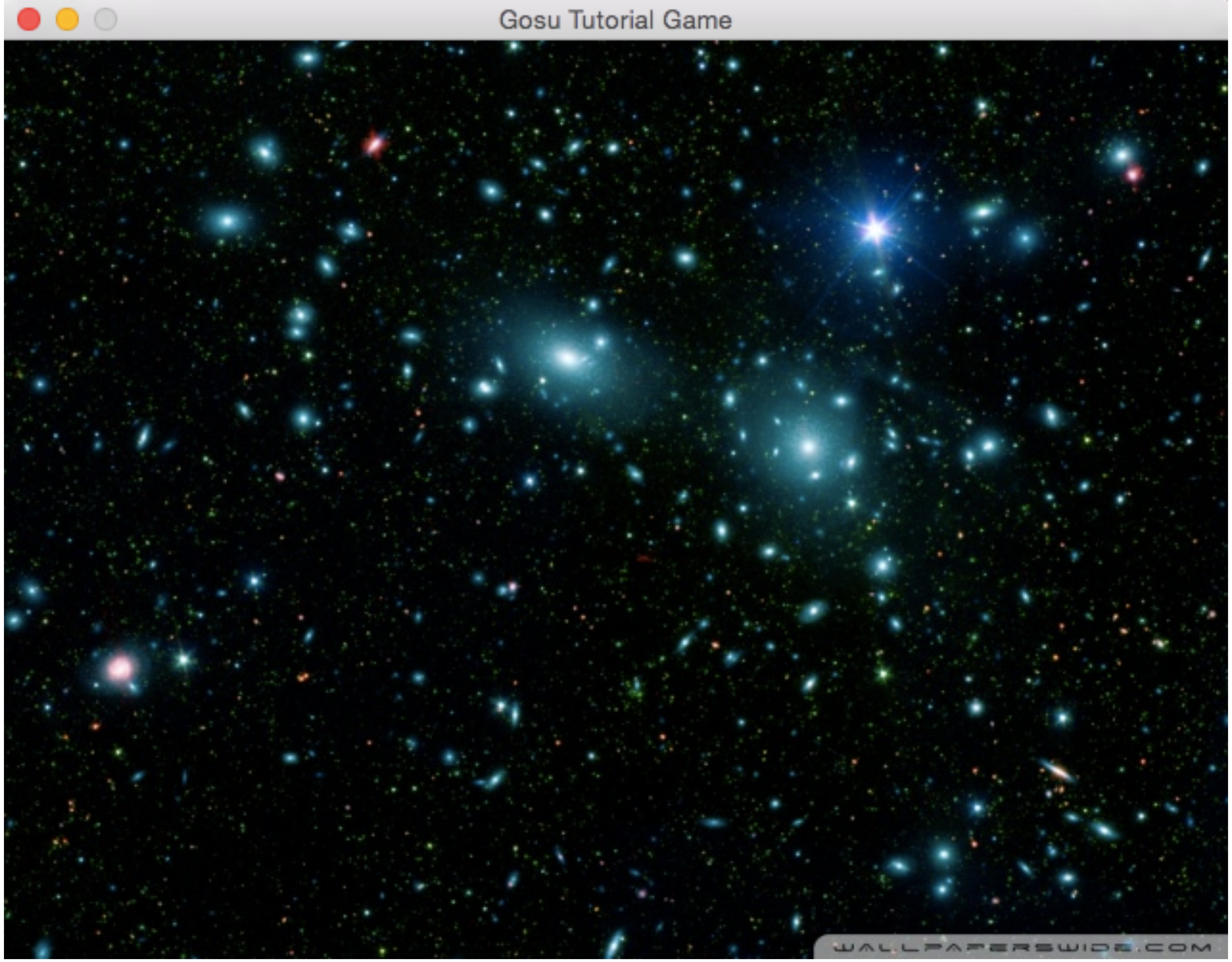


Gosu Tutorial Game

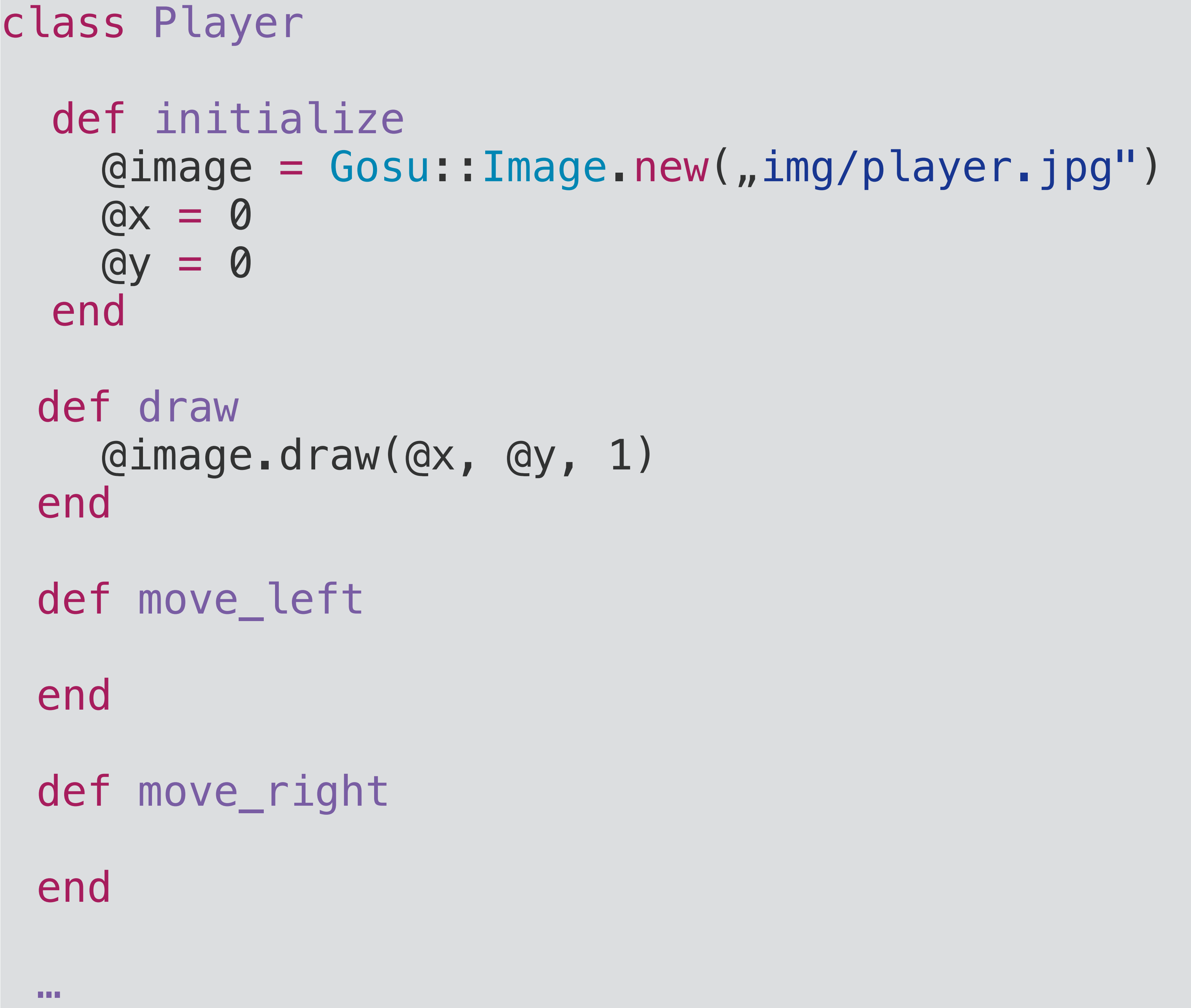
draw()

```
def initialize
…
 @image = Gosu::Image.new("img/background.jpg", :tileable => true)
…
end
```

```
def draw
 @image.draw(0, 0, 0)
end
```

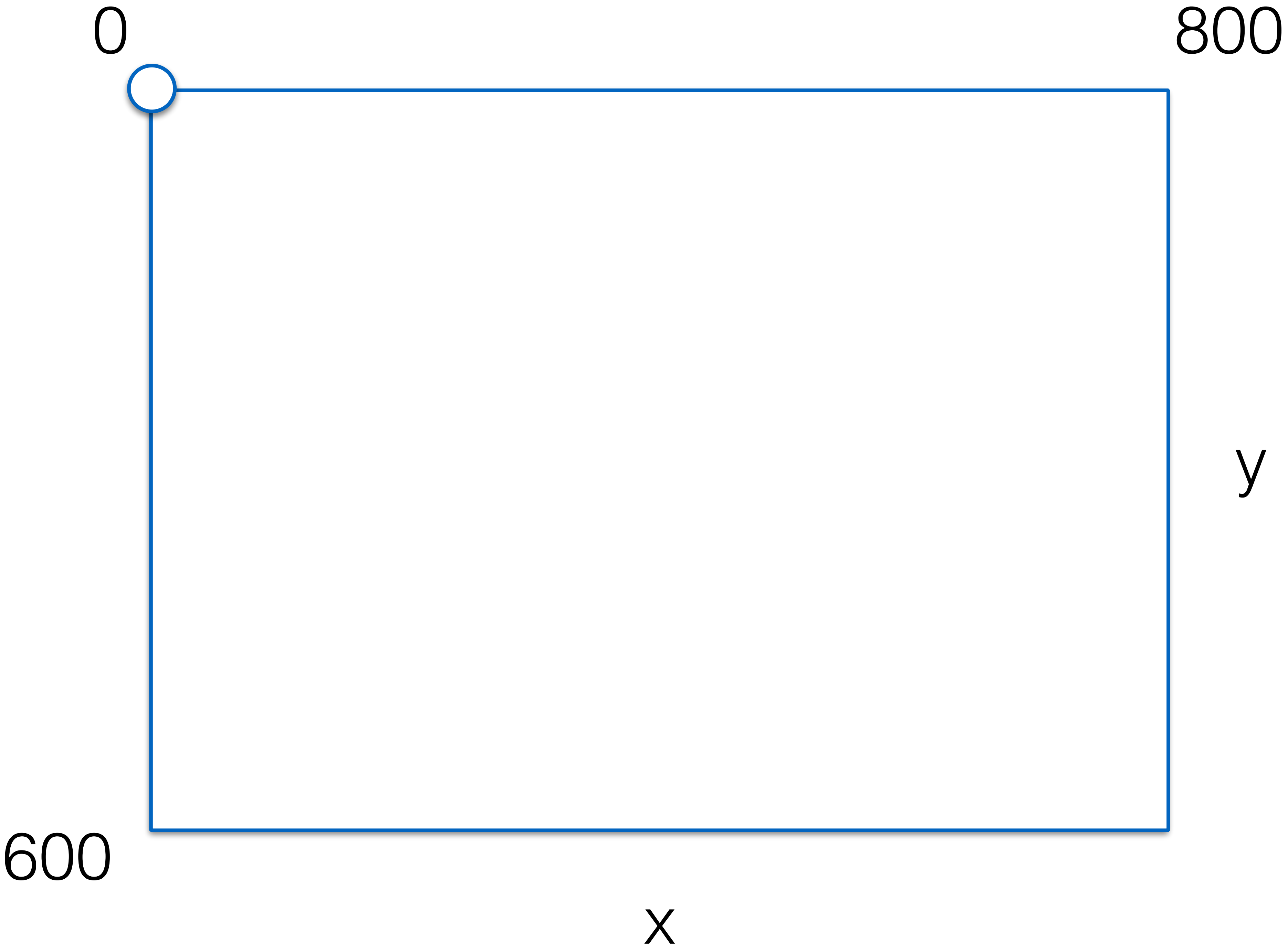

Gosu Tutorial Game
WALLPAPERSWIDE.COM

- obiekty na ekranie - PORO (Plain Old Ruby Object)



```
class Player

  def initialize
    @image = Gosu::Image.new(„img/player.jpg")
    @x = 0
    @y = 0
  end

  def draw
    @image.draw(@x, @y, 1)
  end

  def move_left

  end

  def move_right

  end

  …
```

Globalne współrzędne



- poruszanie się gracza - update()

```
class GameWindow < Gosu::Window

  def initialize
    …
     @player = Player.new
    …
  end

  def update
    if Gosu::button_down? Gosu::KbLeft || Gosu::button_down? Gosu::GpLeft
      @player.move_left
    elsif Gosu::button_down? Gosu::KbRight || Gosu::button_down? Gosu::GpRight
      @player.move_right
    elsif …
  end

  def draw
    …
    @player.draw
    …
  end
end
```

- GUI - prosty napis

```
def draw
    …
    @font.draw("Score: #{@player.score}", 10, 10, ZOrder::UI, 1.0, 1.0, 0xff_ffff00)
    …
end
```

- dźwięki

```
@beep = Gosu::Sample.new(„media/beep.wav")
@beep.play
```

- fizyka (rigid body) - Chipmunk

- http://chipmunk-physics.net

Showcases

https://www.libgosu.org/cgi-bin/mwf/board_show.pl?bid=2

Dzięki