

# Musical Hacking com openFrameworks

Nuno Hespanhol

7 e 8 de Maio 2016



# Digitópia

A Digitópia é uma plataforma de música digital sediada na Casa da Música, no Porto, que incentiva a audição, a performance e a criação musical. Baseando-se em ferramentas digitais, embora não exclusivamente, enfatiza a criação musical colaborativa, o design de software, a educação musical e a inclusão social, promovendo a emergência de comunidades multiculturais de performers, compositores, curiosos e amantes de música.

A Digitópia é uma colaboração entre a Casa da Música, o INESC Porto, a ESMAE e a Escola das Artes – UCP.

# Nuno Hespanhol

## Académico

- Conservatório de Música do Porto
- Mestrado em Engenharia Informática e Computação - FEUP
- Master in Sound and Music Computing - UPF (Barcelona)

## Profissional

- Researcher @ SMC-INESC Porto e SMC-MTG Barcelona
- Digitópia - Casa da Música
- Software Developer @ toFlow Intelligent Computing

## Artístico

- Guitarra @ Conservatório de Música do Porto
- Desenvolvimento de software musical/instalações
- Participação na hackathon de Music Hack Day Barcelona 2015

# Objetivos

- Recorrendo à biblioteca open source openFrameworks, serão abordados de modo simples, conceitos e técnicas de criação de pequenos programas artísticos.
- Segue-se o trabalho com as ferramentas git e Github úteis para desenvolver software colaborativamente.
- Haverá no fim uma hackathon, com a modificação de projectos open source já disponíveis e/ou desenvolvimento de ideias dos participantes.

# Objetivos (disclaimer)

- C++, openFrameworks, Git, OOP, creative programming são todos tópicos muito grandes
- Objetivo é experimentar e dar a conhecer
- Façam perguntas e interrompam à vontade
- Hands-on

?

- background programação?
- C++?
- openFrameworks?
- POO (OOP)?
- Git/Github?

# Plano (sábado)

Manhã 11:30 - 13:00

- openFrameworks
- C++
- Mini projeto: Mondrian

Tarde 14:30 - 17:00

- Programação Orientada a Objetos
- Mini projeto: Pong
- Git e Github

# Plano (domingo)

Manhã 11h30 - 13h

- openFrameworks + Som
  - midi
  - pd
  - outros

Tarde 14h30 - 17h

- Hackathon

*Pode mudar em função de hoje*





- é a linguagem que o openFrameworks usa
- inventado por Bjarne Stroustrup em 1983
- popular onde performance é importante
- continua com desenvolvimento ativo, última versão c++14 e próxima c++17



# openFrameworks

- conjunto de ferramentas open source em c++ para programação criativa
- desenhado para funcionar como a “cola” entre várias bibliotecas dedicadas: OpenGL, GLEW, GLUT, libtess2, cairo, rtAudio, PortAudio, OpenAL, Kiss FFT, FMOD, FreeType, FreeImage, Quicktime, GStreamer, videoInput, Poco, OpenCV, Assimp, ...
- Princípios:
  - **Colaborativa**
  - Simples
  - Consistente e Intuitiva
  - **Multi-plataforma** (osx, windows, linux, rpi, ios, android, emscripten)
  - Potente (“powerful”)
  - **Extensível**
  - **Do It With Others (“DIWO”)**

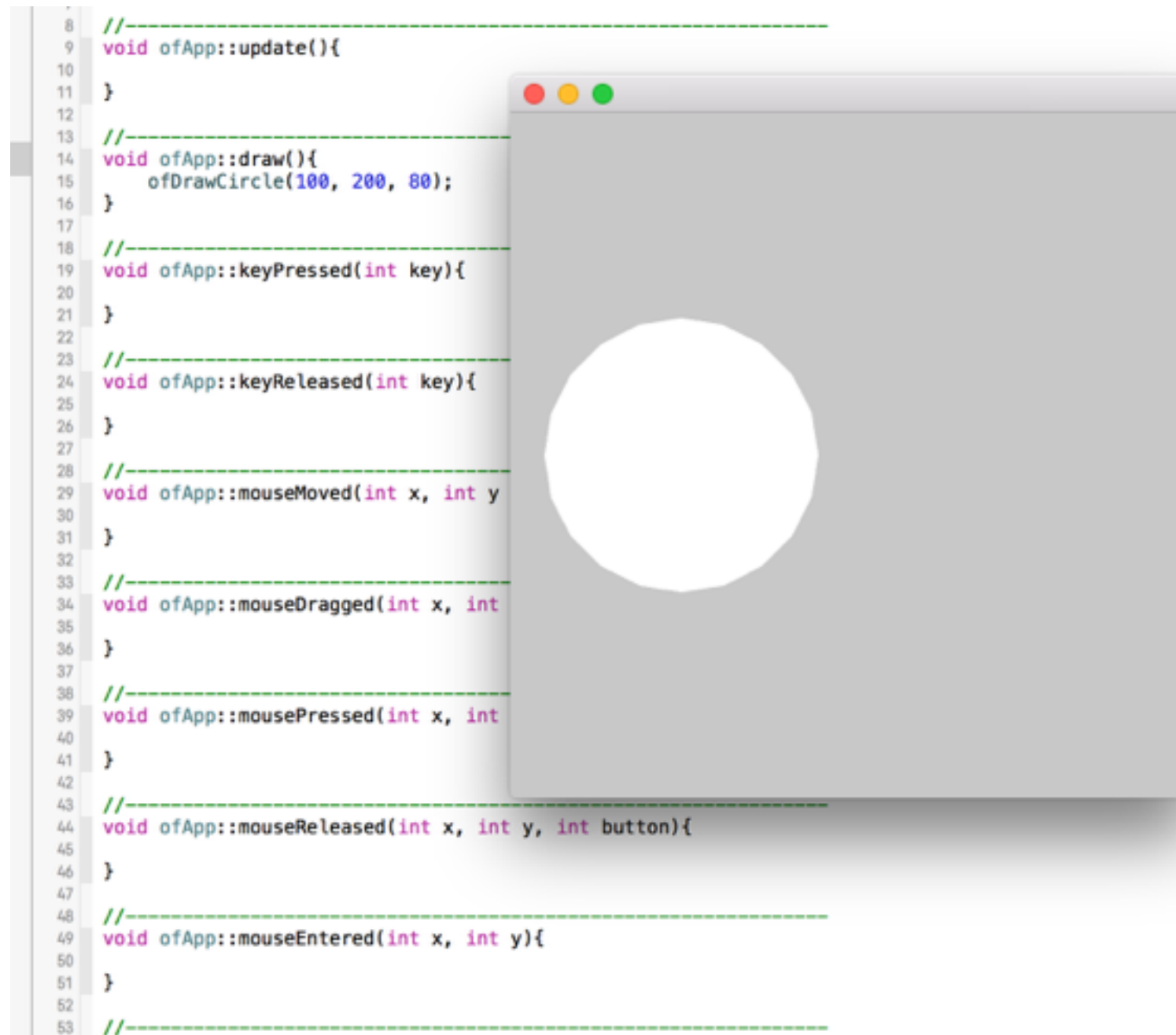
# Hello World em openFrameworks

*Primeiro exemplo de código num dos livros mais importante da história da computação: The C Programming Language (Kernighan and Ritchie, 1974)*

- Instalar IDE (Xcode ou Visual Studio) e descarregar openFrameworks versão 0.9.3
- Tentar correr o exemplo, para ver se tudo ok:  
`examples/3d/3DPrimitivesExample`
- Gerar um projeto novo com o projectGenerator  
(pasta `projectGenerator-osx`)
- Na função `draw()`, desenhar um círculo:  
`ofDrawLine(100, 200, 80)`

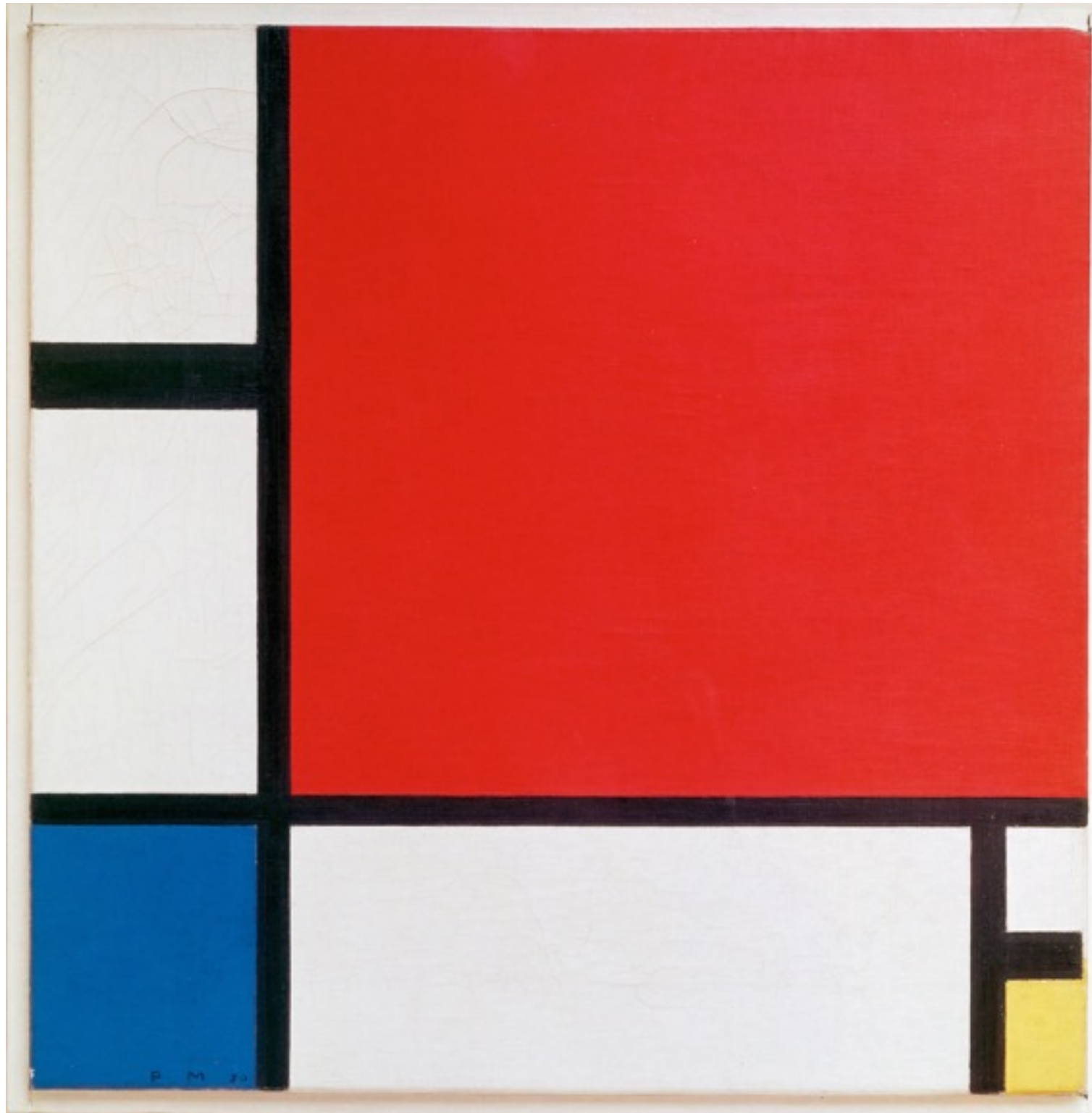


# Hello World em openFrameworks



Pode-se mudar tamanho da janela no main.cpp  
i.e. ofSetupOpenGL(400, 400, OF\_WINDOW)

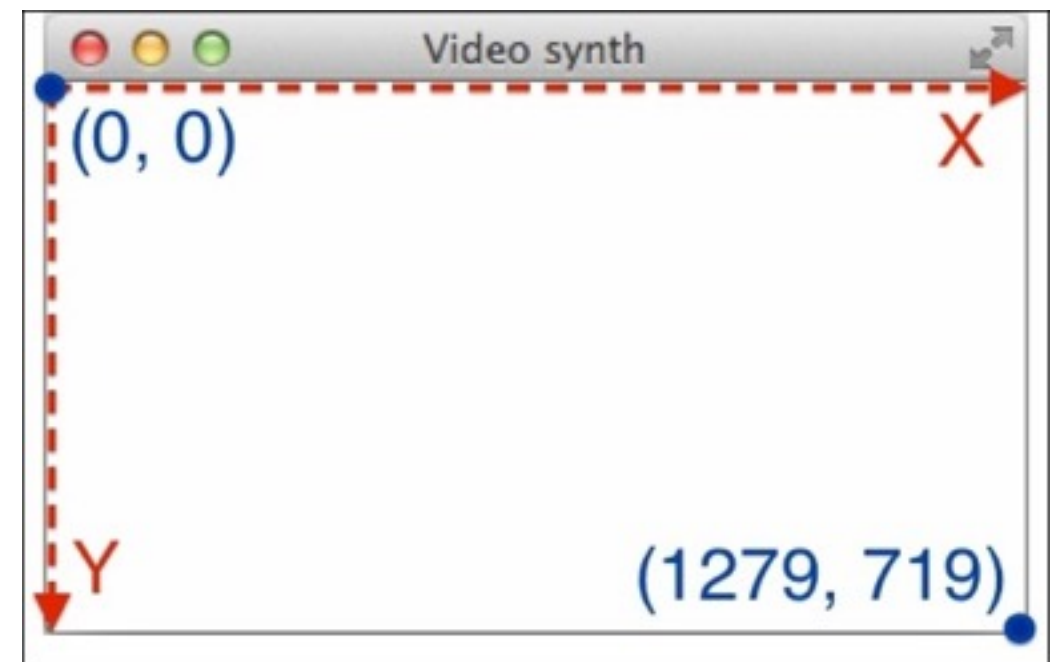
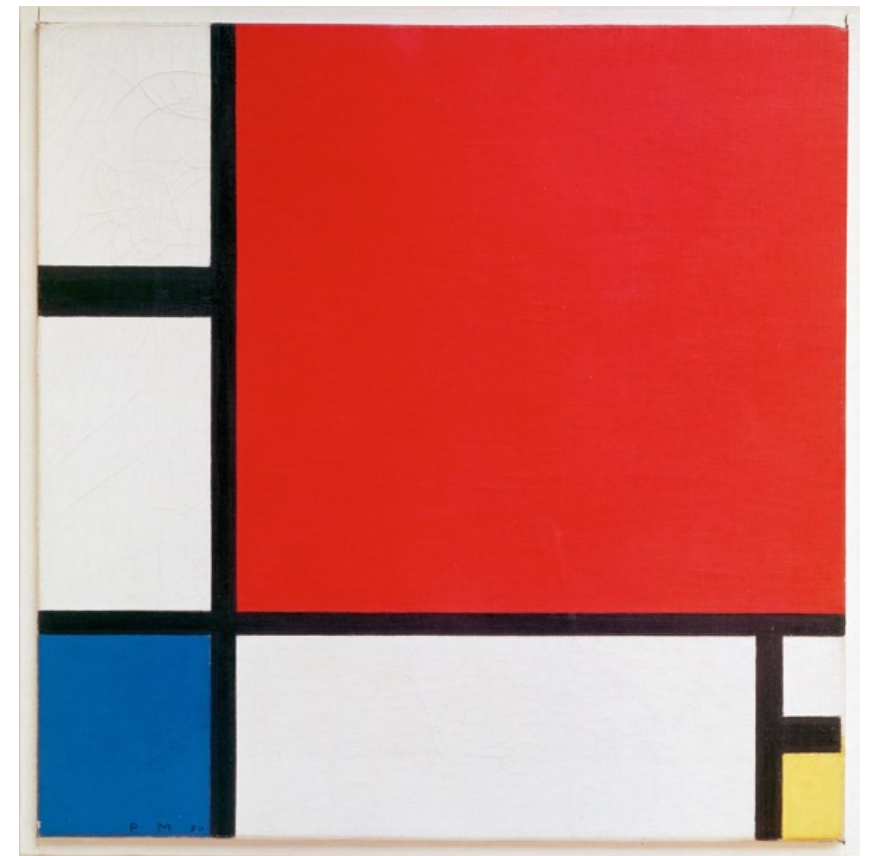
# Primeiro “programa artístico”



retângulos, linhas e cores

- `ofDrawRectangle(x1, y1, w, h)`
- `ofDrawLine(x1, x2, y1, y2)`  
`ofSetLineWidth(width)`

`ofSetColor(r, g, b)`  
`ofSetColor(r, g, b, a)`  
`ofSetColor(grey)`



<http://openframeworks.cc/documentation/>

# C++ - tipos

// um número inteiro

int n = 2;

// número decimal

float pi = 3.14;

// só pode ter valores "true" ou "false"

bool var = false;

// só um carácter

char c = 'c';

// vários caracteres

string s = "olá mundo";

# C++ - estruturas de controlo

```
if (n == 1) {  
    cout << "O valor de n é 1" << endl;  
}  
else {  
    cout << "O valor de n é qualquer um que não 1" << endl;  
}
```

```
for (int i = 0; i < 5; i++) {  
    cout << i << endl;  
}  
// irá imprimir: 0 1 2 3 4
```

```
while (ofGetElapsedTimef() <= 5000) {  
    x++;  
}  
// enquanto não passarem 5 segundos,  
// continua a incrementar a variável x
```



# C++ - funções

tipo de retorno      nome da função      tipo parâmetro      nome parâmetro

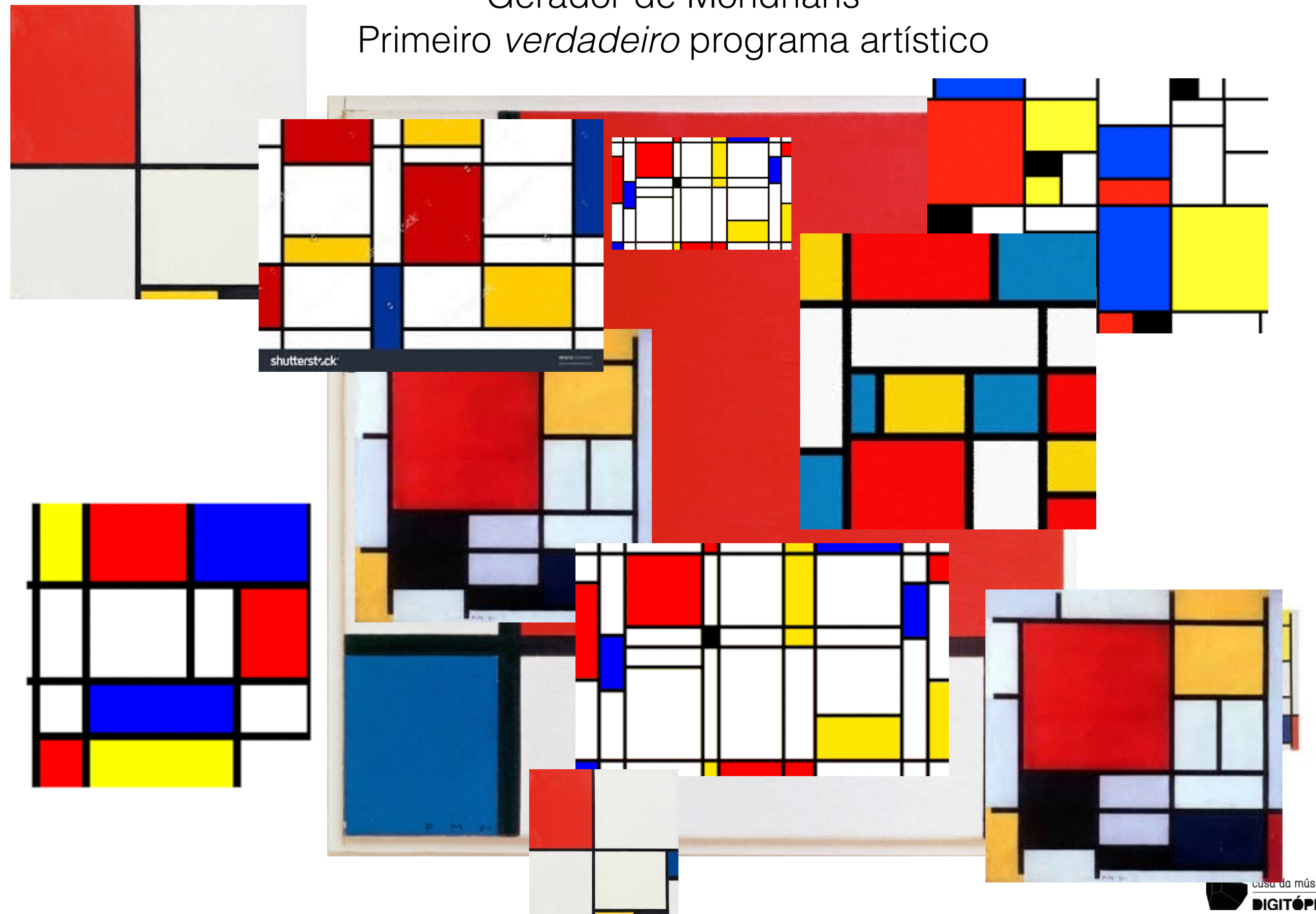
↓                      ↓                      ↓                      ↓

```
float media(int a, int b) {  
    return (a+b)/2.0;  
}
```

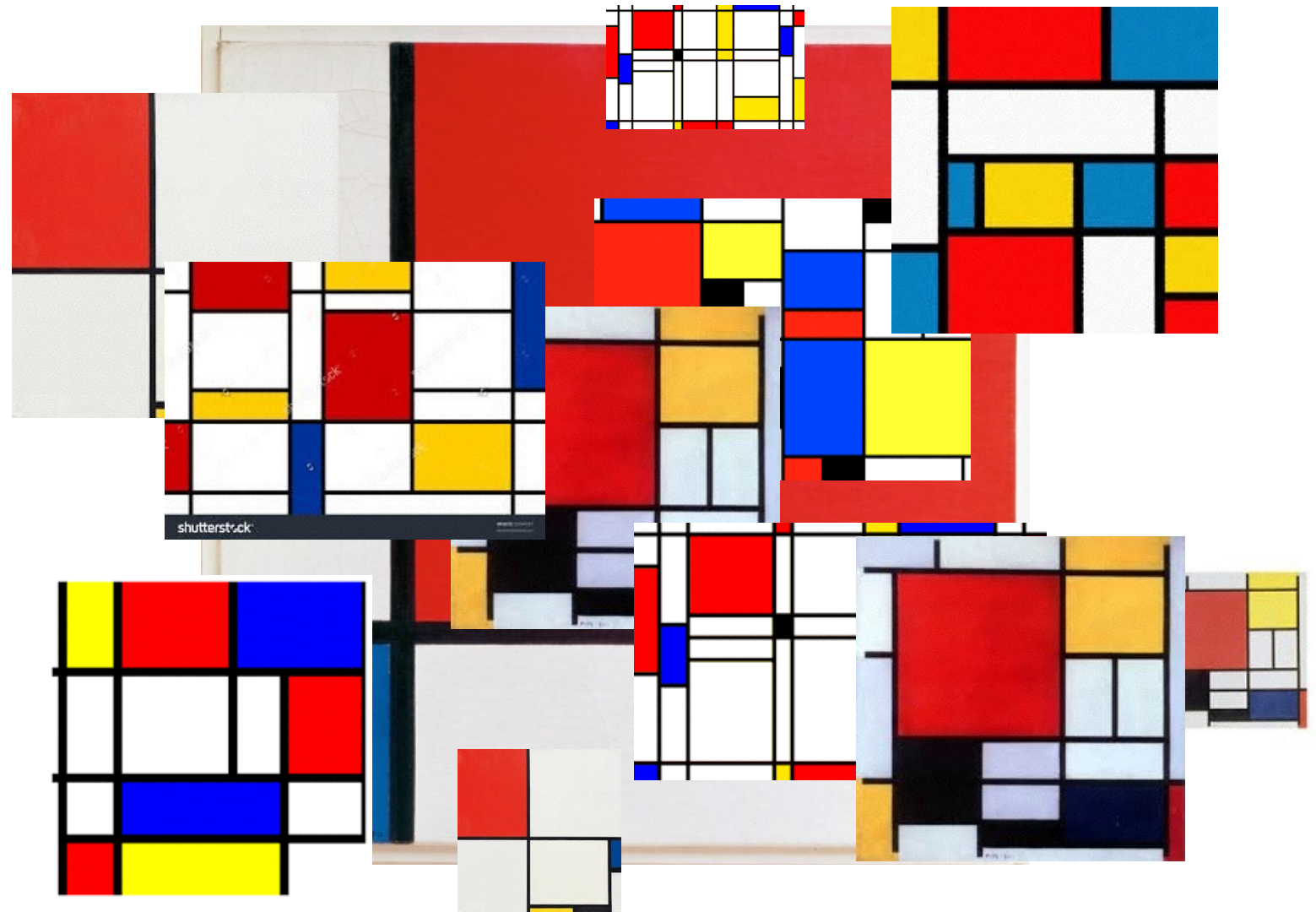
chamadas à função

```
// imprime logo 4  
cout << media(3, 5) << endl;  
  
// guarda o resultado na variável  
float resultado = media(2, 2);
```

# “Gerador de Mondrians” Primeiro *verdadeiro* programa artístico



- `float ofRandom(float val0, float val1)`
- `ciclo for`



# Programação Orientada a Objetos (OOP)

- é um paradigma de programação. Outros exemplos: imperativo, funcional e lógico
- permite modelar problemas complexos de uma maneira simples
- de uma maneira geral, de-facto standard
- conceitos-chave: classe, objeto, dados-membro, método, herança

# Programação Orientada a Objetos (POO)

- **Classes:** é a definição geral de um objeto; e.g.  
“Animal”
- **Objetos:** são o caso concreto de uma classe; e.g.  
“cão”, “gato”
- **Métodos:** são as funções que um objeto pode fazer e.g. “miar”, “comer”
- **Dados-membros** (ou propriedades): são as variáveis de um objeto: “nome”, “dono”

# OOP (ou OOP)

nome da classe

`class Bola {`

`public:`

`Bola(int x, int y);`

`void update();`

`void draw();`

`int x;`

`int y;`

`};`

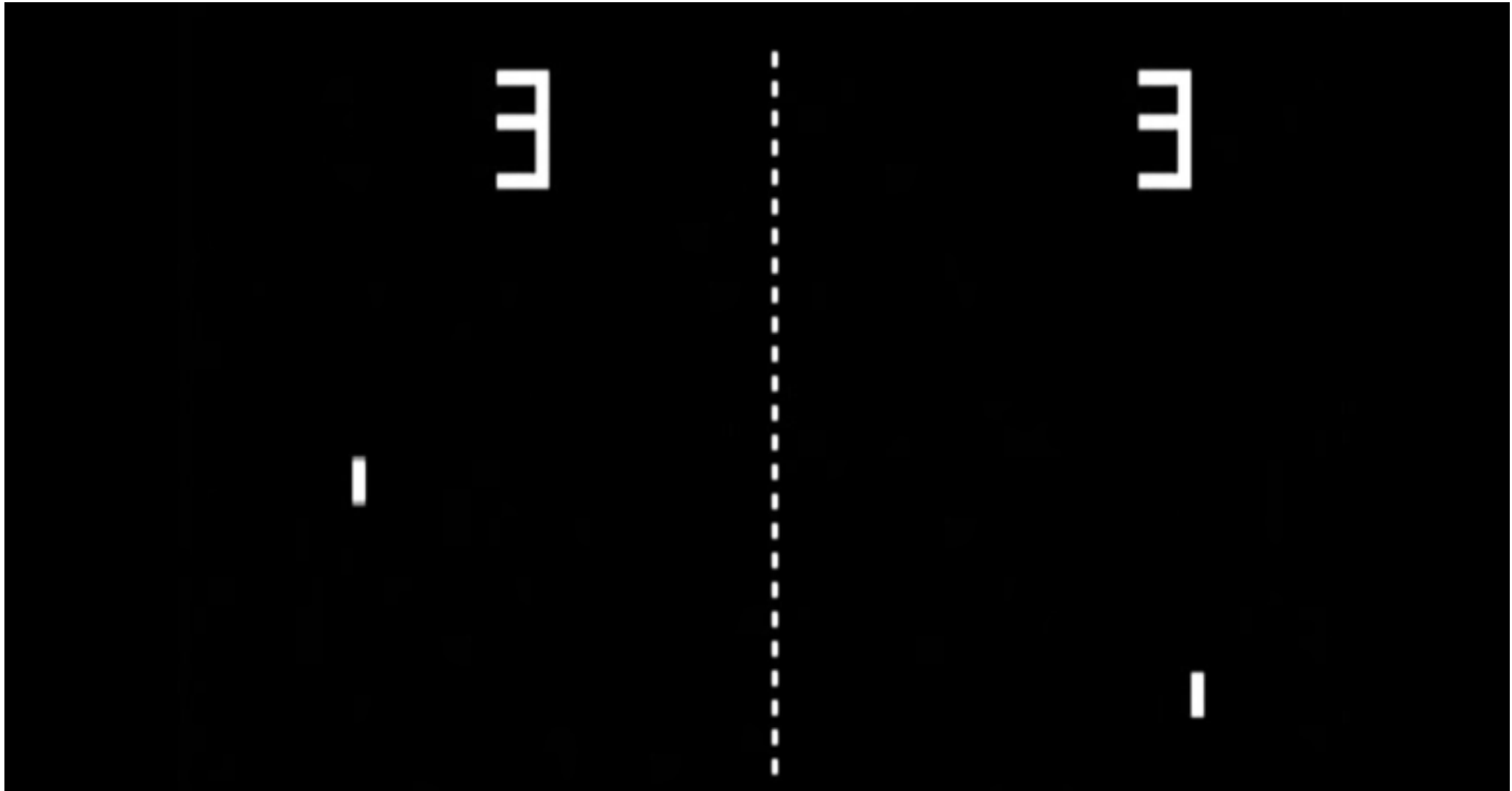
construtor

método/função que cria/instancia um objeto

métodos/funções

dados-membro/propriedades/variáveis

# Pong



<https://www.youtube.com/watch?v=it0sf4CMDDeM>



- Classes: Bola, Plataforma

- Bola: float vx, vy; int radius;

- Plataforma: int x, y, size;

- controlar com o teclado a plataforma ou rato

```
void ofApp::mousePressed(int x, int y, int button){  
}
```

```
void ofApp::keyPressed(int key) {  
    if (key == OF_KEY_UP) {  
    }  
}
```

- quando há golo, tocar som

```
ofSoundPlayer sound;  
sound.load("sound.wav");  
sound.play();
```



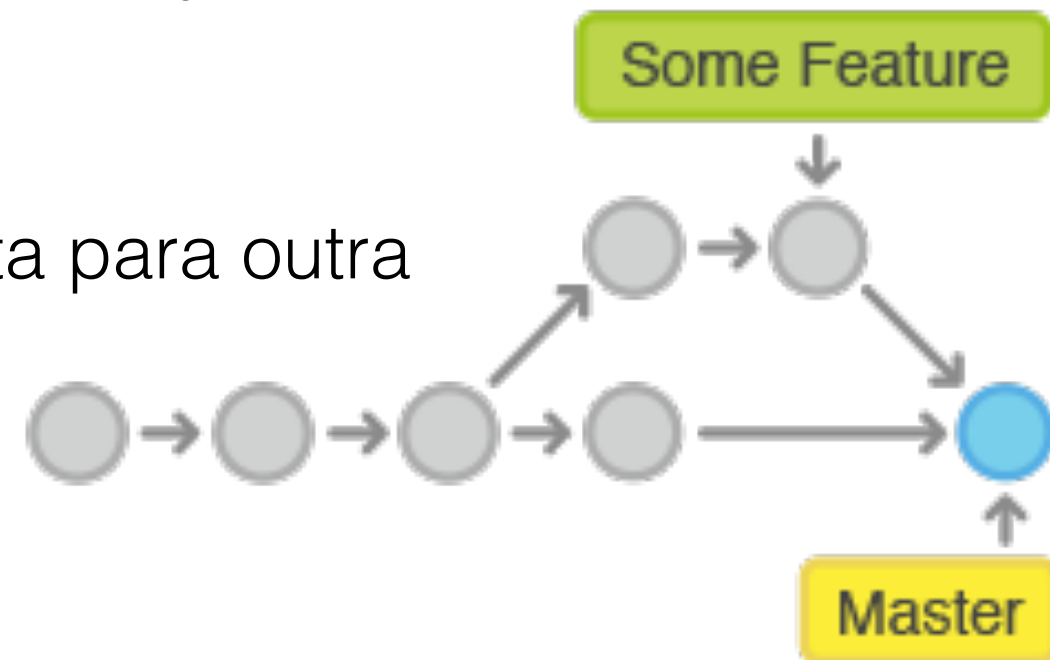


# Git e Github

- Git
  - é sistema de controlo de versões distribuído
  - open source
  - é uma biblioteca de software (à semelhança do openFrameworks)
- Github
  - é uma empresa privada (fundada 2008)
  - que proporciona serviços gratuitos de alojamento de repositórios Git
  - 10 milhões de repositórios

# Git e Github - **conceitos**

- repositório: conjunto de pastas e ficheiros versionado
- commit: um snapshot no tempo de um repositório
- branch: as alterações aos ficheiros de um repositório podem ser feitas em paralelo
- pull request: pedido de incorporação de alterações (novos *commits*) num repositório
- fork: cópia de um repositório de uma conta para outra



# Git e Github - **hands on**

- Como é que o software open source evolui?
  - R: Através de patches ou na terminologia do Github Pull Requests.

Fazer Pull Request com o nome para a secção participantes do ficheiro README.md do repositório deste workshop:

<https://github.com/Digitopia/ws-openframeworks>

# Plano (domingo)

Manhã (11h30 - 13h)

- openFrameworks + Som: midi, pd e outros

Tarde (14h30 - 17h)

- Hackathon

# Som

- ofxMidi
- ofxPd
- exemplos sound/

# Hackathon

- o que cada um quiser
- melhorar um exemplo que tenham gostado
- melhorar um projeto de ontem
- explorar mais exemplos do oF para ideias

Criar repositório Github e deixá-lo disponível em open source :)

# Hackathon

Mui

BACKUPS



# Recursos

- <https://www.kadenze.com/courses/creative-programming-for-audiovisual-art/info>
- ofBook
- <http://openframeworks.cc/>
- <https://forum.openframeworks.cc/>
- <http://ofxaddons.com/categories>
- <https://github.com/openframeworks>