

# ITUS GAME CLUB

Week II

21/11/2014

# TIÊU ĐỀ

Nội dung 01

Hướng dẫn cài đặt logic game  
Flappy Bird



Nội dung 02

C++ Getting Started

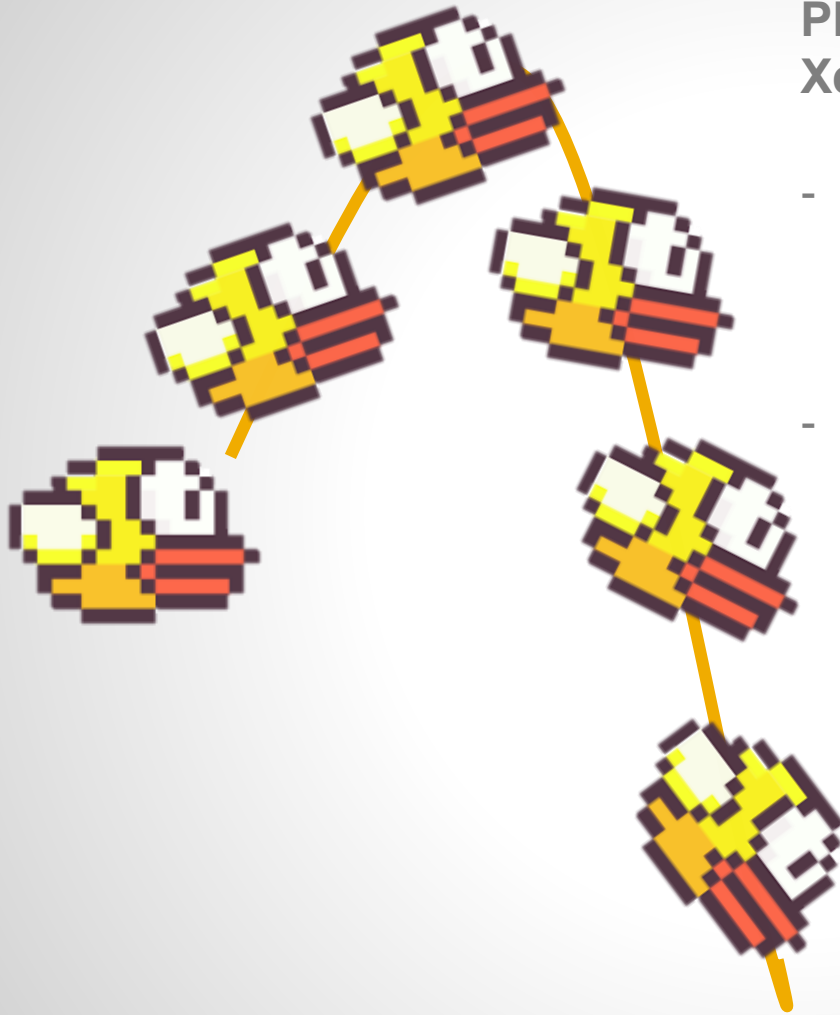


# Cài đặt logic game Flappy Bird



## Chuyển động của flappy trong game?

# Cài đặt logic game Flappy Bird



Phương trình chuyển động:  
Xét theo 2 trục:

- Trục X: chuyển động thẳng đều.
  - $X = x_0 + v_0 \cdot t$ .
- Trục Y: ném thẳng đứng lên trên
  - $y = y_0 + v_0 \cdot t - \frac{1}{2} \cdot g \cdot t^2$

## Chuyển động của vật ném xiên!

# Cài đặt logic game Flappy Bird

Áp dụng tính toán trong game:

Cơ bản cập nhật tọa độ của flappy bird theo thời gian.



Đặt :

```
float time; // thời gian trong kể từ lúc chim  
bay lên, gốc =0;
```

```
Vector2 position;// Tọa độ của chim.
```

Ta có:

Lúc chim bắt đầu bay:

```
x0=position.x;
```

```
y0=position.y;
```

```
time=0;
```

Cập nhật tọa độ của chim theo thời gian:

```
position.x=x0+v0*time;
```

```
position.y=y0+v0*time-0.5*g*t^2.
```

# C++ Getting Started



**KEEP  
CALM  
AND  
CODE  
C++;**

# C++ Getting Started

Biến và hằng

Cấu trúc: struct – enum-union

Hàm và Tham số

Source compile process

# Biến và Hằng

Phạm vi của biến:

- ☐ Local
- ☐ Global
- ☐ Static

```
int bienToanCuc;
```

```
void Ham1(){  
    int a;  
    printf(" Gia tri cua a: %d \n",a)  
}
```

```
void main(){  
    printf(" Gia tri cua bien toan cuc: %d\n",bienToanCuc);  
    Ham1();  
}
```

Giá trị mặc định là 0  
Phạm vi sử dụng toàn  
chương trình

**Không có giá trị mặc  
định**  
Phạm vi sử dụng nội bộ  
hàm



# Biến và Hằng

Biến static:

```
static int a;

void Ham(){
    static int b;
    printf(" Gia tri cua b: %d\n", b);
    b=b+1;
}

void main(){
    printf(" Gia tri cua a: %d\n",a);
    Ham();
    Ham();
}
```

-> Có giá trị mặc định 0.

-> Được khởi tạo trên vùng nhớ khi bắt đầu chạy chương trình và hủy khi kết thúc.

# Biến và Hằng

Biến Extern:

```
int bienExtern=10;
```

```
Void Ham()  
{  
  
}
```

Temp.cpp

```
Extern int bienExtern;
```

```
Void main()  
{  
    printf(" gia tri %d:",bienExtern);  
}
```

Main.cpp

- > Thể hiện biến đã được định nghĩa ở một file khác
- > Đóng vai trò như là biến dùng chung cho nhiều file cpp
- > Tránh trường hợp khởi tạo lặp.

# Biến và Hằng

## Hằng

```
const int a=5;
```

```
Void main()
```

```
{
```

```
    a=100
```

```
}
```

Error: expression must be a modifiable lvalue

-> Thể hiện một giá trị hằng không thay đổi

**Còn tiếp!**