

3D Game Programming game history & 2D intro

Ming-Te Chi
Department of Computer Science,
National Chengchi University

Outline



game history



Coordinate system




Simple 2D game example

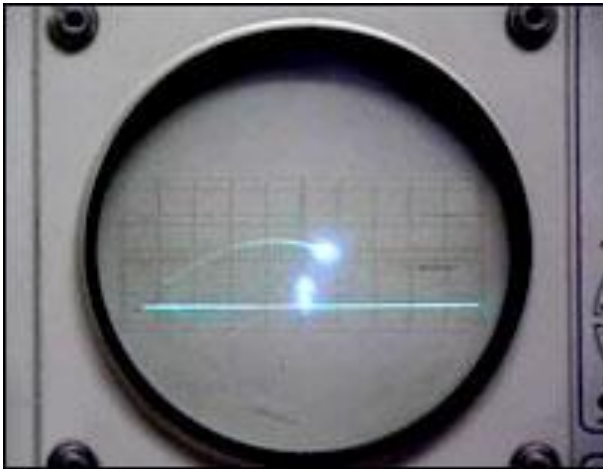


Simple 2D game example (OpenGL)

2D VIDEO GAME HISTORY

The first video game

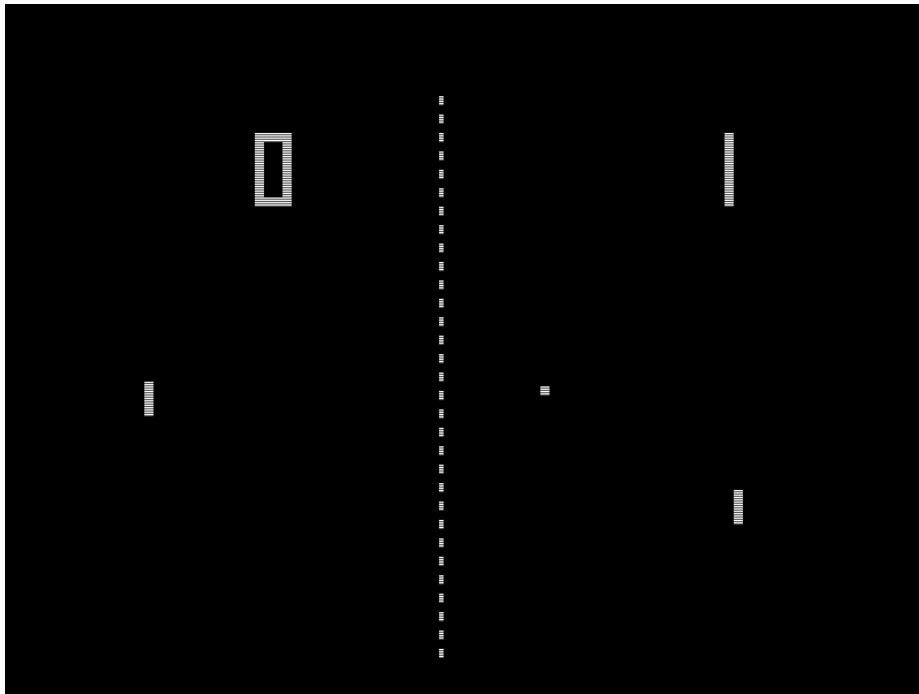
 Tennis for Two was a game developed in 1958 on an analog computer, which simulates a game of tennis or ping pong on an oscilloscope.



2D GAME

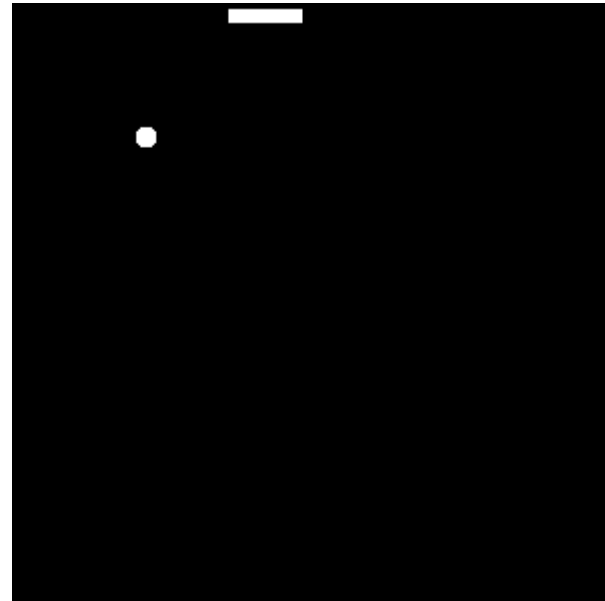
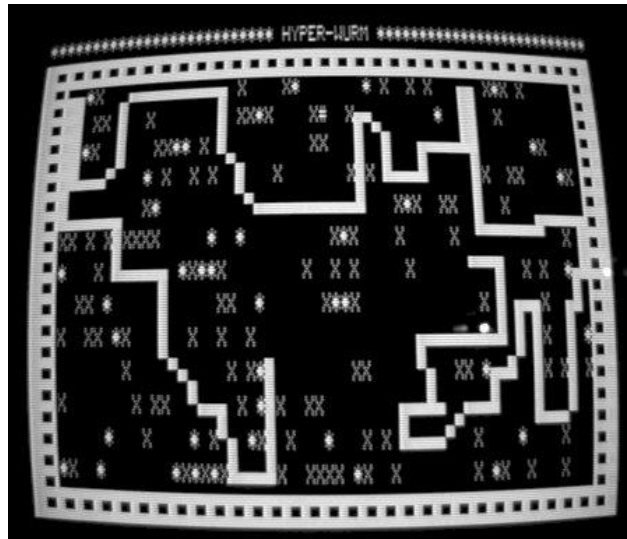


PONG 1972.
– earliest video game

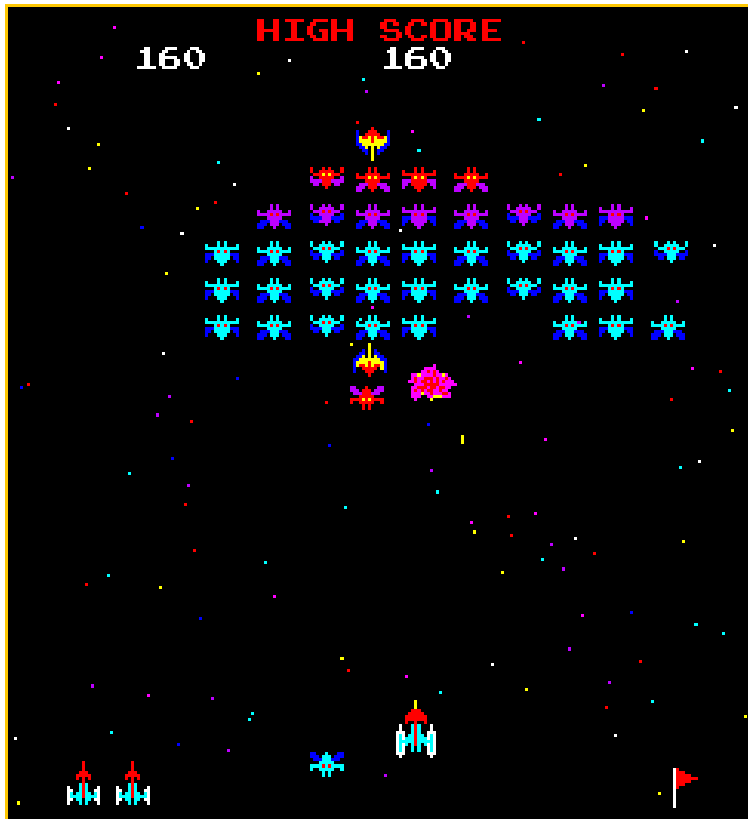


Snake (1970s)

- Control a snake to move, and avoid hitting to wall or its growing tail.



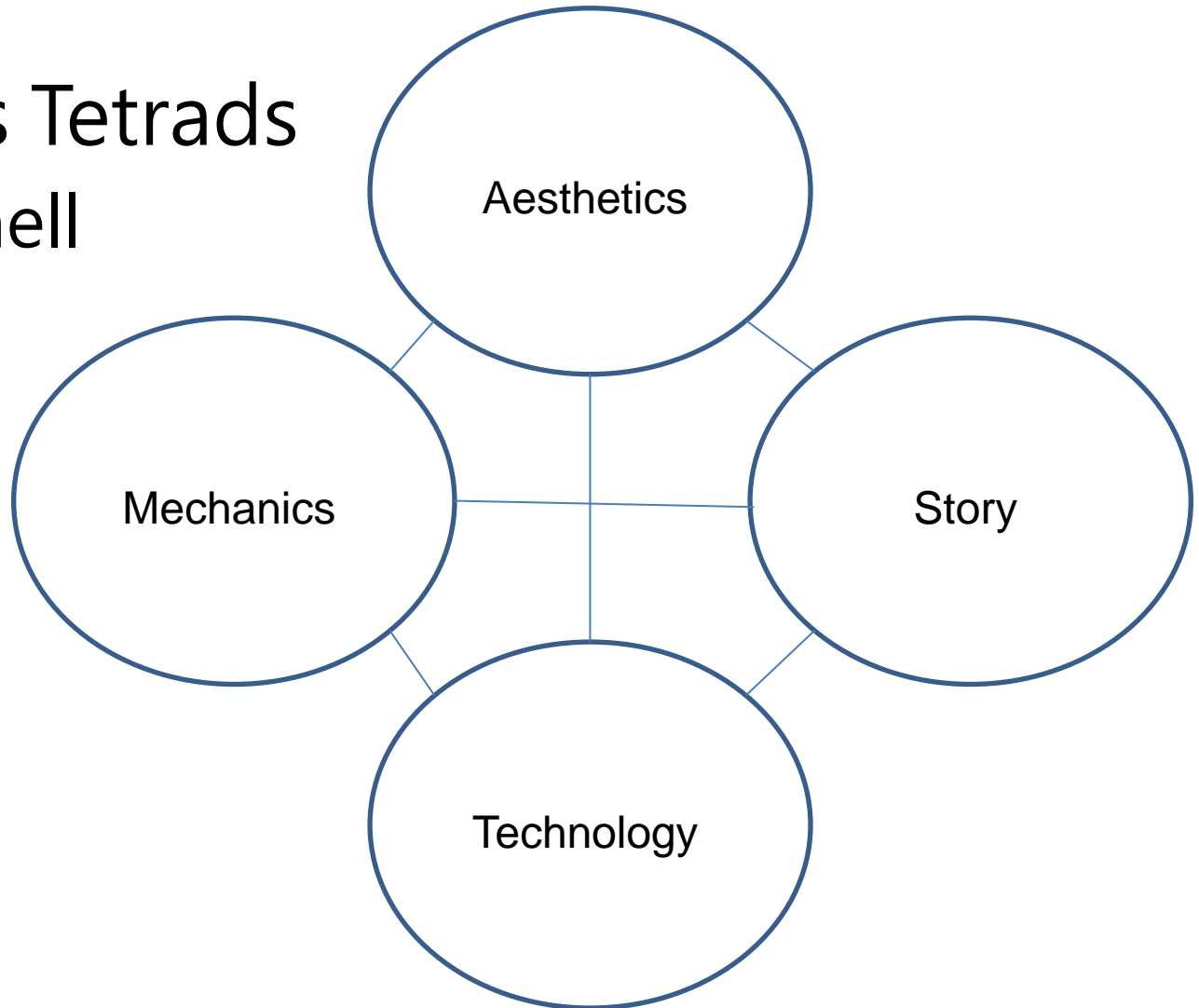
Galaxian (1979 by Namco)



Basic elements



Elements Tetrads – Jesse Schell





Mechanics

- the procedures and rules



Story

- the sequence of events that unfolds in your game



Aesthetics

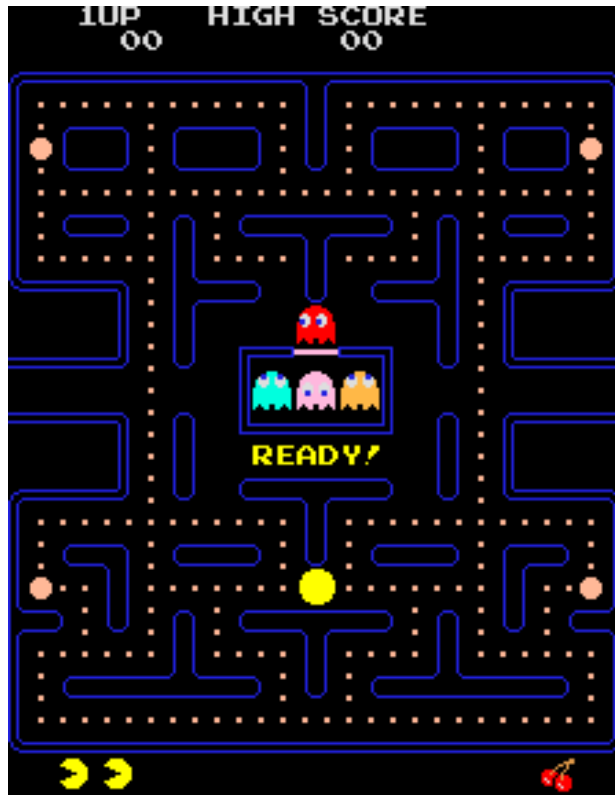
- how your game looks, sounds, tastes, and feels.



Technology

- any materials and interactions that make your game possible

Pac Man 1980 by Namco



Game & Watch 1980

 **Game & Watch** is a line of handheld electronic games produced by Nintendo from 1980 to 1991.



Ball: the first game & watch game



Family Computer(FAMICOM)



Mario series. By Nintendo



Donkey Kong
1981



Mario Bros.
1983



Super Mario Bros.
1985

Tetris

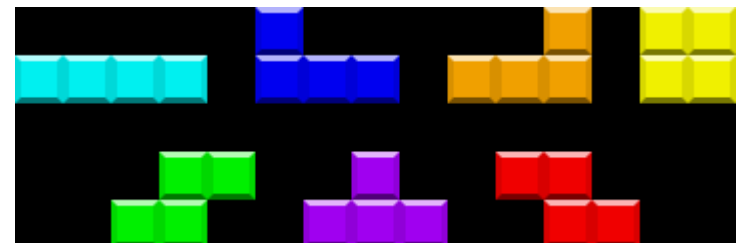
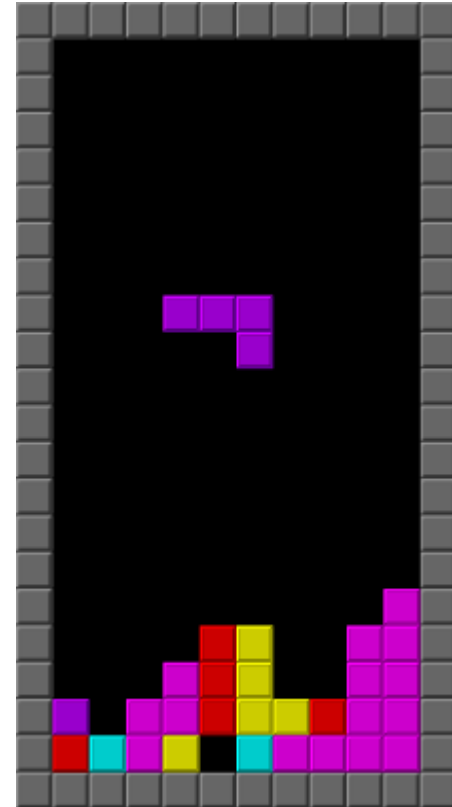
Design by 阿列克謝·帕基特諾夫

(Алексей Леонидович Пажитнов) in 1984

Puzzle game



1984 version



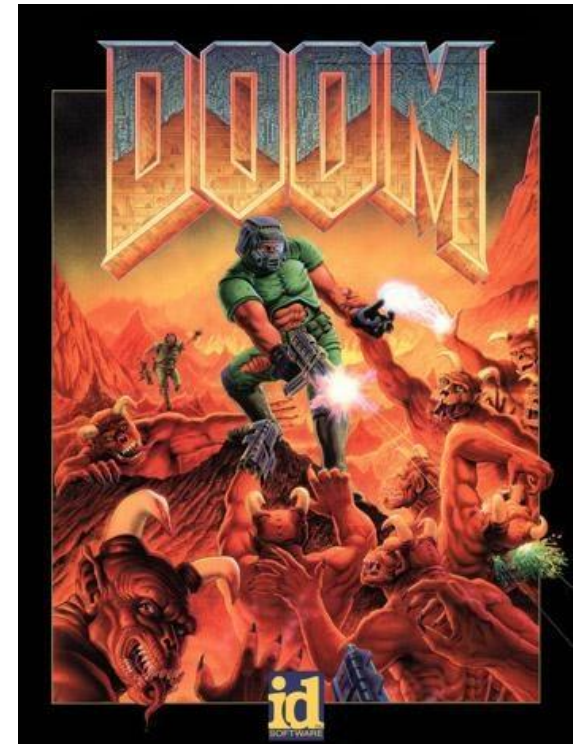
Super Mario World. 1990

- Rich color, Parallax scrolling, zoom and rotate sprite.



Doom 1993

- A landmark 1993 first-person shooter (FPS) video game by *id Software*.



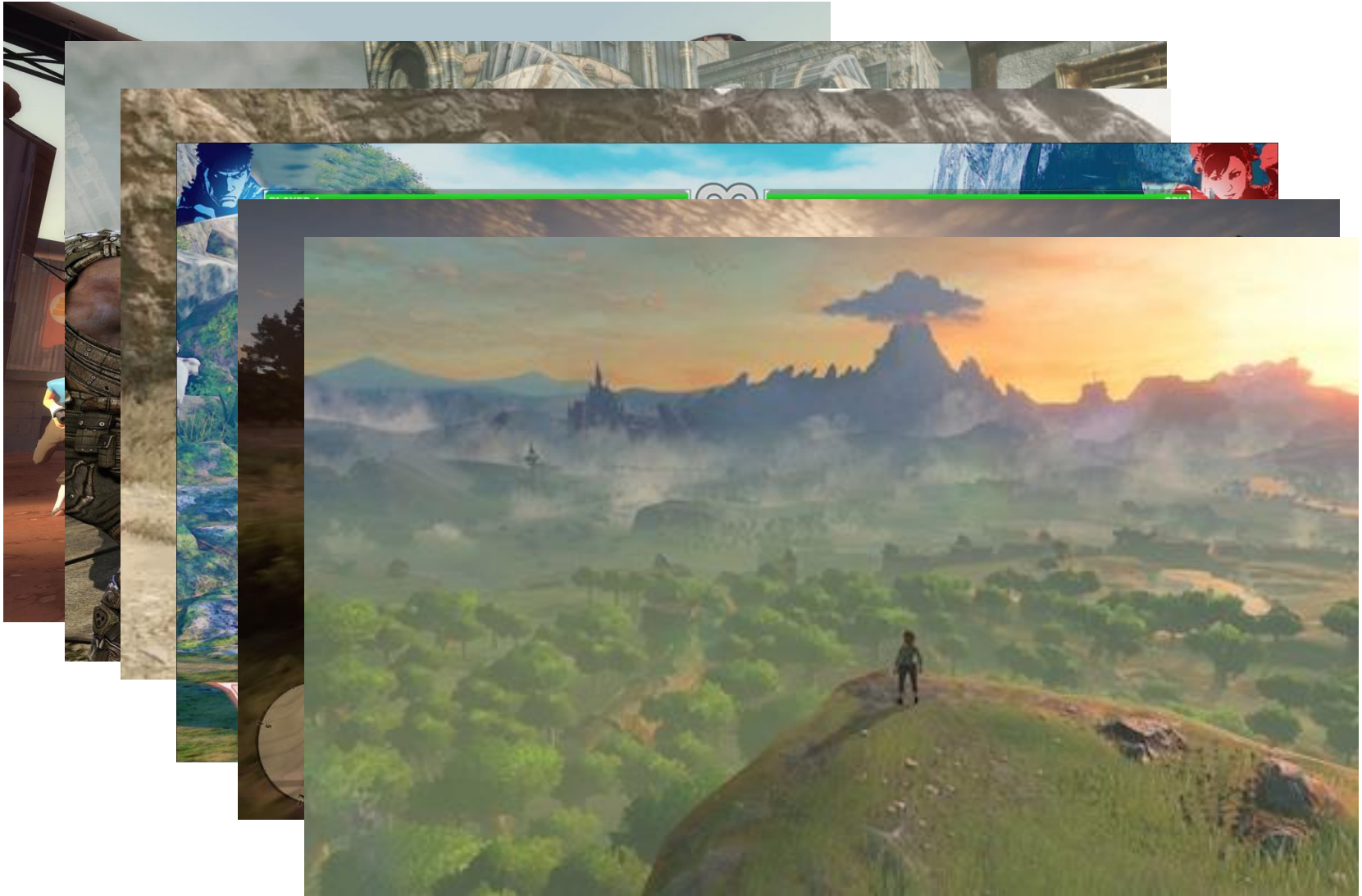
3D Graphics – evolution



3D Graphics – evolution



3D Graphics



Social game



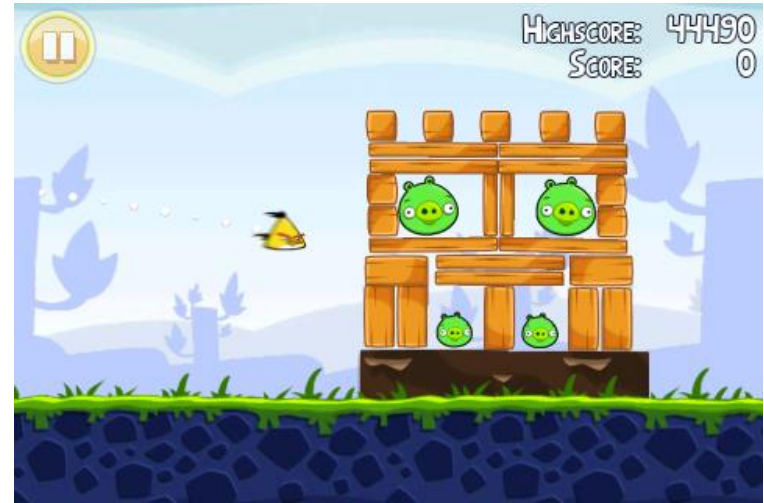
Happy Farm



Mobile game – touch

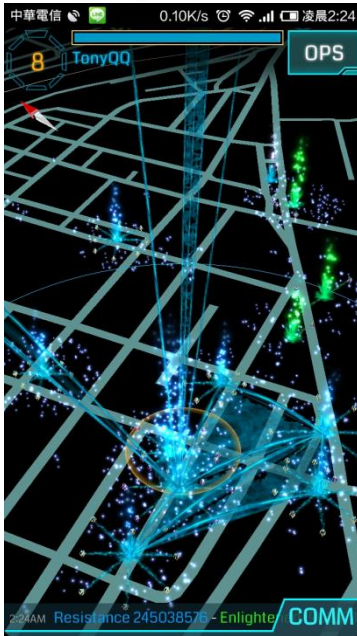


flight control by Firemint



Angry Birds by Rovio

LBS & AR



Snake in 2014



Super Mario Sunshine (2002)

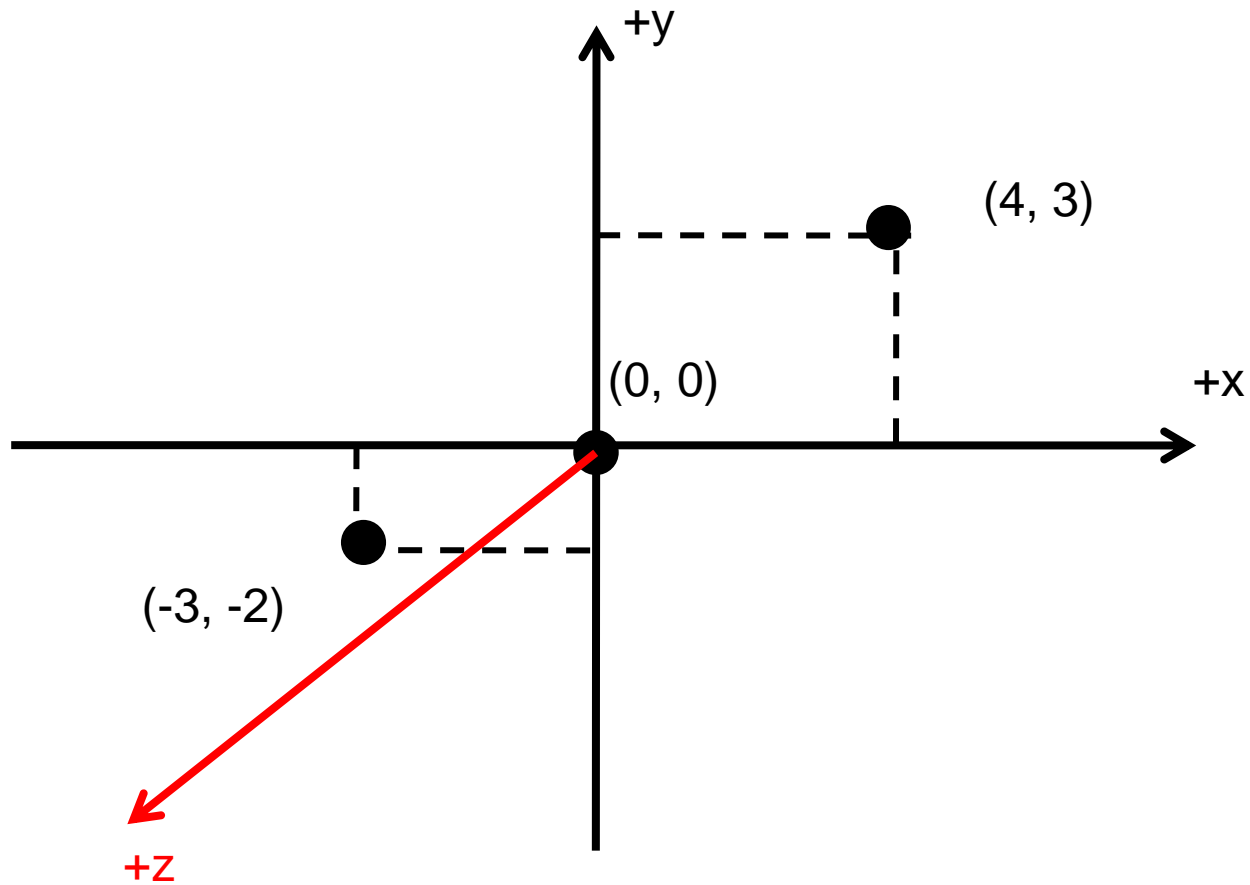


Fall Guys (2020)

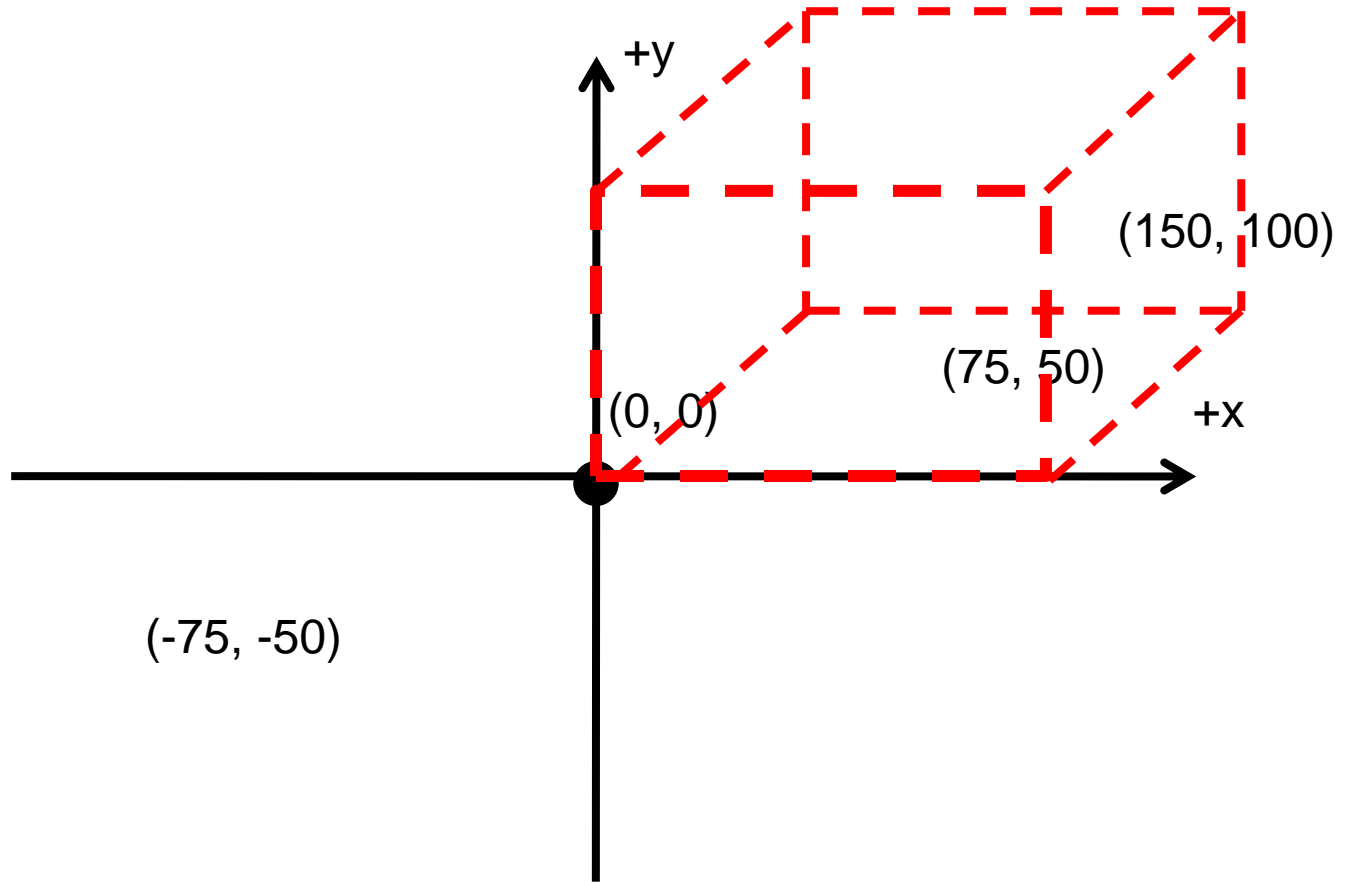


2D BASIC COMCEPT

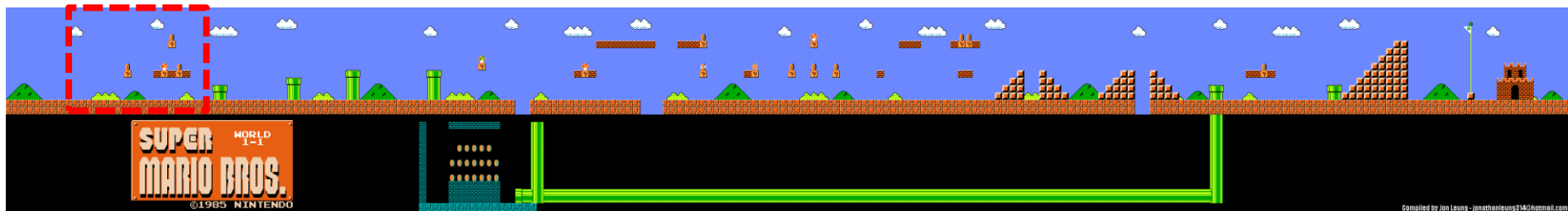
Cartesian Plane



Coordinate Clipping



Game World



Super Mario Bros. Nintendo

scrolling

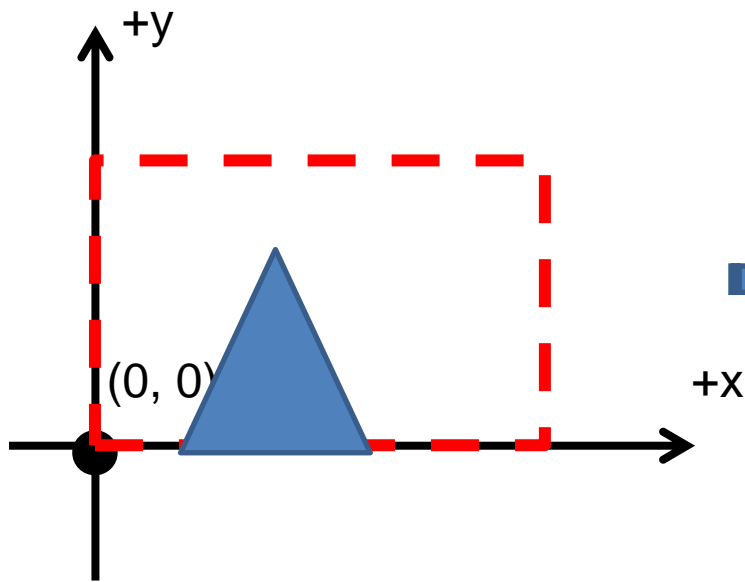


Start from 5:09

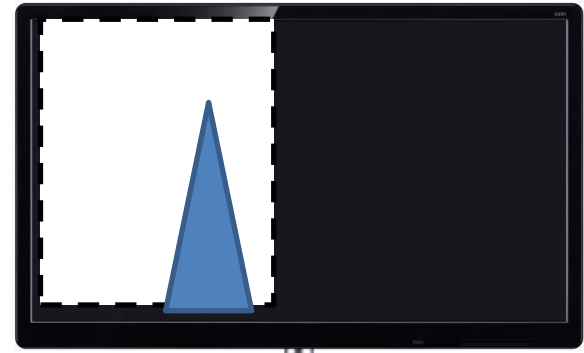
Viewport



Mapping drawing coordinates to windows coordinates



clipping space



window space

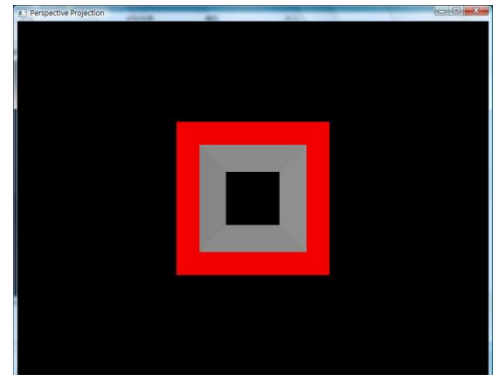
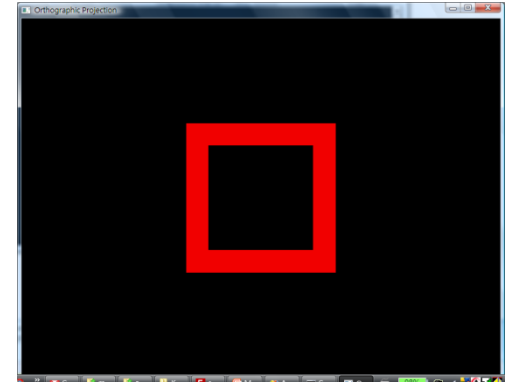
Projection



Getting 3D to 2D

– Orthographic projections

– Perspective projections



2D GAME

What is a Game?

Games are an exercise of voluntary control systems, in which there is a contest between powers, confined by rules in order to produce a disequilibrium outcome.

– Elliot Avedon and Brian Sutton-Smith

Game architecture

Asset Management

loading

saving

caching

Game Loop

Start()

Update(delta)

Display(delta)




I/O System

Keyboard, mouse

audio

storage

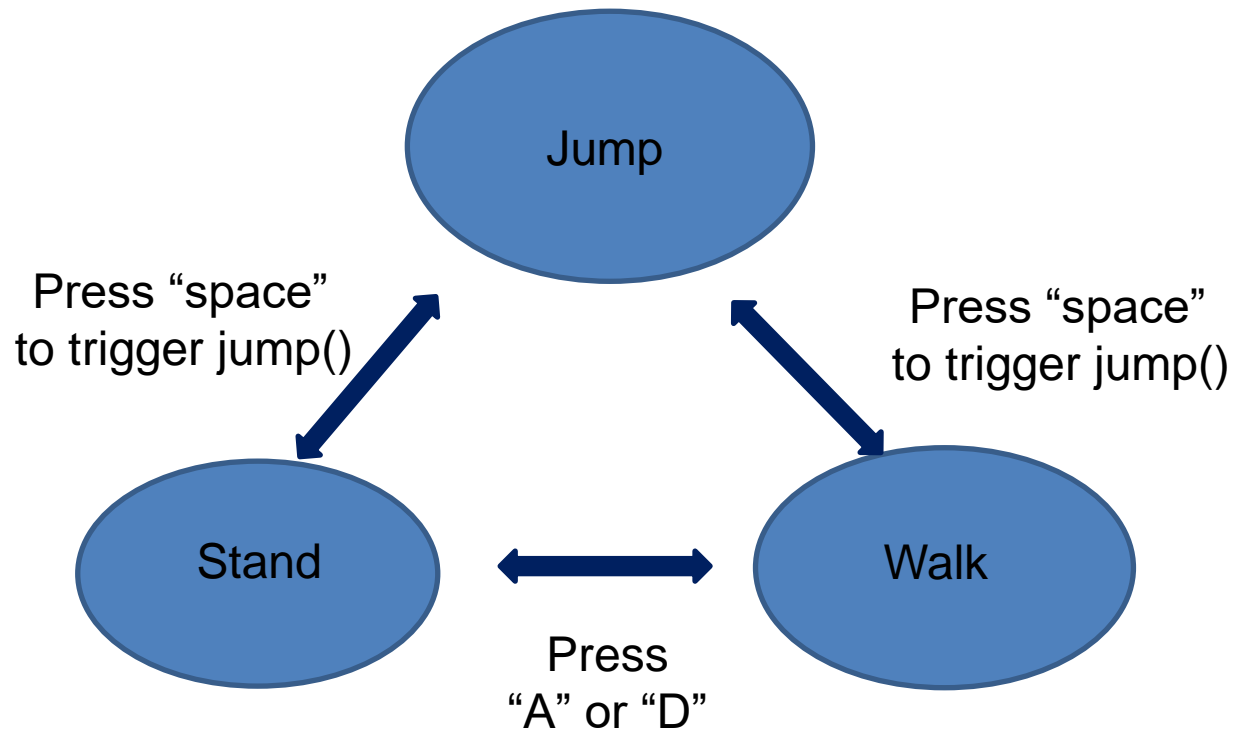
A simple example

-  A character has three states: **stand**, **walk**, and **jump**
-  Use "A" and "D" key to move the character
-  When press "space" , the character will jump, and the score will increase by 1.

State

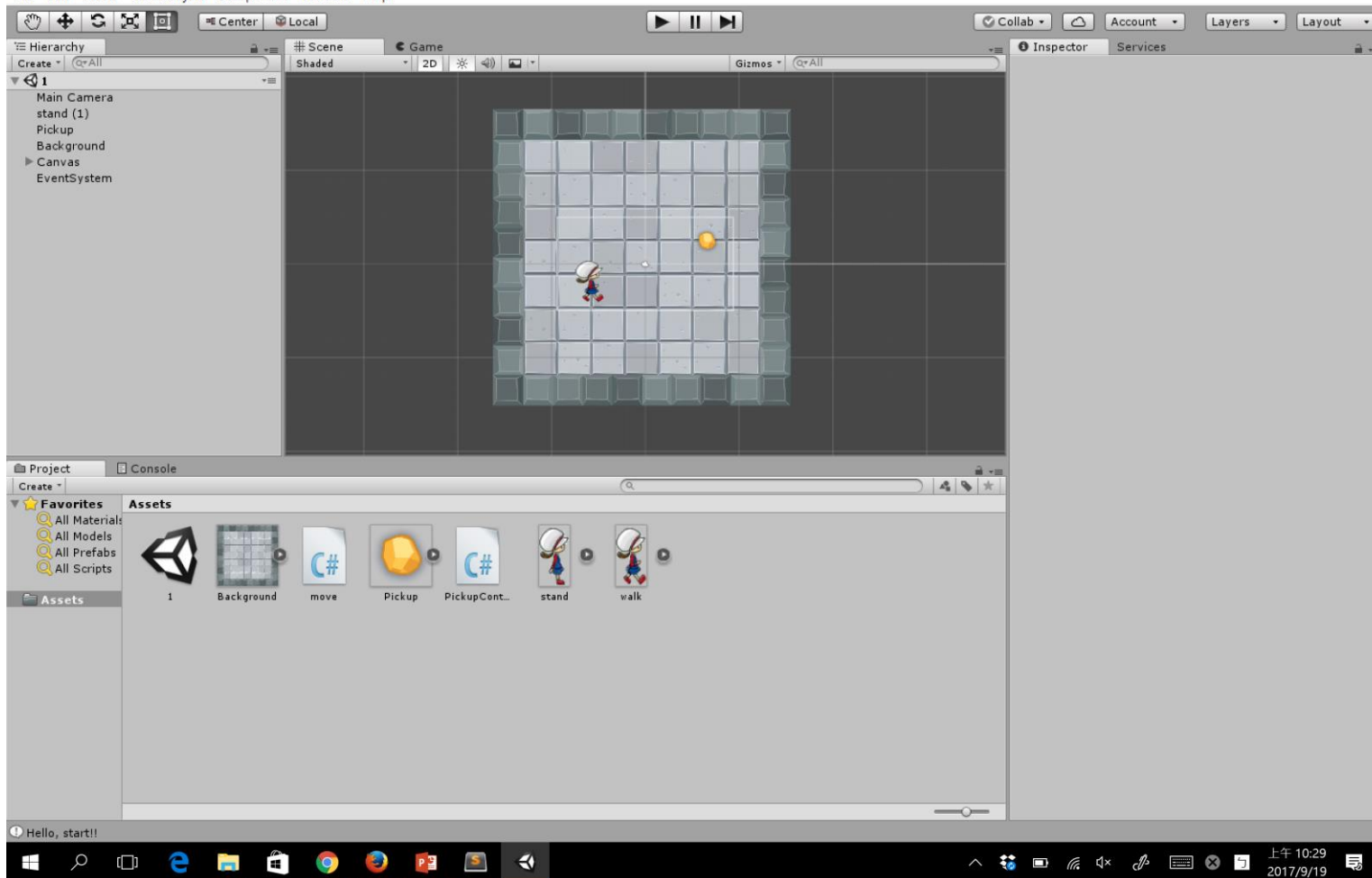


Position, Direction, Gamescore



Unity 2017.1.1f1 Personal (64bit) - 1.unity - 2d-17-test1 - PC, Mac & Linux Standalone <DX11>

File Edit Assets GameObject Component Window Help



Debug & moving

// Use this for initialization

```
void Start () {  
    Debug.Log("Hello, start!!");  
}
```

// Update is called once per frame

```
void Update () {  
    this.transform.position += new Vector3(1, 0, 0);  
}
```

Control speed

```
float speed = 20;
```

```
// Use this for initialization
```

```
void Start () {  
    Debug.Log("Hello, start!!");  
}
```

```
// Update is called once per frame
```

```
void Update () {  
    this.transform.position += new Vector3(speed*Time.deltaTime, 0, 0);  
}
```

Walking

```
void Update () {  
  
    if (Input.GetKey(KeyCode.D))  
    {  
        this.transform.position += new Vector3(speed*Time.deltaTime, 0, 0);  
    }  
  
    ...  
}
```


Jump

```
bool IsJump = false;
int JumpCount = -1;

void Update () {
    if(Input.GetKey(KeyCode.Space))
    {
        if(!IsJump)
        {
            IsJump = true;
            JumpCount = 0;
        }
    }
    ...
}
```

```
float dis = speed * Time.deltaTime;

if (JumpCount >= 0)
{
    if(JumpCount < 20)
    {
        this.transform.position += new Vector3(0, dis, 0);
    }
    else
    {
        this.transform.position += new Vector3(0, -dis, 0);
    }
    JumpCount++;

    if (JumpCount>39)
    {
        IsJump = false;
        JumpCount = -1;
    }
}
```

Sprite switch

```
public Sprite[] sprites;  
int sprites_index = 0;  
  
// Update is called once per frame  
void Update () {  
  
    if (Input.GetKey(KeyCode.D))  
    {  
        this.transform.position += new Vector3(speed * Time.deltaTime, 0, 0);  
        int i = (++sprites_index)%2;  
        this.GetComponent().sprite = sprites[i];  
    }  
}
```

**Art challenges technology;
technology inspires the art.
- John Lasseter**

OPENGL 2D

class RGBApixmap

```
RGBApixmap pic;
```

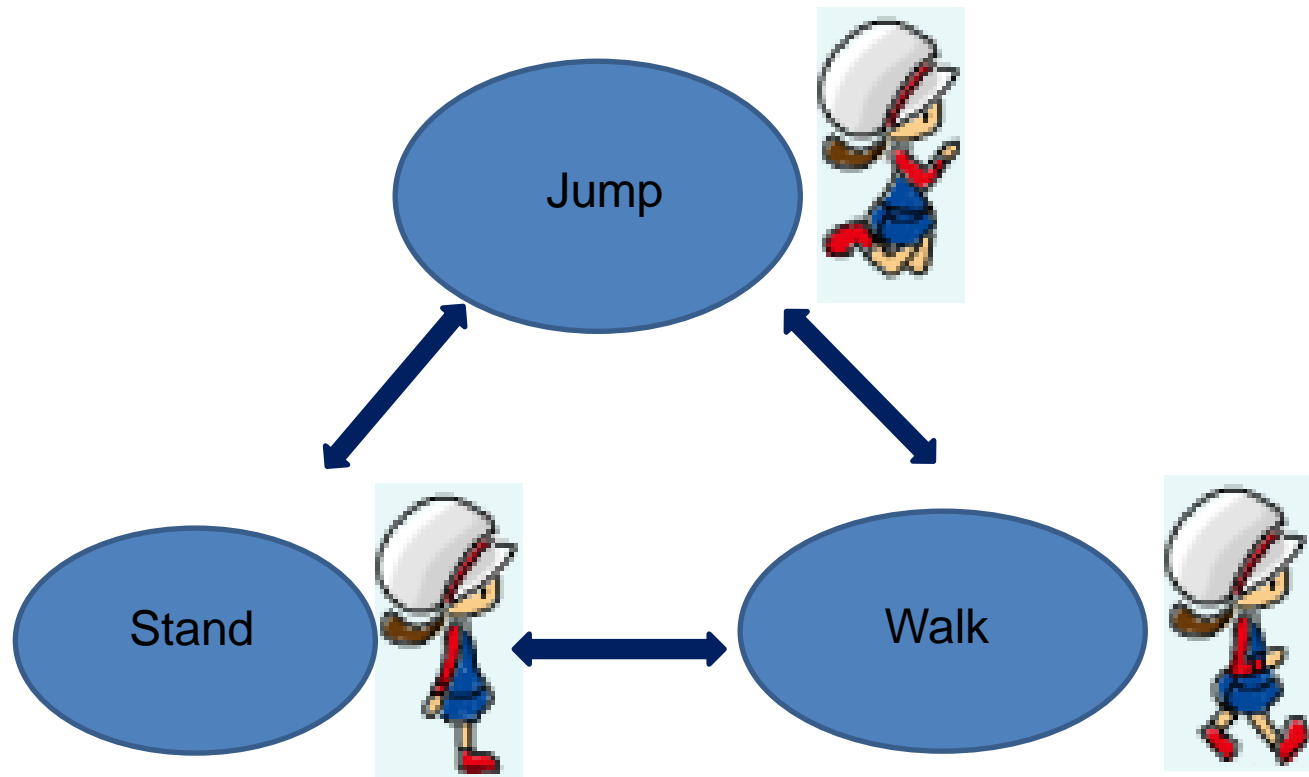
```
pic.readBMPFile( "stand.bmp" );
```

```
pic.setChromaKey(232, 248, 248);
```

```
// draw
```

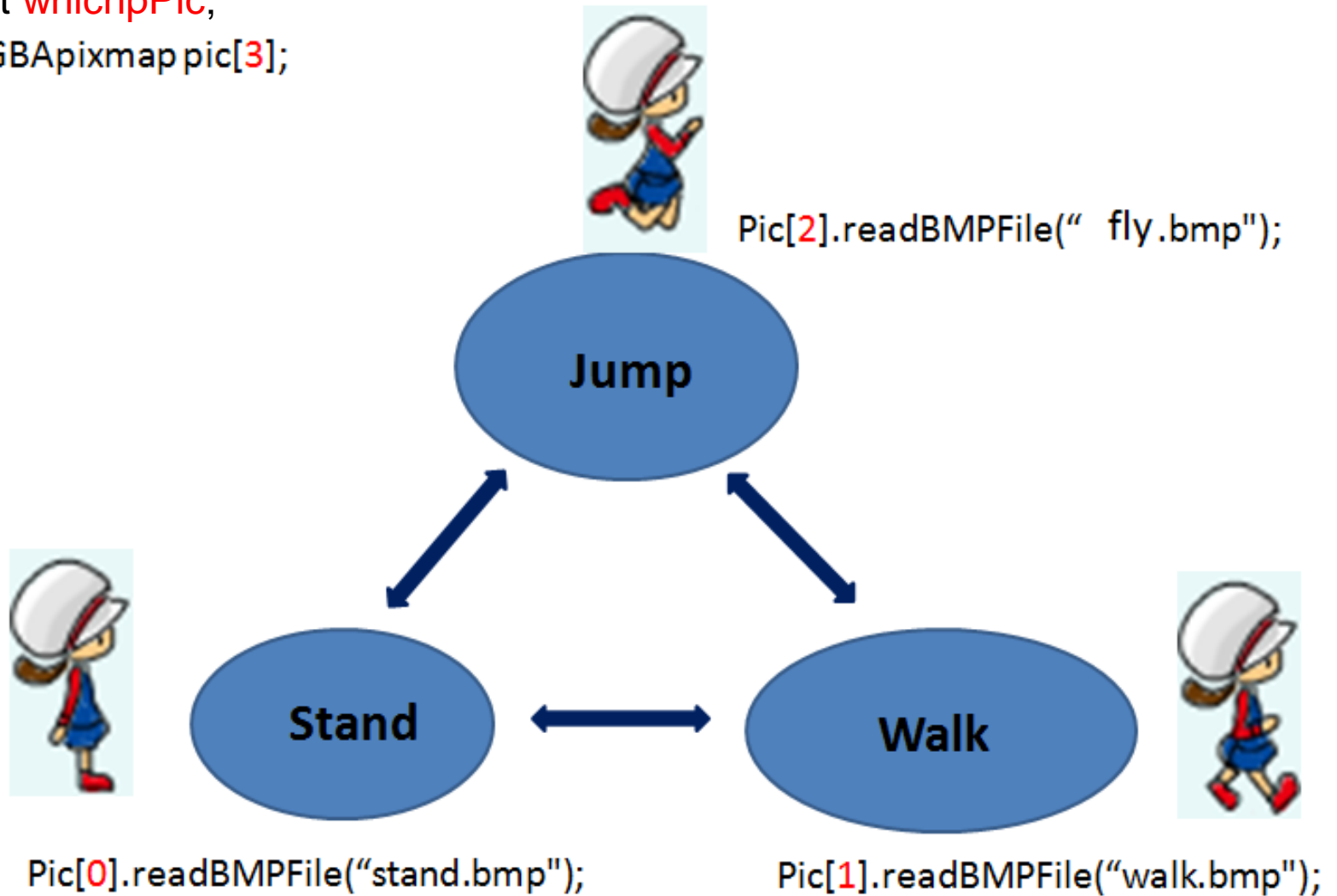
```
pic.blendtex(picX, picY, 1.0, 1.0);
```

State and Image



State

```
int whichPic;  
RGBAixmap pic[3];
```



Change State

```
void SpecialKeys(int key, int x,
int y)
{
    switch(key) {
        case GLUT_KEY_LEFT:
            picX -= 5;
            if (whichPic==0)
                whichPic=1;
            else
                whichPic=0;
            DirectState=1; //left
            break;

        case GLUT_KEY_RIGHT:
            picX += 5;
            if (whichPic==0)
                whichPic=1;
            else
                whichPic=0;
            DirectState=0; //right
            break;
    }
}
```

```
void display() {
    ...
    if (DirectState==0) { //向右
        pic[whichPic].blendTex(picX, picY, 1, 1);
    } else { //向左
        int offset = pic[whichPic].nCols; //圖的
        寬度
        pic[whichPic].blendTex(picX+offset,
        picY, -1, 1);
        //調整x位置，並以x=0為軸翻轉影像
    }
    ...
}
```

Font rendering

```
//Font
```

```
char mss[30];  
sprintf(mss, "Score %d", Gamescore);
```

```
glColor3f(1.0, 0.0, 0.0); //set font color  
glRasterPos2i(10, 550); //set font start position  
void * font = GLUT_BITMAP_9_BY_15;  
for(int i=0; i<strlen(mss); i++) {  
    glutBitmapCharacter(font, mss[i]);  
}
```

Press Space to Trigger jump()

```
void myKeys(unsigned char key, int x, int y)
{
    switch(key)
    {
        case ' ':
            if(jumpState==0) {
                jumpState=1;
                Gamescore++;
                jump(0);
            }
            break;
    }
    glutPostRedisplay();
}
```

Jump motion

```
void jump(int i)
{
    whichPic=2; //switch state

    if(i<=10) {
        if (i<5) picY+=4;
        else picY-=4;

        i++;
        glutTimerFunc( 100, jump, i);

    }else {
        whichPic=0;
        jumpState=0;
    }

    glutPostRedisplay();
}
```

