Tìm hiểu và sử dụng Action Script trong Flash để thiết kế game đơn giản: Apple Harvest

1. The game plan

Sử dụng quy trình thiết kế game flash truyền thống gồm 3 pha: thiết kế (pha 1), develop(pha 2), kiểm thử (pha 3). Pha thiết kế bao gồm xác định mạch truyện(story line), xác định các đối tượng sử dụng trong game, xác định luật chơi, …Pha phát triển chứa quá trình viết code cho các đối tượng trong game. Pha kiểm thử bao gồm việc chạy thử movie clip và sửa các lỗi gặp phải

1. Pha thứ nhất: thiết kế
2. Xác định mạch truyện

Đã đến mùa táo chín tại trang trại Táo Đỏ. Jovi, chú chim trong game, rất thích ăn táo và trang trại nơi đây có vô số cây táo. Trong suốt vụ mùa, chúng rơi xuống đất với số lượng rất lớn. Không may, những quả táo này thuộc loại táo mềm, khi chúng rơi xuống đất, chúng sẽ bị vỡ thành từng mảnh.

Hãy giúp Jovi của chúng ta chén thỏa thuê táo trước khi chúng rơi vỡ. Việc ăn một quá táo hoàn chỉnh luôn ngon hơn ăn những mảnh táo vương vãi trên đất

1. Xác định và tạo các thành phần trong game

Các thành phần cần trong game:

* Jovi



* Táo, cành cây, ..

* Các hình ảnh nền

* Ô điểm số
* Âm thanh nền, ..
* Các button chuyển cảnh, bật/ tắt âm thanh

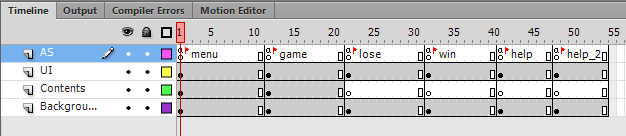
1. Xác định chi tiết game

* Jovi sẽ di chuyển theo di chuyển của con trỏ chuột
* Táo sẽ rơi ngẫu nhiên với tốc độ không đổi từ phía trên màn hình
* Táo khi chạm đất sẽ biến mất, trừ 10 điểm
* Mỗi quả táo Jovi ăn được sẽ được cộng 10 điểm
* Các vật phẩm khác như cành cây, … rơi xuống với tốc độ khác tốc độ rơi của táo; khi Jovi chạm vào sẽ nhận hoặc mất một số điểm
* Khi Jovi ăn no (đạt đến 500 điểm), trò chơi sẽ kết thúc
* Số điểm nhỏ hơn 0, thua màn chơi hiện tại

1. Pha thứ hai: pha phát triển
2. Bố cục cho file fla

Sử dụng 4 layer trong apple\_harvest.fla bao gồm AS, UI, Content, Background.

Layer AS chứa tất cà các kịch bản hành động(actionscript) cần tại frame đó. Ta không đặt bất cứ đối tượng đồ họa hay movieclip nào vào trong layer này. Layer UI chứa các đối tượng người dùng có thể tương tác vào như button, text, … Layer Content chứa các đối tượng đồ họa, movieclip, … được hiển thị trong game. Layer Backgrounnd chứa phông nền cho game



Timeline được chia thành 6 vùng: menu, game, lose, win, help, help\_2 tương ứng với các scene trong game. Khi bắt đầu vào game, timeline sẽ dừng tại vùng menu, bắt sự kiện để chuyển sang các vùng khác



Layer UI chứa 2 button chuyển cảnh, layer AS xử lí khi có thao tác kích chuột vào 2 button (chuyển sang cá vùng khác, tắt bật âm thanh)

stop();

var menuSound = new sndMenuMusic();

var scnMenu = new SoundChannel();

var helpSound = new sndFonMusic();

var scnHelp = new SoundChannel();

scnMenu = menuSound.play(0, int.MAX\_VALUE);

//chuyển sang scene game để chơi

startBtn.addEventListener(MouseEvent.CLICK, startHandling);

function startHandling (myEvent: MouseEvent){

scnMenu.stop();

gotoAndPlay("game");

}

// chuyển sang scene help, tắt âm menu, bật âm help

helpBtn.addEventListener(MouseEvent.CLICK, helpHandling);

function helpHandling (myEvent: MouseEvent){

scnMenu.stop();

gotoAndStop("help");

scnHelp = helpSound.play(0, int.MAX\_VALUE);

}

Tương tự với các vùng lose, win, help, help\_2.

Ta tập trung vào xử lí vùng game do các hoạt động chính đều tập trung ở đây

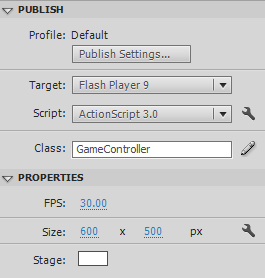
Lớp Background ta đặt một file ảnh cây cỏ, để trống lại một vùng trắng ở trên để xử lí chuyển động của mây, bầu trời trong GameController

Lớp Content chứa đối tượng movieclip: mcGameStage là nơi ta sẽ đặt các xử lí đồ họa vào trong

Lớp AS thực hiện hàm startGame

1. Thiết lập các giá trị khởi tạo trong GameController.as, tạo các đối tượng dạng MovieClip

Tạo liên kết từ file apple\_harvest.fla đến file trỏ tới file .as tương ứng



Hầu hết code ActionScript trong game đều nằm trong file .as này. Hàm startGame() khởi tạo giá trị ban đầu của các biến, trạng thái mặc định của game

public function startGame()

{

scn = mySound2.play(0, int.MAX\_VALUE);

speed = C.PLAYER\_SPEED;

gravity = C.GRAVITY;

score = C.PLAYER\_START\_SCORE;

randomChance = C.APPLE\_SPAWN\_CHANCE;

apples = new Array();

player = new Bird();

stones = new Array();

star = new Array();

mcBackground1 = new StarField();

mcBackground2 = new StarField();

mcBackground2.x = mcBackground1.width;

mcGameStage.addChildAt(mcBackground1,0);

mcGameStage.addChildAt(mcBackground2,0);

mcGameStage.addChild(player);

mcGameStage.addEventListener(Event.ENTER\_FRAME,update);

}

1. Xử lý tương tác

Game sử dụng chuột để chơi, do đó ta cần 2 biến để lưu tọa độ của chuột trong Stage: currMouseX, currMouseY. Ta sử dụng moveX và moveY để xác định hướng di chuyển của player

var moveX = 0;

var moveY = 0;

var currMouseX = mouseX;

var currMouseY = mouseY;

if (currMouseX > player.x)

{

moveX = 1;

}

else if (currMouseX < player.x)

{

moveX = -1;

}

if (currMouseY > player.y)

{

moveY = 1;

}

else if (currMouseY < player.y)

{

moveY = -1;

}

Sau đó, ta dịch chuyển player theo một tốc độ nhất định đến vị trí đặt trỏ chuột hiện tại

if (moveX > 0)

{

if (player.x + C.PLAYER\_SPEED <= currMouseX)

player.x += C.PLAYER\_SPEED;

}

else if (moveX < 0)

{

if (player.x - C.PLAYER\_SPEED > currMouseX)

player.x -= C.PLAYER\_SPEED;

}

if (moveY > 0)

{

if (player.y + C.PLAYER\_SPEED <= currMouseY)

player.y += C.PLAYER\_SPEED;

}

else if (moveY < 0)

{

if (player.y - C.PLAYER\_SPEED > currMouseY)

player.y -= C.PLAYER\_SPEED;

}

Sinh táo ngẫu nhiên, và thêm vào mcGameStage để hiển thị

if (Math.random() < randomChance)

{

var newApple = new Apple();

newApple.x = Math.random() \* C.APPLE\_SPAWN\_END\_X + C.APPLE\_SPAWN\_START\_X;

newApple.y = C.APPLE\_START\_Y;

apples.push(newApple);

mcGameStage.addChild(newApple);

}

Cho táo rơi, xóa khỏi màn hình nếu chạm đất

for (i = apples.length-1; i >= 0; i--)

{

apples[i].y += gravity;

if (apples[i].y > C.APPLE\_END\_Y)

{

mcGameStage.removeChild(apples[i]);

apples.splice(i,1);

score -= 10;

}

}

Kiểm tra xem táo có bị Jovi ăn không: sử dụng hàm hitTestObject () => nếu bị ăn thì loại bỏ khỏi màn hình

var playerPoint = new Point(player.x, player.y);

for (i = apples.length-1; i >= 0; i--)

{

if (player.hitTestObject(apples[i]))

{

score += 10;

mySound.play();

mcGameStage.removeChild(apples[i]);

apples.splice(i,1);

}

}

Xử lý cảnh bầu trời chuyển động ở background: gắn 2 background liền nhau và cho chuyển động cuộn bằng cách gắn điểm đầu của Background2 vào Background1, khi Background1 hoàn toàn nắm ngoài màn hình thì làm ngược lại, gắn điểm đầu của Background2 vào điểm cuối của Background1

mcBackground1.x -= C.SCROLL\_SPEED;

mcBackground2.x -= C.SCROLL\_SPEED;

if (mcBackground1.x <= -mcBackground1.width)

{

mcBackground1.x = mcBackground2.x +mcBackground2.width;

}

else if (mcBackground2.x <= -mcBackground2.width)

{

mcBackground2.x = mcBackground1.x + mcBackground1.width;

}

1. Pha cuối: kiểm thử