**BÁO CÁO ĐỒ ÁN**

1312120-Hà Minh Đạt

1312027-Đặng Gia Bảo

# Giới thiệu:

* Mô phỏng lại game Duck Hunt huyền thoại trên hệ máy NES.
* Các ngôn ngữ sử dụng: HTML, CSS, Javascript.
* Phần mềm sử dụng: Notepad++.
* Thư viện sử dụng: EaselJS.
* Hình ảnh: google image search.

# Thư viên easelJS:

Thư viện EaselJS Javascript cung cấp một chế độ đồ họa thêm vào cho canvas bao gồm một danh sách đầy đủ theo thứ bậc hiển thị, một mô hình tương tác cốt lõi, và các lớp helper để làm việc với đồ họa 2D Canvas dễ dàng hơn nhiều. EaselJS cung cấp các giải pháp thẳng về phía trước để làm việc với đồ họa phong phú và tương tác với HTML5 Canvas ... [*Learn more*](http://www.createjs.com/)



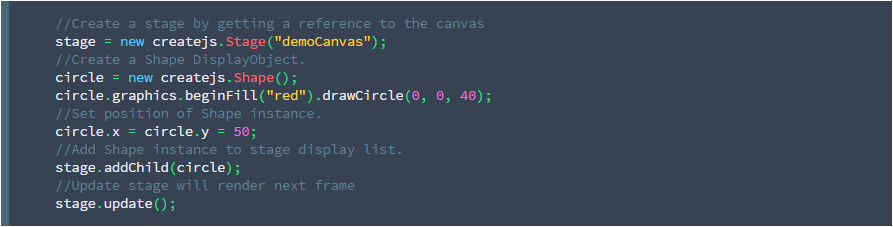
# Hướng dẫn sử dụng:

**Bắt đầu**:

Để bắt đầu với Easel, tạo ra một [*state*](http://www.createjs.com/docs/easeljs/classes/Stage.html) chứa đối tượng CANVAS, và thêm [*DisplayObject*](http://www.createjs.com/docs/easeljs/classes/DisplayObject.html) như 1 lớp con. EaselJS hỗ trợ:

* Hình ảnh sử dụng [*Bitmap*](http://www.createjs.com/docs/easeljs/classes/Bitmap.html)
* Đồ họa vector sử dụng [*Shape*](http://www.createjs.com/docs/easeljs/classes/Shape.html) và [*Graphics*](http://www.createjs.com/docs/easeljs/classes/Graphics.html)
* Bitmap động sử dụng [*SpriteSheet*](http://www.createjs.com/docs/easeljs/classes/SpriteSheet.html) và [*Sprite*](http://www.createjs.com/docs/easeljs/classes/Sprite.html)
* Trường hợp văn bản đơn giản sử dụng [*Text*](http://www.createjs.com/docs/easeljs/classes/Text.html)
* Các thùng giữ *DisplayObjects* khác sử dụng [*container*](http://www.createjs.com/docs/easeljs/classes/Container.html)
* kiểm soát HTML DOM elements sử dụng [*DOMElement*](http://www.createjs.com/docs/easeljs/classes/DOMElement.html)

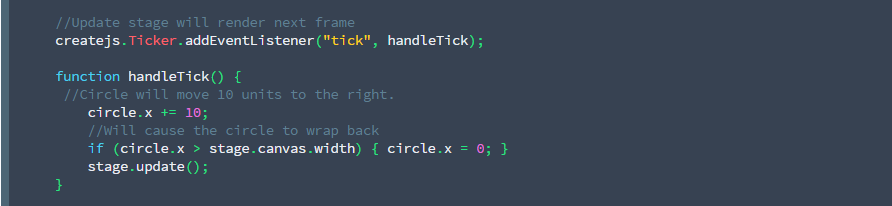
Tất cả các đối tượng hiển thị có thể được thêm vào các giai đoạn như trẻ em, hoặc rút ra một bức tranh trực tiếp.



**Tương tác người sử dụng**

Tất cả các đối tưởng chứa trong *stage* (trừ DOMElement) sẽ cử các sự kiện tương tác với các cách sử dụng một con chuột hoặc chạm vào. EaselJS hỗ trợ di chuột, báo chí, và phát hành sự kiện, cũng như một mô hình kéo-và-thả dễ sử dụng. Kiểm tra [*MouseEvent*](http://www.createjs.com/docs/easeljs/classes/MouseEvent.html) để biết thêm thông tin. [*Learn more*](http://www.createjs.com/docs/easeljs/modules/EaselJS.html)





# Project Solutions:

**Khởi tạo và load dữ liệu:**

* Khởi tạo và sử dụng Easel:

stage = new createjs.Stage("myCanvas");

* Thiết lập các Queue Asset và load âm thanh.

queue = new createjs.LoadQueue(false);

queue.installPlugin(createjs.Sound);

queue.on("complete", queueLoaded, this);

createjs.Sound.alternateExtensions = ["ogg"];

* Tạo và tải các manifest cho tất cả các Asset.

queue.loadManifest([

{id: 'backgroundImage', src: 'assets/background.png'},

{id: 'crossHair', src: 'assets/crosshair.png'},

{id: 'shot', src: 'assets/shot.mp3'},

{id: 'background', src: 'assets/countryside.mp3'},

{id: 'gameOverSound', src: 'assets/gameOver.mp3'},

{id: 'tick', src: 'assets/tick.mp3'},

{id: 'deathSound', src: 'assets/die.mp3'},

{id: 'batSpritesheet', src: 'assets/batSpritesheet.png'},

{id: 'batDeath', src: 'assets/batDeath.png'},

]);

queue.load();

* Tạo một bộ đếm thời gian mà cập nhật một lần mỗi giây

gameTimer = setInterval(updateTime, 1000);

**Tương tác với người dung:**

* Thêm *background image*

var backgroundImage = new createjs.Bitmap(queue.getResult("backgroundImage"))

stage.addChild(backgroundImage);

//Add Score

scoreText = new createjs.Text("1UP: " + score.toString(), "36px Arial", "#FFF");

scoreText.x = 10;

scoreText.y = 10;

stage.addChild(scoreText);

//Ad Timer

timerText = new createjs.Text("Time: " + gameTime.toString(), "36px Arial", "#FFF");

timerText.x = 800;

timerText.y = 10;

stage.addChild(timerText);

* Chạy *background sound*

createjs.Sound.play("background", {loop: -1});

* Tạo *Bat spritesheet*

spriteSheet = new createjs.SpriteSheet({

"images": [queue.getResult('batSpritesheet')],

"frames": {"width": 198, "height": 117},

"animations": { "flap": [0,4] }

});

* Tạo *Bat death spritesheet*

batDeathSpriteSheet = new createjs.SpriteSheet({

"images": [queue.getResult('batDeath')],

"frames": {"width": 198, "height" : 148},

"animations": {"die": [0,7, false,1 ] }

});

// Create bat sprite

createEnemy();

* Tạo *Button Play Game*

var background = new createjs.Shape();

background.name = "background";

background.graphics.beginFill("red").drawRoundRect(0, 0, 200, 60, 10);

var label = new createjs.Text("PLAY GAME", "bold 24px Arial", "#FFFFFF");

label.name = "label";

label.textAlign = "center";

label.textBaseline = "middle";

label.x = 200/2;

label.y = 60/2;

var button = new createjs.Container();

button.name = "button";

button.x = 400;

button.y = HEIGHT/2;

button.addChild(background, label);

stage.addChild(button);

var targets = [button,label,background];

* ListenMouseClick:

for (var i=0; i<targets.length; i++) {

var target = targets[i];

target.on("click", handleClick, null, false, null, false);

target.on("click", handleClick, null, false, null, true);

}

// Add ticker

createjs.Ticker.setFPS(15);

createjs.Ticker.addEventListener('tick', stage);

function handleClick(evt) {

stage.removeChild(button);

createjs.Ticker.addEventListener('tick', tickEvent);

// Set up events AFTER the game is loaded

window.onmousedown = handleMouseDown;

flat = 1;

}

# Link source code:

* Link source: <https://github.com/MinhDat/WebGame_BatHunt>
* Link online: <http://minhdat.github.io/WebGame_BatHunt/>