

CSIT 6910 INDEPENDENT PROJECT

CSIT6910A INDEPENDENT PROJECT

2013 SPRING

TEACHING PHYSICS USING 'ANGRY BIRDS' ELEMENTS

Student: LI Renhao

Supervisor: Prof. David Rossiter

CONTENTS

1. Introduction	1
1.1 Overview	1
1.2 Goals.....	1
2. Develop Environment	1
3. Physics Engine Development.....	1
3.1 Physics engine loop	2
3.2 Physics entity	2
3.3 Collision detection	3
3.4 Collision solver	4
3.5 User Interaction	4
3.6 Physics Engine Modes	5
3.7 Rendering	5
4. Website Structure	7
4.1 Introduction Page.....	7
4.2 Chapter 1 Page: The Basic Motion.....	7
4.3 Chapter 2 Page: Gravity	8
4.4 Chapter 3 Page: Dynamic	9
4.5 Chapter 4 Page: Collision and Momentum.....	9
5. Conclusion.....	10
6. Appendix.....	11
7. References.....	14

1. INTRODUCTION

1.1 OVERVIEW

Physics is mysterious and fascinating, but its textbooks are scary for many young students. A new physics learning system which uses a web game 'angry birds' as its interface may attract kids who are scared by physics formulas.

The learning system is a website which is organized into 4 chapters. All these chapters are essentially a mechanics primer. Chapter 1 is Basic Motion, it helps students understand the uniform rectilinear motion concept. Chapter 2 is Gravity, which aims to give explanations of mass, gravity and acceleration. Chapter 3 is Dynamic. The goal of this chapter is to help students understand force and its relations with acceleration and velocity. Chapter 4 is Collision. This chapter will teach students about momentum and energy concepts.

1.2 GOALS

The goals of the project is:

- Develop a 2d game physics engine based on HTML5 canvas
- Use the physics engine to develop 4 simple games for each chapter
- Organize each chapter into a tutorial in a website

2. DEVELOP ENVIRONMENT

System: windows 8

Web Server: IIS 7.5

Program Language: Javascript, HTML5 and CSS

Framework: jQuery

Browser: Chrome, IE 10

3. PHYSICS ENGINE DEVELOPMENT

A physics engine is computer software that provides an approximate simulation of certain physical systems, such as rigid body dynamics. In this project, I designed and developed my 2d physics engine which is called Thor2d.

3.1 PHYSICS ENGINE LOOP

The structure of Thor2d physics engine:

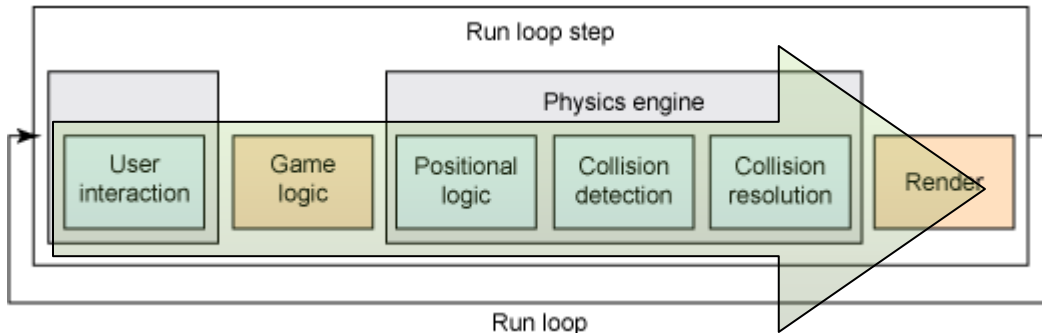


Figure 1

As shown in the figure 1, physics engines are fairly complex computationally but are rather simple to structure. Each calculation in that step consist of its own object or API. In Thor2d, I make use of the requestAnimationFrame() API ⁱⁱto run the main loop. There are 6 steps inside the game loop. They are user interaction, game logic, positional logic, collision detection, collision resolution and render. Since browsers are single-threaded, each step is executed one by one following the order.

3.2 PHYSICS ENTITY

Though the physics entity makes up the smallest and simplest of the components in the engine, it is the most important. As noted previously, the entity will represent each element on the screen. Entities, in both games and physics, represent the object's state by holding all the relevant metadata to that entity.

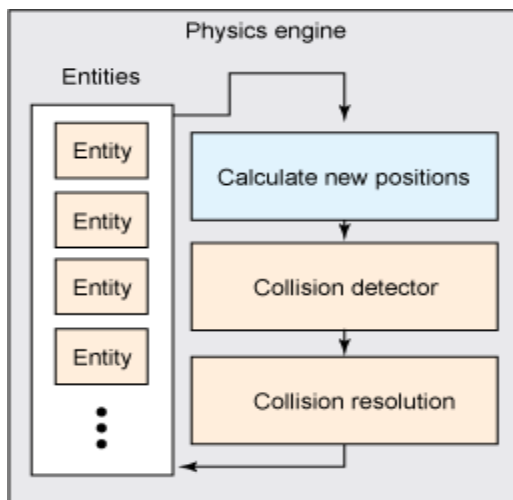


Figure 2

In Thor2d, the engine object maintains an entities array. During each game loop we calculate the new state of each entity according to the rules that we choose to perform in the game. For example, in chapter 1, I only want to simulate the uniform rectilinear motion, no acceleration, no force. So I need only to update the velocity of the player when the user click mouse in the hit area at the beginning. Then in each game loop only the position of each entities need to be updated.

3.3 COLLISION DETECTION

In Thor2d physics engine, I used the simple algorithm to detect collisions among all entities. For every collidable entities, iterate all though other physics entities and use the detection methods to judge whether two entities are collided.

After optimization of dual loop, the algorithm is shown in list 1:

```
//Notice that i< entities.length -1, because you do not need to
//check the last entity
for (i = 0; i < Entities.length - 1; i++)
{
    entityA = entities[i];
    for (j = i + 1; j < Entities; j++)
    {
        entityB = entities[j];
        if (entityA.collide(entityB))
        {
            // do whatever
        }
    }
}
```

List 1

I implemented two types of collision detection method in Thor2d, which are collideRect and collideCircle. They are used to detect collision between two rectangle entities and two circle entities respectively.



Figure 3

In figure 3, we use the position, width and height of two boxes to calculate if one is overlapping another. And we calculate the distance between the centers of two circles to check if $dist \leq r1+r2$.

3.4 COLLISION SOLVER

The collision detection step returns an array `collisions[]`. It is a two dimension array which contains references to two collision entities. This collisions array will serve as an input argument to the collision solver step.

Because we need to perform different behaviors after collision happened in different chapter, different solver function is needed. For example in Chapter 1, we only want to remove the entities after collision happened. However in chapter 3 we want an elastic effect, that is the entities will bounce back subjecting to conservation of momentum and energy. Thus I implemented 3 kinds of collision solver function in Thor2d.

- **resolveElastic:** calculate momentums and of two colliding entities, and change their velocity according to momentum conservation law.
- **resolveDisplace:** just change the positions of two colliding entities so as to eliminate collision.
- **resolveRemove:** immediately stops two entities and remove them from entities array.

3.5 USER INTERACTIONⁱⁱⁱ

In the tutorial all user inputs come from their mouse. User can click, drag and release his/her mouse button to control the behaviors of the bird. In different chapter the click will be handle by different methods.

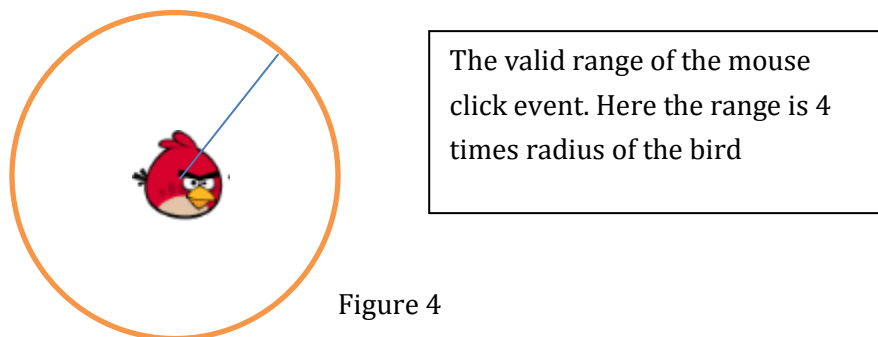


Figure 4

Let's Take a simple example, the code shows the handler function in Chapter 1. First it judge if the click is in valid area. If it is then change the velocity of the player entity.

```
inertiaMouseup: function(mouse) {
```

```

//
//console.log("mouseup event");

var d,dx,dy;
dx = mouse.x-this.getCenterX();
dy = mouse.y-this.getCenterY();
d = Math.sqrt(dx*dx+dy*dy);
//hit test
if(d<=this.radius*4)
{
    //handle the mousedown event here

    this.vx = dx;
    this.vy = dy;
    console.log("mouseup event"+dx+', '+dy);
    return true;
}
return false;
}

```

List 2

3.6 PHYSICS ENGINE MODES

Thor2d implemented 3 modes for updating the state of entities in each step. The three modes are:

- **INERTIA:** used to simulate uniform rectilinear motion. In this mode only the position of entities will change according to the initial velocity of entity during the game loop.
- **KINEMATIC:** used to simulate non-gravity motion. In this mode the calculation of velocities takes no consideration of gravity
- **DYNAMIC:** this mode apply gravity to all the entities.

3.7 RENDERING^{iv}

In brief, rendering means drawing entities on the canvas. Rendering is the last step in the game loop. After updating entities data, collision detection and solving collisions, we have the final version of entities data that we want to show you on the screen. Thor2d engine will use different function to render different types of entity.

- **drawBackground**
- **drawPlayer**
- **drawEnimies**
- **drawStage**

Those 4 function are quite similar to each other, for they just need to draw the right image on the right position. However in Angry Birds, the camera will trace the bird after it has been shot out. The idea to implement this effect is to scroll the background image left.

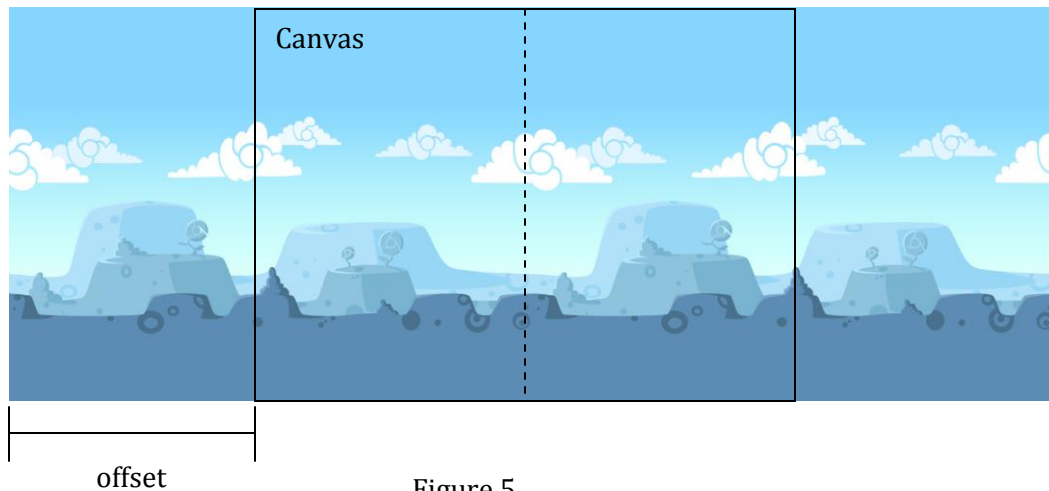


Figure 5

```
self.ctx.translate(-this.backgroundOffset, 0);
self.ctx.drawImage(img, 0, 0);
self.ctx.drawImage(img, bg_img.width, 0);
self.ctx.translate(this.backgroundOffset, 0);
```

List 3

Notice that in figure 5, Angry Birds use an tileable image as its background. Thus in *drawBackground* function we translate the coordinates system according to *backgroundOffset* value and draw two images on after another.

Since we want to focus on the bird and make it always in the middle of the canvas when it is flying, we need to derive the *backgroundOffset* value from the bird's X position.

```
var offset = player.x - canvas_w* .5;

if (offset > 0 && offset < canvas_w) {
    this.backgroundOffset = offset;
}
else if(offset > canvas_w){
    this.backgroundOffset = canvas_w;
}
else {
    this.backgroundOffset = 0;
}
```

List 4

4. WEBSITE STRUCTURE

The name of the website is PHYSICS & ANGRY BIRDS^v. There are 5 pages and the first one is the Introduction page, the rest are each chapters. At the beginning of each chapter, users will be shown a brief description of the chapter and a tips picture which tells them how to play the game.

4.1 INTRODUCTION PAGE

This page guides people to take the tutorial.

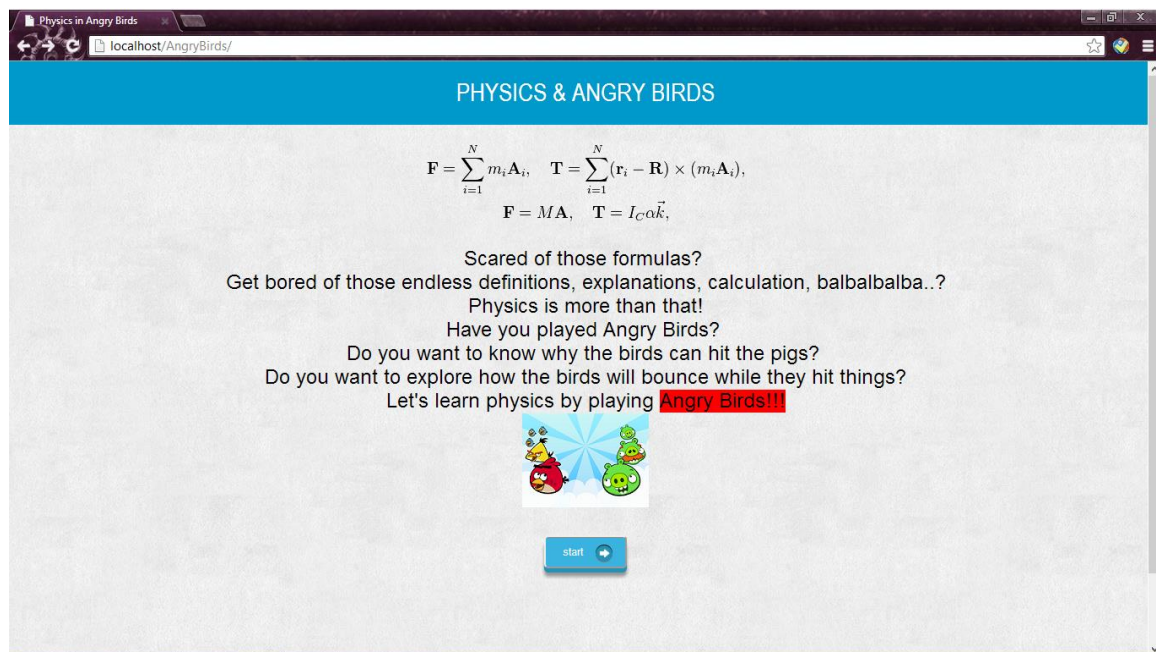


Figure 6

4.2 CHAPTER 1 PAGE: THE BASIC MOTION

The main topic of this page is uniform rectilinear motion. That is the first law of Newton. In this page the player's task is to hit the green pigs by pointing the direction which you want the bird to move to. When the player release the mouse, Thor2d engine will set the initial value of velocity according to the coordinates of the mouse. Then the bird will move in constant speed towards the direction from center to mouse point.

```
Player.vx = (mouseX - centerX) * FACTOR;
Player.vy = (mouseY - centerY) * FACTOR;
```



Figure 7

4.3 CHAPTER 2 PAGE: GRAVITY

First the page will show the concept of gravity. Then give the player hints how to play the game.

The game for this chapter is let user to “create” birds by click in the canvas. One click generates one bird in the coordinates of the mouse. The bird will fall because the engine in this chapter is set to “Dynamic” mode. Which means apply gravity to all the entities.



Figure 8

4.4 CHAPTER 3 PAGE: DYNAMIC

This chapter aims to let students understand why things moves. First I quotes the description of Newton's law. And then introduce the concept of force. The game in this chapter look like the normal Angry Birds game, the user drag the bird and shoots it out to hit the pigs.

Dragging the bird and release means give the bird an initial force. It is that particular force make the bird move.

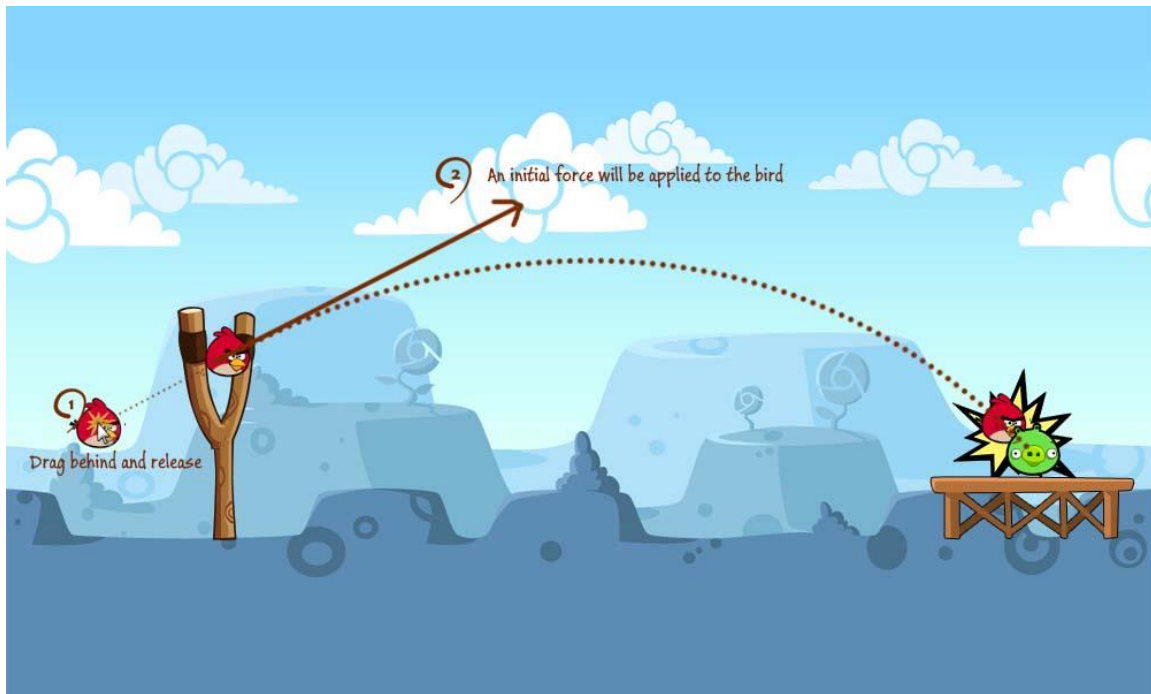


Figure 9

4.5 CHAPTER 4 PAGE: COLLISION AND MOMENTUM

The game in this chapter is identical to the previous one. This is a sophisticated game for I use an open source engine called Box2dWeb^{vi} instead my Thor2d engine to create it. The goal of this chapter is to give students intuitive of momentum conservation concept and energy conservation concept.

As hinted in the tips before starting the game, the user should try to hit the stage and see what happens to the pig. Ideally the motion energy of the bird will transfer to the stage elements which will result in changes of motion of the entities upon the stage. Finally the pig falls due to the movement of stage and dies.

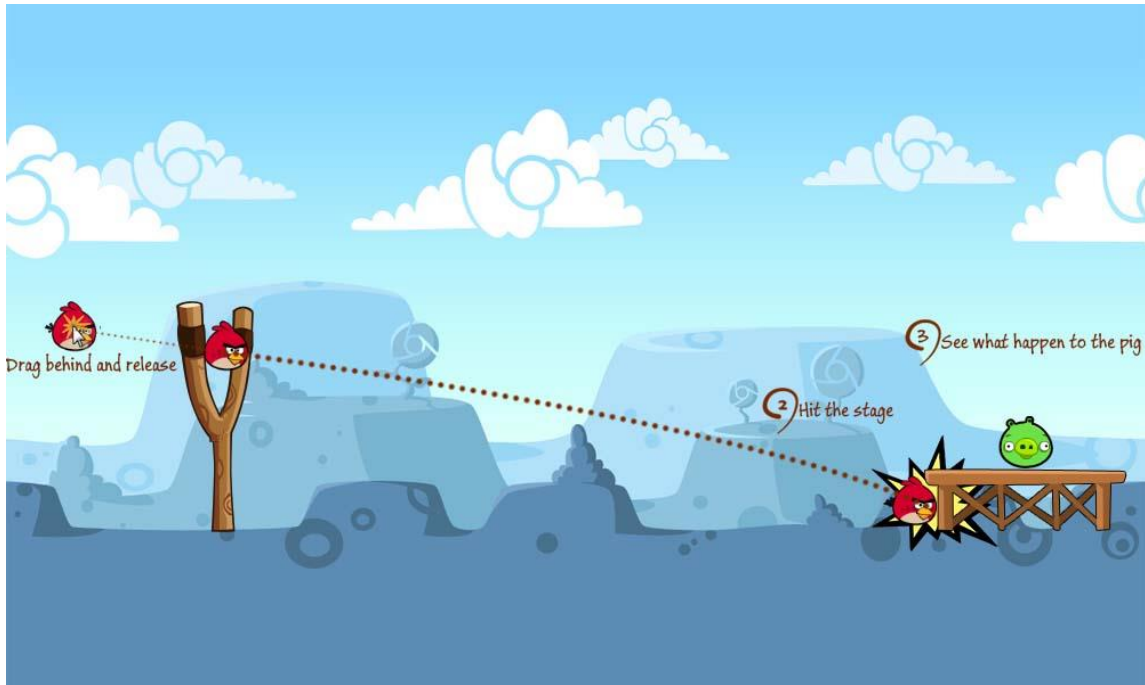


Figure 10

5. CONCLUSION

The most important part of this project is development of physics engine. In this project the engine is called Thor2d. When the physics engine is complete, the rest part of the project is to build up each game level. Because this is the tutorial for physic beginners, the webpage should be well organized.

In terms of engine design, I integrate many algorithms. Since the number of physics entities in all these games are relatively small, I just apply the simple methods. Many advanced algorithms can be used to improve the performance. For example in the collision detection part, other advanced algorithms such as Octree, KD Tree, BSP Tree, Hashing, can be used to replace the one I used.

In terms of web design, I could use CSS3 to make the web page more attractive and design the game more smartly.

Despite the drawbacks, the project reaches its goals which mentioned at the beginning.

6. APPENDIX

MINUTES OF 1ST MEETING

Date: Thursday, 7 March 2013

Time: 11 a.m.

Place: Rm.3512

Attending: Prof. Rossiter, LI Renhao

Absent: None

Recorder: LI, Renhao

1. Approval of Minutes

LI Renhao showed the basic particle parabola animation created by using HTML5 Canvas

2. Report on Progress

Since this is the first meeting, there is no progress to be reported.

3. Discussion Items and Things To Do

- Project idea
- Design the structure of the project website

4. Meeting Adjournment

None

MINUTES OF 2ND MEETING

Date: Wednesday, 27 March 2013

Time: 11:15 a.m.

Place: Rm.3512

Attending: Prof. Rossiter, LI Renhao

Absent: None

Recorder: LI Renhao

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

LI Renhao showed the interactive particle system created by using his own basic javascript 2d physics engine.

3. Discussion Items and Things To Do

- Learning basic 2d game programming
- Design the content of each chapter of the website

4. Meeting Adjournment

None

MINUTES OF 3RD MEETING

Date: Monday, 15 April 2013

Time: 11:30 a.m.

Place: Rm.3512

Attending: Prof. Rossiter, LI Renhao

Absent: None

Recorder: LI Renhao

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

LI Renhao showed one level of Angry Birds game developed by using Box2dweb engine.

LI Renhao finished the basic structure of the website.

3. Discussion Items and Things To Do

- Design games for each chapter
- Build up guidance information for each chapter

4. Meeting Adjournment

None

MINUTES OF 4TH MEETING

Date: Thursday, 16 May 2013

Time: 11:30 a.m.

Place: Rm.3512

Attending: Prof. Rossiter, LI Renhao

Absent: None

Recorder: LI Renhao

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

LI Renhao finished the final javascript physics engine development, and showed the teaching website that he created.

3. Discussion Items and Things To Do

- Finished the report

4. Meeting Adjournment

none

7. REFERENCES

-
- ⁱ http://www.w3schools.com/tags/ref_canvas.asp
- ⁱⁱ <http://paulirish.com/2011/requestanimationframe-for-smart-animating/>
- ⁱⁱⁱ <http://www.ibm.com/developerworks/web/library/wa-games/index.html>
- ^{iv} <http://www.ibm.com/developerworks/web/library/j-html5-game2/index.html>
- ^v <http://www.angrybirds.com/>
- ^{vi} <http://box2d.org/>