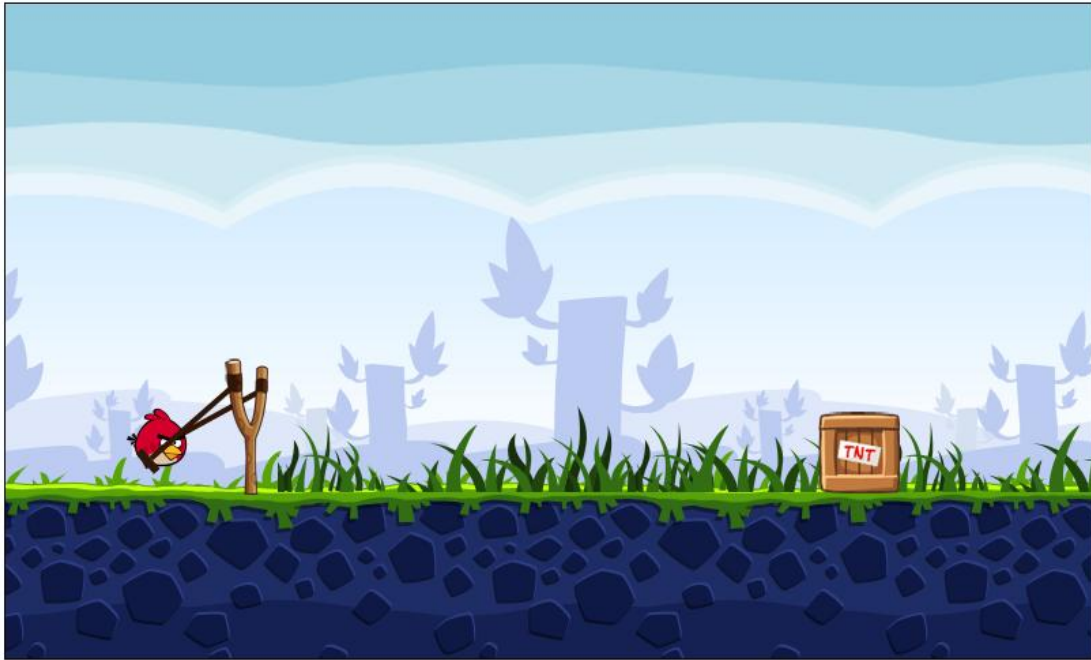


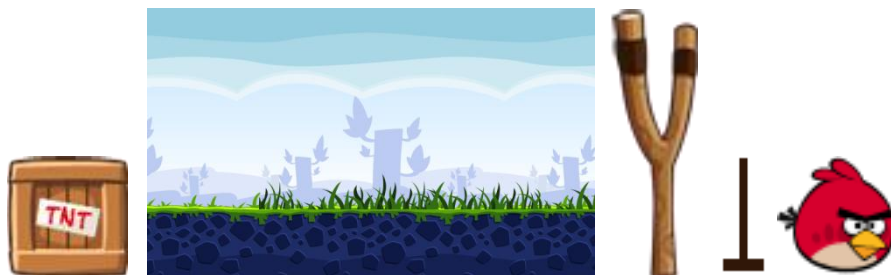
Lab 5a

目標：製作一個簡化了的 angry bird 遊戲，玩家可將左邊的 angry bird 射向右邊木箱。

場景製作：



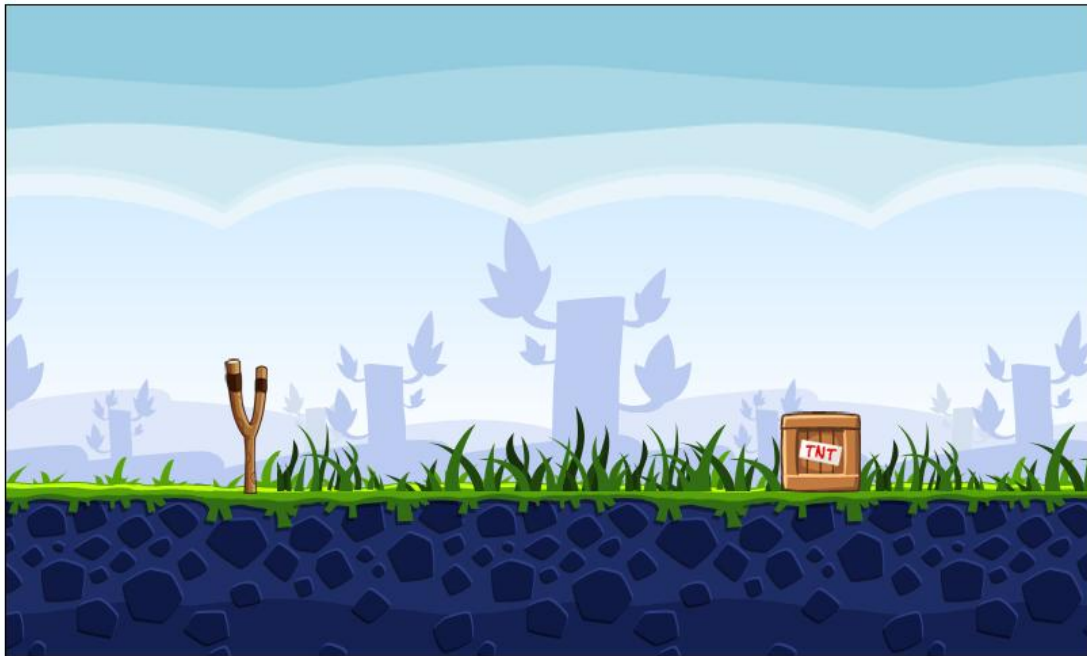
這個畫面主要包括幾個元件：木箱、背景、丫叉、橡筋、憤怒鳥。



1. 木箱、背景、丫叉和憤怒鳥都是用 div 加上 background-image 做的。
2. 橡筋由兩部份組成 rubber_body 和 rubber_holder，rubber_body 是長條形的 div，寬 4px，顏色是#301708，它的高度暫時可隨意設定，因為它之後會用 script 改變的。rubber_holder 則是長 20px，高 6px，顏色是#301708。然後 rubber_body 加入 margin 將它移到中間對齊。
3. 準備好圖片後，便開始組合場景，先將木箱、丫叉、橡筋、憤怒鳥放入背景

的 div，為方便隨意擺放，可將來它們設成 position: absolute。

4. 丫叉位置要垂直移位 260px，水平移位 160px，可通過改變 margin 來固定位置。橡筋和憤怒鳥位置暫時可隨意設定，如不想它顯示在畫面，可設定 top 為負值來隱藏它們。
5. 使用 jQuery 隨機放置木箱，寫一個 function 來處理，在網頁 load 起後執行。用 jQuery 設定木箱的 css，改變它的 top 和 left，top 是距離場景 div 約 300px，而 left 則是在場景的 300px 至 700px 之間。
Math.random() 可以生成 0—1 之間的隨機值。如要取得場景在網頁的絕對位置，可用 \$("#bg").offset().top 和 \$("#bg").offset().left，這便可以用來計算出木箱在網頁的絕對位置。



在 mouse move 時，angry bird 和橡筋會跟住 mouse 拉動。

首先加入 jquery library，便開始加入 script，加入：

```
$(function () {  
});
```

它的意思是當網頁的 element load 完後便開始執行上面的 script。之後我們加入 mouse 在背景移動的 event listener。

```
$(function () {  
    $("#bg").mousemove(function (e) {  
    });  
});
```

然後要做的就是移動 angry bird 到 mouse 的位置，和拉動橡筋，我們先加入 SetbirdPos(x,y) 和 PullRubbers() 兩個 function。而 angry bird 的位置應該是跟什麼位置有關？它是和 mouse 的位置相同，我們可以用 e.pageX, e.pageY 來得到 mouse 的位置。

```
$(function () {  
    $("#bg").mousemove(function (e) {  
        SetbirdPos(e.pageX, e.pageY);  
        PullRubbers();  
    });  
});
```

```
function SetbirdPos(x, y) {  
}
```

```
function PullRubbers() {  
}
```

設定位置的方法很簡單，只要更改 css 的 top 和 left。在 jquery 設 css 的方法是用 \$("#bird").css()，之後在加入 left 和 top

```
function SetbirdPos(x, y) {  
    $("#bird").css({  
        "top": y + "px",  
        "left": x + "px"  
    });  
}
```

再到拉動橡筋，這裡有兩件事要做的，第一是改變長度，第二個改變是什麼？旋轉物件，改變角度。

首先我們找出兩點，第一點是丫叉左邊，第二點是 angry bird 中心



要找出丫叉的座標，我們可以用 `offset` 來取得它的 `top` 和 `left`

```
x1 = $("#sling").offset().left;
```

而 `y` 的座標大約是丫叉向下 16px

```
y1 = $("#sling").offset().top + 16;
```

同樣用 `offset`，可以找出 `angry bird` 的中心，找到 `angry bird` 的位置後要如何計算出中心呢？我們可以再加上 `angry bird` 寬的一半

```
x2 = $("#bird").offset().left + $("#bird").width() / 2;
```

```
y2 = $("#bird").offset().top + $("#bird").height() / 2;
```

用兩個座標找出長度的公式是？

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

```
rubberLength = Math.sqrt(Math.pow((x1 - x2), 2) + Math.pow((y2 - y1), 2))
```

但這還不夠，長度不是去到中心就完結的，我們要再加上 `angry bird` 的半徑，即寬度的一半



```
rubberLength = Math.sqrt(Math.pow((x1 - x2), 2) + Math.pow((y2 - y1), 2)) +  
$("#bird").width() / 2;
```

設定橡筋長度，我們只改變 `rubber_body` 部份，而不是整個 `rubber1`，因為 `rubber holder` 是不需要改變的。

```
$("#rubber1 .rubber_body").height(rubberLength);
```

之後是計算角度，公式是 $\tan^{-1} \frac{x1-x2}{y2-y1}$

```
angle = Math.atan((x1 - x2) / (y2 - y1));
```

計算出來的角度是用弧度表示的。

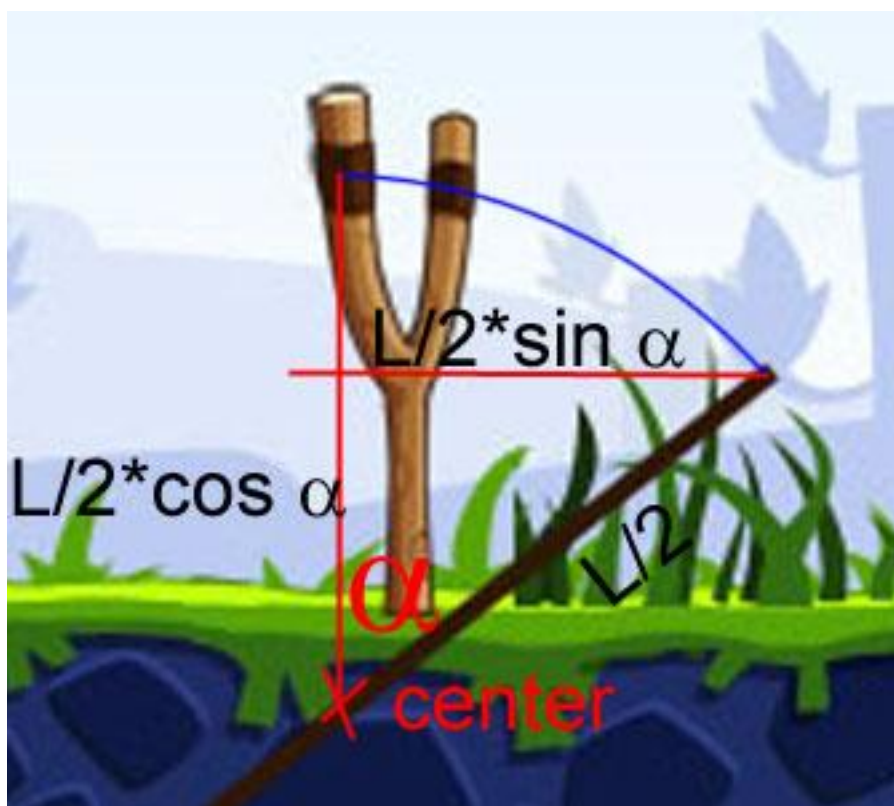
要旋轉物件，我們可以用 `jangle` 這個 plugin

```
$("#rubber1").jangle(angle * 180 / Math.PI);
```

然後將它放到丫叉

```
$("#rubber1").css({  
    "top": y1, "left": x1  
});
```

但這時發現位置不對，主要原因是旋轉的中心不是它的左上角，而是正中心



水平移位是橡筋長度一半再乘 $\sin \alpha$ ，垂直移位是橡筋長度一半減去橡筋長度一半再乘 $\cos \alpha$

$$dx = \frac{L}{2} * \sin \alpha$$

$$dy = \frac{L}{2} - \frac{L}{2} * \cos \alpha$$

```
dx = rubberLength / 2 * Math.sin(angle);
dy = rubberLength / 2 - rubberLength / 2 * Math.cos(angle);
$("#rubber1").css({
    "top": y1 - dy
    , "left": x1 - dx
});
```

後面的橡筋也是同樣的做法，只是丫叉上的點有所不同，會改變的部份有 **x1**, **y1**, **rubber1**，所以我們將以上的步驟寫成 **function**，給後面的橡筋重用。

```
function PullRubbers() {
    //前面的橡筋
    x1 = $("#sling").offset().left;
    y1 = $("#sling").offset().top + 16;
    GenRubber(x1, y1, 1);

    //後面的橡筋
    x1 = $("#sling").offset().left + 20;
    y1 = $("#sling").offset().top + 21;
    GenRubber(x1, y1, 2);
}

function GenRubber(x1, y1, rubberId) {
    x2 = $("#bird").offset().left + $("#bird").width() / 2;
    y2 = $("#bird").offset().top + $("#bird").height() / 2;
    angle = Math.atan((x1 - x2) / (y2 - y1));
    ...
}
```