

Lab 5b

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。

我們要拉動 angry bird 時，並不是只要 mouse 移動時便會動，而是要先按下 mouse，才可以拉動，然後放開 mouse，便會射出 angry bird，所以我們要更改一下，檢查是否先按下 mouse，才可以拉動 angry bird，我們加一個變數來存取現在是否按下 mouse：

```
var aim = false;
```

放下時，我們設定 aim = true，然後便開始放好 angry bird 和橡筋

```
$("#bg").mousedown(function (e) {  
    aim = true;  
    SetbirdPos(e.pageX, e.pageY);  
    PullRubbers();  
});
```

SetbirdPos 和 PullRubbers 我們要兩個一起重用，所以設一個 function

```
function SetPos(e) {  
    SetbirdPos(e.pageX, e.pageY);  
    PullRubbers();  
}
```

mousedown改寫成：

```
$("#bg").mousedown(function (e) {  
    aim = true;  
    SetPos(e);  
});
```

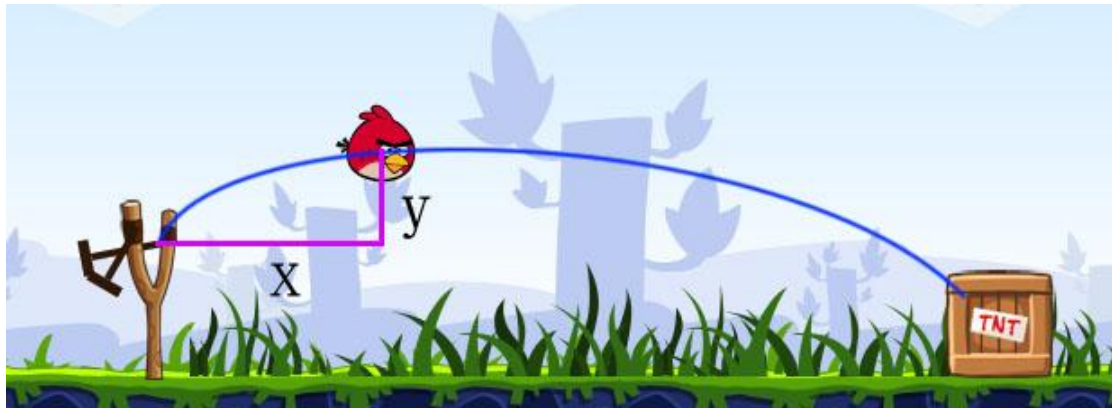
mousemove 時，則要先檢查 aim 是否 true，才放好 angry bird 和橡筋

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

之後放開 mouse，射出 angry bird，這時我們也要將 aim 改回 false

```
$("#bg").mouseup(function (e) {  
    ShootBird();  
    aim = false;  
});
```

射出後，angry bird 會進行拋物線運動，我們可以用以下的公式求得它進行中時的坐標：



$$X = u * \cos(\theta) * t$$

$$Y = u * \sin(\theta) * t - \frac{gt^2}{2}$$

u 是起始速度， θ 是發射角度， g 是地心加速， t 是經過的時間。

首先我們定義起始速度、發射角度、起始位置的全域變量

```
var initPos;  
var initAngle;  
var initSpeed;
```

起始位置則是 mouseup 時 angry bird 的坐標：`$("#bird").offset().left` 和 `$("#bird").offset().top`

發射角度是橡筋的角度的補角，所以我們加上一個變量 `currentAngle`

```
var currentAngle;
```

再在 `GenRubber` 加上：

```
function GenRubber(x1, y1, rubberId) {  
    ...  
    if (rubberId==1) currentAngle = angle;  
}
```

速度則是根據橡筋的長度而變，假設是橡筋長度的 $2/3$

```
function ShootBird() {  
    initPos = [$("#bird").offset().left, $("#bird").offset().top];  
    initAngle = (Math.PI/2 - currentAngle);  
    initSpeed = ($("#rubber1 .rubber_body").height() * 2/3);  
}
```

準備好初始值後，便可以計算出飛行中的座標，而座標是按時間改變，所以我們定一個 function：

```
function shoot(t) {  
}
```

再在 function 入面，用上面的拋物線公式計算座標，地心加速 g 是固定值 9.8

```
g = 9.8;
```

```
dx = initSpeed * Math.cos(initAngle) * t;
```

```
dy = initSpeed * Math.sin(initAngle) * t - (g * t * t) / 2;
```

```
SetbirdPos(initPos[0] + dx, initPos[1] - dy);
```

之後我們增加一下時間 t ，令 angry bird 到下一個位置，我們先給 t 一個數值，再測試一下，直到動作順暢為止

```
t += 0.25;
```

然後我們過一段時間後（例如 50ms），再用新的 t 值，重複再做 shoot：

```
setTimeout("shoot(" + t + ")", 50);
```

這樣，angry bird 便會拋物線地飛起來，但是它不會停下來，所以我們要設定停止的條件：

1. 射中目標
2. 飛出畫面外面

而偵測是否射中目標，我們用 collision 的 plugin，`$("#bird").collision(".block").size()` 如果等於 1，就表示 bird 和 block 兩個相撞，即是射中目標。以上兩個條件同時不成立時，我們才用 `setTimeout` 令 angry bird 繼續飛行。

```
if (($("#bird").offset().top < ($("#bg").height() - ($("#bird").height())) &&
```

```
($("#bird").collision(".block").size() == 0)) {
```

```
    setTimeout("shoot(" + t + ")", 50);
```

```
} else {
```

```
}
```

否則，在停止後，不論是否射中，我們也令 angry bird 消失，然後再在丫叉出現：

```
$("#bird").fadeOut("fast", function () {
```

```
    SetbirdPos($("#sling").offset().left - 30, ($("#sling").offset().top + 20);
```

```
    $("#bird").show();
```

```
    PullRubbers();
```

```
});
```

如果射中木箱，我們就令木箱爆炸，我們用 jQuery UI 的 `explode`，爆完後，再重新在新位置顯示木箱。

```
if ($("#bird").collision(".block").size() == 1) $(".block").effect("explode", function ()  
{
```

```
    ShowTarget();
```

```
    $(".block").delay(500).fadeIn("fast");
```

```
});
```

這個遊戲基本上完成，但有些地方，操作不當，可能會令遊戲出問題，所以我們要修改一下。

如果在angry bird飛行途中，我們又再按下mouse，便會再次觸發mousedown、mousemove和mouseup，可能會做成計算混亂，所以我們在mouseup時，加入shooting的變量來檢查和限制，如在shooting時，就不能觸發mousemove和mouseup

```
$("#bg").mousemove(function (e) {  
    if ((!shooting) && (aim)) {  
        SetPos(e);  
    }  
});
```

```
$("#bg").mouseup(function (e) {  
    if (!shooting) {  
        shooting = true;  
        ShootBird()  
        shoot(0);  
    }  
    aim = false;  
});
```

而射出後，完成飛行停下來時，我們重設 shooting

```
$("#bird").fadeOut("fast", function () {  
    ...  
    shooting = false;  
});
```

我們可以將 angry bird 拉向任何方向，如果直接向前拉到木箱，便必定射中！所以我們也限制 angry bird 只可以向後拉，於是在 SetPos 時加入 CheckBirdPos，如果拉到丫叉前面，就將位置鎖定在丫叉。另外也檢查不要將 angry bird 拉出畫面外面：

```
function SetPos(e) {  
    pos = CheckBirdPos(e.pageX, e.pageY)  
    SetbirdPos(pos[0], pos[1]);  
    PullRubbers();  
}  
function CheckBirdPos(x, y) {  
    if (y > $("#bg").offset().top + $("#bg").height() - $("#bird").height()) {  
        birdY = $("#bg").offset().top + $("#bg").height() - $("#bird").height();  
    } else {  
        birdY = y;  
    }  
}
```

```

    }

    if (x > $("#sling").offset().left - $("#bird").width()) {
        birdX = $("#sling").offset().left - $("#bird").width();
    } else {
        birdX = x;
    }
    return [birdX, birdY];
}

```

最後，當發射angry bird時，我想橡筋也一起跟住動直至angry bird離開丫叉，我們可以在shoot時不停PullRubbers()，這時橡筋的位置也會跟住一起更新。

```

function shoot(t) {
    ...
    if ($("#bird").offset().left < $("#sling").offset().left - 20) {
        PullRubbers();
    }
    ...
}

```

這個遊戲這樣便完成了！趕快試試你的眼界，看看射得中木箱嗎？

