

Физическая модель

bird = { //птица

 x: 60, //позиция птицы
 y: 0,

 frame: 0,
 velocity: 0,
 animation: [0, 1, 2, 1], //загрузка спрайта птицы 0,1,2,1

 rotation: 0,
 radius: 12,

 gravity: 0.25, //гравитация для птицы
 _jump: 4.6, //на сколько атица будет подниматься

 //падение и прыжки птицы

 jump: function() {
 this.velocity = -this._jump;
 },

 //обновление спрайтов,анимации и позиции птицы

 update: function() {

 //движения крылбями вызов спрайтов 0,1,2,1
 var n = currentstate === states.Splash ? 10 : 5;
 this.frame += frames % n === 0 ? 1 : 0;
 this.frame %= this.animation.length;

 //движение птицы на месте в верх и вниз
 if (currentstate === states.Splash) {

 this.y = height - 280 + 5*Math.cos(frames/10);
 this.rotation = 0;
 //ограничение движения и добав гравитации
 } else {

 this.velocity += this.gravity;
 this.y += this.velocity;

 // изменение оков при крсании птицы земли
 if (this.y >= height - s_fg.height-10) {
 this.y = height - s_fg.height-10;
 if (currentstate === states.Game) {
 currentstate = states.Score;
 }
 // сеорость прыжка
 this.velocity = this._jump;
 }

 //действие при косании земли и анимация падения и взлете
 if (this.velocity >= this._jump) {

```

        this.frame = 1;
        this.rotation = Math.min(Math.PI/2, this.rotation + 0.3);

    } else {

        this.rotation = -0.3;

    }

},

```

```

//CanvasRenderingContext2D ctx используется для рисования
draw: function(ctx) {
    ctx.save();
    // переносим ctx систему координат
    ctx.translate(this.x, this.y);
    ctx.rotate(this.rotation);

    var n = this.animation[this.frame];
    // рисуем птицу в центре
    s_bird[n].draw(ctx, -s_bird[n].width/2, -s_bird[n].height/2);

    ctx.restore();
}
},

```