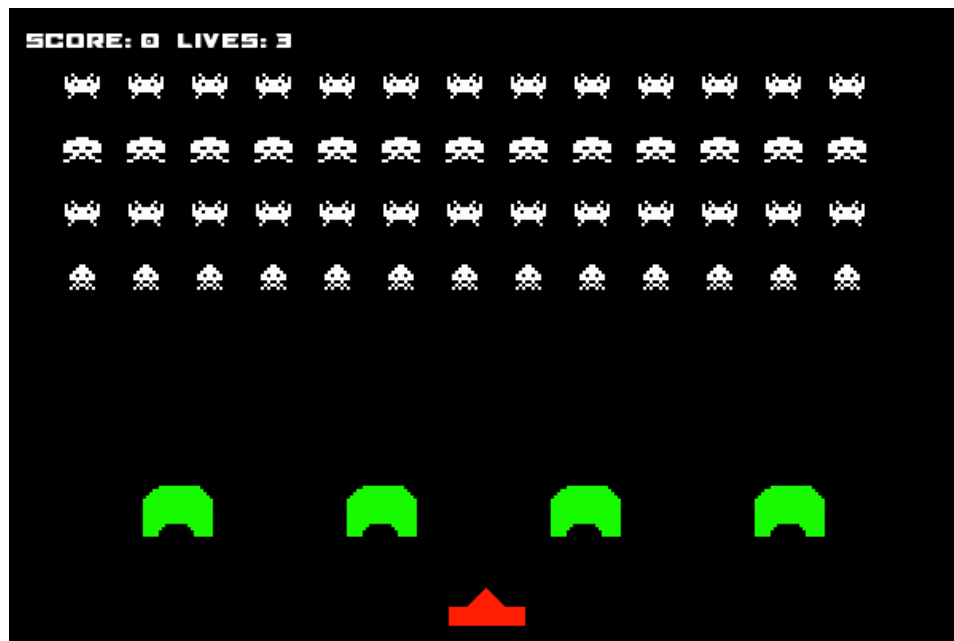


MODUL 14 – GAME PROGRAMMING



Gambar 1. Space Invaders dibuat oleh TAITO Jepang pada tahun 1978

Setelah kita berhasil membuat Game PUZZLE, maka tentu Anda penasaran untuk bereksplorasi membuat game-game jenis lainnya dengan menggunakan Adobe Flash. Pada dasarnya Adobe flash adalah lingkungan yang nyaman untuk game programming. Game Puzzle yang telah kita buat, menggunakan konsep DRAG and DROP dan tidak menggunakan karakter yang biasanya dijadikan sebagai tokoh utama dalam sebuah game. Sesuai dengan judulnya, game Puzzle tergolong dalam jenis "Puzzle Game".

Membuat karakter yang akan digunakan dalam game membutuhkan kreativitas dan keterampilan menggambar. Oleh karena itu, untuk membuat karakter dibutuhkan seorang Character Designer yang professional atau telah terbiasa membuat atau menciptakan karakter yang dapat digunakan, digerakkan, dan dianimasikan di dalam game. Dewasa ini Character Designer adalah sebuah profesi "mahal" yang banyak dibutuhkan oleh Game Developer dan Studio Animasi di seluruh dunia. Itulah sebabnya, kita sekarang mengenal banyak karakter game sebagai hasil karya Character Designer.

Hal lainnya yang juga diperlukan dalam sebuah game adalah gambar latar belakang. Seperti halnya gambar karakter, gambar latar belakang juga membutuhkan kreativitas dan keterampilan menggambar. Banyak orang yang bisa membuat gambar dengan menggunakan software vector seperti Adobe Flash, Adobe Illustrator, CorelDRAW, atau Adobe Photoshop, dan software lainnya. Namun, belum tentu dapat membuat gambar yang sesuai dengan tema yang ingin diciptakan di dalam game.

Jika kita melihat jauh ke belakang ketika industri game baru mulai dikembangkan, kita bisa mengetahui bagaimana tampilan game pada saat itu yang sangat sederhana jika

dibandingkan dengan tampilan game masa sekarang. Namun, di balik semua keterbatasan software dan hardware, banyak game yang digemari pada saat itu dan dimainkan oleh banyak orang, contohnya “Space Invaders” yang menjadi “wabah” baru yang menjangkiti dunia. Karakter alien pada game tersebut hingga kini tetap dikenal dan menjadi ikon yang mengingatkan orang pada game tersebut.

Selain karakter dan gambar background, music dan sound effect juga menjadi salah satu hal yang dibutuhkan oleh game. Itulah sebabnya, kita sekarang mengenal jenis music yang dimainkan dalam “Super Mario Bros” atau Sound Effect yang dimainkan oleh “Pacman”. Banyak orang yang kemudian memasang keduanya (music atau sound effect) sebagai Ringtone pada perangkat gadget yang dimilikinya.

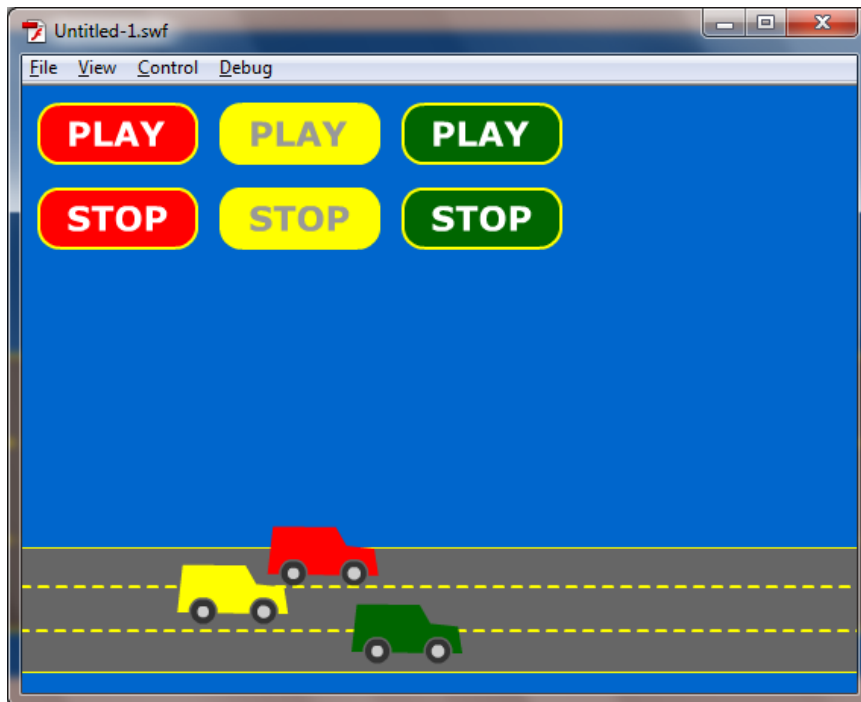
MENGENDALIKAN KARAKTER DENGAN OBJEK LAIN

Salah satu cara untuk mengendalikan karakter adalah dengan menggunakan objek lain sebagai controller. Objek pengendali tersebut dapat berupa Tombol atau Movie Clip yang diberi tugas khusus sebagai pengendali karakter.

Pada dasarnya, karakter pada game dibuat dengan Movie Clip sehingga bisa diberi beberapa gambar di dalamnya, atau bisa juga diberi animasi. Movie Clip adalah objek yang biasa dipergunakan untuk animasi interaktif. Gambar diam atau gambar dengan animasi jika kita jadikan sebagai sebuah Movie Clip, maka gambar atau animasi tersebut dapat dikendalikan dengan menyebutkan nama instance pada Action Script.

Movie Clip memiliki sejumlah method dan properties, sehingga dengan cara menjalankan method atau mengganti nilai properties maka Movie Clip akan bereaksi sesuai dengan method dan properties yang kita ubah.

Contoh berikut dapat menjelaskan bagaimana sebuah Movie Clip dapat dikendalikan dengan menggunakan tombol yang bekerja dengan cara mengubah property dari Movie Clip:



Pergerakan Movie Clip dari kiri ke kanan dapat diatur dengan mengubah nilai property X, yakni subordinat dari X dan Y. Seperti yang telah kita pelajari sejak di bangku SMP, koordinat terdiri atas sumbu X dan sumbu Y. Dengan demikian, agar Movie Clip dapat bergerak ke arah kanan, maka kita harus menambahkan secara terus menerus nilai property X.

Contoh:

```
var speed:Number = 10;

merah_mc.addEventListener(Event.ENTER_FRAME, merah);

function merah(e:Event) {
    merah_mc.x += speed;
}
```

Script di atas akan membuat movie clip yang diberi nama instance “merah_mc” akan bergerak ke arah kanan dengan kecepatan 10 pixel / frame.

Jika kita tambahkan dengan tombol STOP dan PLAY, maka kita dapat membuat mobil dengan nama instance merah_mc dapat berhenti dan berjalan dengan menggunakan script yang mengatur “speed” sebagai berikut:

```
stop_btn.addEventListener(MouseEvent.CLICK, berhenti);

function berhenti(e:Event) {
    speed = 0;
}
```

```
play_btn.addEventListener(MouseEvent.CLICK, jalan);
```

```
function jalan(e:Event){
    speed = 10;
}
```

Nah, dengan kedua tombol tersebut kita dapat mengendalikan Movie Clip untuk berhenti dan berjalan.

Bagaimana jika karakter dapat bergerak ke arah yang lain? Misalnya ke arah kiri, kanan, atas, dan bawah?

Untuk menggerakkan Movie Clip kita dapat menggunakan kedua sumbu X dan Y secara bersamaan. Contoh sebagai berikut:

```
var speedX:Number = 0;
var speedY:Number = 0;

merah_mc.addEventListener(Event.ENTER_FRAME, merah);

function merah(e:Event){
    merah_mc.x += speedX;
    merah_mc.y += speedY;
}

kanan_btn.addEventListener(MouseEvent.CLICK, kanan);

function kanan(e:Event){
    speedX = 10;
    speedY = 0;
}
```

```
kiri_btn.addEventListener(MouseEvent.CLICK, kiri);

function jalan(e:Event){
    speedX = -10;
    speedY = 0;
}
```

```
atas_btn.addEventListener(MouseEvent.CLICK, atas);

function atas(e:Event){
    speedX = 0;
    speedY = -10;
}
```

```
bawah_btn.addEventListener(MouseEvent.CLICK, bawah);

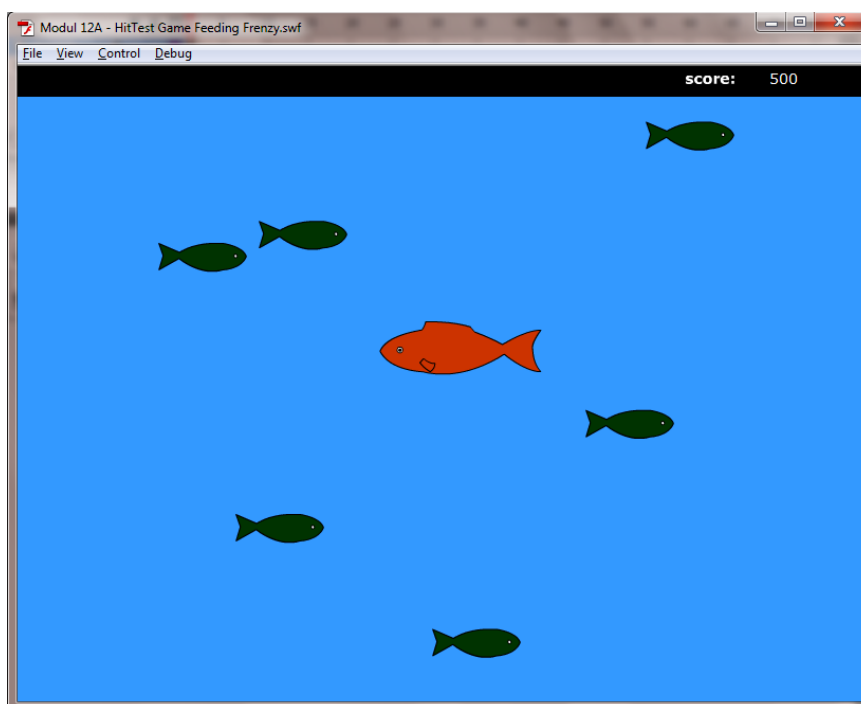
function bawah(e:Event){
    speedX = 0;
    speedY = 10;
}
```

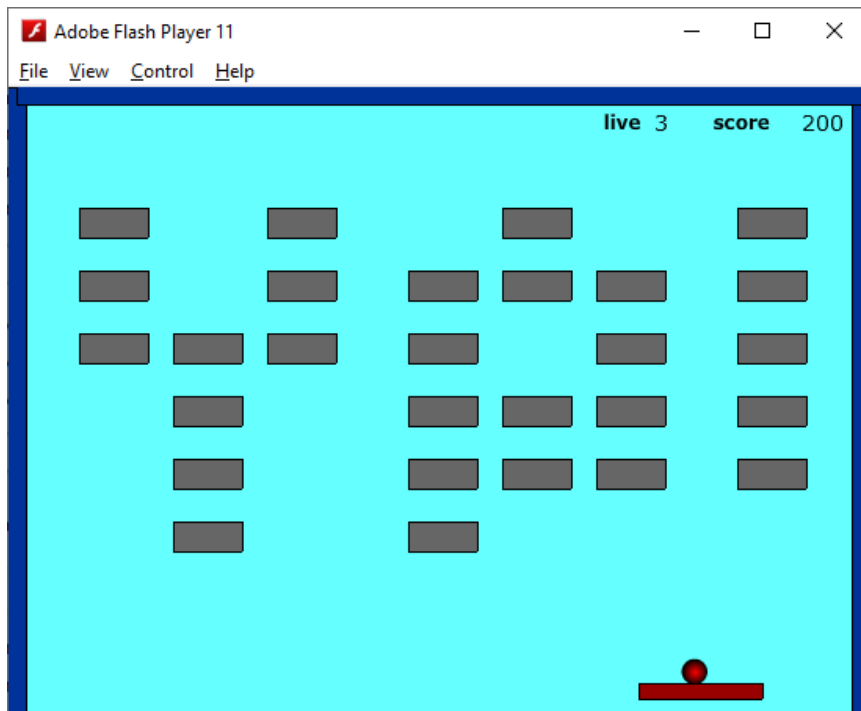
MENGENDALIKAN KARAKTER DENGAN MOUSE

Mengendalikan karakter dengan menggunakan Objek lain (Symbol Button) dapat dengan mudah dilakukan pada Mobile Game dengan Operating System Android atau iOS. Perangkat Mobile tidak menggunakan Mouse atau Trackpad akan tetapi menggunakan layar sentuh yang dapat dengan mudah dikendalikan dengan menggunakan jari. Setiap tombol yang disediakan dapat dengan mudah dijangkau dan di-klik karena jari kita memiliki mekanisme yang sempurna.

Pada computer yang tidak menggunakan layar sentuh, cara tersebut cenderung merepotkan karena kita harus menggerakkan pointer untuk bisa menjangkau tombol tertentu dan kemudian menekan tombol tersebut (Klik). Oleh karena itu, salah satu cara untuk mengendalikan karakter pada computer adalah dengan menggunakan Mouse. Cara ini banyak digunakan pada game yang membutuhkan pergerakan karakter ke segala arah. Biasanya penggunaan Mouse ini bersifat opsional karena juga dapat dilakukan dengan keyboard. Pilihan menggunakan Mouse atau Keyboard dapat diserahkan kepada pengguna.

Contoh berikut ini memperlihatkan bagaimana karakter dapat bergerak mengikuti gerakan Mouse:





Adapun script yang dapat menggerakkan karakter dengan menggunakan Mouse adalah sebagai berikut:

```
ikan_mc.addEventListener(Event.ENTER_FRAME, swim);

function swim(e:Event) {
    ikan_mc.x = mouseX;
    ikan_mc.y = mouseY;
}
```

Jika Movie clip hanya digerakkan di sepanjang sumbu X saja, maka script-nya adalah sebagai berikut:

```
papan_mc.addEventListener(Event.ENTER_FRAME, geser);

function geser(e:Event) {
    papan_mc.x = mouseX;
}
```

Yang harus dicatat adalah kursor berbentuk panah pada saat movie Flash dijalankan harus dihilangkan dengan menambahkan satu baris script sebagai berikut:

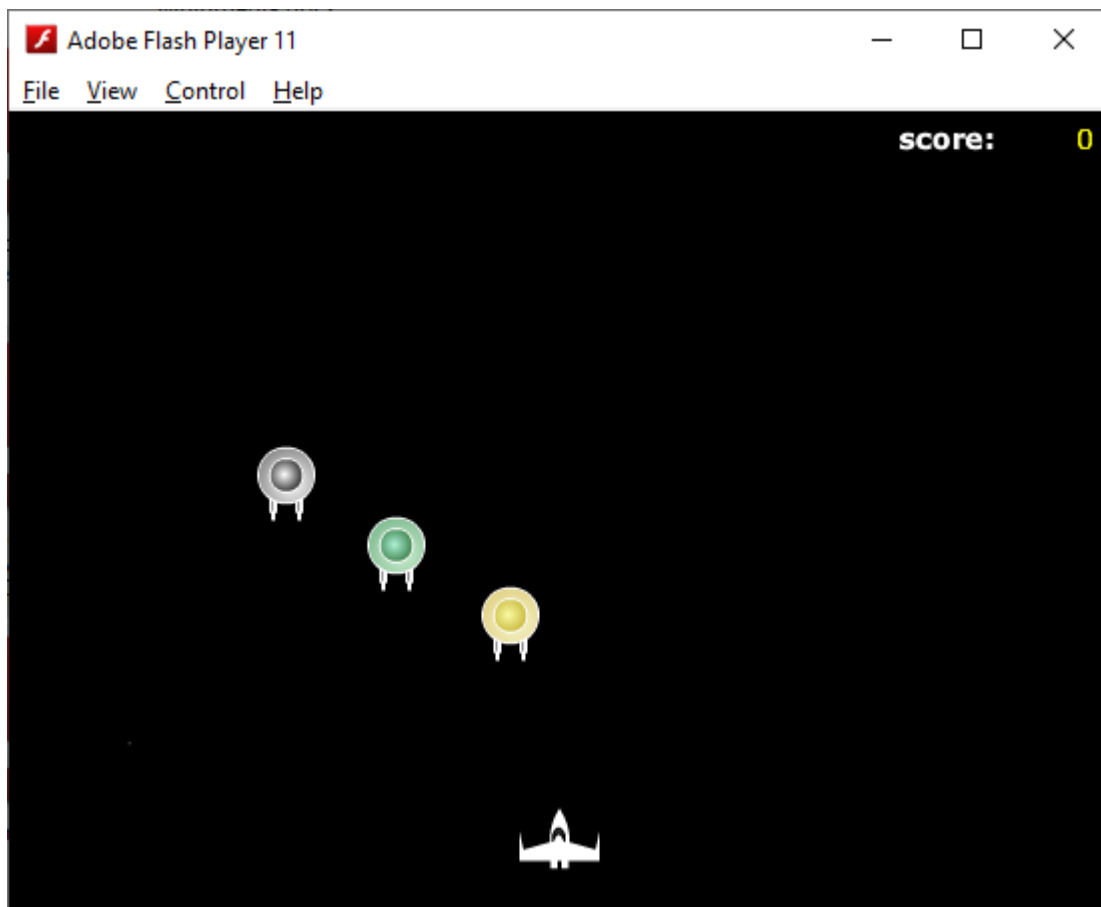
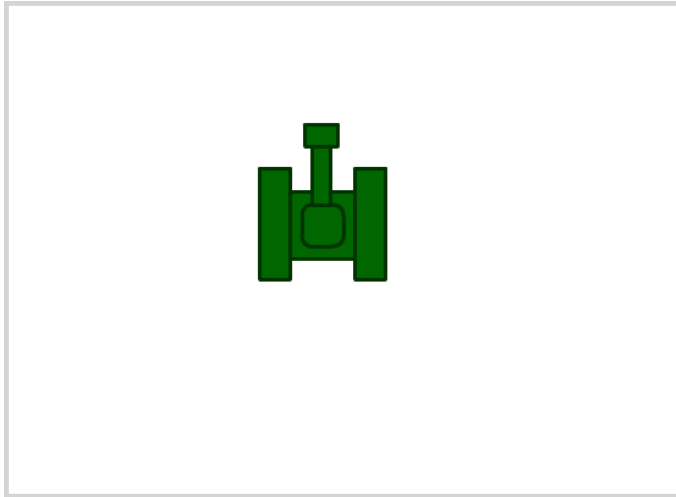
```
Mouse.hide();
```

Dengan script tersebut di atas, maka kita dapat membuat Movie clip yang dapat digerakkan dengan menggunakan Mouse.

MENGENDALIKAN KARAKTER DENGAN KEYBOARD

Selain menggunakan Mouse, Movie Clip juga dapat dikendalikan dengan menggunakan Keyboard. Metode ini sangat sering dipergunakan pada Game Interaktif. Seperti halnya dengan menggunakan Mouse, interaktivitas dengan Keyboard harus ditempatkan pada posisi Frame yang menunggu perintah dari waktu ke waktu (EnterFrame).

Contohnya adalah sebuah Movie Clip “tank_mc” atau “pesawat_mc” di bawah ini yang akan digerakkan dengan menggunakan keyboard sebagai berikut:



Terlebih dahulu kita harus menambahkan EventListener pada stage. Dengan demikian ketika pengguna menekan salah satu kunci pada keyboard, maka program akan mengetahuinya. Demikian halnya ketika pengguna melepaskan tekanan pada kunci tersebut, maka program pun akan mengetahui Event tersebut.

Jadi, ada 2 Event yang terjadi, yakni ketika pengguna menekan salah satu kunci (KeyboardEvent.KEY_DOWN) dan kemudian melepaskan tekanan pada kunci tersebut (KeyboardEvent.KEY_UP).

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDown);
stage.addEventListener(KeyboardEvent.KEY_UP, onKeyUp);
```

Mengetahui kedua Event ini menjadi penting karena kita akan menggerakkan Movie Clip ketika tombol keyboard ditekan dan menghentikan gerakan Movie Clip ketika tombol pada keyboard dilepaskan.

Berikut ini adalah cara mendeteksi dan mengirimkan kode kunci keyboard yang ditekan dan mengirimkan kode null ketika keyboard dilepaskan.

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDown);
stage.addEventListener(KeyboardEvent.KEY_UP, onKeyUp);

var keys:Array = [];

function onKeyDown(e:KeyboardEvent) {
    keys[e.keyCode] = e.keyCode;
}

function onKeyUp(e:KeyboardEvent) {
    keys[e.keyCode] = null;
}
```

Selanjutnya adalah dengan menambahkan EventListener pada Movie Clip “tank_mc” atau “pesawat_mc” tersebut. Adapun Event-nya adalah ENTER_FRAME yang akan dieksekusi beberapa kali per detik tergantung pada frame rate.

Berikut ini adalah script pada Movie Clip “tank_mc” yang akan bekerja menggerakkan dan memutar arah “tank_mc” sesuai dengan arah panah pada Keyboard:

```
tank_mc.addEventListener(Event.ENTER_FRAME, runGame);

function runGame(e:Event) {
    if(keys[Keyboard.UP]) {
        tank_mc.y -= 10;           //bergerak ke atas
        tank_mc.rotation = 0;      //mengarah ke atas
    }
}
```

```

if(keys[Keyboard.DOWN]){
    tank_mc.y += 10;           //bergerak ke bawah
    tank_mc.rotation = 180; //mengarah ke bawah
}
if(keys[Keyboard.LEFT]){
    tank_mc.x -= 10;           //bergerak ke kiri
    tank_mc.rotation = 270; //mengarah ke kiri
}
if(keys[Keyboard.RIGHT]){
    tank_mc.x += 10;           //bergerak ke kanan
    tank_mc.rotation = 90;    //mengarah ke kanan
}
}

```

MENDETEKSI TUMBUKAN ANTAR MOVIE CLIP

Untuk mengetahui apakah sebuah Movie Clip bertumbukan dengan Movie Clip lainnya, kita bias menggunakan method **hitTestObject()**. Metode ini sering dipergunakan pada banyak game seperti shooting game, racing game, fighting game, dan lain-lain. Dengan menggunakan method **hitTestObject()** kita dapat membuat script yang dibangkitkan oleh event tersebut sehingga memudahkan kita untuk mengendalikan Movie Clip disesuaikan dengan jalan cerita yang telah dirancang sebelumnya.

Contohnya misalnya pada game Feeding Frenzy, Movie Clip berupa ikan dapat digerakkan dengan menggunakan Mouse. Kemudian, ketika bertumbukan dengan ikan yang lebih kecil maka ikan kecil menghilang “dimakan” dan skor bertambah.

```

stage.addEventListener(Event.ENTER_FRAME, runGame);

function runGame(e:Event) {
    player_mc.x = mouseX;
    player_mc.y = mouseY;
    Mouse.hide();
    addChild(ikan);
    ikan.x += speed;
    if(player_mc.hitTestObject(ikan)) {
        ikan.x = -100;
        ikan.y = 25 + Math.random()*350;
        skor +=50;
        nilai.text = skor.toString();
    }
}

```

Dengan menggunakan method **hitTestObject()**, maka kita dapat membuat banyak game. Contohnya shooting game, racing game, fighting game, dan sebagainya.

Method **hitTestObject()** adalah method yang tidak terlalu presisi untuk digunakan pada Object dengan bentuk yang rumit, misalnya bentuk tubuh manusia. Oleh karena itu, pada

prakteknya bentuk yang rumit harus dipecah-pecah lagi ke dalam beberapa bagian dengan bentuk-bentuk yang lebih sederhana.