

CSE 190 M Flash Sessions

Session 3



Game Programming!!!!

(Finally)



(Finally)





Games

- Write Sprite classes for all of the objects in our game.
- Update all of the game objects every frame (ENTER_FRAME).
- Have event listeners for mouse clicks / key presses.
- Store state about the game (Is the game running or paused? How many points does the player have?)

Events

- Just like events in JavaScript
- Allows us to process mouse clicks and key presses.

Event	Description
MouseEvent.MOUSE_DOWN MouseEvent.MOUSE_UP	Triggered when mouse is pressed/unpressed.
KeyboardEvent.KEY_DOWN KeyboardEvent.KEY_UP	Triggered when key is pressed/unpressed.
Event.ENTER_FRAME	Triggered multiple times a second.
Event.ADDED_TO_STAGE Event.REMOVED_FROM_STAGE	Triggered when Sprite is added/removed from the display list (using addChild/removeChild).
Event.RESIZE	Triggered when movie is resized.

Events

Mouse Clicks

```
package {
    import flash.display.Sprite;
    import flash.events.MouseEvent;

    public class ClickProgram extends Sprite {
        public function ClickProgram() {
            var myCar:Car = new Car();
            addChild(myCar);
            myCar.addEventListener(MouseEvent.CLICK, onMouseDown);
        }

        private function onMouseDown(e:MouseEvent):void {
            // code that runs on mouse click
        }
    }
}
```

Events

Mouse Clicks

```
package {
    import flash.display.Sprite;
    import flash.events.MouseEvent;

    public class ClickProgram extends Sprite {
        public function ClickProgram() {
            var myCar:Car = new Car();
            addChild(myCar);
            myCar.addEventListener(MouseEvent.CLICK, onMouseDown);
        }

        private function onMouseDown(e:MouseEvent):void {
            // code that runs on mouse click
        }
    }
}
```


Events

Mouse Clicks

A slightly different way:

```
package {
    import flash.display.Sprite;
    import flash.events.MouseEvent;

    public class ClickProgram extends Sprite {
        public function ClickProgram() {
            var myCar:Car = new Car();
            addChild(myCar);
            myCar.addEventListener(MouseEvent.CLICK, function() {
                // code that runs on mouse click
            });
        }
    }
}
```

Events

Key Presses

```
package {
    import flash.display.Sprite;
    import flash.events.KeyboardEvent;

    public class KeyProgram extends Sprite {
        public function KeyProgram() {
            stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDown);
        }

        private function onKeyDown(e:MouseEvent):void {
            // code that runs on key press
        }
    }
}
```

Events

Key Presses

How do we know what key is pressed?

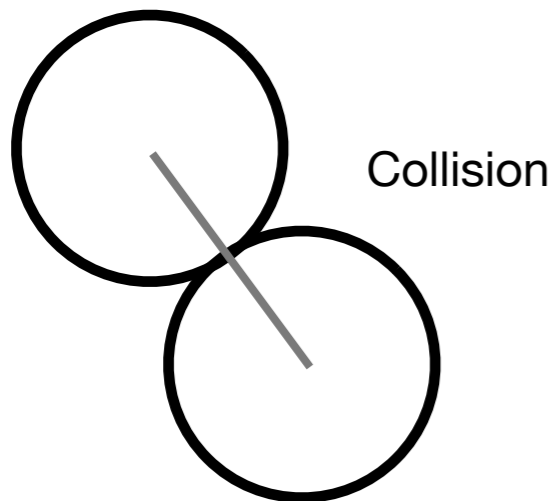
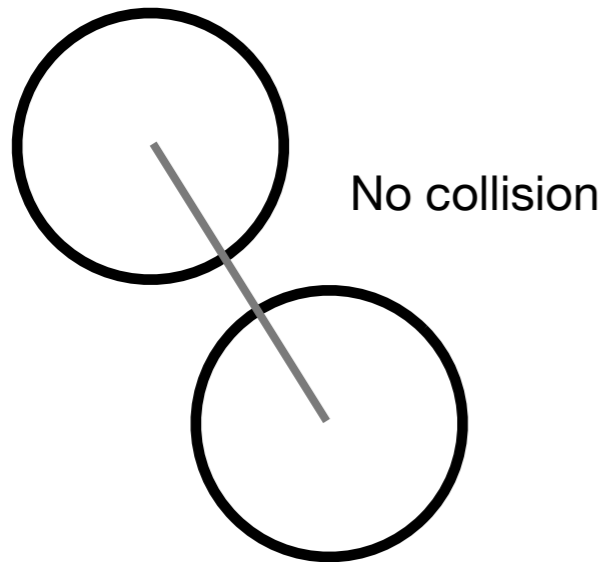
```
import flash.ui.Keyboard;

...

private function onKeyDown(e:KeyboardEvent):void {
    if (e.keyCode == Keyboard.LEFT) {
        // code that runs when left key is pressed
    } else if (e.keyCode == Keyboard.A) {
        // code that runs when "A" key is pressed
    }
}
```

Collision

Circles



```
var dx:Number = s1.x - s2.x;  
var dy:Number = s1.y - s2.y;  
var dist:Number = Math.sqrt(dx*dx + dy*dy);  
var rad1:Number = s1.width / 2;  
var rad2:Number = s2.width / 2;  
if (dist <= rad1 + rad2) {  
    // collide!  
}
```

Timer

```
import flash.utils.Timer;
import flash.events.TimerEvent;

...

// create a new timer that "ticks" every 1000 milliseconds
var myTimer:Timer = new Timer(1000);
myTimer.addEventListener(TimerEvent.TIMER, function() {
    // code that runs every tick
});
```

Text Fields

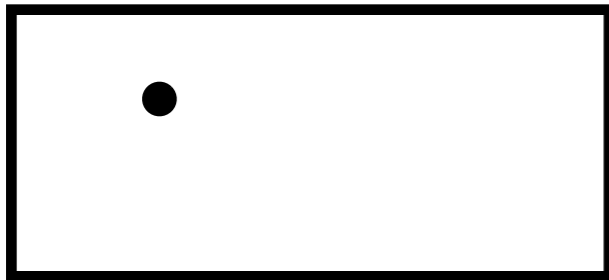
```
var textFormat:TextFormat = new TextFormat();  
textFormat.size = 24;
```

```
var myTextField:TextField = new TextField();  
myTextField.x = 10;  
myTextField.y = 20;  
myTextField.width = 100;  
myTextField.defaultTextFormat = textFormat;  
addChild(myTextField);
```

```
myTextField.text = "Hello World!"
```

More Collision

Point vs. Rectangle



```
if (p.x > rect.x &&  
    p.x < rect.x + rect.width &&  
    p.y > rect.y &&  
    p.y < rect.y + rect.height) {  
    // collide!  
}
```

More Collision

Collision Reaction

When two things collide, we could:

- End the game
 - hero dies, game is paused
 - easy!
- Bounce
 - easy for vertically or horizontally aligned objects, just reverse vx or vy
 - hard for circles or complex shapes
- Place one colliding object directly outside the other
 - it's doable for point vs. rectangle, but very difficult for everything else
- You can also use a physics library for more realistic collisions:

box2d: <http://box2df1ash.sourceforge.net>

AWESOME box2d video tutorials: <http://www.kerp.net/box2d/>

Motion & Animation

Common strategies

- Math.sin() can be used for oscillating or circular motion.

```
// every frame:  
theta += 0.1;  
mySprite.x = Math.sin(theta) * 100;
```

- Cos/sin functions are useful for spring and gravity effects / following points.
- If you add to position every frame, you get constant velocity.
- If you add to velocity every frame, you get constant acceleration.
- For advanced animation, check out the Tweeners library:

Tweeners: <http://code.google.com/p/tweeners/>

Other resources

- A couple common game libraries:
 - <http://flixel.org>
 - <http://flashpunk.net/>
- Books:
 - <http://www.amazon.com/Essential-ActionScript-3-0-Colin-Moock/dp/0596526946>
 - <http://www.amazon.com/Foundation-ActionScript-Animation-Making-Things/dp/1590595181>
- Tutorials:
 - <http://www.gotoandlearn.com/index.php>
 - <http://www.kongregate.com/games/Kongregate/shootorial-0>