Game Events

Most games are full of things that happen, and other things that result:

- Move a joystick → a player runs forward
- Two planets collide → an explosion occurs
- A race car rounds a corner → a flag waves
- A character sees the player → becomes angry
- A gun fires → an echo sound is produced

We have seen event handling associated with input devices. SAGE supports these in its Input Manager.

Event Handling

Complex games may have thousands of game objects. It isn’t feasible to check every object for every relevant possible state change.

We need an efficient way of:

- Abstracting events as “event objects”
- Associating listeners with types of event objects
- Triggering the production of event objects when events occur
- Handling those events in the event listeners

Event Handling in SAGE

SAGE package sage.event contains:

Interfaces:
- IEventManager
- IEventListener
- IGameEvent

Classes:
- EventManager (a global singleton)
- AbstractGameEvent

A reference to the EventManager singleton can be obtained by calling:

EventManager.getInstance()
Defining an Event Type

```java
import sage.event.*;

public class CrashEvent extends AbstractGameEvent {
   // Defines an event object that is passed to all
   // observers of this event type.
   private int whichCrash; // may contain local data
   public CrashEvent(int c) { whichCrash = c; } // an example constructor
   public int getCrashCount() { return whichCrash; } // another example function
}
```

Defining an Event Listener

```java
import sage.event.*;
import java.lang.String;

public class MyListener implements IEventListener {
   public boolean handleEvent(IGameEvent event) {
      // if the event has programmer-defined information in it,
      // it must be cast to the programmer-defined event type.
      CrashEvent c = (CrashEvent) event;
      int crashCount = c.getWhichCrash();
      if (crashCount % 2 == 0)
         this.setColorBuffer(colorBuf1);
      else
         this.setColorBuffer(colorBuf2);
      return true;
   }
}
```

Associating an Event with a Listener

```java
import sage.event.*;
...
EventManager eventMgr = EventManager.getInstance();
MyListener aListenerObject = new MyListener();
eventMgr.addListener(aListenerObject, CrashEvent.class);
```

Triggering an Event

```java
CrashEvent newCrash = new CrashEvent(numCrashes);
eventMgr.triggerEvent(newCrash);
```