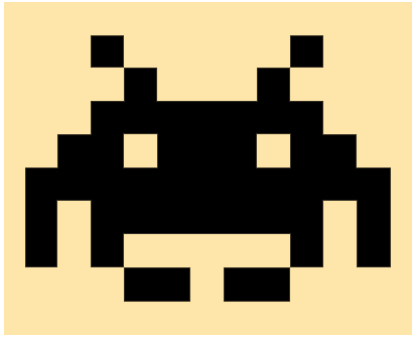
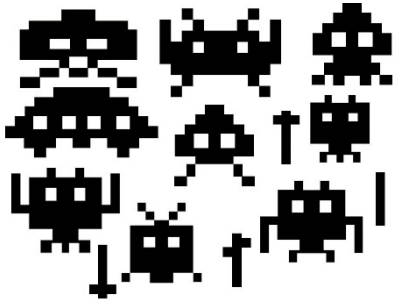
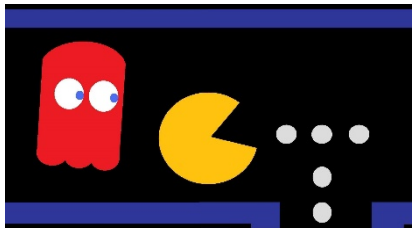
CSCI 1106
Lecture 2



Introduction to Game Architecture





Announcements

- Special lecture on **Team Working** this Friday!
  - Dr. Susan Holmes
  - **Attendance is highly recommended**
- Today's Topics
  - Introduction to the Game Design Module
  - Components of a game
  - Introduction to event driven programming
  - Introduction to Scratch

The Game Design Module

Topics

- Structure of a game
- Game mechanics
  - Collision Detection
  - Player movement
  - Autonomous Game Elements
  - Randomness
  - Controls
- Playability and play testing

To Do List

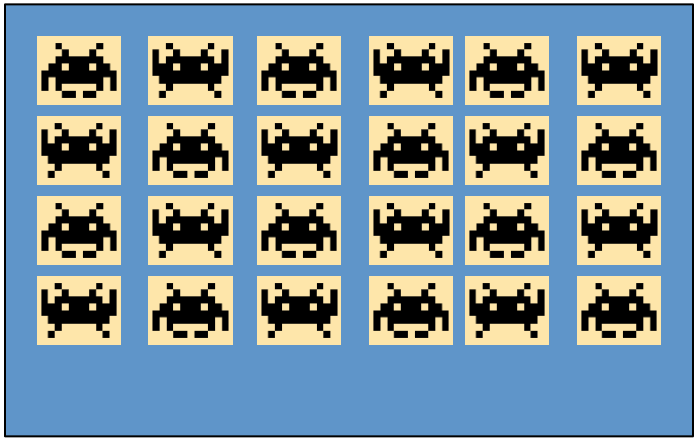
- Five tutorials:
  - Implement a game
  - Learn about game design
- One play-testing session
- Game Design Project
  - Design your own game
  - Implement the game
  - Write a technical manual
  - Write a user manual

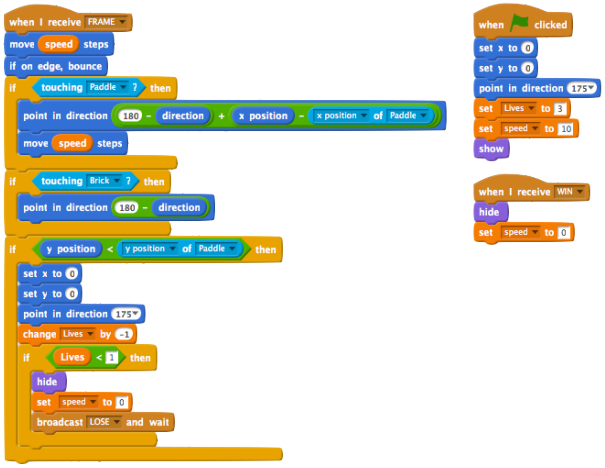
"All the World's a Stage" -As You Like It
William Shakespeare

- A theatrical play consists of:
  - A stage, where the action takes place
  - Actors, who move and recite based on a script
  - A script, which specifies the action and dialogue
- A movie consists of:
  - A screen, where the action takes place
  - Actors and Animations, who move and recite based on a screenplay
  - A screenplay, which specifies the action and dialogue

Components of a Game

- Stage: Displays (renders) the game
- Sprites:
  - Graphical objects that interact on the stage
  - Represent various artifacts in the game
    - Characters
    - Projectiles
    - Power-ups, obstacles, etc
- Game Code:
  - Governs interactions between sprites
  - Governs interactions between player and sprites
  - Implements the rules of the game
  - Contains *event handlers* that respond to events in the game
  - Updates the sprites on the stage





The Movie Metaphor

- In a movie the screen is updated 24 times per second
- In a game the stage is updated 30 times per second
- The update is called a *frame*
- A frame occurs every 1/30th of a second
- When a frame occurs
  - Sprites modify their properties
    - Position
    - Look
    - Sound
    - Etc
  - Sprites are redrawn on stage in each frame
- Key Idea: A game is simply an interactive movie!
- What interaction?

Our Event-based World

- Question: How do you know when to do something?
  - "something" happens because "something else" happened
- Examples:
  - We wake up *when* the alarm goes off
  - We respond *when* someone asks us a question
  - We duck *when* something is thrown at us
  - We cease talking *when* the lecture begins
- Answer: We *respond* to *events*
- Analogy: Actors *act* on *cues*

In the Game World

- A game performs "some action" when "something" happens
- Examples:
  - Character moves *when* the mouse is moved
  - An object explodes *when* it is hit by a laser
  - The stage is updated *after* 1/30th of a second
  - The stage is populated *when* the game starts up
- The "something" are called events

The Event-Driven Paradigm

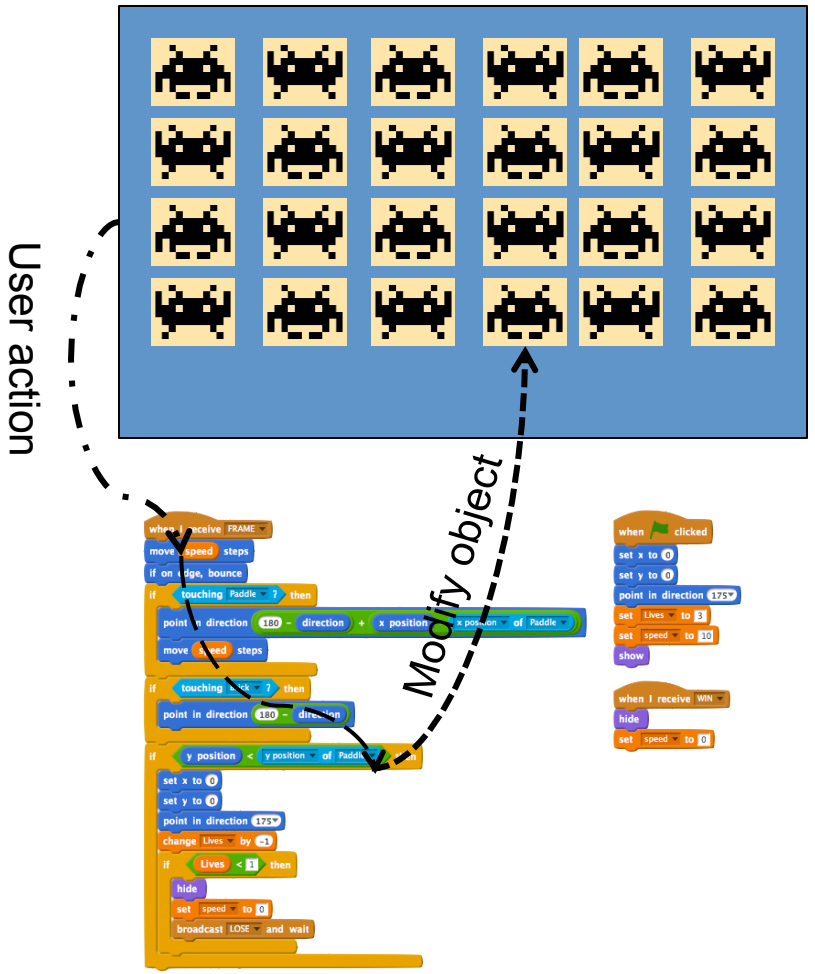
- Idea: Game code simply responds to events
- Possible events:
  - External events
    - Player movement (mouse, keyboard, kinect, etc)
  - Internal events
    - Start of game
    - Frame (stage update every 1/30th of a second)
    - Timer expired
    - Sprites cloned
- Each event is handled by an *event handler*
- The game code simply consists of event handlers that handle all aspects (behaviours) of the game!

The Main Loop

- Idea: The main loop is implemented for you

  GAME ENGINE

- Main Loop:
  - Event (action) occurs
  - Handle (respond to) event
  - Update (modify) object(s)



- All you need to do is
  - generate events and
  - write the event handlers!

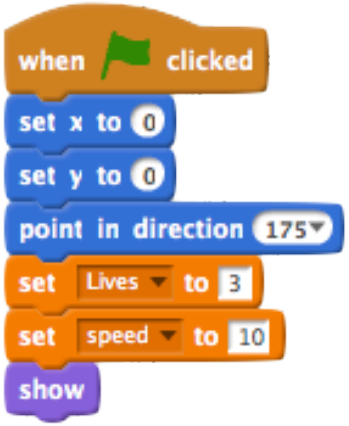
Events in Scratch

- When flag is clicked: game is started
- When I receive: programmer specified event
  - All programmer specified events such as "FRAME" fall into this category
- When key pressed: keyboard event
- When sprite clicked: mouse click event
- There are more!

Scratch in a Nutshell

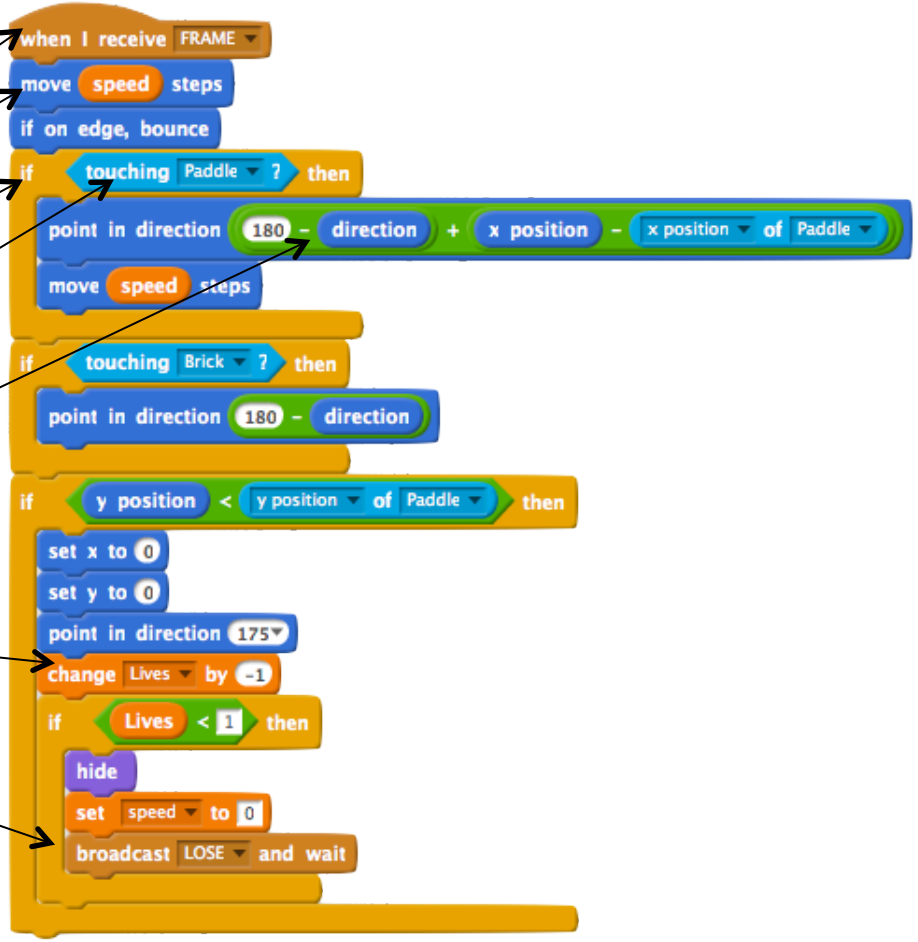
- A Scratch program consists of
  - A *stage* on which sprites are displayed
  - One or more *sprites*
    - *g*raphical objects that interact on the stage

  - Zero or more *scripts associated with the sprites*
- A *sprite* has
  - *Properties such as position, direction, size, etc.*
  - *Zero or more variables* used to store values
  - *One or more costumes*, describing how it looks
  - *Zero or more sounds* that it can emit
  - Zero or more *scripts* that respond to events
- A *script* responds to an event
  - These scripts are also called event handlers



A Scratch Script

- Is a sequence of blocks
- Starts on a *when* block
- Contains
  - *motion* blocks
  - *control* blocks
  - *sensing* blocks
  - *operator* blocks
  - *data* blocks
  - *event* blocks
- Is executed when an event occurs



Making Your Game Run

- Idea: Your game will need a FRAME event
  - 30 times per second
  - Allows sprites to update themselves
  - Generated by a script associated with the stage
  - Generated when game is running
- Use the following script
  - when game starts
  - repeat forever
    - wait 1/30th of a second
    - generate FRAME event