Lecture 8:

Engines and Content
Traditional Way to Break Up a Game

• Rules and Mechanics
• Game Engine
• User Interface
• Content
Traditional Way to Break Up a Game

- Rules and Mechanics
- **Game Engine**
- User Interface
- Content
Game Engine

• Component that powers the
  • graphics and sound
  • physics
  • artificial intelligence
  • game mechanics
  • interactions

• Game environment is
  • simulated by the engine
  • populated by the content
Game Engines: Systems

- Physics is an example of a game **system**
- Specifies the *space of possibilities* for a game
- But not the *specific parameters* of elements
Systems: *Super Mario Bros.*

- **Levels**
  - Fixed height scrolling maps
  - Populated by blocks and enemies

- **Enemies**
  - Affected by stomping or bumping
  - Different movement/AI schemes
  - Spawn projectiles or other enemies

- **Blocks**
  - Can be stepped on safely
  - Can be bumped from below

- **Mario (and Luigi) can be small, big, or fiery**
Systems: *Solitaire*
History of Engines

Score: 42 of 185

Engines and Content
Doom (1994)
Unreal (1998)
Forces

Engines and Content
Velocity
Collision Detection

Box 1

Box 2
Collision Detection

Box 1

Box 2
Collision Detection
Collision Detection

Engines and Content
Ways to do physics

- Do math
- Use an existing engine
Box2D – in Flash!

30 fps average
Physics step: 0 ms (3333 f
15 MB used

Box2DFlashAS3 2.0.1
'Left'/ 'Right' arrows to go to previous/next example.
'R' to reset.

Ragdolls
Box2D – in Flash!
FlixeL

SCORE: 500

Aww, you died!

Engines and Content
Flixel

Engines and Content
Flixel
public class MyGame extends FlxGame
{
    super(width, height, MyState);
}

Engines and Content
public class MyState extends FlxState {
    public override function create(): void {
        // create stuff
    }
    public override function update(): void {
        // update stuff
    }
    public override function destroy(): void {
        // destroy stuff
    }
}
FlxSprite

- Similar to Sprite but also includes
  - acceleration
  - velocity
  - maxVelocity
  - drag
  - scale
public override function create():void
{
    player = new FlxSprite(FlxG.width/2 - 5);
    player.makeGraphic(10,12,0xffaa1111);
    player.maxVelocity.x = 80;
    player.maxVelocity.y = 200;
    player.acceleration.y = 200;
    player.drag.x = player.maxVelocity.x*4;
    add(player);
}
public override function update(): void
{
    player.acceleration.x = 0;

    if(FlxG.keys.LEFT)
        player.acceleration.x = -player.maxVelocity.x*4;

    if(FlxG.keys.RIGHT)
        player.acceleration.x = player.maxVelocity.x*4;
}

Engines and Content
// size of game
FlxG.width
FlxG.height

// useful methods!
FlxG.overlap(coins, player, getCoin);
FlxG.overlap(exit, player, win);
FlxG.collide(level, player);
A note on animations
Blitting (sprite sheet)

http://www.adobe.com/devnet/flex/articles/actionscript_blitting.html
Game Engine

• Component that powers the
  • underlying game system
  • physics
  • artificial intelligence

• Game environment is
  • created by the engine
  • populated by the content
Traditional Way to Break Up a Game

- Rules and Mechanics
- Game Engine
- User Interface
- Content
Content

• **Everything else**
  • Levels
  • Art assets
  • Story messages
  • Sound effects
  • Music
  • Tutorial messages
Level Editor
Timeline

9/18 Now

10/7 1st Prototype

10/21 2nd Prototype

10/28 1st Release

Engine

Interface

Content

Logging

Engines and Content