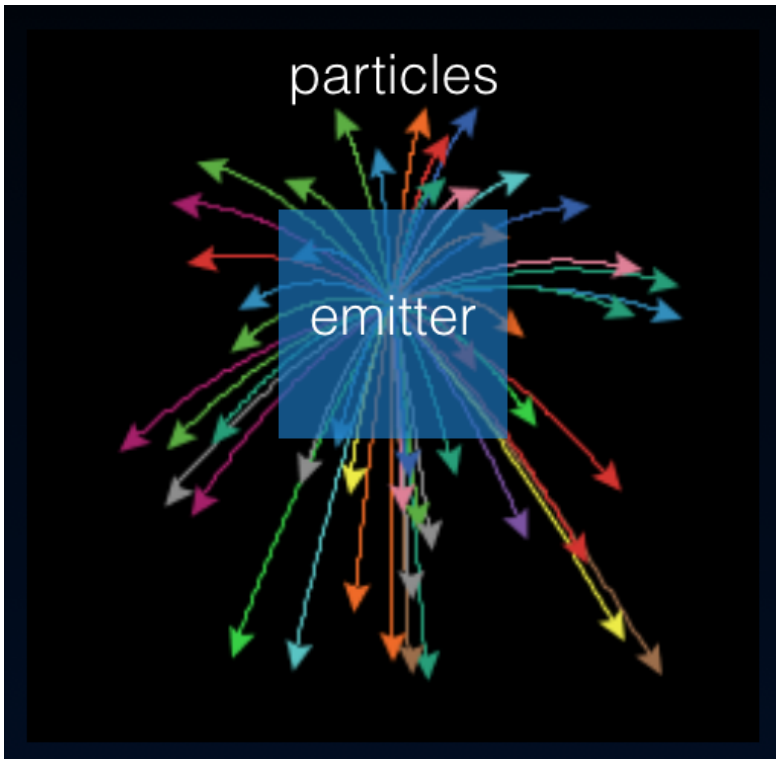
Week 11: Advanced Phaser Features

Working with Advanced Phaser Features

This week, we wrap up our Phaser investigation with some advanced features that the Phaser game engine offers us.

Particle Systems

Phaser supports particle systems with emitters and particles.



This topic is discussed in textbook section 6.20.

Signals

A signal is a form of event-driven programming. A signal calls a function when something specified happens, including mouse clicks, collisions and sound-related events.

```
this.countdownStarted = new Phaser.Signal();
this.countdownStarted.add(myFunction, this);
```

This topic is discussed in textbook sections 6.21-6.23.

Prefabs

“Too many moving parts!” is a common complaint from every game developer. Phaser Prefabs let you extend a Phaser class like a sprite to isolate code for specific game characters and gameplay.

```
export default class SpecialSprite
 extends Phaser.Sprite {
   constructor(game, x, y, frame) {
      super(game, x, y, ‘spriteKey’, frame);
      // init all properties, etc.
      this.speed = 2;
   }
   update() {
      this.x += this.speed;
   }
}
```

This topic is discussed in textbook section 6.24.

Exists Flag

The **exists** property of every Phaser sprite and other display object can be set to **false** to remove the element from any display or logic processing.

This topic is discussed in textbook section 6.25.

Object Pooling

Rendering many sprites slows down the system. Creating new ones and removing old ones is time and memory intensive. A strategy used to avoid slow game experiences is the creation of a fixed number of sprites in a pool to use as needed on screen and then temporarily retired to reuse later.

```
sprite.kill();
// halts rendering, but object still exists

sprite.reset();
// reactivates object so renders again
```

Animation

Keyframes and frame-based animation is traditional. Creature and Spine are 2D animation authoring apps.

Phaser also supports a time-based sprite expiration property: **lifespan**.

```
sprite.lifespan = 1000;
// sprite expires after 1000ms (1sec)
```

Topic Lecture Screencast

0:00

Class **(http://ethan.com/srjc/_videos/CS74-42A_Week_06_Part_1.m4v)** (.m4v)