Splitting up a Project

Working with objects

Topics

- What are objects?
- Objects in JavaScript
- Creating new objects
  - Using multiple files
  - Extending objects
- Working with objects
- Bonus Content – Source Control with GIT

What's wrong with what we've been doing?

- We've been using a style of programming called Procedural Programming
- This is using the act of programming using procedures (functions)
- OK for small programs, but hard to keep track of variables

What are Objects?

- We can group related variables and functions together into objects
- This is known as Object Oriented Programming
- Objects are usually the nouns in our game
  - A person, a place, or a thing

What are the objects in this image?

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

What are Objects?

- What were the objects in our Asteroids game?
  - Ship
  - Asteroid
  - Bullet
- These objects have their own properties that can be grouped together
  - Ship: speed, direction, rotation, position, radius
  - Asteroid: speed, direction, position, radius,
  - Bullet: speed, direction, position, radius,

Objects in JavaScript

- A collection of variables and functions is called an object
- Objects can be defined and created by the programmer

We've already used objects in our games

```
var player = document.createElement("img");
player.src = "ship.png";
```

- The player variable is an image object
- We can read and write to object properties using the 'dot' notation

Objects in JavaScript

- In JavaScript, we can create objects in a couple of different ways:
  - Create the object variable and define the properties at the same time
  - Create a function to return new objects

Creating new Objects

- Create the object variable and add properties to it at the same time

```
// this creates the player object and assigns it some properties
var player = {
   image: document.createElement("img"),
   x: SCREEN_WIDTH/2,
   y: SCREEN_HEIGHT/2,
   width: 93,
   height: 80,
   velocityX: 0,
   velocityY: 0,
   angularVelocity: 0,
   rotation: 0
};

player.image.src = "hero.png";
```

Creating new Objects

- Create a function to return new objects

```
var Player = function() {
   this.image = document.createElement("img");
   this.x = canvas.width/2;
   this.y = canvas.height/2;
   this.width = 159;
   this.height = 163;
   this.velocityX = 0;
   this.velocityY = 0;
   this.angularVelocity = 0;
   this.rotation = 0;

   this.image.src = "hero.png";
};

var player = new Player();
```

Using a Function to Create Objects

- This function is called a constructor
- Any time you read/write to a property of the object, you must use the this keyword
- When you want to create a new object using your constructor, you must use the new keyword
- Easy to create many objects using the same definition

```
var Player = function() {
   this.image = document.createElement("img");
   this.image.src = "hero.png";
};

var player = new Player();
```

Using Multiple Files

- We can define our objects in their own files
- Breaks up a program into manageable pieces
- Easily locate code for a specific object
  1. Create a new .js file
  2. Add your object definition to this file
  3. Add a new <script> statement in the HTML file

Using Multiple Files

index.html

```
<html>
  <body>
    <canvas id=“gameCanvas” width=“640” height=“480”>
    </canvas>
    <script src=“player.js”></script>
    <script src=“main.js”></script>
  </body>
</html>
```

main.js

```
var player = new Player();

function run()
{
  context.fillStyle = "#ccc";
  context.fillRect(0, 0, canvas.width, canvas.height);

  context.drawImage(player.image, player.x, player.y);
}
```

player.js

```
var Player = function()
{
 this.image = document.createElement("img");
 this.x = canvas.width/2;
 this.y = canvas.height/2;
 this.width = 159;
 this.height = 163;
 this.velocityX = 0;
 this.velocityY = 0;
 this.angularVelocity = 0;
 this.rotation = 0;

 this.image.src = "hero.png";
};
```

Extending Objects

- Add a new property to an object at any time
  - (but it's not good practice)
- Add a new method by modifying the object prototype

```
var Player = function() {
    this.image = document.createElement("img");
    this.x = canvas.width/2;
    this.y = canvas.height/2;
    this.width = 159;
    this.height = 163;

    this.image.src = "hero.png";
};

Player.prototype.update = function(deltaTime)
{
     if( typeof(this.rotation) == "undefined" )
         this.rotation = 0;                // hang on, where did this variable come from!
     this.rotation += deltaTime;
}

Player.prototype.draw = function()
{
     context.save();
        context.translate(this.x, this.y);
        context.rotate(this.rotation);
        context.drawImage(this.image, -this.width/2, -this.height/2);
     context.restore();
}
```

Working with Objects

- Create reusable objects
  - (x, y) can become a Vector2 object
- Use new when creating an object
- Always use this when defining the object properties/methods
  - Otherwise global variables are created
- Avoid cyclic dependencies
- Create each object in its own file
- List source files in the order used (in the HTML file)

Bonus Content – Source Control with GIT

- Now that your programs are getting larger, you'll want to make sure you have proper back-ups
- What is source control?
  - Manages changes to documents
  - Changes identified by an identifier (number)
  - Revisions marked with timestamp, name of author
  - Revisions can be compared, restored or merged

GIT

- Most widely adopted version control system for software development
- Free software
- Complete history and version tracking
- https://github.com/
  - Free to create an account
- Your platformer assignment must be submitted via GIT

GIT

- Signup for a free account at https://github.com
- Download and install the GitHub software
  - Windows and Mac versions available
- Create a repository (using GitHub software)
- Publish your repository
- Sync your code

GIT - Syncing

- Anything in your local repository folder can be sync'ed with the GIT server
  - Right-click on your repo, then select 'Open in Explorer'
  - Copy / move your project files to this directory
- New / modified files appear as uncommitted changes
- Enter a summary and description, press 'commit'
  - Changes are marked for upload, but not yet sent
- Press Sync button to upload to / download from server

GIT

- You can return to a previous version at any time
  - Select it in your history and press ‘Revert’
- More advanced users may wish to use SourceTree
  - https://www.atlassian.com/software/sourcetree/overview

Summary

- Group related variables and functions into Objects
- Objects are the nouns in our game
- Objects help break down our code into manageable pieces
- Create each object in its own file
- GIT is a great tool for making sure all versions of your project are backed up
  - Also useful when working as a team on the same project

Questions?

aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX