

Lesson 3.1 Extension Worksheet — Adding Baddies

Goal: Add a dangerous baddie with collision + consequences.

1. Block A — A Single Baddie with Collision and Result

Add this to your *index.html* (add inside `<div id="game-area">...</div>` Note that we can also refer to this *div* as `#game-area` which is what we use in *CSS*):

```
<!-- Baddie -->  

```

That puts the baddie in your game. You will need to save a *baddie.png* in your project folder — remember Lesson 4: transparent PNG!

Put this outside `#game-area`, next to your scoreboard:

```
<!-- Lives display -->  
<h2 id="lives-board">Lives: 3</h2>
```

CSS (add to *style.css*)

Put this in your *style.css* file:

```
#baddie {  
  position: absolute;  
  top: 50px;  
  left: 350px;  
  width: 40px;  
  height: 40px;  
}
```

JavaScript (add to script.js)

This is the more complex part. Work through these instructions carefully!

First, near the top of your JavaScript file, add these baddie variables:

```
// --- BADDIE VARIABLES (put near your other variables at the top) ---  
  
let baddieX = 350;  
let baddieY = 50;  
let lives = 3;  
let isGameOver = false;
```

Now find your *checkCollision* function. Identify the `}` that signals the end of the *checkCollision* block of code (**tip:** if you click on the opening `{` in VSCode or similar app, it will highlight the closing `}` for you). Put the following function after that final `}` of *checkCollision*:

```
// --- BADDIE COLLISION FUNCTION (put below checkCollision) ---  
  
function checkBaddieCollision() {  
    const baddie = document.getElementById('baddie');  
  
    // Measure the distance between player and baddie  
    const xDistance = Math.abs(playerX - baddieX);  
    const yDistance = Math.abs(playerY - baddieY);  
  
    // If they overlap...  
    if (xDistance < 35 && yDistance < 35) {  
        lives = lives - 1;  
        document.getElementById('lives-board').textContent = 'Lives: ' + lives;
```

```

// Teleport baddie to a random new spot
baddieX = Math.floor(Math.random() * 460);
baddieY = Math.floor(Math.random() * 260);
baddie.style.left = baddieX + 'px';
baddie.style.top = baddieY + 'px';

// Send the player back to the start
playerX = 100;
playerY = 100;

// Out of lives?
if (lives <= 0) {
    isGameOver = true;
    document.getElementById('game-status').textContent = "You LOSE! Press F5 to restart.";
    document.getElementById('player').style.display = 'none';
}
}
}

```

Now you need to call your new *checkBaddieCollision* function from your keydown listener. To do this, find *checkCollision()* in your code and add one line below it like this:

```

checkCollision();
checkBaddieCollision(); // <-- add this line

```

The Result

- Touching the baddie costs a life, teleports the baddie, and resets the player to start.
- At 0 lives the player disappears and a "You LOSE!" message appears.
- *isGameOver* is a flag (true/false) — you'll use it in Block C. You can also use it to stop scoring by adding this:

At the top of your *checkCollision()* function (after the opening { but before any more of the function's code), add:

```
if (isGameOver) return;
```

Block A challenges

1. Make the baddie chase the player: in the keydown listener, use:

```
if (baddieX < playerX) { baddieX += 2; } else { baddieX -= 2; }
```

Do the same for Y too.

2. Change the life cost to 1 point and a life lost so the player loses a point when they lose a life. Look again at how *score = score + 1;* works and add the minus version in the lose a life part of the code.
3. Add a "You WIN!" check in checkCollision: if (score >= 10) { ... } (see the Lesson 3 fast-finisher challenges).

Coming next... Spawning Multiple Baddies... Things are getting complex 😊