| **Content Type** | **Tutorial** |
| --- | --- |
| **Title** | **Lesson 5.1 - Clicky Mouse** |
| **Summary** | **Overview:**  It’s time for the final unit! We will start off by creating a new project and importing the starter files, then switching the game’s view to 2D. Next we will make a list of target objects for the player to click on: Three “good” objects and one “bad”. The targets will launch spinning into the air after spawning at a random position at the bottom of the map. Lastly, we will allow the player to destroy them with a click!  **Project Outcome:**  A list of three good target objects and one bad target object will spawn in a random position at the bottom of the screen, thrusting themselves into the air with random force and torque. These targets will be destroyed when the player clicks on them or they fall out of bounds. |
| **Overview Video** | [Lesson 5.1 - Clicky Mouse](https://youtu.be/wabxBLCTcxI) |
| **Cover Image** |  |
| **Unity Version** | 2018.4 - 2020.3 |
| **Difficulty Level** | Beginner |
| **Estimated Time** | 1h 00min |
| **Skills** | Use common logic structures to control the execution of code.  Write code that utilizes the various Unity APIs  Implement appropriate data types  Write code that integrates into an existing system  Implement a code style that is efficient and easy to read |
| **XP Categories and Values** | Programming - 10 xp |
| **Unity editor packages** | - |
| **Tags** | - |
| **Topics** | For Educators  Scripting  User Interface |
| **Industries** | - |
| **Video transcripts** | - |
| **Files and assets** |  |
| **Unity docs links** | - |
| **Asset store links** | - |
| **Learn group link** | - |
| **Hidden / Visible** | Hidden |

## Step 1: Create project and switch to 2D view

*One last time… we need to create a new project and download the starter files to get things up and running.*

**CWC\_U5.L1.S1\_T1\_v01-2020LTS -** [**https://youtu.be/cgGPd9qOPQc**](https://youtu.be/cgGPd9qOPQc)

***CwC 5.1.1 Create project and switch to 2D view***

1. Open **Unity Hub** and create an empty “Prototype 5” project in your course directory on the correct Unity version.   
   If you forget how to do this, refer to the instructions in [Lesson 1.1 - Step 1](https://learn.unity.com/tutorial/set-up-your-first-project-in-unity?uv=2018.4&courseId=5cf96c41edbc2a2ca6e8810f&projectId=5caccdfbedbc2a3cef0efe63#5cb7a1acedbc2a10b7261d15)
2. Click to download the [Prototype 5 Starter Files](https://connect-prd-cdn.unity.com/20210507/c7c2c2ce-f2f4-492e-819c-58096e11ab9f/Prototype%205%20-%20Starter%20Files.zip?_ga=2.33909089.1186801097.1620052249-59568313.1601905412), **extract** the compressed folder, and then **import** the .unitypackage into your project.   
   If you forget how to do this, refer to the instructions in [Lesson 1.1 - Step 2](https://learn.unity.com/tutorial/set-up-your-first-project-in-unity?uv=2018.4&courseId=5cf96c41edbc2a2ca6e8810f&projectId=5caccdfbedbc2a3cef0efe63#5cca0230edbc2a635ca5d6d2)
3. Open the **Prototype 5** scene, then delete the **sample scene** without saving
4. Click on the **2D icon** in Scene view to put Scene view in **2D.**
5. (optional) Change the texture and color of the **background** and the color of the **borders**

## Step 2: Create good and bad targets

*The first thing we need in our game are three good objects to collect, and one bad object to avoid. It’ll be up to you to decide what’s good and what’s bad.*

**CWC\_U5.L1.S2\_T1\_v01 -** [**https://youtu.be/DAGtUT9oIiY**](https://youtu.be/DAGtUT9oIiY)

***CwC 5.1.2 Create good and bad targets***

1. From the **Library**, drag 3 “good” objects and 1 “bad” object into the Scene, rename them “Good 1”, “Good 2”, “Good 3”, and “Bad 1”
2. Add **Rigid Body** and **Box Collider** components, then make sure that Colliders surround objects properly
3. Create a new Scripts folder, a new “Target.cs” script inside it, attach it to the **Target objects**
4. Drag all 4 targets into the **Prefabs folder** to create “original prefabs”, then **delete** them from the scene

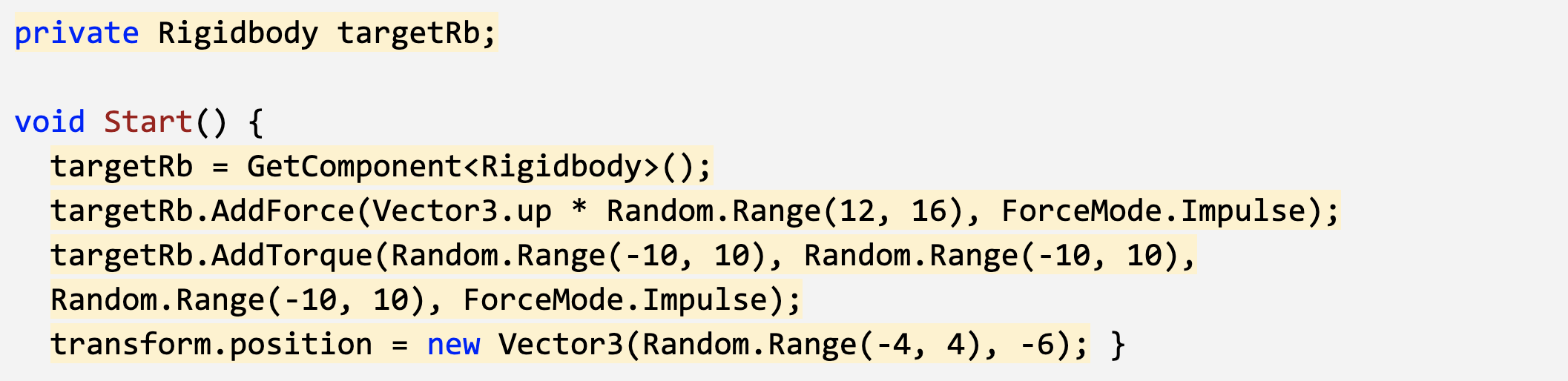
## Step 3: Toss objects randomly in the air

*Now that we have 4 target prefabs with the same script, we need to toss them into the air with a random force, torque, and position.*

**CWC\_U5.L1.S3\_T1\_v01 -** [**https://youtu.be/F79pNzl6cBA**](https://youtu.be/F79pNzl6cBA)

***CwC 5.1.3 Toss objects randomly in the air***

1. In **Target.cs**, declare a new ***private Rigidbody targetRb;*** and initialize it in **Start()**
2. In **Start()**, add an **upward force** multiplied by a **randomized speed**
3. Add a **torque** with randomized **xyz values**
4. Set the **position** with a randomized **X value**

****

**Alt text:** *-*

## Step 4: Replace messy code with new methods

*Instead of leaving the random force, torque, and position making our Start() function messy and unreadable, we’re going to store each of them in brand new clearly named custom methods.*

**CWC\_U5.L1.S4\_T1(2)\_v01 -** [**https://youtu.be/9k\_xFklXZ0U**](https://youtu.be/9k_xFklXZ0U)

***CwC 5.1.4 Replace messy code with new methods***

1. Declare and initialize new private float variables for ***minSpeed, maxSpeed, maxTorque,*** ***xRange,*** and ***ySpawnPos***;
2. Create a new function for ***Vector3 RandomForce()*** and call it in **Start()**
3. Create a new function for ***float RandomTorque()***, and call it in ***Start()***
4. Create a new function for ***RandomSpawnPos()***, have it return a new ***Vector3*** and call it in ***Start()***



**Alt text:** *-*

## Step 5: Create object list in Game Manager

*The next thing we should do is create a list for these objects to spawn from. Instead of making a Spawn Manager for these spawn functions, we’re going to make a Game Manager that will also control game states later on.*

**CWC\_U5.L1.S5\_T1-2020LTS -** [**https://youtu.be/4mSbW7f1J7U**](https://youtu.be/4mSbW7f1J7U)

***CwC 5.1.5 Create object list in Game Manager***

1. Create a new “Game Manager” **Empty object**, attach a new **GameManager.cs** script, then open it
2. Declare a new ***public List<GameObject> targets;***, then in the Game Manager inspector, change the list **Size** to 4 and assign your **prefabs**

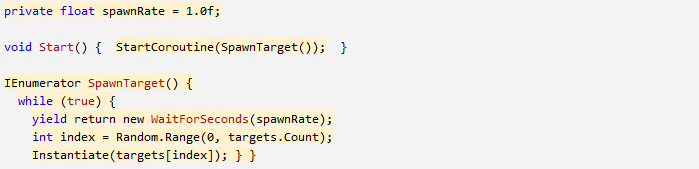
## Step 6: Create a coroutine to spawn objects

*Now that we have a list of object prefabs, we should instantiate them in the game using coroutines and a new type of loop.*

**CWC\_U5.L1.S6\_T1\_v01 -** [**https://youtu.be/TAKp3P\_-8vU**](https://youtu.be/TAKp3P_-8vU)

***CwC 5.1.6 Create a coroutine to spawn objects***

1. Declare and initialize a new ***private float spawnRate*** variable
2. Create a new ***IEnumerator SpawnTarget ()*** method
3. Inside the new method, ***while(true),*** wait **1 second**, generate a **random index**, and spawn a random **target**
4. In **Start()**, use the ***StartCoroutine*** method to begin spawning objects



**Alt text:** *-*

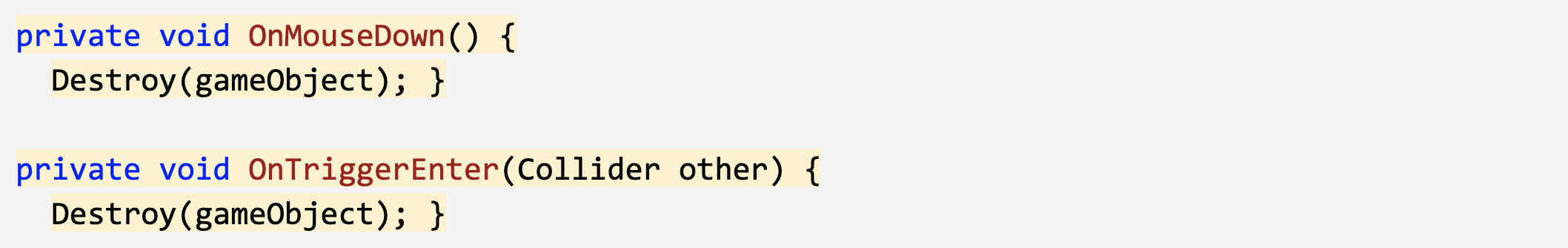
## Step 7: Destroy target with click and sensor

*Now that our targets are spawning and getting tossed into the air, we need a way for the player to destroy them with a click. We also need to destroy any targets that fall below the screen.*

**CWC\_U5.L1.S7\_T1\_v02 -** [**https://youtu.be/hHsOHAzmbwc**](https://youtu.be/hHsOHAzmbwc)

***CwC 5.1.7 Destroy target with click and sensor***

1. In **Target.cs**, add a new method for ***private void OnMouseDown() { }*** , and inside that method, destroy the gameObject
2. Add a new method for ***private void OnTriggerEnter(Collider other)*** and inside that function, destroy the gameObject



**Alt text:** *-*

## Step 8: Lesson Recap

**CWC\_U5.L1.SRecap\_T1\_v01 -** [**https://youtu.be/vYgwJLj8Y5I**](https://youtu.be/vYgwJLj8Y5I)

***CwC 5.1.8 Lesson Recap***

New Functionality**:**

* Random objects are tossed into the air on intervals
* Objects are given random speed, position, and torque
* If you click on an object, it is destroyed

New Concepts and Skills**:**

* 2D View
* AddTorque
* Game Manager
* Lists
* While Loops
* Mouse Events

Next Lesson**:**

* We’ll add some effects and keep track of score!