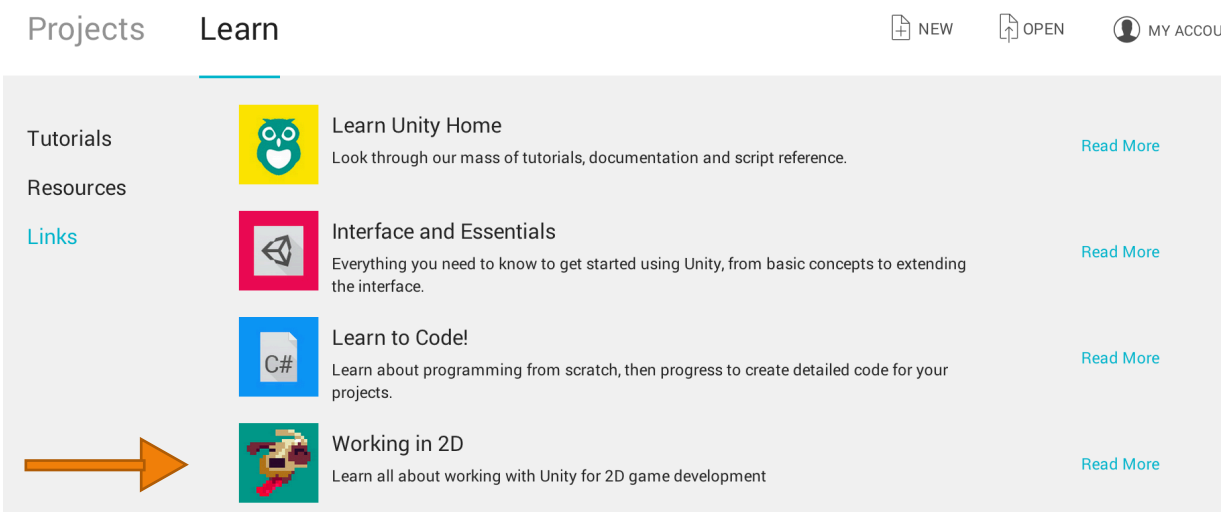
GAME PRODUCTION - Tutorial 2

1. **Unity – C# Coding** - Work through the code examples in Chapter 22 (Loops) and Chapter 23 (List and Arrays).

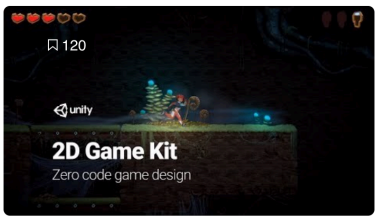
    a. This requires you to set up a project in Unity for each chapter and copy and test the code from the textbook. (Note you will also need to have been through Chapters 20, 21 to understand the required C# syntax used in the code)

    b. If you haven't got the textbook you can try and implement the code at the end of this tutorial – just set up an empty project (like Hello World) – the important thing to do is get familiar with how to use this type of code. You will also see lots of examples in the demo projects you will build

2. **2D Game Development** – Go to the "Working in 2D" page in "Unity – Learn".



You will find the following two resources. One is a video, the second is a 2D project – you can try them in either order.



Video: 2D Game Kit Walkthrough

Tutorial · Beginner · 45 Mins

This is a video version of the written 2D Game Kit Walkthrough. In this live session with guest host Aurore Dimopoulos, we will explore the 2D Game Kit project from Unity Technologies. 2D Game Kit is designed to allow new users to explore working in Unity and making games without first learning how to code in a friendly drag and drop environment. The kit consists of a

Unity Technologies

Part of: 2D Game Kit

2D Game Kit Walkthrough

Tutorial · Beginner · 1 Hour

This guide will walk you through setting up an empty Scene to start creating a new level with the Game Kit. This will walk you through some of the fundamentals used in this Kit to create gameplay. The Kit comes with a premade game, which contains examples of every part of the Kit in use if you get stuck for ideas. If you're already familiar with the Kit, and need some quick

Unity Technologies

Part of: 2D Game Kit

3. Working in 2D Topics
   Make yourself familiar with the concept of sprites and working with them (you will be using this concept in your 2D game). A good place to start is the Unity manual.
   https://docs.unity3d.com/Manual/Sprites.html

Sprites

Leave feedback

Sprites are 2D Graphic objects. If you are used to working in 3D, **Sprites** are essentially just standard textures but there are special techniques for combining and managing sprite textures for efficiency and convenience during development.

Unity provides a placeholder Sprite Creator, a built-in Sprite Editor, a Sprite Renderer and a Sprite Packer

See **Importing and Setting up Sprites** below for information on setting up assets as **Sprites** in your Unity project.

Sprite Tools

Sprite Creator

Use the Sprite Creator to create placeholder sprites in your project, so you can carry on with development without having to source or wait for graphics.

Sprite Editor

The Sprite Editor lets you extract sprite graphics from a larger image and edit a number of component images within a single texture in your image editor. You could use this, for example, to keep the arms, legs and body of a character as separate elements within one image.

Sprite Renderer

Code – Chapter 22 – Loops

Loops.cs – page 350

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Loops : MonoBehaviour {
5
6      // Use this for initialization
7      void Start () {
8
9          int i = 0;
10
11         while (i < 3) {
12             //print ("While Loop: " + i);
13             i++;
14
15         } //Start
16
17         for (int j=5; j>2; j--) {
18             //print("For Loop: " + j);
19         }
20
21         string str  = "Hello";
22         foreach (char chr in str) {
23                 print(chr);
24         }
25
26     }
27
28     // Update is called once per frame
29     void Update () {
30
31     } //Update
32 }
33
```

Code – Chapter 23 – Lists and Arrays

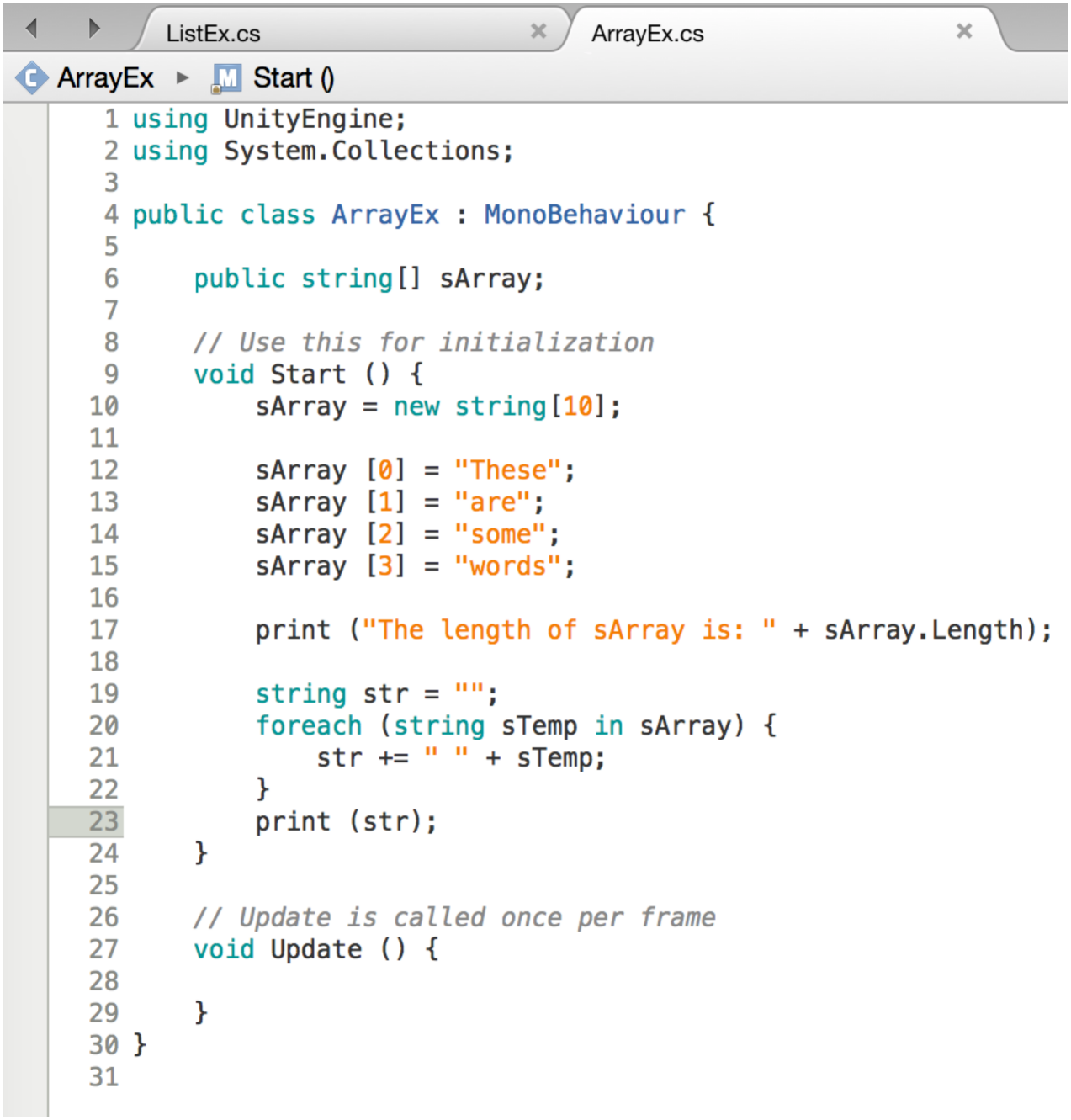
ListEx.cs – page 363

ListEx.cs ArrayEx.cs

No selection

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5  public class ListEx : MonoBehaviour {
6      public List<string> aList;
7
8      void Start () {
9          aList = new List<string> ();
10         aList.Add ("Experience");
11         aList.Add ("is");
12         aList.Add ("what");
13         aList.Add ("you");
14         aList.Add ("get");
15         aList.Add ("when");
16         aList.Add ("you");
17         aList.Add ("didn't");
18         aList.Add ("get");
19         aList.Add ("what");
20         aList.Add ("you");
21         aList.Add ("wanted");
22         print ("aList Count = " + aList.Count);
23         print ("The 0th element is : " + aList [0]);
24         print ("The 1st element is : " + aList [1]);
25         print ("The 3rd element is : " + aList [3]);
26         print ("The 8th element is : " + aList [8]);
27
28         string str = "";
29         foreach (string sTemp in aList) {
30             str += sTemp + " ";
31         }
32         print (str);
33
34     }
35
36     // Update is called once per frame
37     void Update () {
38
39     }
40 }
```

ArrayEx.cs page 371




```
1  using UnityEngine;
2  using System.Collections;
3
4  public class ArrayEx : MonoBehaviour {
5
6      public string[] sArray;
7
8      // Use this for initialization
9      void Start () {
10         sArray = new string[10];
11
12         sArray [0] = "These";
13         sArray [1] = "are";
14         sArray [2] = "some";
15         sArray [3] = "words";
16
17         print ("The length of sArray is: " + sArray.Length);
18
19         string str = "";
20         foreach (string sTemp in sArray) {
21             str += " " + sTemp;
22         }
23         print (str);
24     }
25
26     // Update is called once per frame
27     void Update () {
28
29     }
30 }
31
```

jaggedListTest.cs page 379

JaggedListTest.cs ListEx.cs ArrayEx.cs

No selection

```
1  using UnityEngine;
2  using System.Collections.Generic;
3
4  public class JaggedListTest : MonoBehaviour {
5
6      public List<List<string>> jaggedList;
7
8      // Use this for initialization
9      void Start () {
10         jaggedList = new List<List<string>>();
11
12         jaggedList.Add (new List<string> ());
13         jaggedList.Add (new List<string> ());
14
15         jaggedList [0].Add ("Hello");
16         jaggedList [0].Add ("World");
17         //jaggedList [1].Add ("Test");
18
19         jaggedList.Add (new List<string> (new string[] { "complex", "initialisation" }));
20
21         string str = "";
22         int i = 0;
23         foreach (List<string> sL in jaggedList) {
24             print ("i=" + i++);
25             foreach (string sTemp in sL) {
26                 print ("sTemp=" + sTemp);
27                 if (sTemp != null) {
28                     str += " | " + sTemp;
29                 } else {
30                     str += " | ";
31                 }
32             }
33             str += " | \n";
34         }
35         print (str);
36     }
37
38     // Update is called once per frame
39     void Update () {
40
41     }
42 }
43
```

CubeSpawner3.cs page 384

CubeSpawner3.cs HelloWorld.cs Cub

No selection

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5  public class CubeSpawner3 : MonoBehaviour {
6      public GameObject cubePrefabVar;
7      public List <GameObject> gameObjectList;
8      public float scalingFactor = 0.95f;
9      public int numCubes = 0;
10
11     // Use this for initialization
12     void Start () {
13         gameObjectList = new List<GameObject>();
14     }
15
16     // Update is called once per frame
17     void Update () {
18         numCubes++;
19         GameObject gObj = Instantiate (cubePrefabVar) as GameObject;
20         gObj.name = "Cube " + numCubes;
21         Color c = new Color (Random.value, Random.value, Random.value);
22
23         //note this next line replaces gObj.renderer.material.color = c;
24         gObj.GetComponent<Renderer> ().material.color = c;
25
26         gObj.transform.position = Random.insideUnitSphere;
27         gameObjectList.Add (gObj);
28
29         List<GameObject> removeList = new List<GameObject> ();
30
31         foreach (GameObject goTemp in gameObjectList) {
32             float scale = goTemp.transform.localScale.x;
33             scale *= scalingFactor;
34             goTemp.transform.localScale = Vector3.one * scale;
35             if (scale <= 0.1f) {
36                 removeList.Add (goTemp);
37             }
38         } //foreach
39
40         foreach (GameObject goTemp in removeList) {
41             gameObjectList.Remove (goTemp);
42             Destroy (goTemp);
43         } //foreach
44
45
46     } //Update
47 } //class
```