CHAPTER 24

DEBUGGING







Getting Started with DebuggingTypes of Bugs



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs
 - Bugs Attaching Scripts



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs
 - Bugs Attaching Scripts
 - Runtime Errors



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs
 - Bugs Attaching Scripts
 - Runtime Errors
- Stepping Through Code with the Debugger



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs
 - Bugs Attaching Scripts
 - Runtime Errors
- Stepping Through Code with the Debugger
 - Attaching the Debugger to Unity



- Getting Started with Debugging
- Types of Bugs
 - Compile-Time Bugs
 - Bugs Attaching Scripts
 - Runtime Errors
- Stepping Through Code with the Debugger
 - Attaching the Debugger to Unity
- Watching Variables in the Debugger





 Debugging is a way to step through and watch your code as it is running



- Debugging is a way to step through and watch your code as it is running
- This can help you



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies
- Debugging is built in to Unity via MonoDevelop



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies
- Debugging is built in to Unity via MonoDevelop
 - The MonoDevelop debugger can attach to the Unity process to debug your code



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies
- Debugging is built in to Unity via MonoDevelop
 - The MonoDevelop debugger can *attach* to the Unity process to debug your code
 - And, the MonoDevelop debugger can connect to an iOS or Android device and debug code running on the device!!!



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies
- Debugging is built in to Unity via MonoDevelop
 - The MonoDevelop debugger can *attach* to the Unity process to debug your code
 - And, the MonoDevelop debugger can connect to an iOS or Android device and debug code running on the device!!!
 - This is very helpful for finding issues with touch interfaces



- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies

Debugging is built in to Unity via MonoDevelop

- The MonoDevelop debugger can *attach* to the Unity process to debug your code
- And, the MonoDevelop debugger can connect to an iOS or Android device and debug code running on the device!!!
 - This is very helpful for finding issues with touch interfaces
 - Can be done over either a cable or WiFi!

- Debugging is a way to step through and watch your code as it is running
- This can help you
 - Better understand code
 - Find errors and bugs
 - Track down inefficiencies
- Debugging is built in to Unity via MonoDevelop
 - The MonoDevelop debugger can *attach* to the Unity process to debug your code
 - And, the MonoDevelop debugger can connect to an iOS or Android device and debug code running on the device!!!
 - This is very helpful for finding issues with touch interfaces
 - Can be done over either a cable or WiFi!
 - The book has detailed instructions for using the debugger





Compile-Time Bugs

A bug found in the syntax of your code



- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling



- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
 - Makes it unusable in Unity until the bug is resolved



- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
 - Makes it unusable in Unity until the bug is resolved
- Compile-time bugs usually cause very specific errors



- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
 - Makes it unusable in Unity until the bug is resolved
- Compile-time bugs usually cause very specific errors
 - The error below is on line 4, character 14 of CubeSpawner1.cs



- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
 - Makes it unusable in Unity until the bug is resolved
- Compile-time bugs usually cause very specific errors
 - The error below is on line 4, character 14 of CubeSpawner1.cs

🗄 Con	sole						* = .	
Clear	Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log	
As 🚺	sets/Cub	eSpawner1.c	s(4,14): erro	or CS0101: The namespace	`global::' alread	y contains a def	inition for `Cube	
	×.							
Assets/	CubeSpav	wner1.cs(4.14	1): error CS0)101: The namespace `glob	al::' already con	tains a definitior	n for	
`CubeS	pawner'			j	,,			
Log	in file: As	sets/CubeSp	awner1.cs a	it line: 4			1	
Asse	ets/CubeS	pawner1.cs(4,14): error (CS0101: The namespace `g	global::' already (contains a defini	ition for `CubeSp	awner'

- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
 - Makes it unusable in Unity until the bug is resolved
- Compile-time bugs usually cause very specific errors
 - The error below is on line 4, character 14 of CubeSpawner1.cs

E Console						*=	
Clear Colla	apse Clear on play	Error pause			Open Player Log	Open Editor Log	
Assets	/CubeSpawner1.c	s(4,14): error	CS0101: The namespace	`global::' alread	y contains a def	inition for `Cub	e
1.							
Assets /Cube	Spawner1.cs(4,14): error CS01	01: The namespace `glo	bal::' already con	tains a definition	n for	
Cubespawn	er						
🕕 Log in fil	e: Assets/CubeSp	awner1.cs at l	ine: 4			1	
Assets/C	ubeSpawner1.cs(4	4,14): error CS	\$0101: The namespace `	global::' already (contains a defin	ition for `CubeS	pawner'

🗄 Cor	isole						*=	
Clear	Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log	
1 As	sets/Cub	eSpawner1.c	s(4,14): erro	or CS0101: The name	space `global::' alread	y contains a def	inition for `Cube	
	ú.							
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `CubeSpawner'								
🕕 Log	in file: As	sets/CubeSp	awner1.cs a	t line: 4			! 1	
Asse	ets/CubeS	Spawner1.cs(4	4,14): error	CS0101: The namesp	ace "global::' already o	contains a defin	ition for `CubeS	pawner'



Console						
Clear Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log
Assets/Cub	eSpawner1.c	s(4,14): erroi	r CS0101: The name	espace `global::' alread	ly contains a def	inition for `Cube
sets/CubeSnav	wher 1 cs (4.14): error CS01	101. The names nace	e `alobal' already con	tains a definitio	a for
CubeSpawner'	witer 1.03(4,14	n. en or coo	101. The namespace	e global alleady con	tants a demition	
Log in file: As	sets/CubeSp	awner1.cs at	: line: 4			1
Assats /Cubas	nawner1.cs(1.14): error C	S0101 The names	nace `olobals' already	contains a defin	ition for `CubeSn

🗄 Console						*=
Clear Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log
Assets/Cub	eSpawner1.c	s(4,14): error	r CS0101: The name	space "global::' alre	ady contains a dei	finition for `Cub
ssets/CubeSpav CubeSpawner'	wner1.cs(4,14): error CS01	101: The namespace	e `global::' already co	ontains a definitio	n for
🕖 Log in file: As	sets/CubeSp	awner1.cs at	line: 4			1
Assets/Cubes	Spawner1.cs(4	4.14): error C	S0101: The namesr	bace `global::' alread	v contains a defin	ition for `CubeS

Click the error message to get more information



🗉 Con	sole						*≡	
Clear	Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `Cube								
	Ĵ.							
Assets/(`CubeSp	CubeSpav pawner'	wner1.cs(4,14	l): error CS0	101: The namespace	`global::' already cont	tains a definition	n for	
🕕 Log	in file: As	sets/CubeSp	awner1.cs a	t line: 4			! 1	
🚺 Asse	ts/CubeS	pawner1.cs(4	4,14): error	CS0101: The namesp	ace `global::' already o	contains a defin	ition for `CubeS	pawner

Click the error message to get more information



E Console		
Clear Collapse Clear on play Error pause Open Player Log C	Open Editor Log	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a define	nition for `Cube	
	_	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition `CubeSpawner'	for	
Use in file: Assets/CubeSpawner1.cs at line: 4	1	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition of the namespace `global::' al	ion for `CubeSpav	wner'

Click the error message to get more information


	-					
E Console						=
Clear Collapse	Clear on play	Error pause		Open Player Log	Open Editor Log	
Assets/Cub	eSpawner1.c	s(4,14): erro	or CS0101: The namespace `global::' already	y contains a def	inition for `Cເ	be
N.						
Assets/CubeSpa	wner1.cs(4,14	4): error CS0	101: The namespace `global::' already cont	ains a definitior	n for	
CubeSpawner'						
Log in file: As	sets/CubeSp	awner1.cs a	t line: 4		•	1

- Click the error message to get more information
- The full error text can usually tell you what's wrong



E Con	sole						*≡	
Clear	Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log	
l As	sets/Cub	eSpawner1.c	s(4,14): erro	or CS0101: The name	space `global::' alread	y contains a def	inition for `Cube	
	£.							
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `CubeSpawner'								
🕕 Log	in file: As	sets/CubeSp	awner1.cs a	t line: 4			1	
🕛 Asse	ets/CubeS	pawner1.cs(4	4,14): error	CS0101: The namesp	ace "global::' already o	contains a defin	ition for `CubeSp	awner'

- Click the error message to get more information
- The full error text can usually tell you what's wrong



E Console			
Clear Collapse Clear on play Error pause	Open Player Log	Open Editor Log	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' alread	y contains a def	inition for `Cube	
N			
Assets/CubeSpawner1.cs(4,14): erro CS0101 The namespace `global::' already con `CubeSpawner'	tains a definition	n for	
Log in file: Assets/CubeSpawner1.cs at line: 4		1	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already (contains a defin	ition for `CubeSpaw	vner'

- Click the error message to get more information
- The full error text can usually tell you what's wrong



E Cons	sole						
Clear	Collapse	Clear on play	Error pause		Open Player Log	Open Editor Log	
Ass Ass	sets/Cub	eSpawner1.c	s(4,14): erro	r CS0101: The namespace `global::' alread	y contains a def	finition for `Cube	
1	ē.						
Assets/C `CubeSp	CubeSpav Dawner'	vner1.cs(4,14	4): erro CSO	101 The namespace `global::' already con	tains a definitio	n for	
🕕 Log i	in file: As	sets/CubeSp	awner1.cs a	t line: 4		1	
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `CubeSpaw							

- Click the error message to get more information
- The full error text can usually tell you what's wrong
 - If not, search the Internet for the error number



E Console							
Clear Collapse Clear on play Error pause Open Player Log Open Editor Log							
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `Cube							
Assets/CubeSpawner1.cs(4,14): erro CS0101 The namespace `global::' already contains a definition for `CubeSpawner'							
Log in file: Assets/CubeSpawner1.cs at line: 4							
Assets/CubeSpawner1.cs(4,14): error CS0101: The namespace `global::' already contains a definition for `CubeSpaw							

- Click the error message to get more information
- The full error text can usually tell you what's wrong
 - If not, search the Internet for the error number
 - Example: "Unity error CS0101"



E Console						*=	
Clear Collapse	Clear on play	Error pause			Open Player Log	Open Editor Log	
Assets/Cub	eSpawner1.c	s(4,14): erro	r CS0101: The namespace	`global::' alread	y contains a def	inition for `Cube	
<i>n</i>							
Assets/CubeSpav	wner1.cs(4,14	4): erro CSO	101 The namespace `glob	al::' already con	tains a definition	n for	
'CubeSpawner'		_	_				
🕕 Log in file: As	sets/CubeSp	awner1.cs at	t line: 4			1	
Assets/Cubes	spawner1.cs(4,14): error (CS0101: The namespace `	global::' already	contains a defin	ition for `CubeSp	awner'

- Click the error message to get more information
- The full error text can usually tell you what's wrong
 - If not, search the Internet for the error number
 - Example: "Unity error CS0101"
 - Unity forums and StackOverflow.com have some of the best answers



error CS0101: The namespace 'global::' already contains a definition for '____'



error CS0101: The namespace 'global::' already contains a definition for '____'

Two scripts are trying to define the same class



error CS0101: The namespace 'global::' already contains a definition for '____'

- Two scripts are trying to define the same class
 - Change the name of the class in one of the scripts



error CS0101: The namespace 'global::' already contains a definition for '____'

Two scripts are trying to define the same class

Change the name of the class in one of the scripts

error CS1525: Unexpected symbol '}'



- error CS0101: The namespace 'global::' already contains a definition for '____'
 - Two scripts are trying to define the same class
 - Change the name of the class in one of the scripts
- error CS1525: Unexpected symbol '}'
 - Many "Unexpected symbol" errors are caused by a semicolon missing on a previous line or a misplaced brace



- error CS0101: The namespace 'global::' already contains a definition for '____'
 - Two scripts are trying to define the same class
 - Change the name of the class in one of the scripts
- error CS1525: Unexpected symbol '}'
 - Many "Unexpected symbol" errors are caused by a semicolon missing on a previous line or a misplaced brace
 - Check line endings for semicolons ;

- error CS0101: The namespace 'global::' already contains a definition for '____'
 - Two scripts are trying to define the same class
 - Change the name of the class in one of the scripts
- error CS1525: Unexpected symbol '}'
 - Many "Unexpected symbol" errors are caused by a semicolon missing on a previous line or a misplaced brace
 - Check line endings for semicolons ;
 - Check to make sure all braces have a mate { }







 Error occurs when attempting to attach a script to a GameObject





- Error occurs when attempting to attach a script to a GameObject
 - Caused by the name of the script not matching the name of the defined class





- Error occurs when attempting to attach a script to a GameObject
 - Caused by the name of the script not matching the name of the defined class
- Example





- Error occurs when attempting to attach a script to a GameObject
 - Caused by the name of the script not matching the name of the defined class

Example

- Script filename: CubeSpawner1 (or CubeSpawner1.cs)



- Error occurs when attempting to attach a script to a GameObject
 - Caused by the name of the script not matching the name of the defined class

Example

- Script filename: CubeSpawner1 (or CubeSpawner1.cs)
- Class name: public class CubeSpawner : MonoBehaviour { ... }



- Error occurs when attempting to attach a script to a GameObject
 - Caused by the name of the script not matching the name of the defined class

Example

- Script filename: CubeSpawner1 (or CubeSpawner1.cs)
- Class name: public class CubeSpawner : MonoBehaviour { ... }
- To Fix: Match the names to each other





Runtime Errors

A bug that occurs when your code is running



- A bug that occurs when your code is running
- Unity has no way of predicting these



- A bug that occurs when your code is running
- Unity has no way of predicting these
- Most common types of Runtime Errors



- A bug that occurs when your code is running
- Unity has no way of predicting these
- Most common types of Runtime Errors
 - UnassignedReferenceException



- A bug that occurs when your code is running
- Unity has no way of predicting these
- Most common types of Runtime Errors
 - UnassignedReferenceException
 - NullReferenceException



E Console	
Clear Collapse Clear on play Error pause Open Player Log Open Editor You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector. Open Player Log Open Editor	Log
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	\cap
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	•
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector. UnityEngine.Object.Instantiate (UnityEngine.Object original) CubeSpawner2.Update () (at Assets/CubeSpawner2.cs:14)	
Log in file: Assets/CubeSpawner2.cs at line: 14	970
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned.	

• UnassignedReferenceException

E Console	
Clear Collapse Clear on play Error pause Open Player Log Open Editor You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector. Open Player Log Open Editor	Log
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector.	Ţ
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned. You probably need to assign the cubePrefabVar variable of the CubeSpawner2 script in the inspector. UnityEngine.Object.Instantiate (UnityEngine.Object original) CubeSpawner2.Update () (at Assets/CubeSpawner2.cs:14)	
Log in file: Assets/CubeSpawner2.cs at line: 14	070
UnassignedReferenceException: The variable cubePrefabVar of 'CubeSpawner2' has not been assigned.	

UnassignedReferenceException

A variable in the Inspector has not been set



UnassignedReferenceException

A variable in the Inspector has not been set

Most commonly GameObject prefabs for Instantiate() calls



UnassignedReferenceException

A variable in the Inspector has not been set

Most commonly GameObject prefabs for Instantiate() calls

- To Fix: Assign the variable in the Inspector





Null Reference Exception



Null Reference Exception

Unity has been asked to access something that doesn't exist


Null Reference Exception

- Unity has been asked to access something that doesn't exist
- Example:



Null Reference Exception

- Unity has been asked to access something that doesn't exist
- Example:

7 void Start () {

- 8 GameObject[] goArray = new GameObject[10];
- 9 print (goArray[5].transform.position);
- 10 } // on line 9, goArray[5] is null, so it has no transform



Null Reference Exception

Unity has been asked to access something that doesn't exist

- Example:

```
7 void Start () {
```

```
GameObject[] goArray = new GameObject[10];
```

```
print (goArray[5].transform.position);
```

9 10

8



Null Reference Exception

Unity has been asked to access something that doesn't exist

- Example:

```
7 void Start () {
```

```
GameObject[] goArray = new GameObject[10];
```

```
print (goArray[5].transform.position);
```

} // on line 9, goArray[5] is null, so it has no transform





8

9

10

Null Reference Exception

Unity has been asked to access something that doesn't exist

- Example:

```
7 void Start () {
```

```
GameObject[] goArray = new GameObject[10];
```

```
print (goArray[5].transform.position);
```

10 3 //

8

9



Error can only tell you the line number

Null Reference Exception

Unity has been asked to access something that doesn't exist

- Example:

```
7 void Start () {
```

```
GameObject[] goArray = new GameObject[10];
```

- print (goArray[5].transform.position);
- 9 10

8





- Error can only tell you the line number
- These are difficult to debug!



Step 1: Set a Breakpoint in your code



Step 1: Set a Breakpoint in your code

000	4	ssembly-	CSharp - CubeSpawner2.cs - MonoDevelop-Unity	
Debug 💠 Default		¢	MonoDevelop-Unity Q Press	%.' to search
Solution • Debug Project	Ξ×	◀ → No select	CubeSpawner2.cs ×	¥ 着 Too
Assembly-CSharp Beferences CubeSpawner2.cs	0.	1 2 3 4	<pre>using UnityEngine; using System.Collections; public class CubeSpawner2 : MonoBehaviour {</pre>	ibox
		5 6 7 8 9	<pre>public GameObject cubePrefabVar; // Use this for initialization void Start () { }</pre>	Properties
Document Outline	Ξ×	10 11 12 13 14 15 16	<pre>// Update is called once per frame void Update () { SpellItOut(); Instantiate(cubePrefabVar); } void SpellItOut () {</pre>	Unit Tests
 CubeSpawner2 CubeSpawner2() void SpelltOut() void Start() void Update() GameObject cubePrefabVar 		18 19 20 21 22 23 24	<pre>string sA = "Hello World!"; string sB = ""; for (int i=0; i<sa.length; +="sA[i];" <="" i++)="" pre="" sb="" {="" }=""></sa.length;></pre>	
		25 26 27 28	<pre>print(SB); }</pre>	
				Errors 🖾 Tasks

Step 1: Set a Breakpoint in your code



Step 2: Attach the Debugger to the Unity process



Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)



Step 2: Attach the Debugger to the Unity process

- Much more detail in the book (about a potential bug)
- Click the Attach to Process button in MonoDevelop



Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)

Click the Attach to Process button in MonoDevelop

● ● ●	,	Assembly-C	Sharp – CubeSpawner2.cs – MonoDevelop-Unity		
Debug	♦ Default	\$	MonoDevelop-Unity	۹ Press '೫.' to search	
Solution	□×	< > /	CubeSpawner2.cs ×	~	-
🕶 🗖 Debua Proiect		No soloctio	n		1



Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)

Click the Attach to Process button in MonoDevelop

6 / 0		Assembly-C	Sharp - CubeSpawner2.cs - MonoDevelop-Unity		
Debug	♦ Default	\$	MonoDevelop-Unity	Q Press '%.' to search	
Solution		1× 4 > /	CubeSpawner2.cs ×	~	-
🕶 🗖 Debua Proiect		No soloctio	n		1



Step 2: Attach the Debugger to the Unity process

- Much more detail in the book (about a potential bug)
- Click the Attach to Process button in MonoDevelop

60		Assembly-0	Sharp - CubeSpawner2.cs - MonoDevelop-Unity		
Debug	♦ Default	\$	MonoDevelop-Unity	Q Press '%.' to search	
Solution		□× ∢ →	CubeSpawner2.cs ×	~	÷
🕶 🗖 Debua Proiect		No colocti	20		7



Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)

Click the Attach to Process button in MonoDevelop

60		Assembly-C	Sharp - CubeSpawner2.cs - MonoDevelop-Unity		
Debug	♦ Default	\$	MonoDevelop-Unity	۹ Press '೫.' to search	
Solution	□ ×	4 >	CubeSpawner2.cs ×	~	
🝷 🥅 Debua Proiect		No soloctio	n		1

00	Attach to Process
Attach to:	
PID Process N	ame
368 null (mon	0)
363 Unity Edit	or (Unity)
Debugger: Uni	ty Debugger
	Cancel Attach

Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)

Click the Attach to Process button in MonoDevelop

600		Assembly-0	Sharp - CubeSpawner2.cs - MonoDevelop-Unity		
Debug	♦ Default	* *	MonoDevelop-Unity	Q Press '%.' to search)
Solution		•× • •	CubeSpawner2.cs ×	~	-
🝷 🗖 Debua Proiect		No solocti	20		1

000	Attach to Process
Attach t	
PID PI	rocess Name
368 n	ull (mono)
363 U	nity Editor (Unity)
Debugg	er: Unity Debugger 🗘 Cancel Attach



Step 2: Attach the Debugger to the Unity process

Much more detail in the book (about a potential bug)

Click the Attach to Process button in MonoDevelop

600		Assembly-0	Sharp - CubeSpawner2.cs - MonoDevelop-Unity		
Debug	♦ Default	* *	MonoDevelop-Unity	Q Press '%.' to search)
Solution		•× • •	CubeSpawner2.cs ×	~	-
🝷 🗖 Debua Proiect		No solocti	20		1

PID Process Name		
368 null (mono)		
363 Unity Editor (Unity)		
Debugger: Unity Debugger	0	





Step 3: Click Play in Unity

The Debugger will halt code execution at the Breakpoint





Step 3: Click Play in Unity

– The Debugger will halt code execution at the Breakpoint



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from
	Over	Into	Out	Process
►	•	÷ +	1	



Step 3: Click Play in Unity

- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach fron
	Over	Into	Out	Process
•	. .	÷	1.1	

- Each controls the Debugger's execution



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from
	Over	Into	Out	Process
•	a.	÷	1	

- Each controls the Debugger's execution
 - Run Continues playing the project until another breakpoint is hit



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from	
	Over	Into	Out	Process	
	4	÷	1		

- Each controls the Debugger's execution
 - Run Continues playing the project until another breakpoint is hit
 - If Run doesn't advance to the next frame, switch back to Unity



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

R	Run	Step Over	Step Into	Step Out	Detach from Process
	▶ 1	4	÷	1	

- Each controls the Debugger's execution
 - Run Continues playing the project until another breakpoint is hit
 - If Run doesn't advance to the next frame, switch back to Unity
 - Step Over Continues to the next line, stepping over function calls



- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from
	Over	Into	Out	Process
•	A	÷	1	

- Each controls the Debugger's execution
 - Run Continues playing the project until another breakpoint is hit
 - If Run doesn't advance to the next frame, switch back to Unity
 - Step Over Continues to the next line, stepping over function calls
 - Step In Continues to the next line, stepping into function calls

Step 3: Click Play in Unity

- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from
	Over	Into	Out	Process
•	4	÷	1	

- Each controls the Debugger's execution

- Run Continues playing the project until another breakpoint is hit
 - If Run doesn't advance to the next frame, switch back to Unity
- Step Over Continues to the next line, stepping over function calls
- Step In Continues to the next line, stepping into function calls
- Step Out Exits the current function but continues debugging

Step 3: Click Play in Unity

- The Debugger will halt code execution at the Breakpoint
- Unity will be <u>completely frozen</u> while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window

Run	Step	Step	Step	Detach from		
	Over	Into	Out	Process		
•	4	÷	1			

- Each controls the Debugger's execution

- Run Continues playing the project until another breakpoint is hit
 - If Run doesn't advance to the next frame, switch back to Unity
- Step Over Continues to the next line, stepping over function calls
- Step In Continues to the next line, stepping into function calls
- Step Out Exits the current function but continues debugging
- Detach Process Stops debugging altogether

Watching Variables in the Debugger

Panes at the bottom of MonoDevelop have more info

👬 Watch 🔒 Loo	als 🗆 🛪 🛁 Threads 🛛 🎯 Breakp	oints	ĺ	Call Stack 🛛 🗙 📄 Imm	nediate
Name	Value	Туре		Name	File
▼ 🖸 this	{Main Camera (CubeSpawner2)}	CubeSpawner	\$	CubeSpawner2.Update ()	/Users/jrgibson/Dropbox/Book Chapter Drafts/
🕨 🖸 base	{UnityEngine.MonoBehaviour}	UnityEngine.			
🕨 📴 cubePrefabVar	{Cube (UnityEngine.GameObject)}	UnityEngine.(
	III			III	
					💽 Errors 🛛 😰 Task



Watching Variables in the Debugger

Panes at the bottom of MonoDevelop have more info

Locals - Allows you to see all local variables




Panes at the bottom of MonoDevelop have more info

Locals - Allows you to see all local variables

• this is a reference to the current class instance



Panes at the bottom of MonoDevelop have more info

- Locals - Allows you to see all local variables

- this is a reference to the current class instance
- Watch Allows you to enter specific variables to watch

📸 Watch 🔒 Lo	cals 🗆 🗙 < Threads 🛛 🎯 Breakp	oints	É	Call Stack 🛛 🗙 📄 Ir	nmediate
Name	Value	Туре		Name	File
▼ 🥃 this	{Main Camera (CubeSpawner2)}	CubeSpawner	\$	CubeSpawner2.Update	e () /Users/jrgibson/Dropbox/Book Chapter Drafts/1
Isotation Isotation	{UnityEngine.MonoBehaviour}	UnityEngine.			
🕨 📔 cubePrefabVar	{Cube (UnityEngine.GameObject)}	UnityEngine.			
	III			III	
					🚺 Errors 🛛 🖾 Tasks



Panes at the bottom of MonoDevelop have more info

- Locals Allows you to see all local variables
 - this is a reference to the current class instance
- Watch Allows you to enter specific variables to watch
- Call Stack Shows you which functions have been called to get to this point in the code

Match	ᡖ Locals 🗆 🗶 🐗 Threads 🛛 🎯 Breakpoi			oints	Call Stack 🖙 🔲 Immediate							
Name		Value			Туре	Name		File				
🝷 💽 this		{Main C	amera (CubeSpawr	ner2)}	CubeSpawner	\$ CubeSpawner2.L	Jpdate ()	/Users/j	rgibson/Dro	pbox/Book	Chapte	r Drafts/1
🕨 🖸 base		{UnityEr	ngine.MonoBehavio	our}	UnityEngine.M							
• 🖪 cubePrefal	bVar	{Cube (I	UnityEngine.Game(Dbject)}	UnityEngine.C							
			III			111						
										🛃 E	rrors	🖾 Tasks

Panes at the bottom of MonoDevelop have more info

- Locals Allows you to see all local variables
 - this is a reference to the current class instance
- Watch Allows you to enter specific variables to watch
- Call Stack Shows you which functions have been called to get to this point in the code
 - Click a function to jump to it's local scope

👬 Watch 🚽 Locals 🗆 x 🚳 Threads 💿 Breakpoi			oints	ĺ	Call Stack 🗆 🗙	📃 Imm	rediate						
Name	ſ	Value			Туре		Name		File				
👻 💽 this		{Main C	amera (CubeSpaw	ner2)}	CubeSpawner	\$	CubeSpawner2.L	Jpdate ()	/Users/j	rgibson/Drop	box/Book C	hapter D)rafts/1
🕨 💽 base	+	{UnityEr	ngine.MonoBehavio	our}	UnityEngine.M								
• 🖪 cubePrefat	bVar	{Cube (I	UnityEngine.Game	Object)}	UnityEngine.C								
			III										
						_					🔂 Erro	ors 💈] Tasks



 Debugging is one of the most important processes in coding



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - · Use it on the code from the book if you're ever confused



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger
 - Lots of information online about how to set this up



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger
 - Lots of information online about how to set this up
- Next Chapter: Classes



- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger
 - Lots of information online about how to set this up
- Next Chapter: Classes
 - Learn about how classes combine data and functionality

- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger
 - Lots of information online about how to set this up
- Next Chapter: Classes
 - Learn about how classes combine data and functionality
 - All the code you write in Unity C# will be in classes

- Debugging is one of the most important processes in coding
 - The MonoDevelop Debugger is one of the most powerful tools for you to learn
 - It's also surprisingly easy to learn to use
 - The Debugger can also help you understand complex code
 - Use it on the code from the book if you're ever confused
- You can also code for Unity using Microsoft Visual Studio, which has its own debugger
 - Lots of information online about how to set this up
- Next Chapter: Classes
 - Learn about how classes combine data and functionality
 - All the code you write in Unity C# will be in classes
 - Classes are also the key to Object-Oriented Programming