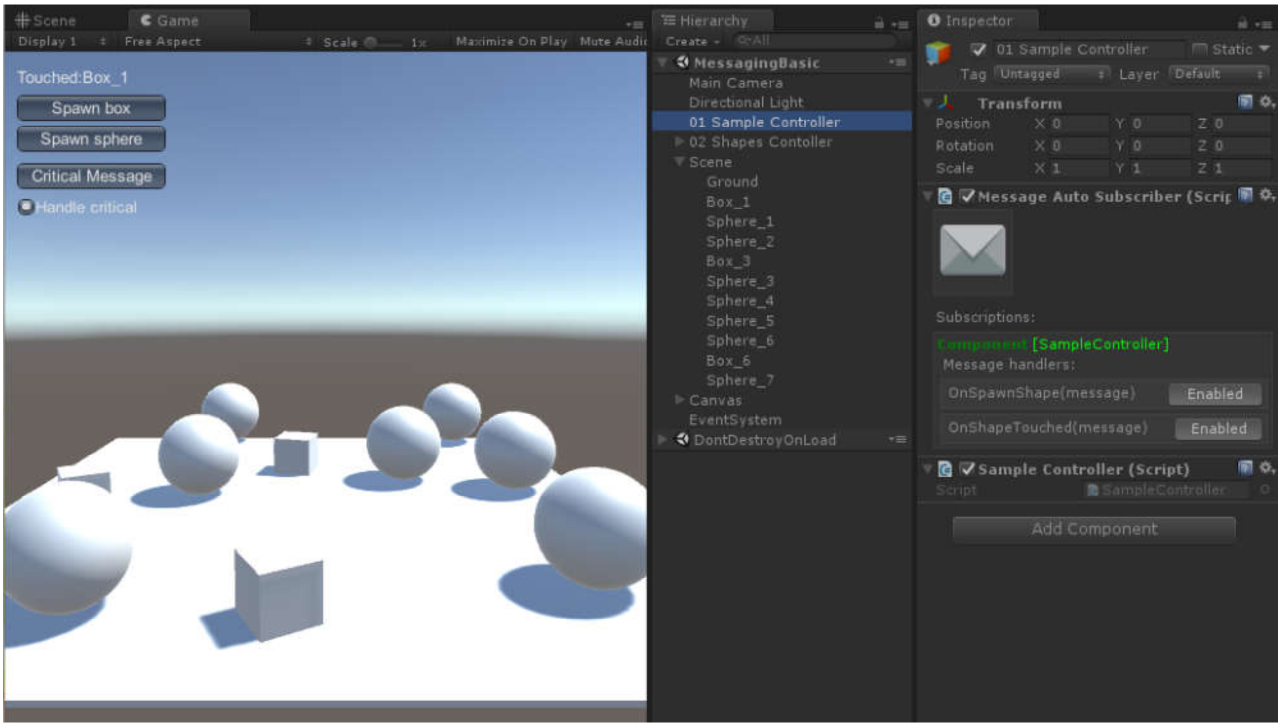
Messaging sample

Sample description.

Sample demonstrate how components can interact each other using Messaging subsystem.



Messaging sample

The sample files are located within *Vdev/Modules/Messaging/samples* directory

Sample file	Description	Messaging details
MessagingBasic.unity	Scene of the sample	Scene
SampleController.cs	Draw and handle UI, handle screen touch.	Broadcast **SpawnShape** and **ShapeTouched** messages.
ShapesController.cs	Create instances of the shapes.	Handle **SpawnShape** message and instantiate shape objects.
SampleShape.cs	Implement with shape interaction.	Handle **ShapeTouched** message.

Instantiating the shape.

When pressing "Spawn box" or "Spawn sphere" button, *SampleController* component broadcasts the message **SpawnShape** that are handled by *ShapesController* component.

Touching the shape.

When user "touch" the screen (in play mode) *SampleController* component determine which shape was touched and broadcasts the message **ShapeTouched** if something. **ShapeTouched** are handled by *SampleShape* components.

Critical Message.

There is special **CriticalMessage** within the sample.

```
[MessageDescription(HandlerRequirement.Required)]
public sealed class CriticalMessage : Message<CriticalMessage>
{}
```

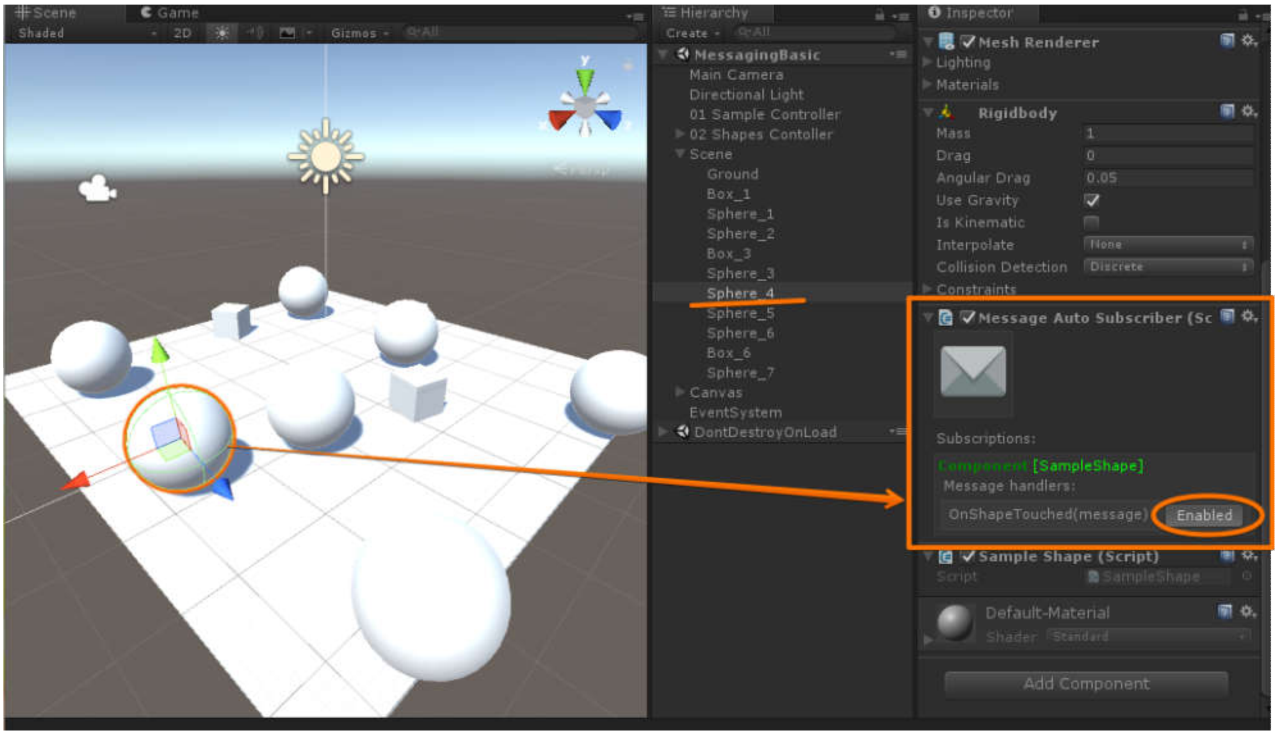
It annotated by **MessageDescription** attribute and specify **handler requirement** as 'Required'. So it must be handled or exception will be throwed. 'Handle critical' checkbox control handler availability.

Messages tools usage.

While sample is started you can 'play' with **development time tools**.

MessageAutosubscriber Component

Select any GameObject that has MessageAutosubscriber assigned onto it and see its inpector.



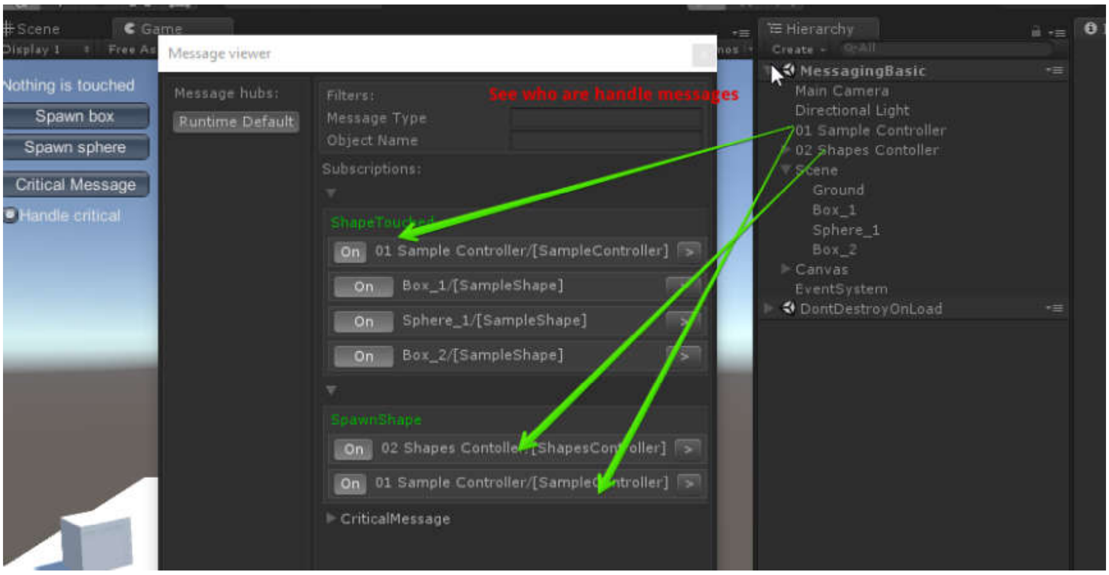
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ☑
Is Kinematic □
Interpolate None
Collision Detection Discrete
Script SampleShape
Shader Standard

MessageAutosubscriber

Toggle 'Enable' option for message that subscribed for components of this GameObject and see how changed sample behaviour.

Messages viewer

Open the **messages viewer tool** from Window -> **Vdev** -> Message viewer menu. Here you can see the messages for which there is a subscription at the moment.



ShapeTouched
On 01 Sample Controller/[SampleController]
On Box_1/[SampleShape]
On Sphere_1/[SampleShape]
On Box_2/[SampleShape]

SpawnShape
On 02 Shapes Contoller/[ShapesContoller]
On 01 Sample Controller/[SampleController]

CriticalMessage

Messages Viewer