DIVISION OF
DIGITAL CONTENTS
DONGSEO UNIVERSITY

Unity Physics

Add Impulse to Rigidbody

jintaeks@dongseo.ac.kr

May 10, 2019

- ✓ **Linear momentum**, **translational momentum**, or simply **mo mentum**(pl. momenta) is the product of the mass and velocit y of an object.
- ✓ It is a vector quantity, possessing a magnitude and a direction in three-dimensional space.
- ✓ If *m* is an object's mass and **v** is the velocity (also a vector), th en the momentum is

$$p = mv$$

Many particles

- ✓ The momentum of a system of particles is the vector sum of t heir momenta. If two particles have respective masses *m_1* an d *m_2*, and velocities *v_1* and *v_2*, the total momentum is

$$\begin{aligned} p &= p_1 + p_2 \\ &= m_1 v_1 + m_2 v_2 \,. \end{aligned}$$

- ✓ The momenta of more than two particles can be added more generally with the following:

$$p = \sum_i m_i v_i \,.$$

- ✓ If the net force *F* applied to a particle is constant, and is appli ed for a time interval Δ*t*, the momentum of the particle chang es by an amount

$$\Delta p = F \Delta t \,.$$

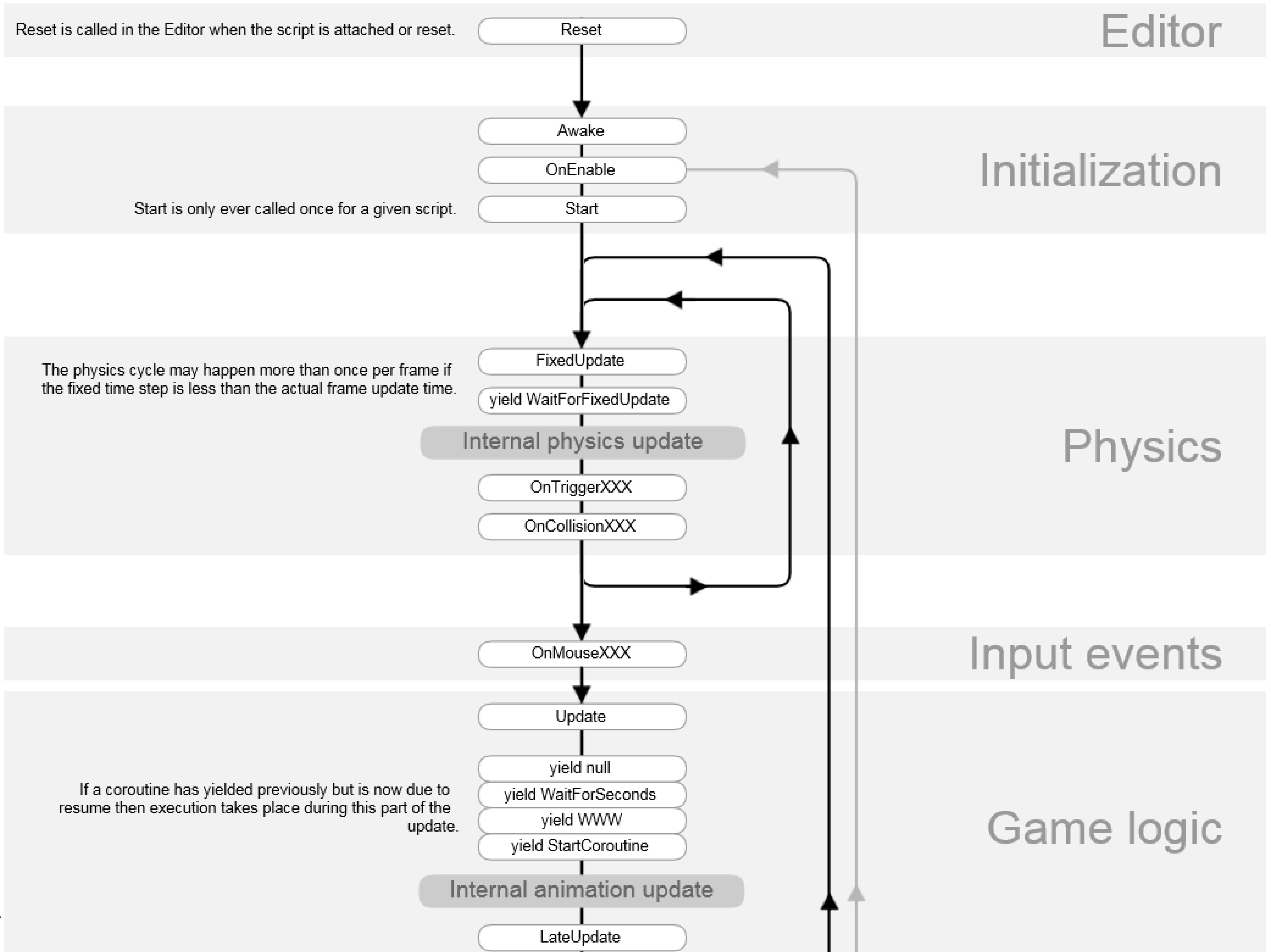
- ✓ In differential form, this is Newton's second law; the rate of ch ange of the momentum of a particle is equal to the instantan eous force *F* acting on it,

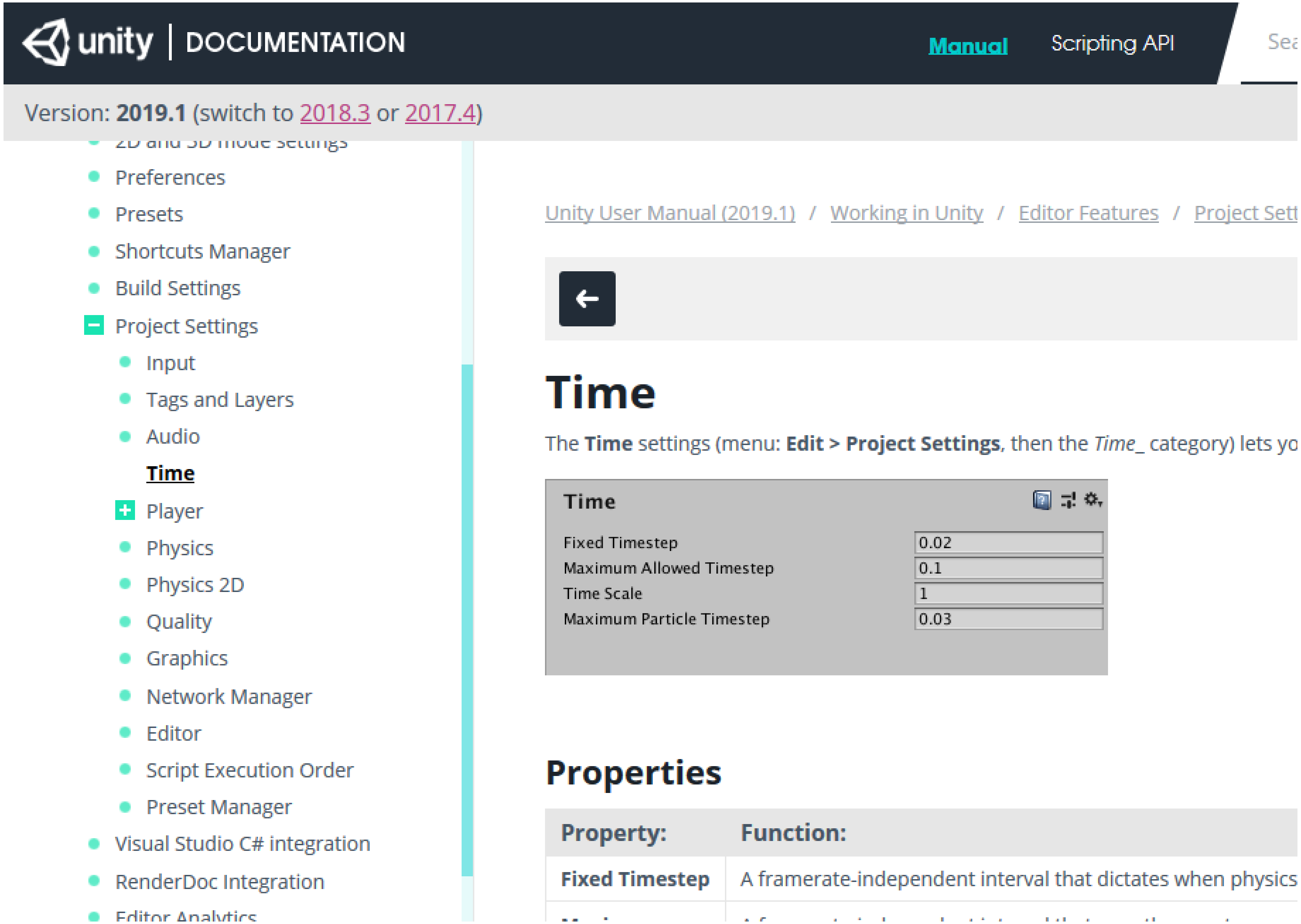
$$F = \frac{dp}{dt}.$$

- ✓ If the net force experienced by a particle changes as a functio n of time, *F(t)*, the change in momentum (or impulse *J*) betwe en times t_1 and t_2 is

$$\Delta p = J = \int_{t_1}^{t_2} F(t)\, dt\,.$$

Unity Demo


Editor
Reset is called in the Editor when the script is attached or reset.
Reset
Initialization
Awake
OnEnable
Start is only ever called once for a given script.
Start
Physics
The physics cycle may happen more than once per frame if the fixed time step is less than the actual frame update time.
FixedUpdate
yield WaitForFixedUpdate
Internal physics update
OnTriggerXXX
OnCollisionXXX
Input events
OnMouseXXX
Game logic
Update
If a coroutine has yielded previously but is now due to resume then execution takes place during this part of the update.
yield null
yield WaitForSeconds
yield WWW
yield StartCoroutine
Internal animation update
LateUpdate



unity | DOCUMENTATION   Manual   Scripting API   Sea

Version: **2019.1** (switch to 2018.3 or 2017.4)

- 2D and 3D mode settings
- Preferences
- Presets
- Shortcuts Manager
- Build Settings
- Project Settings
  - Input
  - Tags and Layers
  - Audio
  - Time
  - Player
  - Physics
  - Physics 2D
  - Quality
  - Graphics
  - Network Manager
  - Editor
  - Script Execution Order
  - Preset Manager
- Visual Studio C# integration
- RenderDoc Integration
- Editor Analytics

Unity User Manual (2019.1) / Working in Unity / Editor Features / Project Sett

Time

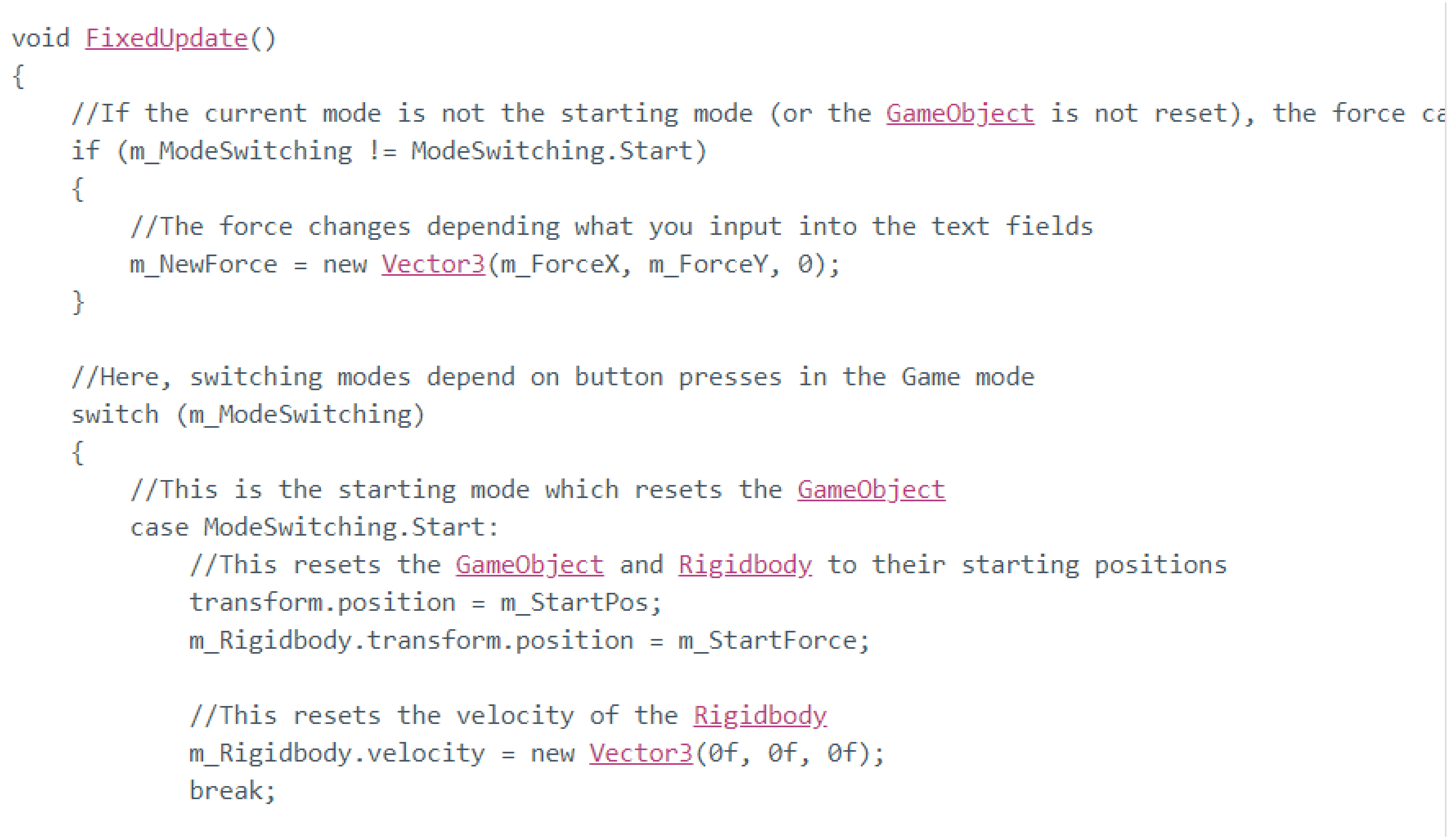
The **Time** settings (menu: **Edit > Project Settings**, then the *Time*_ category) lets yo

Time

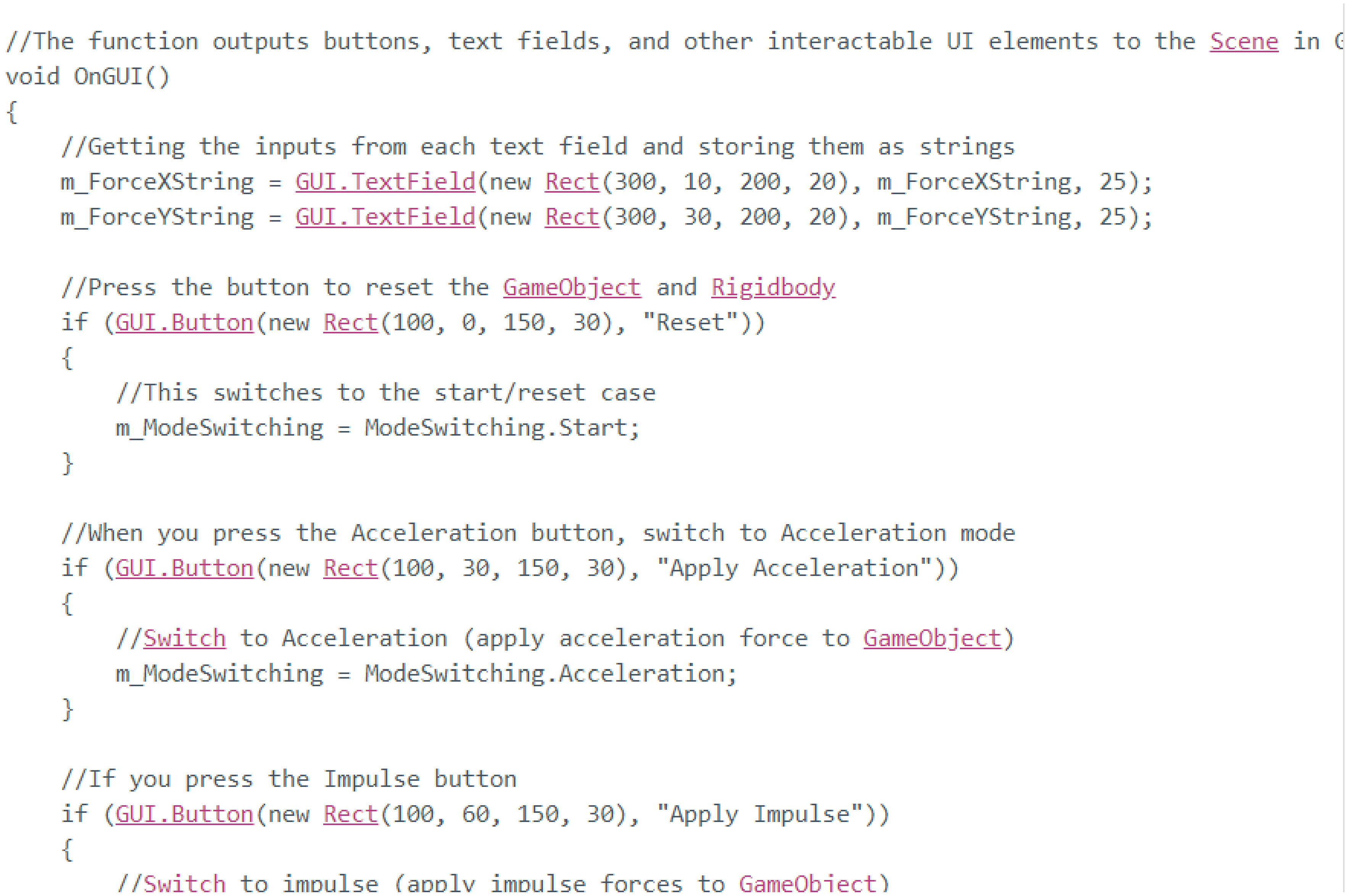
Fixed Timestep	0.02
Maximum Allowed Timestep	0.1
Time Scale	1
Maximum Particle Timestep	0.03

Properties

Property:	Function:
Fixed Timestep	A framerate-independent interval that dictates when physics

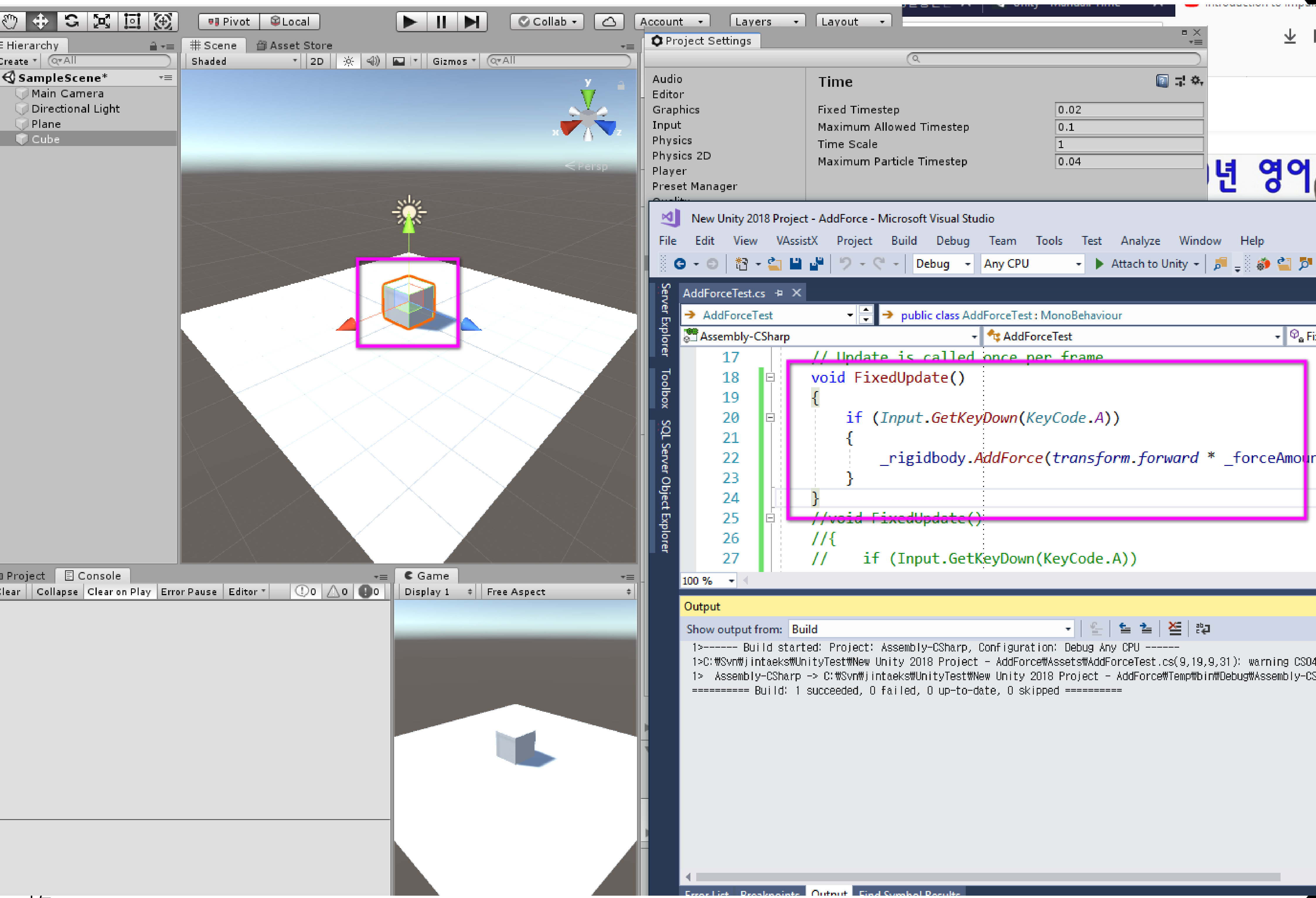


```
void FixedUpdate()
{
    //If the current mode is not the starting mode (or the GameObject is not reset), the force ca
    if (m_ModeSwitching != ModeSwitching.Start)
    {
        //The force changes depending what you input into the text fields
        m_NewForce = new Vector3(m_ForceX, m_ForceY, 0);
    }

    //Here, switching modes depend on button presses in the Game mode
    switch (m_ModeSwitching)
    {
        //This is the starting mode which resets the GameObject
        case ModeSwitching.Start:
            //This resets the GameObject and Rigidbody to their starting positions
            transform.position = m_StartPos;
            m_Rigidbody.transform.position = m_StartForce;

            //This resets the velocity of the Rigidbody
            m_Rigidbody.velocity = new Vector3(0f, 0f, 0f);
            break;
```



```
//The function outputs buttons, text fields, and other interactable UI elements to the Scene in
void OnGUI()
{
    //Getting the inputs from each text field and storing them as strings
    m_ForceXString = GUI.TextField(new Rect(300, 10, 200, 20), m_ForceXString, 25);
    m_ForceYString = GUI.TextField(new Rect(300, 30, 200, 20), m_ForceYString, 25);

    //Press the button to reset the GameObject and Rigidbody
    if (GUI.Button(new Rect(100, 0, 150, 30), "Reset"))
    {
        //This switches to the start/reset case
        m_ModeSwitching = ModeSwitching.Start;
    }

    //When you press the Acceleration button, switch to Acceleration mode
    if (GUI.Button(new Rect(100, 30, 150, 30), "Apply Acceleration"))
    {
        //Switch to Acceleration (apply acceleration force to GameObject)
        m_ModeSwitching = ModeSwitching.Acceleration;
    }

    //If you press the Impulse button
    if (GUI.Button(new Rect(100, 60, 150, 30), "Apply Impulse"))
    {
        //Switch to impulse (apply impulse forces to GameObject)
```

Demo



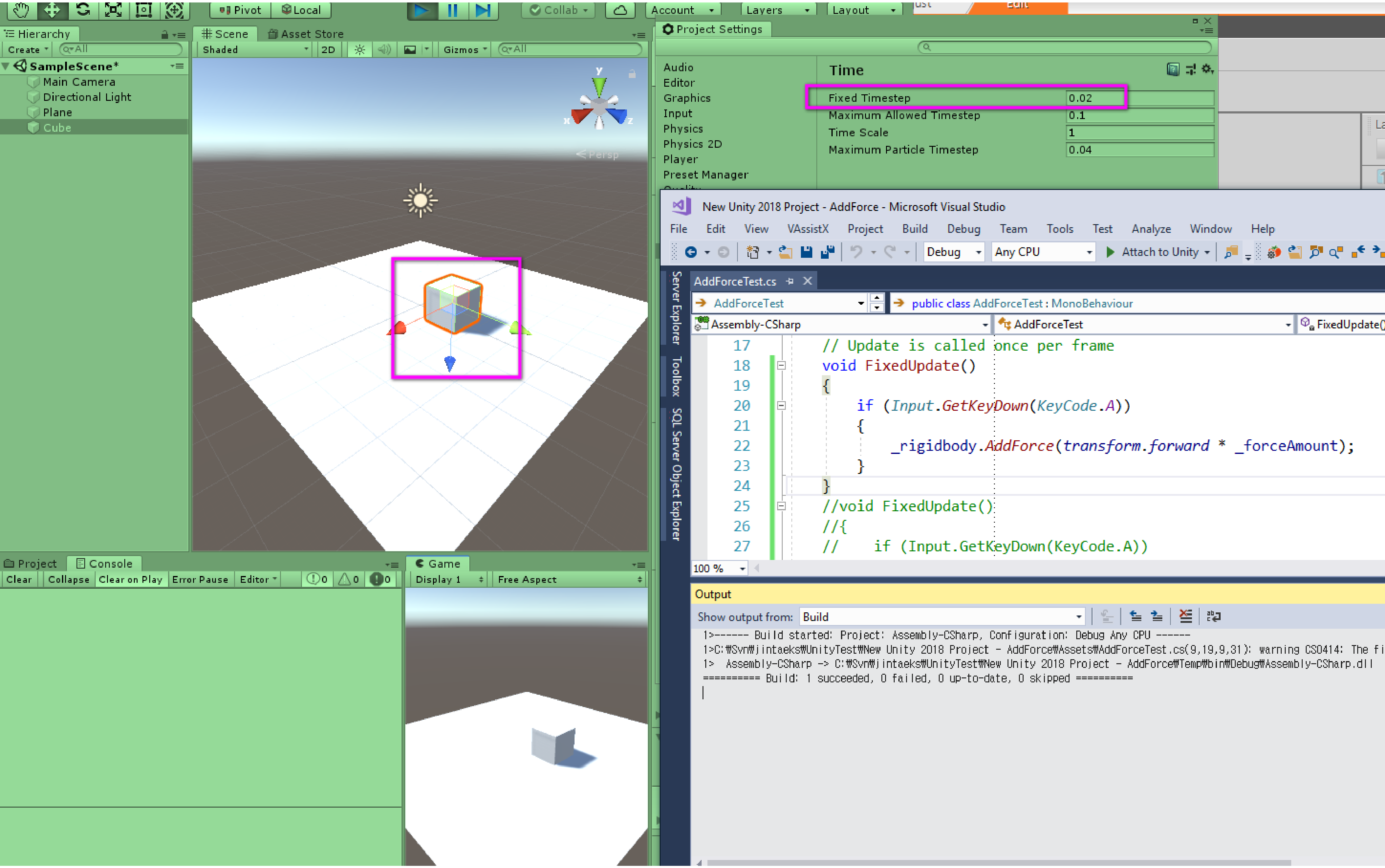
Time

Fixed Timestep	0.02
Maximum Allowed Timestep	0.1
Time Scale	1
Maximum Particle Timestep	0.04

```
    // Update is called once per frame
    void FixedUpdate()
    {
        if (Input.GetKeyDown(KeyCode.A))
        {
            _rigidbody.AddForce(transform.forward * _forceAmou
        }
    }
    //void FixedUpdate()
    //{
    //    if (Input.GetKeyDown(KeyCode.A))
```

Show output from: Build

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Assets\AddForceTest.cs(9,19,9,31): warning CS0
1>  Assembly-CSharp -> C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Temp\bin\Debug\Assembly-CS
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



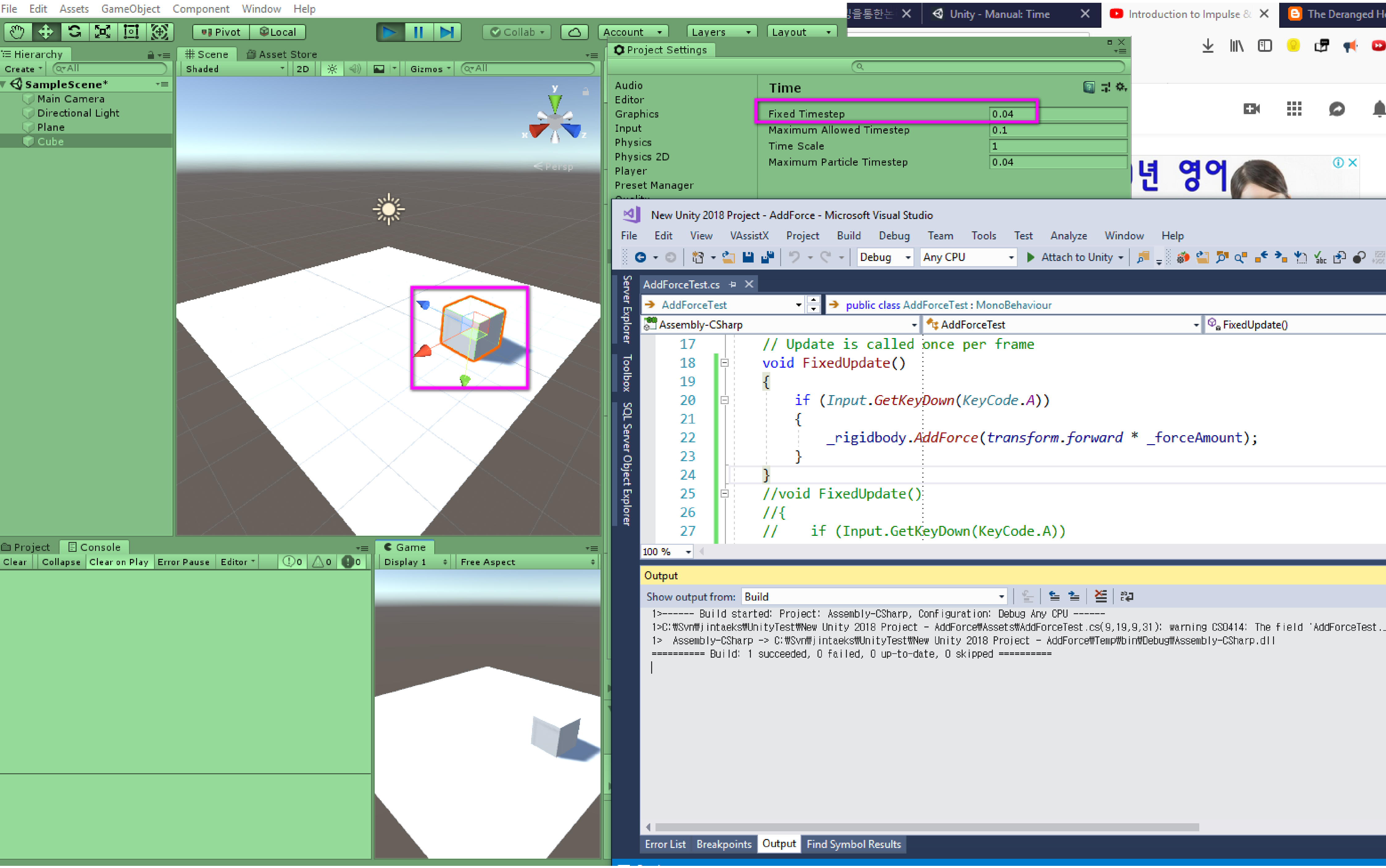
Time

Fixed Timestep	0.02
Maximum Allowed Timestep	0.1
Time Scale	1
Maximum Particle Timestep	0.04

```
public class AddForceTest : MonoBehaviour
```

```
    // Update is called once per frame
    void FixedUpdate()
    {
        if (Input.GetKeyDown(KeyCode.A))
        {
            _rigidbody.AddForce(transform.forward * _forceAmount);
        }
    }
    //void FixedUpdate()
    //{
    //    if (Input.GetKeyDown(KeyCode.A))
```

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>C:₩Svn₩jintaeks₩UnityTest₩New Unity 2018 Project - AddForce₩Assets₩AddForceTest.cs(9,19,9,31): warning CS0414: The fi
1>  Assembly-CSharp -> C:₩Svn₩jintaeks₩UnityTest₩New Unity 2018 Project - AddForce₩Temp₩bin₩Debug₩Assembly-CSharp.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

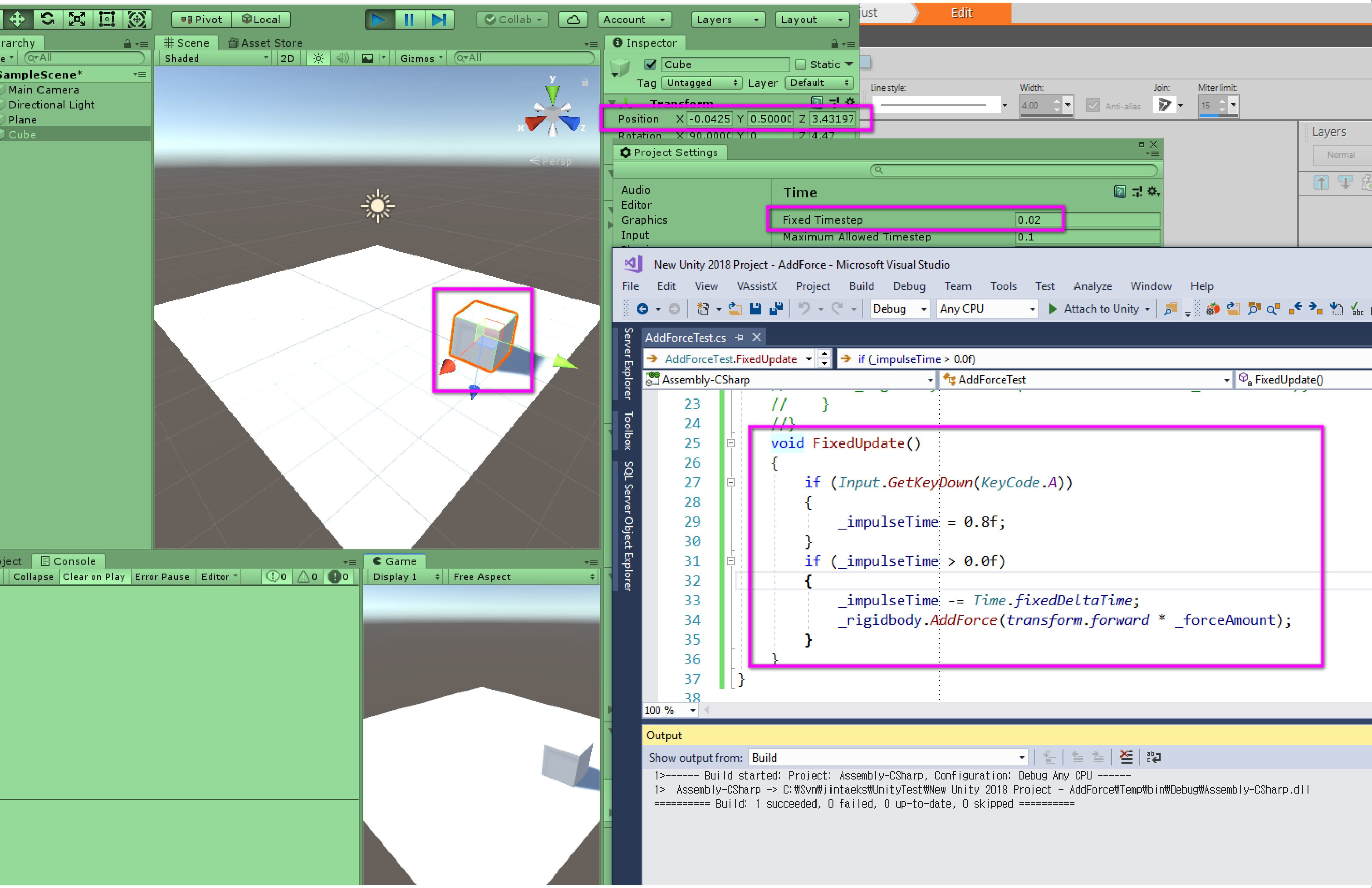


Time

Fixed Timestep	0.04
Maximum Allowed Timestep	0.1
Time Scale	1
Maximum Particle Timestep	0.04

```
    // Update is called once per frame
    void FixedUpdate()
    {
        if (Input.GetKeyDown(KeyCode.A))
        {
            _rigidbody.AddForce(transform.forward * _forceAmount);
        }
    }
    //void FixedUpdate()
    //{
    //    if (Input.GetKeyDown(KeyCode.A))
```

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Assets\AddForceTest.cs(9,19,9,31): warning CS0414: The field 'AddForceTest._
1>  Assembly-CSharp -> C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Temp\bin\Debug\Assembly-CSharp.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



Cube

Tag Untagged Layer Default

Position X -0.0425 Y 0.50000 Z 3.43197

Rotation X 90.0000 Y 0 Z 4.47

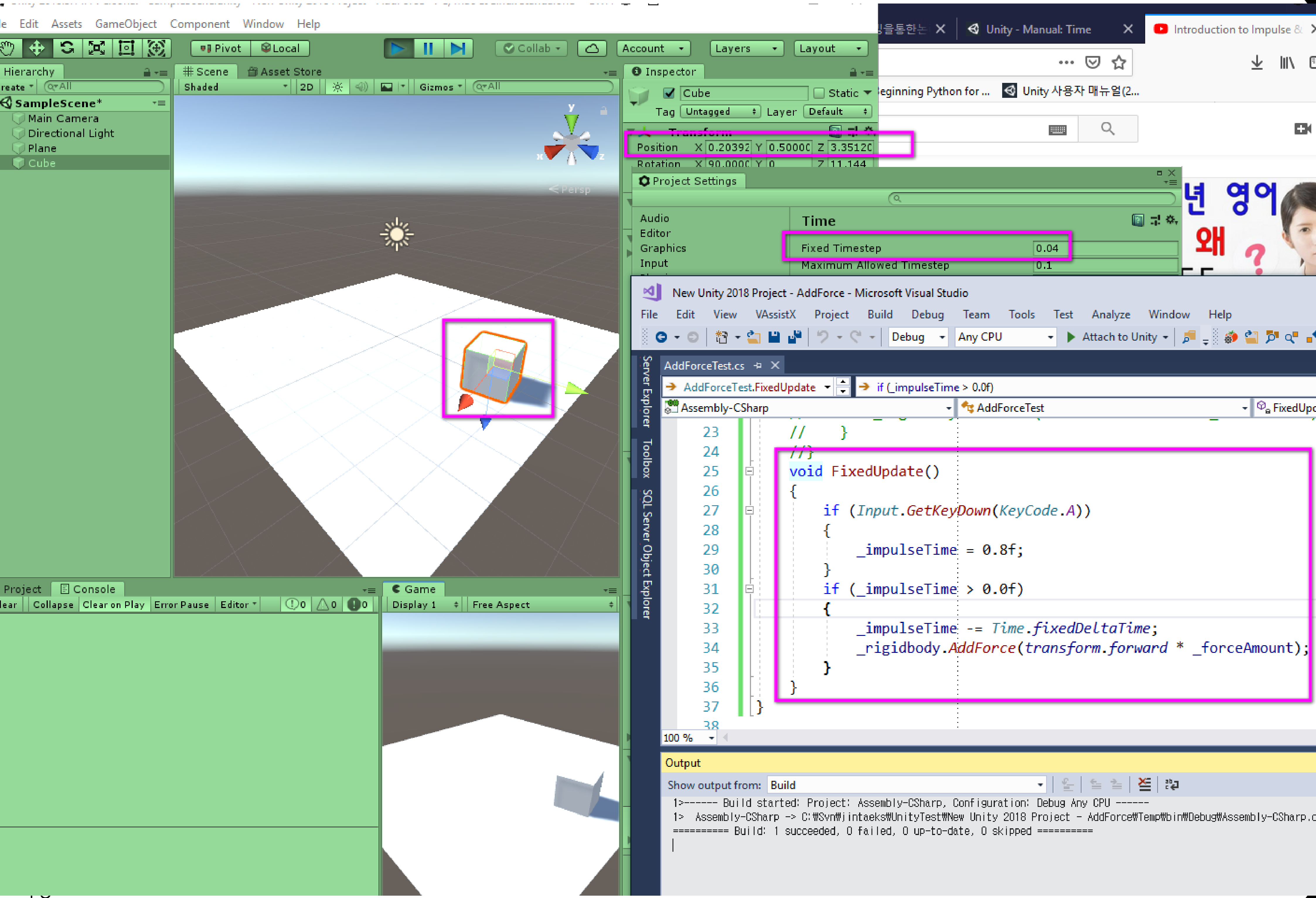
Fixed Timestep 0.02

Maximum Allowed Timestep 0.1

```
    //     }
    //}
    void FixedUpdate()
    {
        if (Input.GetKeyDown(KeyCode.A))
        {
            _impulseTime = 0.8f;
        }
        if (_impulseTime > 0.0f)
        {
            _impulseTime -= Time.fixedDeltaTime;
            _rigidbody.AddForce(transform.forward * _forceAmount);
        }
    }
}
```

Show output from: Build

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>  Assembly-CSharp -> C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Temp\bin\Debug\Assembly-CSharp.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



Cube

Tag Untagged Layer Default

Position X 0.20392 Y 0.50000 Z 3.35120

Rotation X 90.0000 Y 0 Z 11.144

Fixed Timestep 0.04

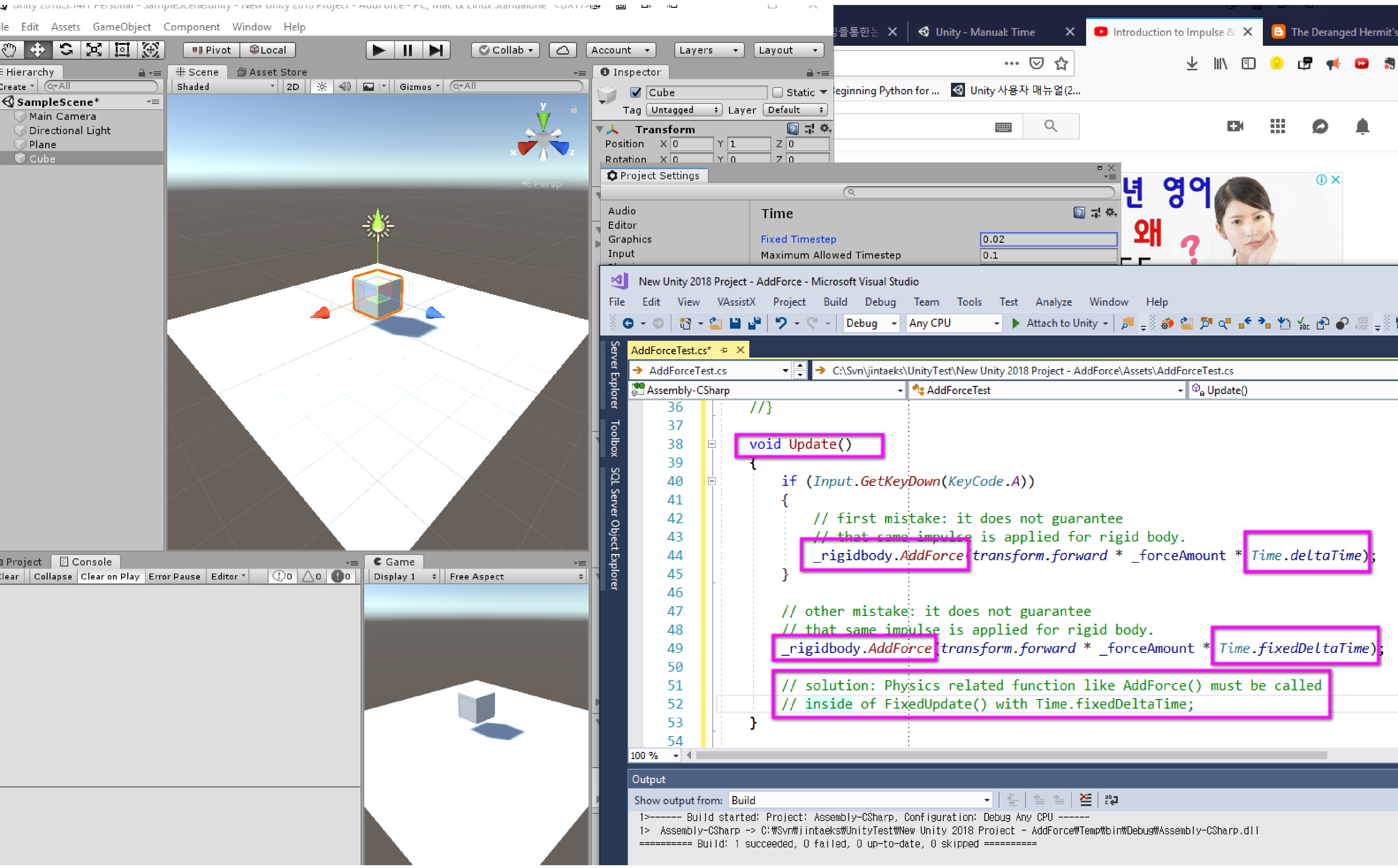
Maximum Allowed Timestep 0.1

```
//     }
//}
void FixedUpdate()
{
    if (Input.GetKeyDown(KeyCode.A))
    {
        _impulseTime = 0.8f;
    }
    if (_impulseTime > 0.0f)
    {
        _impulseTime -= Time.fixedDeltaTime;
        _rigidbody.AddForce(transform.forward * _forceAmount);
    }
}
}
```

Output

Show output from: Build

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>  Assembly-CSharp -> C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Temp\bin\Debug\Assembly-CSharp.d
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```



Inspector: Cube — Tag Untagged, Layer Default
Transform — Position X 0 Y 1 Z 0; Rotation X 0 Y 0 Z 0

Project Settings — Time
Fixed Timestep 0.02
Maximum Allowed Timestep 0.1

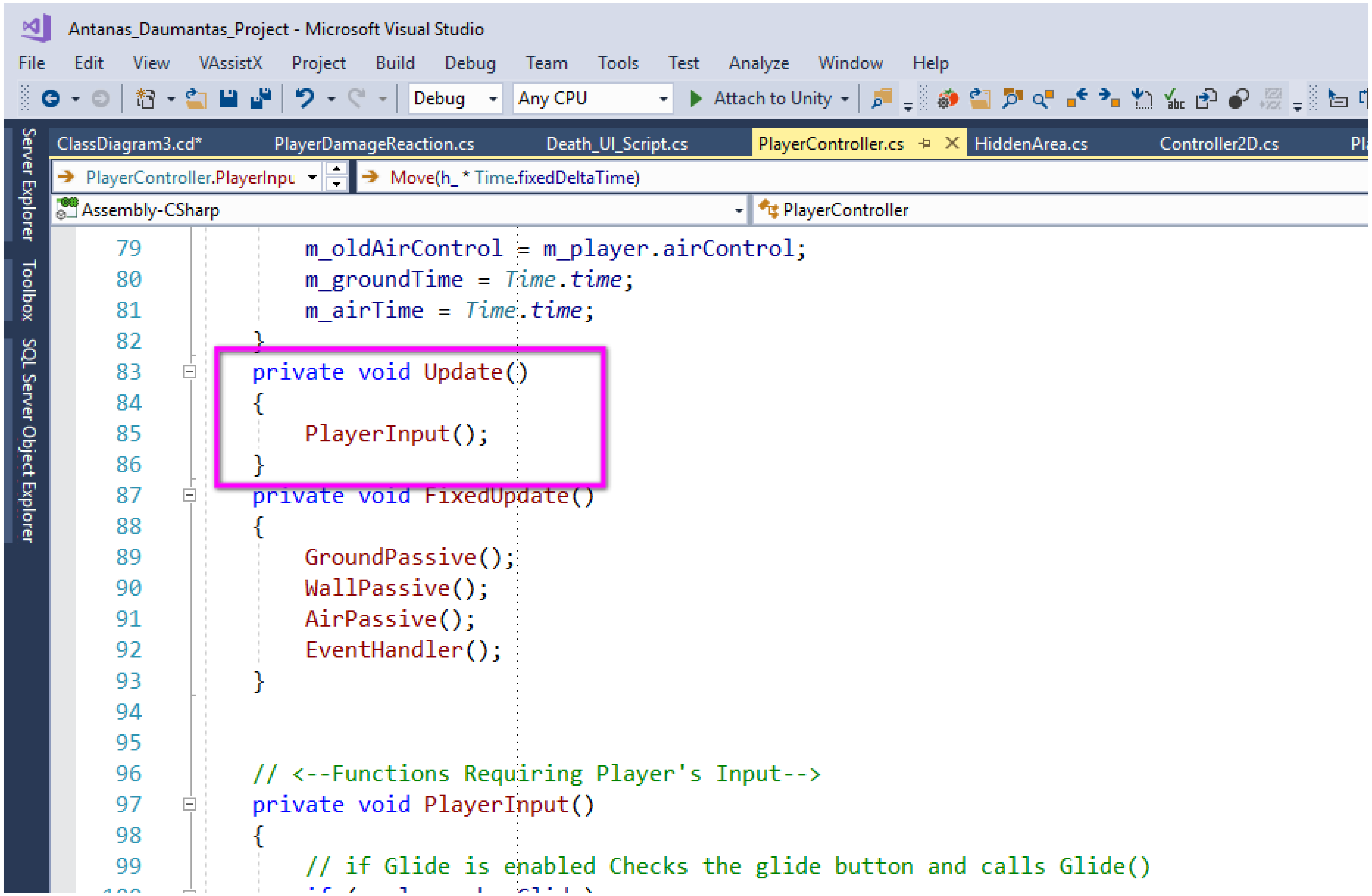
C:\Svn\jintaeks\UnityTest\New Unity 2018 Project - AddForce\Assets\AddForceTest.cs

```
    //}

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.A))
        {
            // first mistake: it does not guarantee
            // that same impulse is applied for rigid body.
            _rigidbody.AddForce(transform.forward * _forceAmount * Time.deltaTime);
        }

        // other mistake: it does not guarantee
        // that same impulse is applied for rigid body.
        _rigidbody.AddForce(transform.forward * _forceAmount * Time.fixedDeltaTime);

        // solution: Physics related function like AddForce() must be called
        // inside of FixedUpdate() with Time.fixedDeltaTime;
    }
```

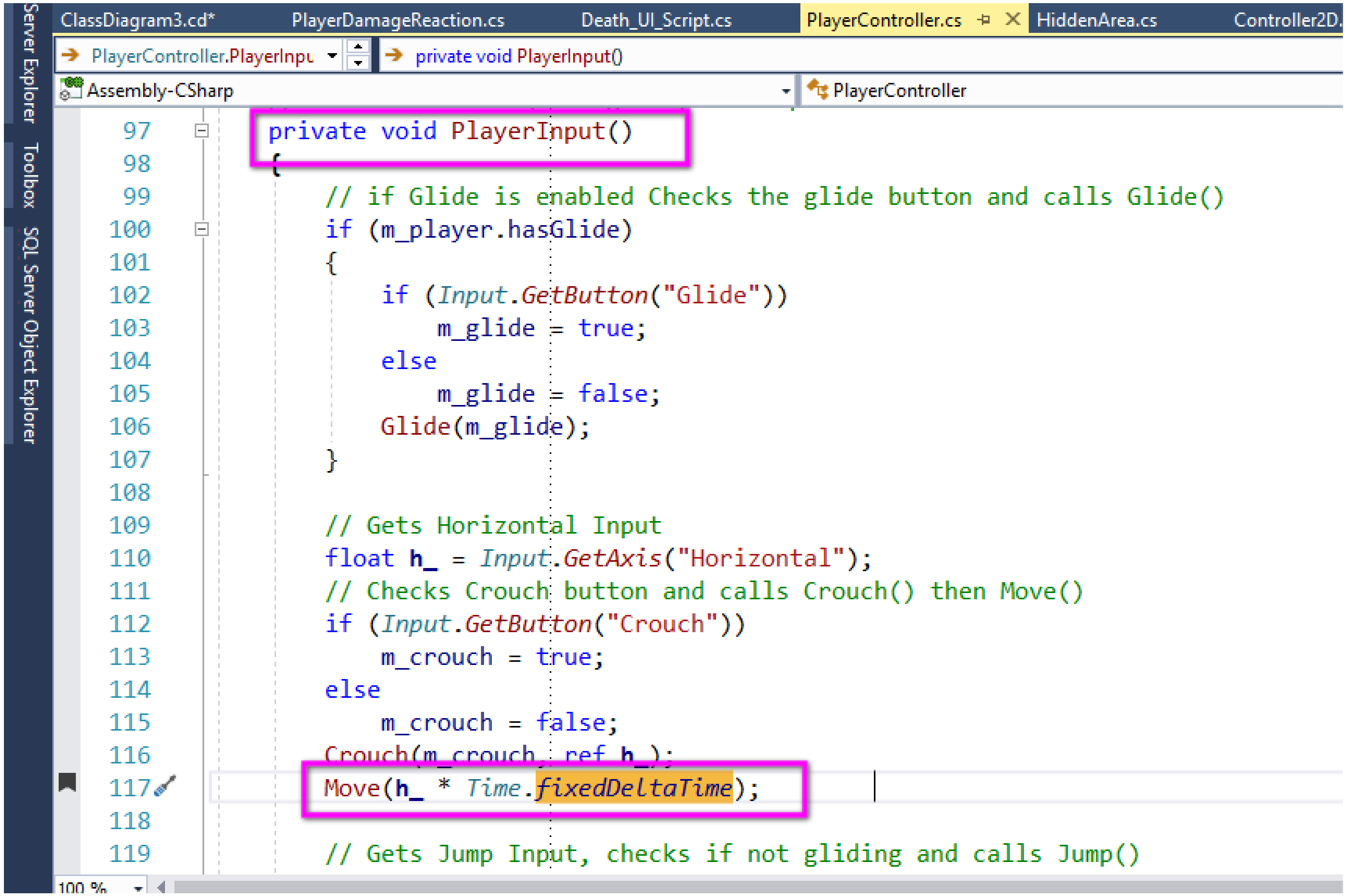
Output — Show output from: Build

```
1>------ Build started: Project: Assembly-CSharp, Configuration: Debug Any CPU ------
1>  Assembly-CSharp -> C:₩Svn₩jintaeks₩UnityTest₩New Unity 2018 Project - AddForce₩Temp₩bin₩Debug₩Assembly-CSharp.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

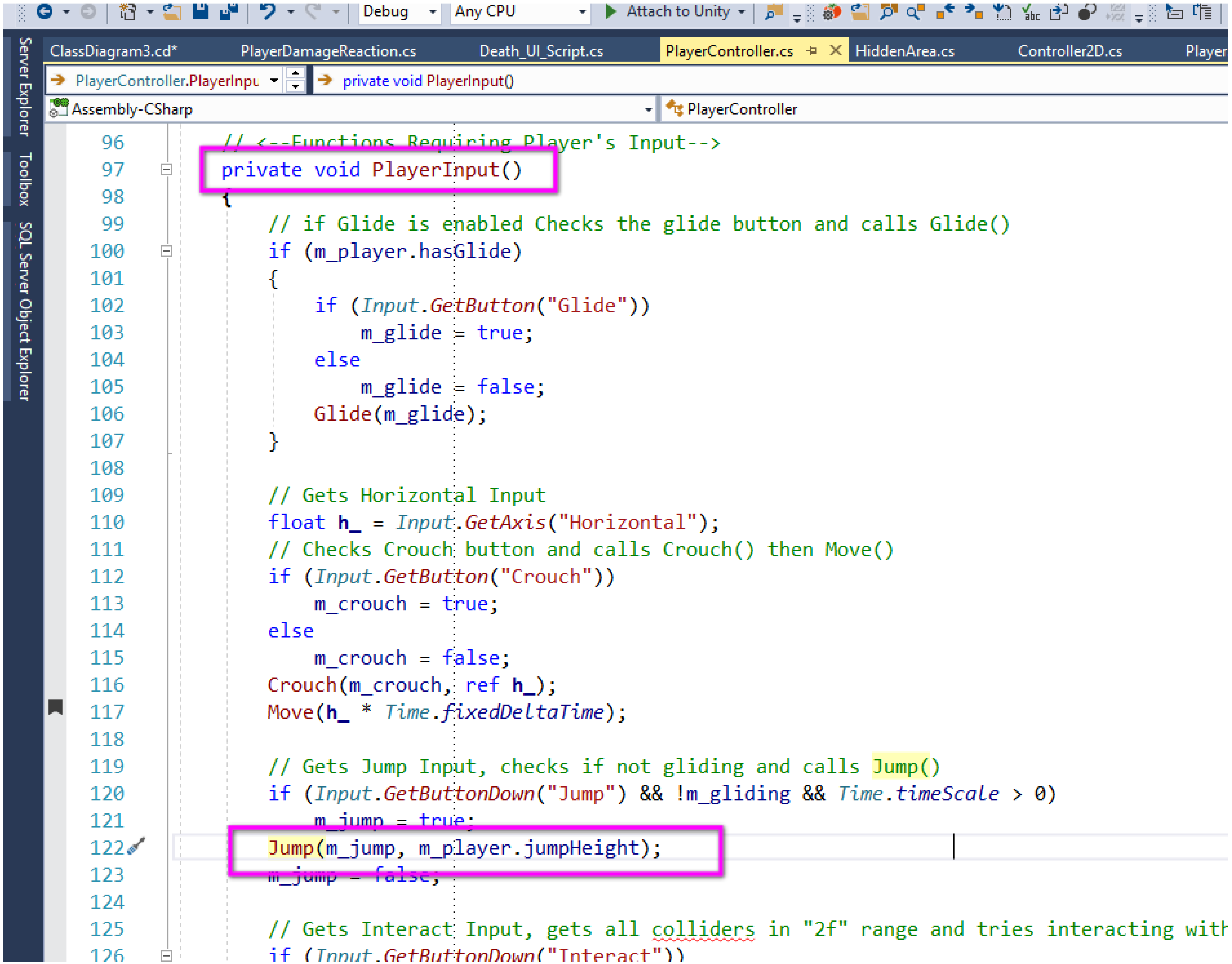
Wrong Usage Example


Antanas_Daumantas_Project - Microsoft Visual Studio
File Edit View VAssistX Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Attach to Unity
Server Explorer
Toolbox
SQL Server Object Explorer
ClassDiagram3.cd*
PlayerDamageReaction.cs
Death_UI_Script.cs
PlayerController.cs
HiddenArea.cs
Controller2D.cs
PlayerController.PlayerInpu
Move(h_ * Time.fixedDeltaTime)
Assembly-CSharp
PlayerController

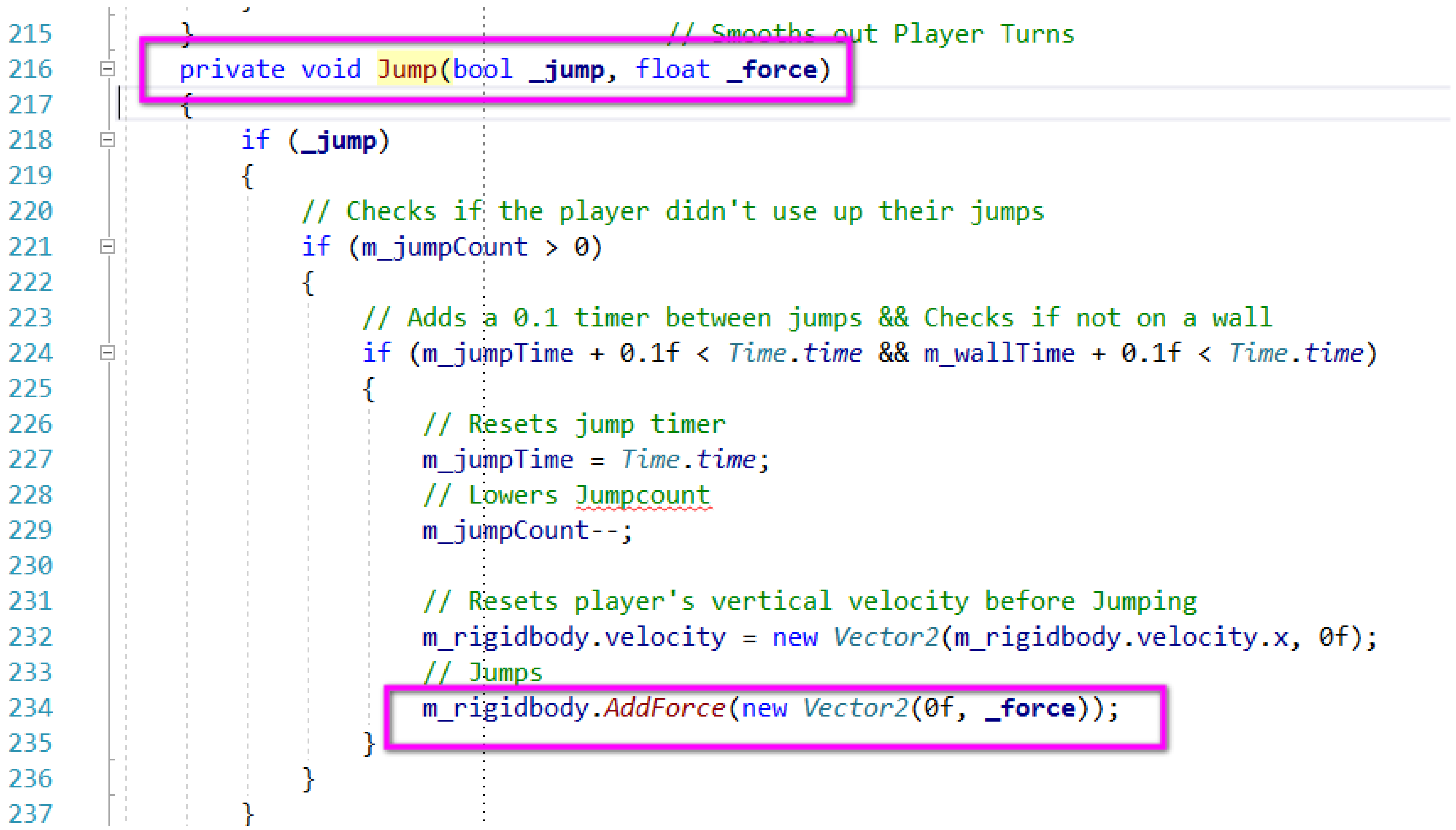
```
            m_oldAirControl = m_player.airControl;
            m_groundTime = Time.time;
            m_airTime = Time.time;
        }
        private void Update()
        {
            PlayerInput();
        }
        private void FixedUpdate()
        {
            GroundPassive();
            WallPassive();
            AirPassive();
            EventHandler();
        }


        // <--Functions Requiring Player's Input-->
        private void PlayerInput()
        {
            // if Glide is enabled Checks the glide button and calls Glide()
```



```
private void PlayerInput()
{
    // if Glide is enabled Checks the glide button and calls Glide()
    if (m_player.hasGlide)
    {
        if (Input.GetButton("Glide"))
            m_glide = true;
        else
            m_glide = false;
        Glide(m_glide);
    }

    // Gets Horizontal Input
    float h_ = Input.GetAxis("Horizontal");
    // Checks Crouch button and calls Crouch() then Move()
    if (Input.GetButton("Crouch"))
        m_crouch = true;
    else
        m_crouch = false;
    Crouch(m_crouch, ref h_);
    Move(h_ * Time.fixedDeltaTime);

    // Gets Jump Input, checks if not gliding and calls Jump()
```



```
    // <--Functions Requiring Player's Input-->
    private void PlayerInput()
    {
        // if Glide is enabled Checks the glide button and calls Glide()
        if (m_player.hasGlide)
        {
            if (Input.GetButton("Glide"))
                m_glide = true;
            else
                m_glide = false;
            Glide(m_glide);
        }

        // Gets Horizontal Input
        float h_ = Input.GetAxis("Horizontal");
        // Checks Crouch button and calls Crouch() then Move()
        if (Input.GetButton("Crouch"))
            m_crouch = true;
        else
            m_crouch = false;
        Crouch(m_crouch, ref h_);
        Move(h_ * Time.fixedDeltaTime);

        // Gets Jump Input, checks if not gliding and calls Jump()
        if (Input.GetButtonDown("Jump") && !m_gliding && Time.timeScale > 0)
            m_jump = true;
        Jump(m_jump, m_player.jumpHeight);
        m_jump = false;

        // Gets Interact Input, gets all colliders in "2f" range and tries interacting with
        if (Input.GetButtonDown("Interact"))
```



```
    }                                   // Smooths out Player Turns
    private void Jump(bool _jump, float _force)
    {
        if (_jump)
        {
            // Checks if the player didn't use up their jumps
            if (m_jumpCount > 0)
            {
                // Adds a 0.1 timer between jumps && Checks if not on a wall
                if (m_jumpTime + 0.1f < Time.time && m_wallTime + 0.1f < Time.time)
                {
                    // Resets jump timer
                    m_jumpTime = Time.time;
                    // Lowers Jumpcount
                    m_jumpCount--;

                    // Resets player's vertical velocity before Jumping
                    m_rigidbody.velocity = new Vector2(m_rigidbody.velocity.x, 0f);
                    // Jumps
                    m_rigidbody.AddForce(new Vector2(0f, _force));
                }
            }
        }
```

References

- ✓ https://en.wikipedia.org/wiki/Momentum
- ✓ https://docs.unity3d.com/ScriptReference/ForceMode.Impulse.html
- ✓ https://answers.unity.com/questions/713217/exact-difference-between-fixeddeltatime-and-deltat.html

QnA

MY **BRIGHT** FUTURE

DSU **Dongseo** University
동서대학교