

TRM62117

PEMROGRAMAN GAME 2D

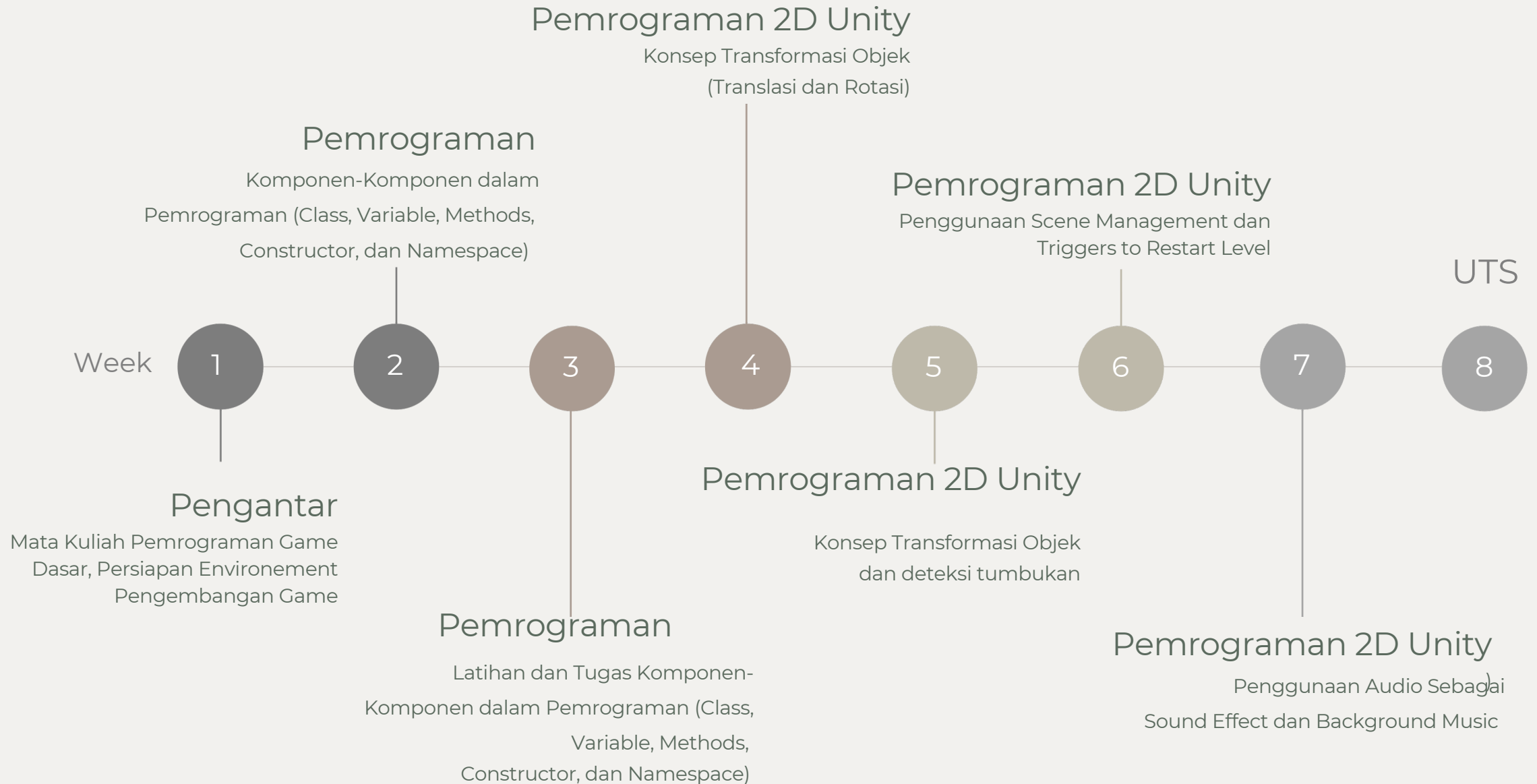
Dosen Pengampu :
Yeni Nurhasanah, S.Pd., M.T.

Penyusun :
Yeni Nurhasanah, S.Pd., M.T
Tita Aprilliawati Putri Santoso, S.ST., M.T.

Semester 4 Prodi Teknologi
Rekayasa Multimedia
Politeknik Negeri Media Kreatif
Tahun Akademik 2024 2025



Timeline



Transformasi Objek

Terdapat 3 properti transform di Unity, yaitu Position, Rotation, dan Scale

Transform merupakan Class yang sudah disediakan oleh Unity :

- localScale adalah properti yang terdapat pada Class Transform

Translate() dan Rotate() adalah Method yang terdapat pada Class Transform

Untuk dapat memanipulasi transformasi objek, kita dapat menggunakan input system

01 Input System



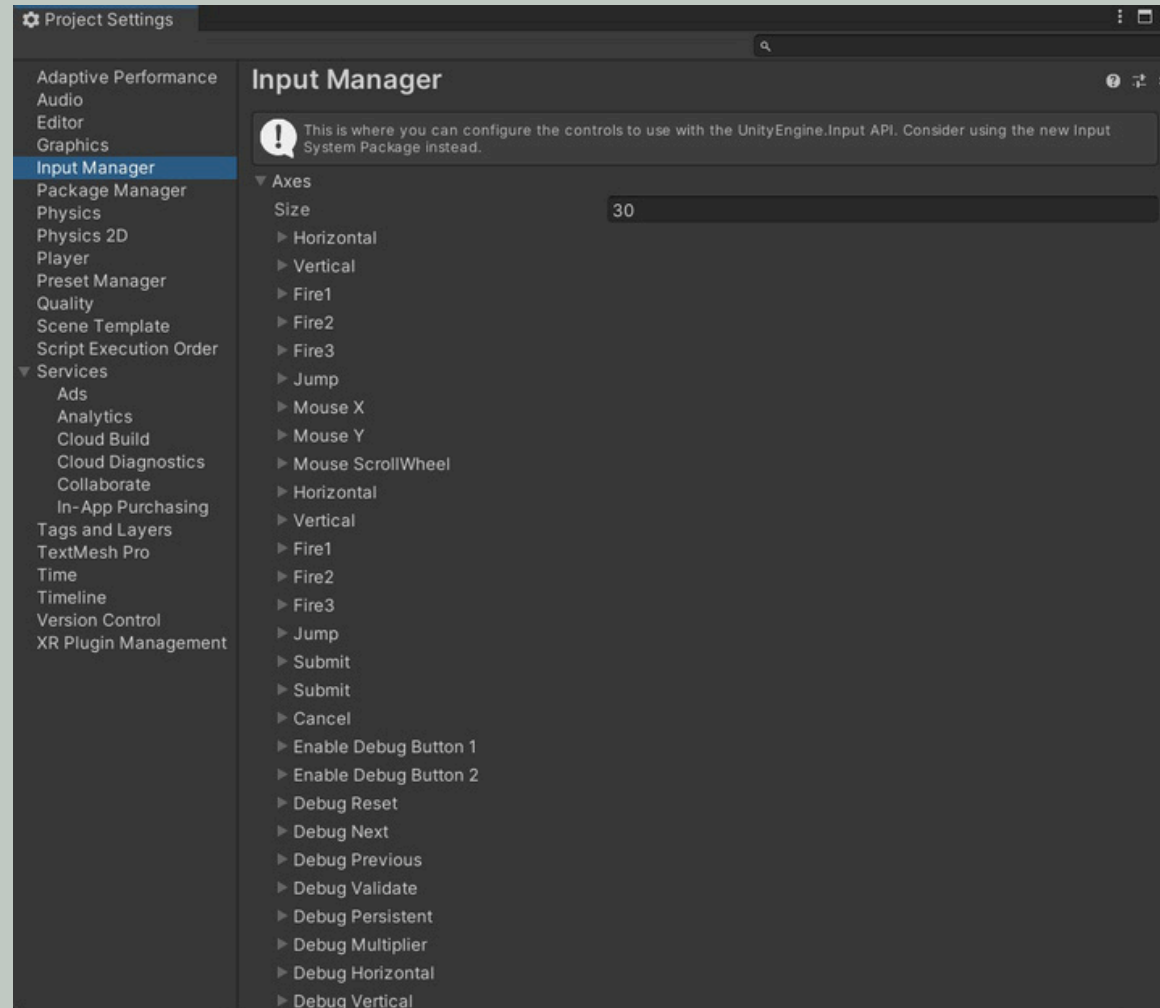
Input System akan mengubah physical action pada pemain/pengguna menjadi informasi untuk dilakukan di dalam game (misal : key press, button press, mouse button press)

Input adalah Class yg disediakan oleh unity dengan beberapa properties di dalamnya, seperti :

- .GetAxis
- .GetKeyDown
- .GetMouseButtonDown

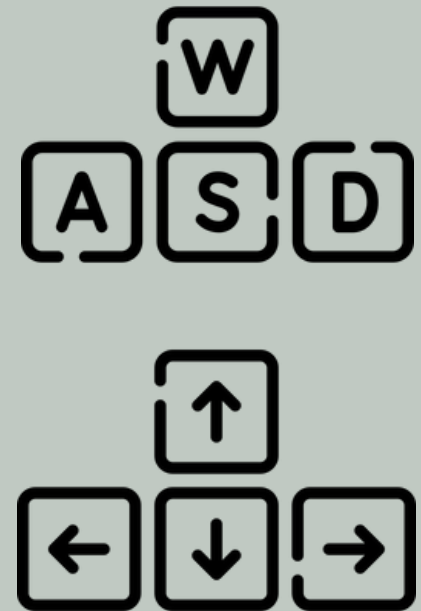
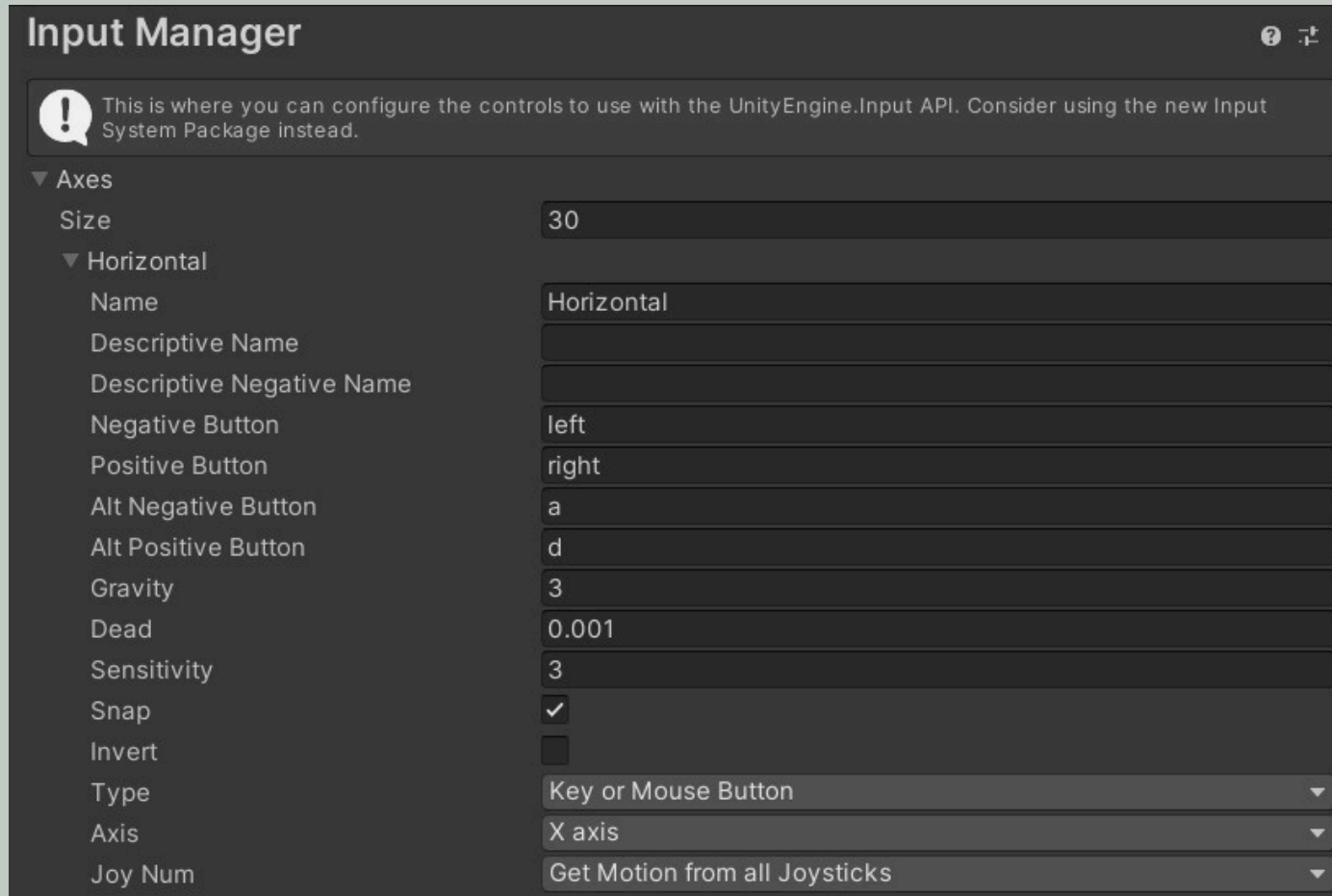
01 Input System (GetAxis)

Edit > Project Settings > Input Manager



01 Input System (GetAxis)

Edit > Project Settings > Input Manager > Horizontal



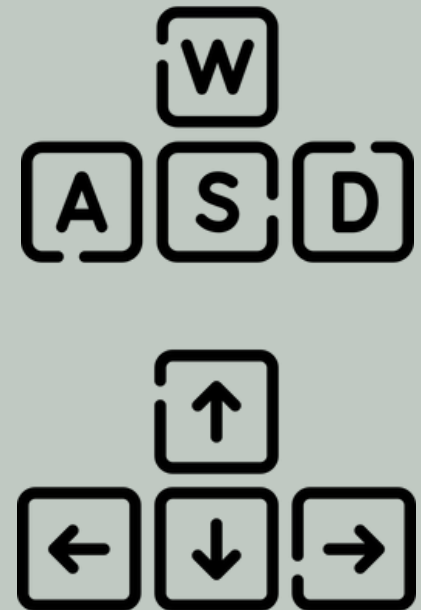
01 Input System (GetAxis)

Edit > Project Settings > Input Manager > Vertical

Input Manager

▼ Vertical

Name	Vertical
Descriptive Name	
Descriptive Negative Name	
Negative Button	down
Positive Button	up
Alt Negative Button	s
Alt Positive Button	w
Gravity	3
Dead	0.001
Sensitivity	3
Snap	<input checked="" type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks



01 Input System (GetAxis)

```
Drive.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 Unity Script | 0 references
6 public class Drive : MonoBehaviour
7 {
8     [SerializeField] float steerSpeed = 0.5f;
9     [SerializeField] float moveSpeed = 0.01f;
10
11     // Start is called before the first frame update
12     Unity Message | 0 references
13     void Start()
14     {
15     }
16
17     // Update is called once per frame
18     Unity Message | 0 references
19     void Update()
20     {
21         float steerAmount = Input.GetAxis("Horizontal") * steerSpeed;
22         float moveAmount = Input.GetAxis("Vertical") * moveSpeed;
23
24         transform.Rotate(0, 0, -steerAmount);
25         transform.Translate(0, moveAmount, 0);
26     }
27 }
```

Variable baru untuk menambahkan input system

Melakukan transformasi objek (rotasi dan translasi)

01 Input System (GetKeyDown)

```
Drive.cs [X]
Assembly-CSharp [Drive]
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [Unity Script | 0 references]
6  public class Drive : MonoBehaviour
7  {
8      [SerializeField] float steerSpeed = 0.5f;
9      [SerializeField] float moveSpeed = 0.01f;
10
11     // Start is called before the first frame update
12     [Unity Message | 0 references]
13     void Start()
14     {
15     }
16
17     // Update is called once per frame
18     [Unity Message | 0 references]
19     void Update()
20     {
21         if (Input.GetKeyDown(KeyCode.R))
22         {
23             transform.Rotate(0, 0, steerSpeed);
24         }
25         if (Input.GetKeyDown(KeyCode.Space))
26         {
27             transform.Translate(0, moveSpeed, 0);
28         }
29     }
30 }
```

Menggunakan tombol "R" pada keyboard untuk berotasi

Menggunakan tombol "spasi" pada keyboard untuk bertranslasi

01 Input System (GetMouseButtonDown)

```
Drive.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [Unity Script | 0 references]
6 public class Drive : MonoBehaviour
7 {
8     [SerializeField] float steerSpeed = 0.5f;
9     [SerializeField] float moveSpeed = 0.01f;
10
11     // Start is called before the first frame update
12     [Unity Message | 0 references]
13     void Start()
14     {
15     }
16
17     // Update is called once per frame
18     [Unity Message | 0 references]
19     void Update()
20     {
21         if (Input.GetMouseButtonDown(0))
22         {
23             transform.Rotate(0, 0, steerSpeed);
24         }
25         if (Input.GetMouseButtonDown(1))
26         {
27             transform.Translate(0, moveSpeed, 0);
28         }
29     }
30 }
```

GetMouseButtonDown(0) → left-click

GetMouseButtonDown(1) → right-click

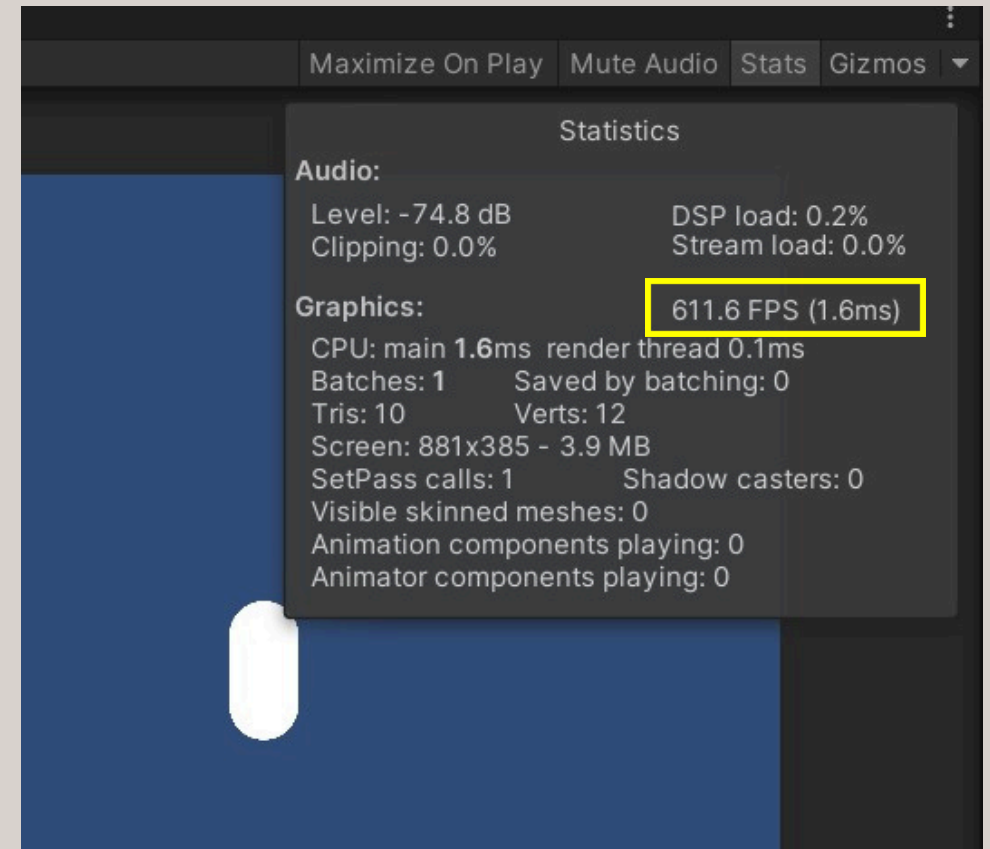
GetMouseButtonDown(2) → middle-click

Menggunakan left-click pada mouse untuk berotasi

Menggunakan right-click pada mouse untuk bertranslasi

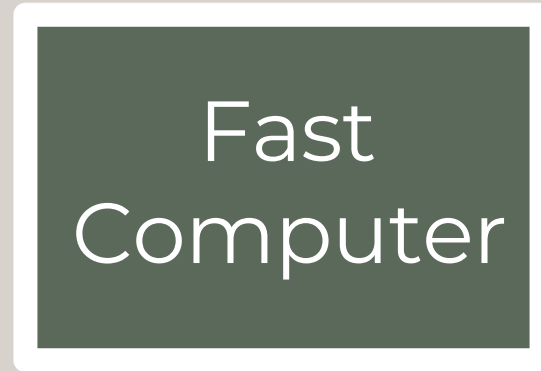
02 Time.deltaTime()

- FPS (Frames per Second) berbeda antar komputer
- Menggunakan Time.deltaTime dapat memberi tahu kita berapa lama waktu yang dibutuhkan untuk mengeksekusi setiap frame
- Saat kita mengalikan sesuatu dengan Time.deltaTime, hal ini menjadikan game kita “frame rate independent”, sehingga game akan berperilaku sama pada komputer yang cepat maupun lambat



02 Time.deltaTime()

Example : On Update (each frame) move 1 unit to the left



Frames per second

10

100

Duration of frame

0.1 s

0.01 s

Distance per second

$$1 \times 10 \times 0.1 = 1$$

$$1 \times 100 \times 0.01 = 1$$

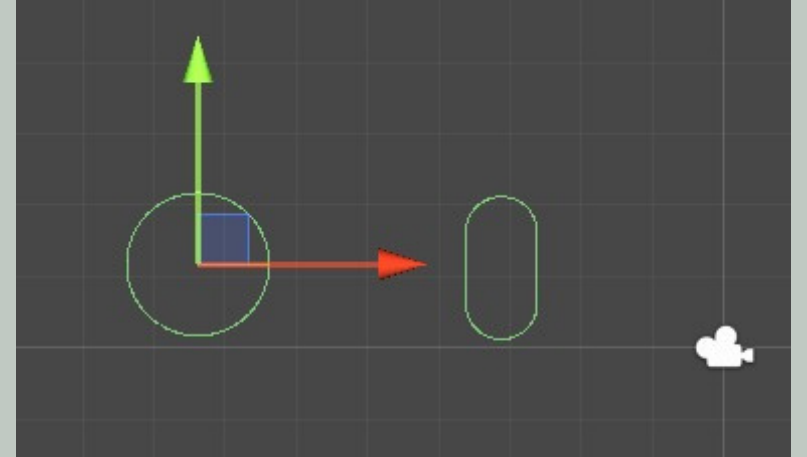
02 Time.deltaTime()

```
Drive.cs [X]
Assembly-CSharp [v] Drive
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  Unity Script | 0 references
6  public class Drive : MonoBehaviour
7  {
8      [SerializeField] float steerSpeed = 0.5f;
9      [SerializeField] float moveSpeed = 0.01f;
10
11     // Start is called before the first frame update
12     Unity Message | 0 references
13     void Start()
14     {
15     }
16
17     // Update is called once per frame
18     Unity Message | 0 references
19     void Update()
20     {
21         float steerAmount = Input.GetAxis("Horizontal") * steerSpeed * Time.deltaTime;
22         float moveAmount = Input.GetAxis("Vertical") * moveSpeed * Time.deltaTime;
23
24         transform.Rotate(0, 0, -steerAmount);
25         transform.Translate(0, moveAmount, 0);
26     }
27 }
```

Mengalikan nilai variable dengan Time.deltaTime

03 Colliders and Rigidbodies

Collider merupakan komponen tidak terlihat (invisible) yang menentukan bentuk suatu objek untuk tujuan tumbukan secara fisik (physical collisions)

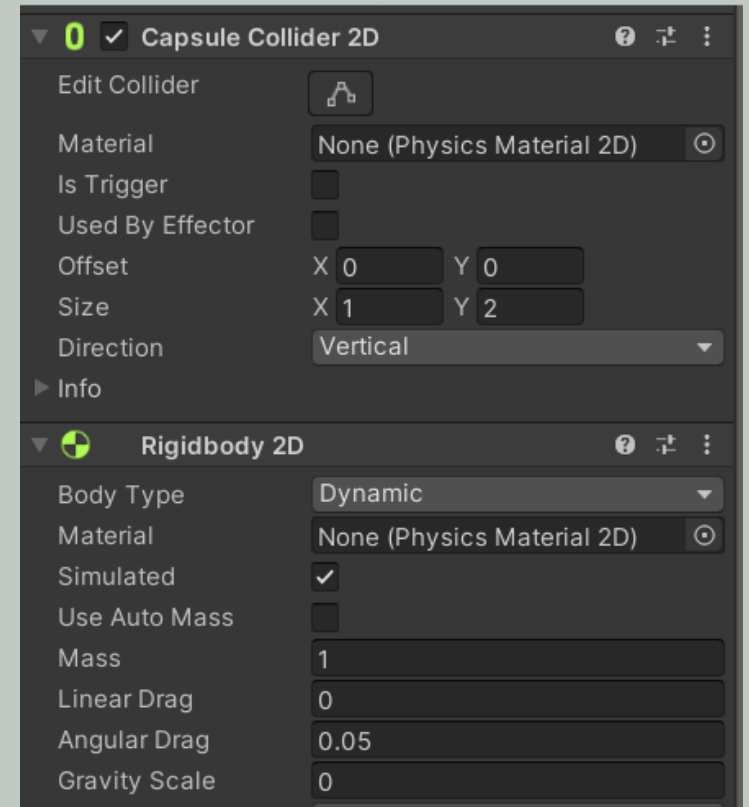
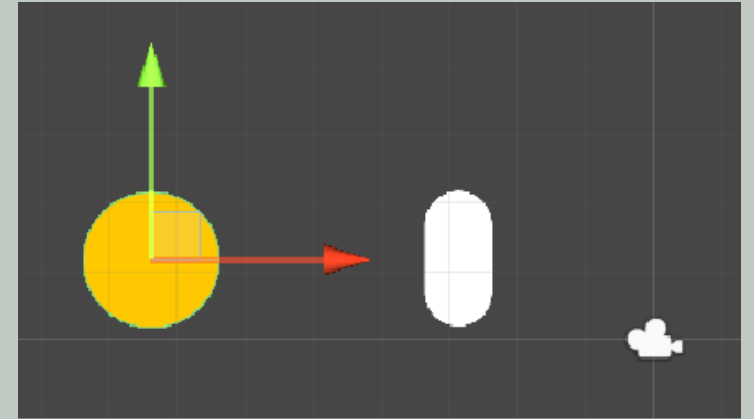


MengapamembutuhkanRigidbody? Untuk membuat gerakan suatu objek berada di bawah kendali Unity's physics engine.

Bahkan tanpa menambahkan kode apa pun, objek dengan Rigidbody akan ditarik ke bawaholeh gravitasidan akanbereaksiterhadap tumbukan dengan objek lain

03 Colliders and Rigidbodies

1. Buat 2D sprite baru berbentuk lingkaran dan beri warna
2. Tambahkan collider pada objek capsule
(Add Component > Physics2D > Capsule Collider2D)
3. Tambahkan collider pada objek circle
(Add Component > Physics2D > Circle Collider 2D)
4. Tambahkan rigidbody pada objek capsule
(Add Component > Physics 2D > Rigidbody 2D)
5. Ubah Gravity Scale menjadi 0



04 OnCollisionEnter2D()

Digunakan untuk memberikan informasi ketika collider sebuah objek melakukan kontak dengan collider objek lain (2D physics only)

void OnCollisionEnter2D(Collision2D collision)

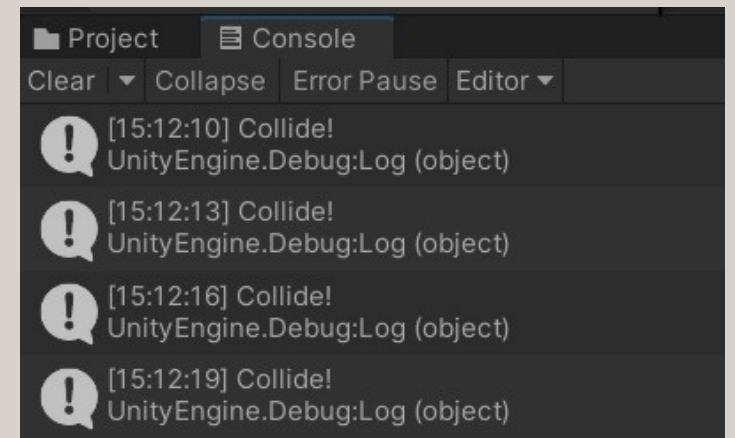
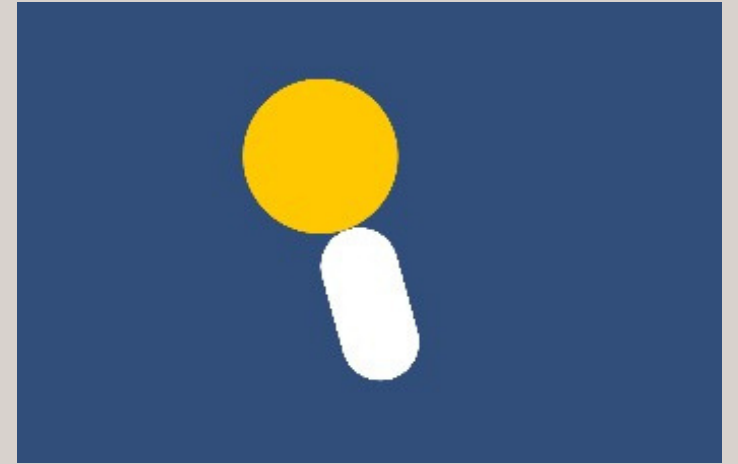


Parameter

(Collision2D yang terkait dengan tumbukan ini)

04 OnCollisionEnter2D()

1. Buat C# script baru bernama "Collision" dan tambahkan pada komponen objek capsule
2. Buat sebuah method baru bernama "OnCollisionEnter2D"
3. Tampilkan pesan pada console menggunakan "Debug.Log()" ketika objek capsule bertumbukan dengan objek lain



04 OnCollisionEnter2D()

```
Collision.cs [X]
Assembly-CSharp Collision
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 Unity Script | 0 references
6 public class Collision : MonoBehaviour
7 {
8     Unity Message | 0 references
9     void OnCollisionEnter2D(Collision2D collision)
10     {
11         Debug.Log("Collide!");
12     }
13 }
```

Menggunakan method
OnCollisionEnter2D dan
menampilkan pesan pada console

OnCollisionEnter2D →

Bertujuan untuk mendeteksi tumbukan fisik antar objek (ketika collider antar objek melakukan kontak, akan terjadi akibat fisik yang ditimbulkan)

VS

OnTriggerEnter2D →

Bertujuan untuk mendeteksi peristiwa yang dipicu (trigger) saat terjadi kontak antar objek (ketika collider antar objek melakukan kontak, tidak terjadi akibat fisik yang ditimbulkan)

05 OnTriggerEnter2D()

Digunakan untuk memberikan informasi ketika collider objek lain memasuki trigger collider yang terpasang pada sebuah objek (2D physics only)

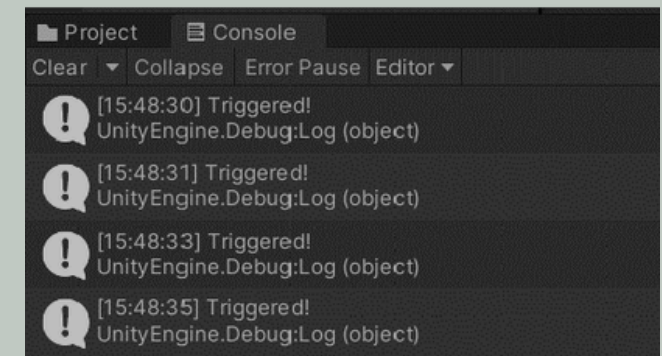
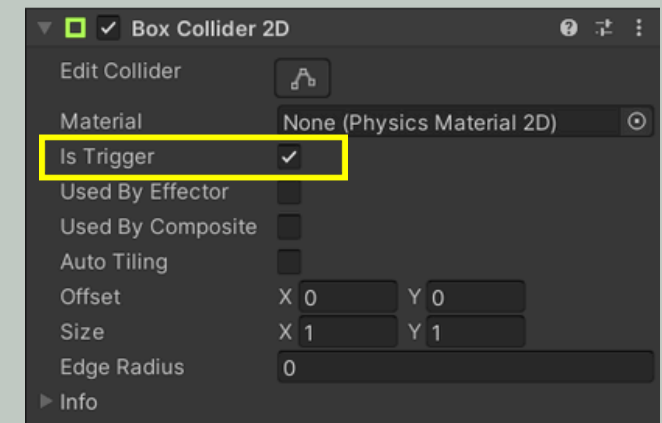
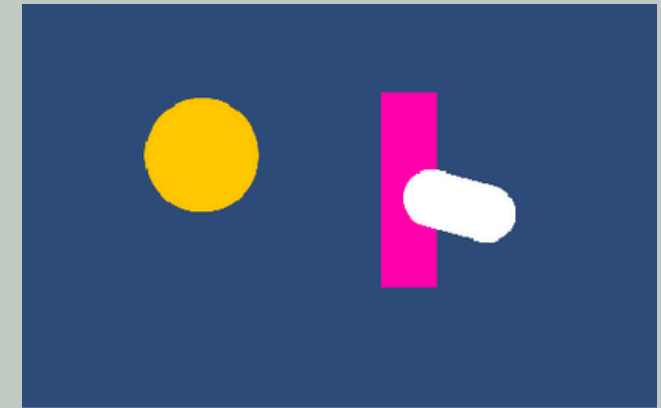
```
void OnTriggerEnter2D(Collider2D collision)
```



Parameter

05 OnTriggerEnter2D()

1. Tambahkan 2D sprite baru berbentuk rectangular dan beri warna
2. Tambahkan collider pada objek rectangular
3. Aktifkan komponen “isTrigger”
4. Buat sebuah method baru bernama “OnTriggerEnter2D”
5. Tampilkan pesan pada console menggunakan “Debug.Log()” ketika objek capsule trigger dengan objek lain



05 OnTriggerEnter2D()

```
Collision.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Collision : MonoBehaviour
6 {
7     void OnCollisionEnter2D(Collision2D collision)
8     {
9         Debug.Log("Collide!");
10    }
11
12    void OnTriggerEnter2D(Collider2D collision)
13    {
14        Debug.Log("Triggered!");
15    }
16 }
17
```

Menggunakan method
OnTriggerEnter2D dan
menampilkan pesan pada console

Question?

Kesimpulan (1)

- Input System akan mengubah physical action pada pemain/pengguna menjadi informasi untuk dilakukan di dalam game, misal :
 - `Input.GetAxis` → Menggunakan input manager yang telah disediakan oleh Unity
 - `Input.GetKeyDown` → Menggunakan input dari tombol keyboard
 - `Input.GetMouseButtonDown` → Menggunakan input dari mouse (left, right, middle-click)
- `Time.deltaTime` membuat game menjadi “frame rate independent”, sehingga game akan berperilaku sama pada komputer yang cepat maupun lambat

Kesimpulan (2)

- Collider dan Rigidbody adalah 2 hal yang saling berkaitan dalam collision (untuk dapat menggunakan rigidbody, kita harus memiliki collider, karena collider yang akan mendeteksi terjadinya tumbukan antar objek)
- Perbedaan OnCollisionEnter2D dan OnTriggerEnter2D adalah akibat yang ditimbulkan. OnCollisionEnter2D akan menimbulkan akibat fisik, tetapi tidak pada OnTriggerEnter2D. Namun, keduanya tetap membutuhkan collider untuk mendeteksi terjadinya kontak antar objek.

Let's Practice!

1. Buat 5 objek 2D sprite dengan warna yang berbeda-beda, berupa :
 - Capsule (sebagai Car, dapat digerakkan)
 - Circle
 - Square
 - Rectangle
 - Diamond

2. Gunakan method OnCollisionEnter2D / OnTriggerEnter2D untuk membuat kondisi-kondisi berikut :
 - Circle dan Capsule saling memantul saat bertumbukan
 - Capsule akan tembus saat bertumbukan dengan Square (tampilkan pesan pada console)
 - Rectangle akan hilang (destroy) saat bertumbukan dengan Capsule
 - Diamond dan Capsule akan sama-sama hilang (destroy) saat bertumbukan

Hasil tugas berupa screen capture program/script dan
video screen record saat play mode

Kumpulkan di Google Classroom pada link “Tugas Pertemuan 05” :

- Hasil screen capture dalam bentuk PDF dengan nama file “Kelas_NIM>Nama_TP05”
- Video screen record dalam bentuk mp4 dengan nama file “Kelas_NIM>Nama_TP05_Video”

References

<https://docs.unity3d.com/ScriptReference/Input.html>

<https://docs.unity3d.com/ScriptReference/Input.GetAxis.html>

<https://docs.unity3d.com/ScriptReference/Input.GetKeyDown.html>

<https://docs.unity3d.com/ScriptReference/Input.GetMouseButtonDown.html>

<https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>

<https://docs.unity3d.com/Manual/CollidersOverview.html>

<https://docs.unity3d.com/560/Documentation/Manual/CollidersOverview.html>

<https://docs.unity3d.com/ScriptReference/Rigidbody.html>

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnCollisionEnter2D.html>

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerEnter2D.html>