

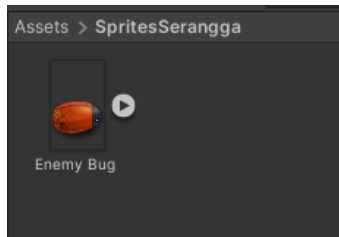
Pertemuan Ke- 4

Laporan

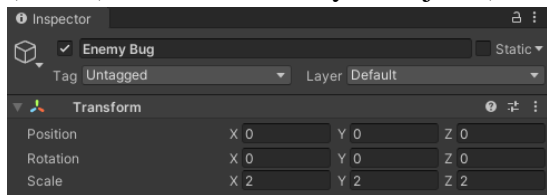
LANGKAH-LANGKAH Pengerjaan**1. Flipping Sprite secara Horizontal**

1 Buat Project Unity 2D

2 Import Import gambar EnemyBug.png yang ada di folder SpritesSerangga kedalam project

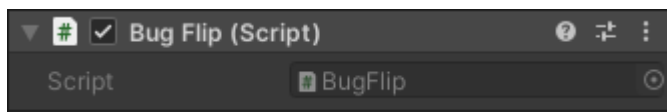


3 Drag gambar Enemy Bug ke scene. Atur position (x,y,z) GameObject tersebut menjadi (0, 0, 0) dan ubah scale-nya menjadi (2, 2, 2).

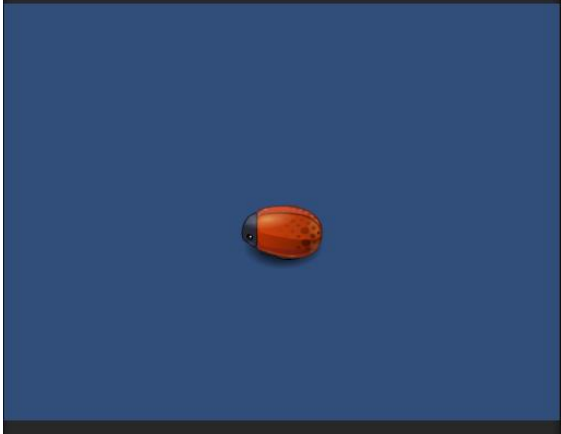
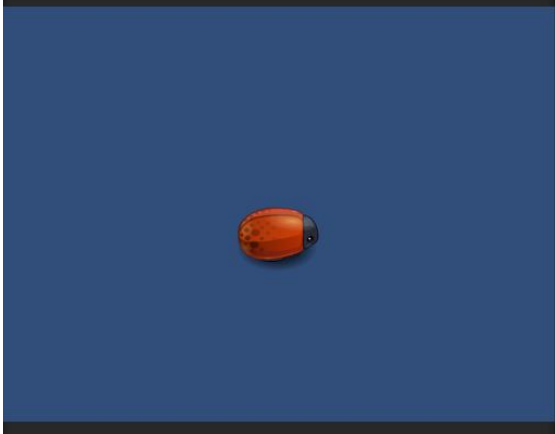
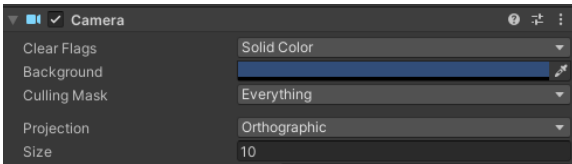
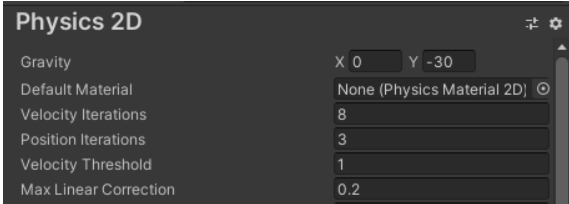
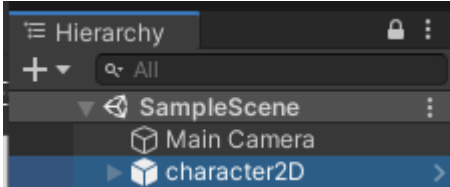



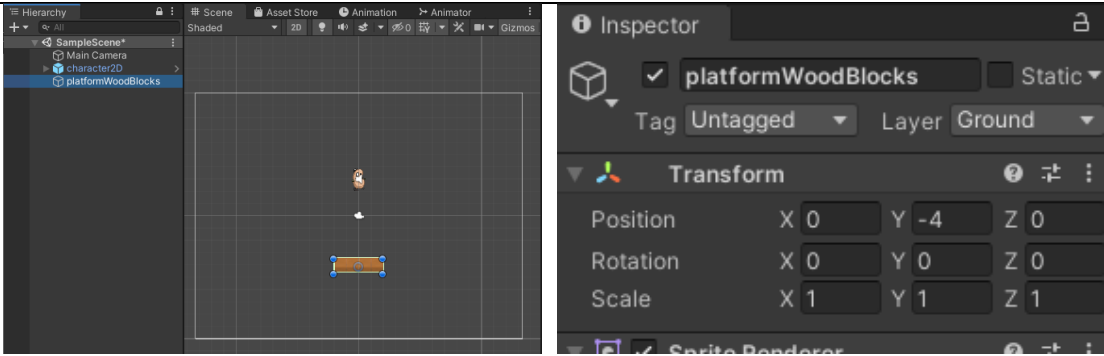
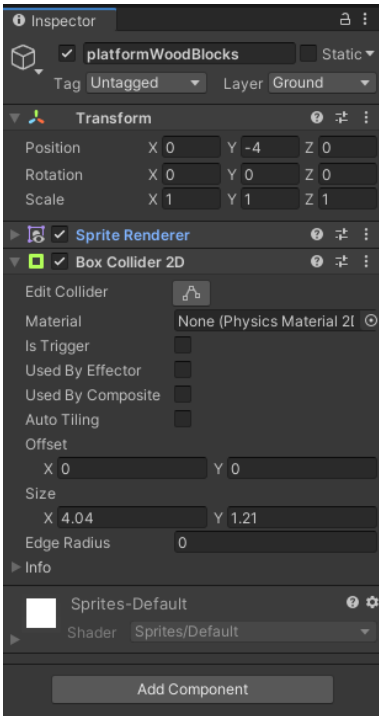
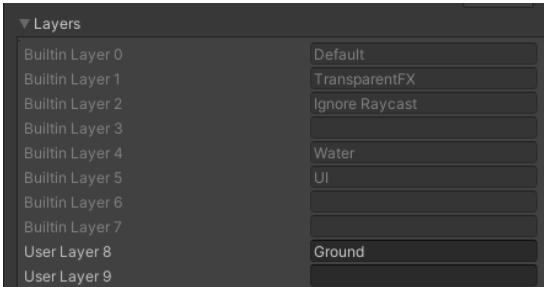
4 Buat C# script dengan nama BugFlip dengan code seperti pada table dibawah ini, kemudian tambahkan code tersebut ke GameObject Enemy Bug.

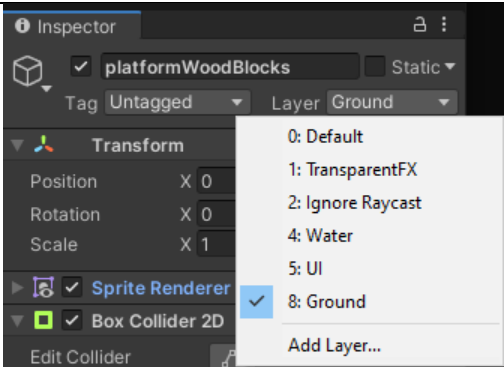
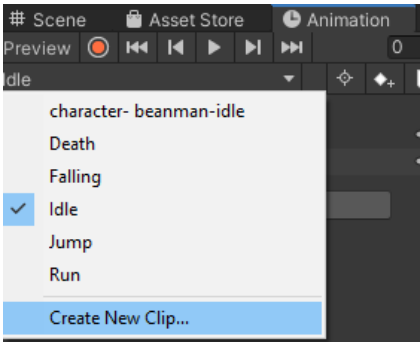

Jika script C# berhasil ditambah pada GameObject Enemy Bug hasilnya adalah seperti berikut:


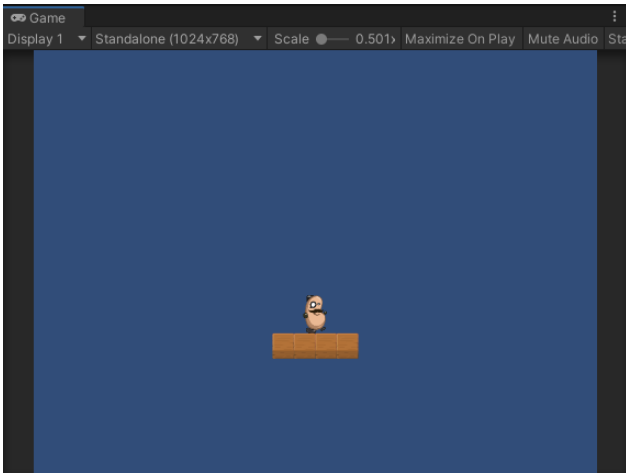
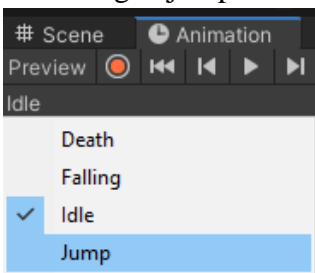


5 Play kemudian tekan button arah ke kiri dan kanan maka anda akan melihat GameObject berubah arah sesuai dengan button yang ada tekan.

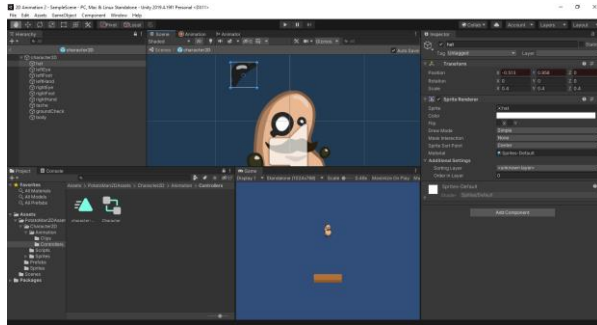
| | |
|---|---|
| | <div>  <p><i>ketika diklik kiri</i></p> </div> <div>  <p><i>ketika diklik kanan</i></p> </div> |
| 6 | Langkah terakhir simpan scene yang telah dibuat. |
| 2. Membuat animasi bagian tubuh ketika karakter bergerak | |
| 1 | Buat project unity 2D (2D Animation 2). |
| 2 | Import asset yang ada di folder PotatoMan2DAssets. |
| 3 | <p>Tambah size Main Camera menjadi 10.</p>  |
| 4 | <p>Mengatur 2D gravity dengan cara pilih Edit Project Settings Physics 2D, kemudian ubah nilai Y pada bagian paling atas menjadi -30</p>  |
| 5 | <p>Drag character2D dari folder Prefabs ke scene. Atur position (x,y,z) GameObject menjadi (0, 3, 0).</p> <div>   </div> |
| 6 | <p>Drag sprite platformWallBlocks dari folder Project Sprites ke scene. Atur position GameObject menjadi (0, -4, 0).</p> |

| | |
|---|---|
| |  |
| 7 | <p>Tambahkan Box Collider 2D component pada GameObject platformWallBlocks dengan cara pilih Add Component Physics 2D Box Collider 2D.</p>  |
| 8 | <p>Ubah layer GameObject platformWallBlocks dengan cara pilih Add layer kemudian ketik Ground.</p>  <p>Pilih kembali platformWallBlocks pada hierarchy kemudian pilih layer Ground seperti pada gambar di bawah ini.</p> |

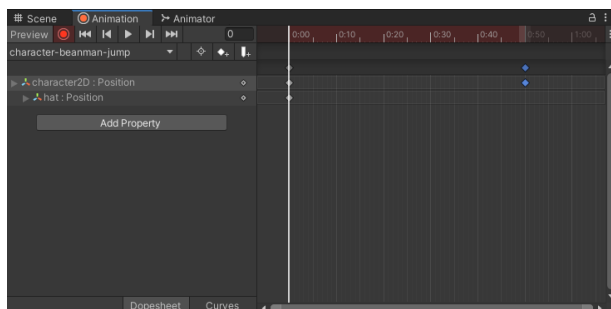
| | |
|----|--|
| |  |
| 9 | Pilih GameObject character2D pada Hierarchy, kemudian buka Animation panel dengan cara buka Window Animation, kemudian pilih button Create. Simpan hasilnya dengan nama Character2D di folder Character2D Animation. |
| 10 | <p>Klik dropdown menu pada panel Animation kemudian pilih menu item [Create New Clip].</p>  |
| 11 | Simpan new clip di folder Character2D Animation, beri nama idle. Anda pada tahap ini telah membuat clip Animation untuk 'idle' character state (tidak bergerak). |
| 12 | <p>Amati pada folder Character2D Animation yang ada di Project panel , seharusnya dapat dilihat terdapat 2 file yaitu Animation clip yang telah dibuat dengan nama (idle) dan Animator controller dari GameObject character2D dengan nama character2D.</p> <p>Note : Jika ada clip character2D maka delete saja karena tidak memerlukan file tersebut.</p>  |
| 13 | Pilih GameObject character2D di Hierarchy, kemudian buka Animator panel (Window |

| | |
|----|---|
| | <p>Animator). Anda dapat melihat State Machine untuk mengatur animation dari character. Pada tahap ini kita hanya memiliki satu Animation clip (character-beanman-idle) maka tambahkan clip character-beanman-idle pada entry State Machine.</p> <p>Dengan cara klik kanan pada Entry kemudian pilih Set StateMachine Default State arahkan ke clip idle.</p>  |
| 14 | <p>Play scene – maka anda akan melihat character selalu pada 'idle' state, untuk sementara ini anda tidak dapat menggerakkan character tersebut sampai membuat clip yang lain. Note: karakter bergerak turun disebabkan gravity bernilai 1 pada Rigidbody 2D. Dan character berhenti pada platformWoodBlocks karena sama-sama memiliki Box Collider 2D.</p>  |
| 15 | <p>Pada tahap selanjutnya kita akan membuat 'jump' Animation clip untuk animasi hat. Dengan cara klik the empty dropdown menu pada Animation panel, kemudian buat clip baru dengan jump dan simpan pada folder Animation folder.</p>  |
| 16 | <p>Klik button Add Property, dan pilih Transform Position yang merupakan hat child object, dengan cara klik '+' plus-sign button. Pada tahap ini kita melakukan recording posisi (X, Y, Z) pada GameObject hat pada clip animation.</p> |
| 17 | <p>Dapat anda amati terdapat 2 'keyframes' pada 0.0 dan at 1.0. Hal ini direpresentasikan dengan diamonds pada Timeline area di sisi sebelah kanan Animation panel.</p> |
| 18 | <p>Pilih keyframe pertama pada waktu (0,0) kemudian pada Scene panel ubah posisi hat ke atas dan ke kiri, menjauhi kepala. Amati nilai X,Y,Z pada inspector memiliki background merah hal ini menunjukkan bahwa perubahab pada Transform component di</p> |

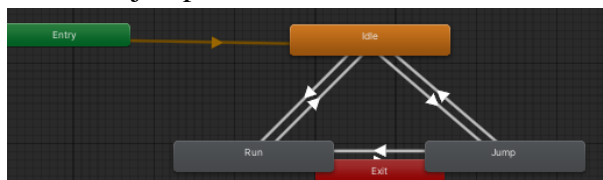
record / rekam pada animation clip.



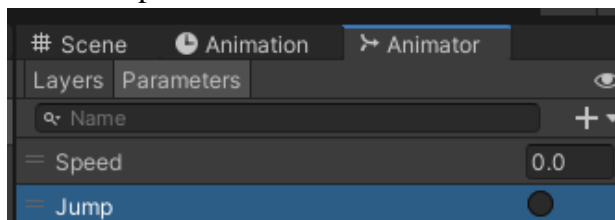
- 19 Karena 1 detik terlalu lama untuk jump animation, drag keyframe kedua ke kiri pada waktu ke 0.5



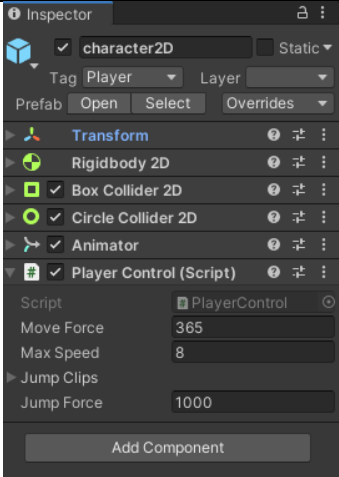
- 20 Pada langkah selanjutnya mengatur transisi dari 'idle' state ke 'jump' state. Pada Animator panel pilih idle selanjutnya buat transisi ke state jump. Dengan cara klik kanan dan pilih menu Make Transition, kemudian drag transition arrow ke state character-beanman-jump.

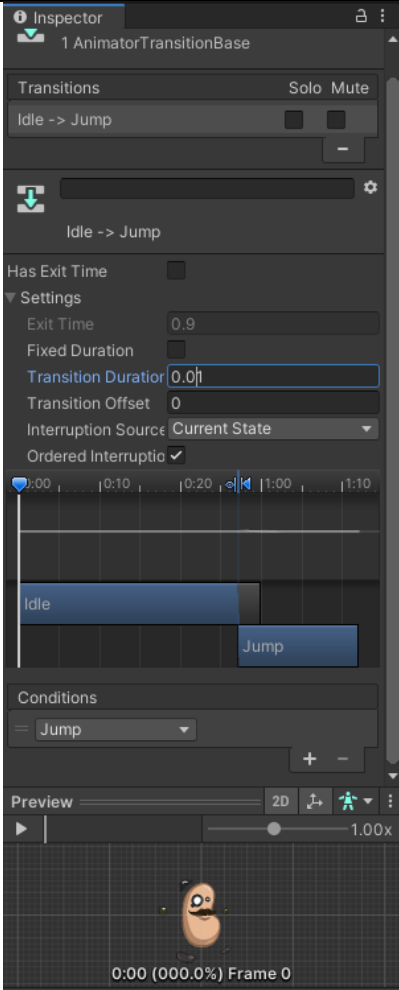
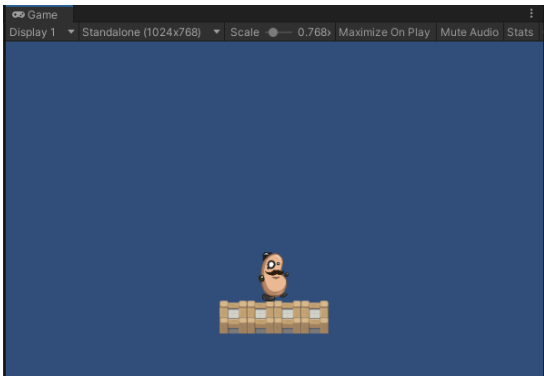


- 21 Kemudian tambahkan Trigger parameter dengan nama 'Jump', dengan cara klik button add parameter plus-sign "+" pada bagian kiri atas Animator panel, pilih Trigger dan beri nama Jump.



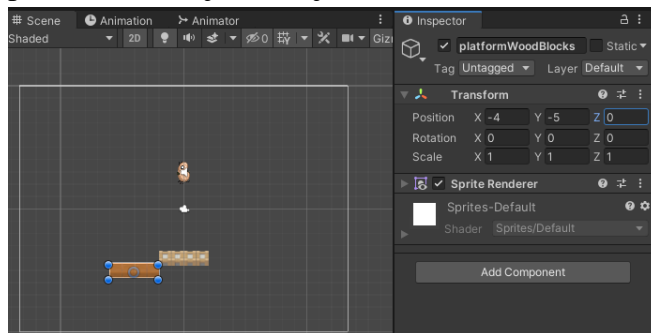
- 22 Pastikan script PlayerControl yang ada di folder script di add ke GameObject character2D.

| | |
|----|--|
| |  |
| 23 | <p>Masih pada Animator Panel, kemudian kita atur properties untuk menentukan kapan karakter melakukan Transition dari idle ke jump. Hal ini dapat dilakukan dengan cara klik Transition arrow, kemudian ubah 4 nilai pada properties di Inspector panel, yaitu:</p> <ol style="list-style-type: none">Has Exit Time: uncheck.Transition Duration: 0.01Interruption State: Current State.Conditions: Add Jump (click plus-sign '+' button at bottom) |

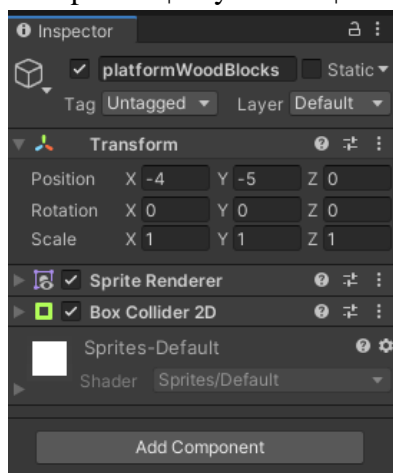
| | |
|----|---|
| |  |
| 24 | <p>Simpan dan run, scene yang telah dibuat. Ketika character mendarat kemudian tekan spasi untuk jump / lompat. Anda dapat melihat character hat lompat menjauhi kepala character dan perlahan kembali. Pada tahap ini kita tidak membuat transisi untuk meninggalkan/berpindah dari Jump state maka Animation clip ini akan berulang, sehingga character hat akan tetap bergerak ketika lompatan character telah berakhir.</p>  |
| 25 | <p>Di Animator panel pilih state jump dan tambahkan Transition untuk kembali ke state idle. Pilih Transition arrow pada Inspector panel ubah propertiesnya seperti berikut:</p> <ol style="list-style-type: none"> Has Exit Time: check time :0.5 (nilai ini harus sama dengan nilai pada second keyframe pada clip Jump) |

| | |
|---|--|
| | <p>animation)</p> <p>c. Transition Duration: 0.01</p> <p>d. Interruption State: Current State</p> |
| 26 | <p>Simpan dan run scene. Sekarang ketika lompat topinya hanya menjauh dari kepalanya sekali, setelah itu character kembali pada Idle state</p> |
| <p>Membuat 3-Frame Animation Clip Sehingga Animasi Bergerak Secara Terus Menerus</p> | |
| 1 | <p>Melanjutkan praktikum pada sub bab kedua, langkah pertama buat scene dengan nama yang berbeda dengan scene sebelumnya, dengan cara tekan File save scene as ... beri nama misalnya 3FrameAnimation.</p> |

- 2 Drag sprite platformWoodBlocks dari folder Project | Sprites ke dalam scene. Ubah posisi GameObject menjadi (-4, -5, 0)



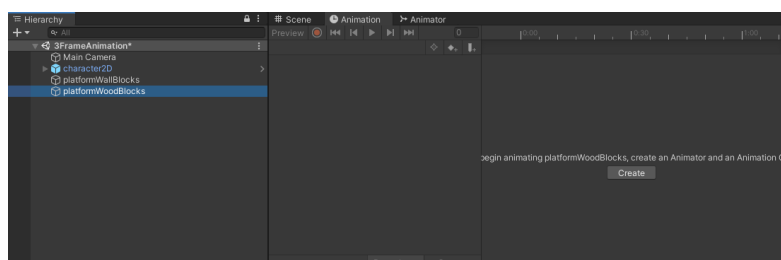
- 3 Tambahkan Box Collider 2D component pada GameObject platformWoodBlocks sehingga character2D juga dapat berdiri pada GameObject ini. Dengan cara Add Component | Physics 2D | Box Collider 2D.

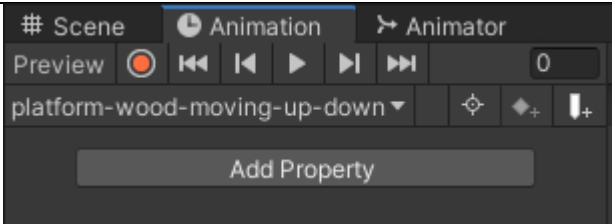
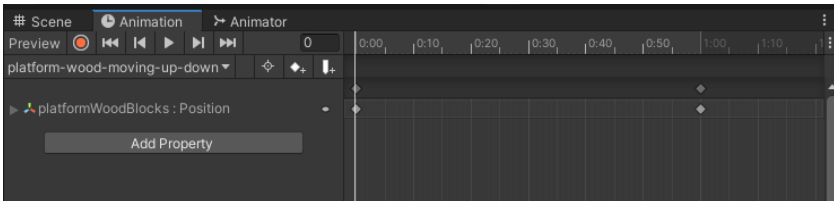
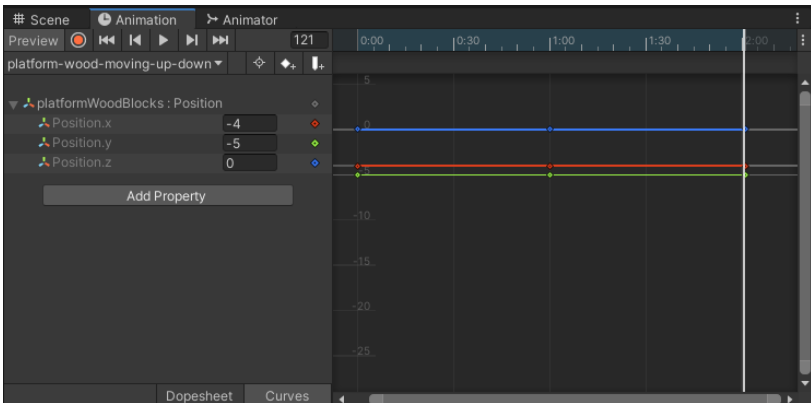


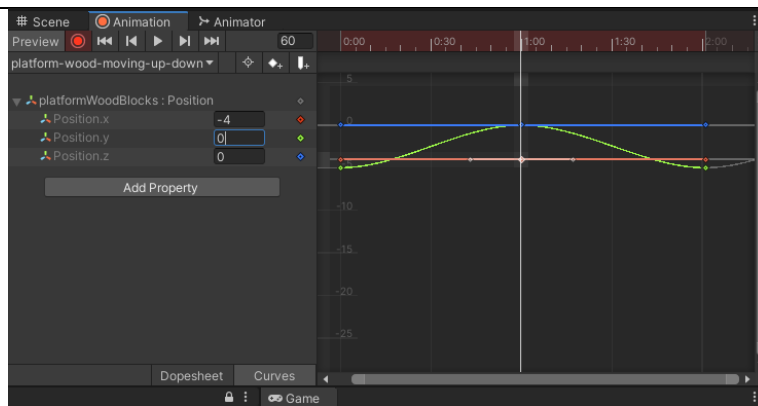
- 4 Buat folder baru dengan nama Animations yang akan digunakan untuk menyimpan animation clip dan controller.

- 5 Pastikan GameObject platformWoodBlocks dipilih pada hierarchy kemudian buka Animation panel (Window | Animation).

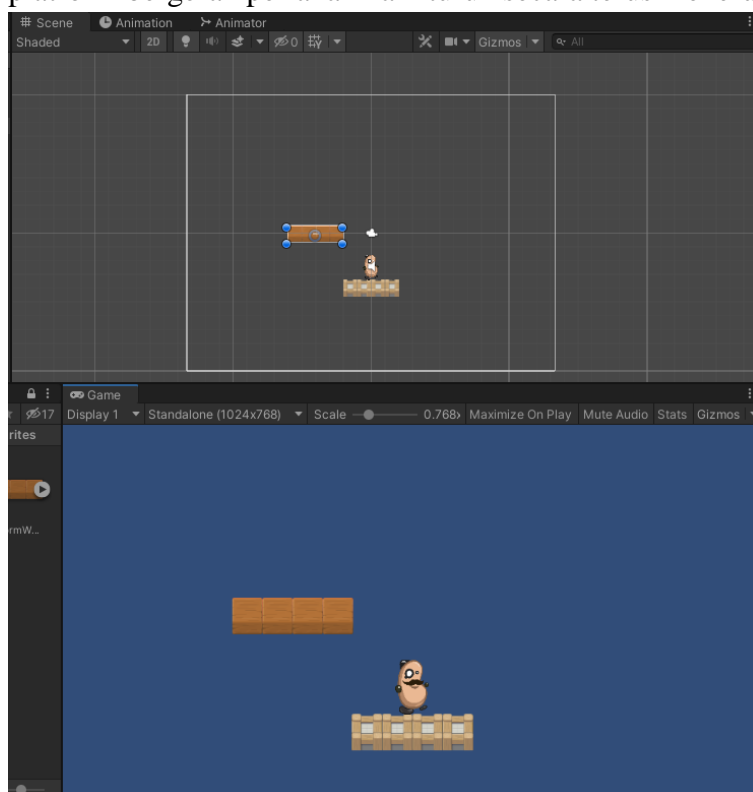
- 6 Kemudian buat clip dengan cara tekan button create pada Animation Pannel , beri nama clip baru dengan nama platform-wood-moving-up-down dan simpan di folder Animations.



| | |
|----|---|
| |  |
| 7 | <p>Klik button Curve pada Animation Panel kemudian pilih Transform Position kemudian klik tanda '+'. Pada tahap ini, kita merekam perubahan posisi (X, Y, Z) pada GameObject platformWoodBlocks.</p>  |
| 8 | <p>Anda akan melihat 2 'keyframes' pada posisi 0.0 dan 1.0. Keyframe direpresentasikan oleh diamonds.</p> |
| 9 | <p>Kita memerlukan 3 keyframes, sehingga perlu menambah 1 keyframe lagi dengan cara klik 2:00 pada Timeline kemudian klik button diamond+ untuk membuat keyframe yang baru.</p>  |
| 10 | <p>Keyframe pertama dan ketiga sudah benar nilai Y = -5 karena keyframe tersebut merekam/record tinggi dari wood platform. Akan tetapi kita mengubah keyframe yang di tengah sehingga merekam ketinggian wood platform ketika berada di puncak gerakannya. Dengan cara pilih keyframe yang kedua (pada timeline ke 1:00).</p> |
| 11 | <p>Setelah memilih keyframe ke 1:00 kemudian ubah nilai Y=0. Dan yang perlu diperhatikan nilai X,Y,Z pada inspector memiliki background merah yang menandakan Transform component sedang direkam pada animation clip.</p> |

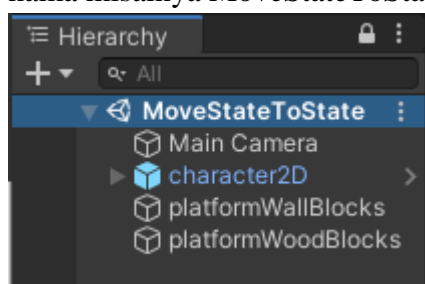


- 12 Pada langkah terakhir simpan dan run scene yang telah dibuat. Anda dapat amati wood platform bergerak perlahan naik turun secara terus menerus.

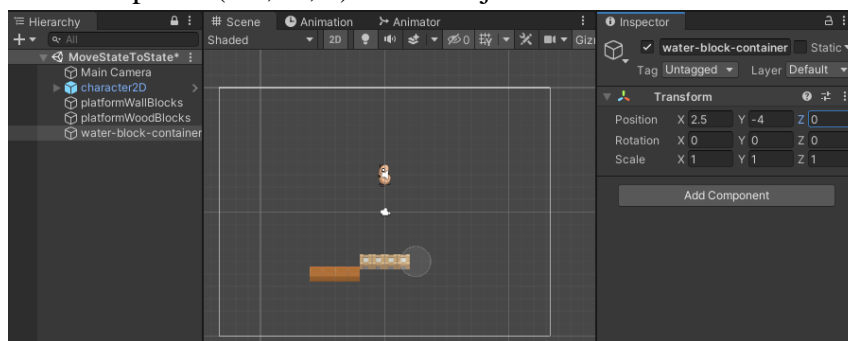


4. Membuat Animasi yang dari Satu State ke State yang lain

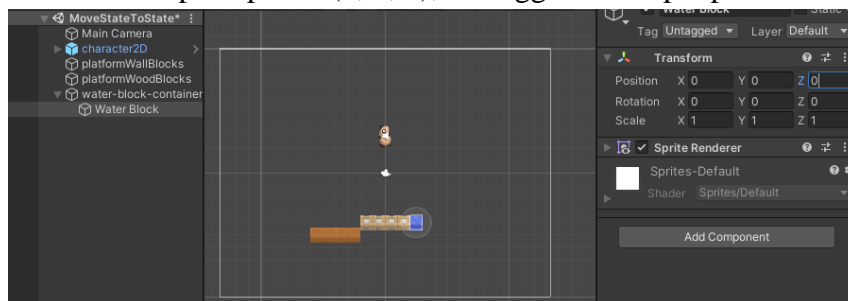
- 1 Melanjutkan praktikum pada sub bab ketiga, langkah pertama buat scene dengan nama yang berbeda dengan scene sebelumnya, dengan cara tekan File | save scene as ... beri nama misalnya MoveStateToState.



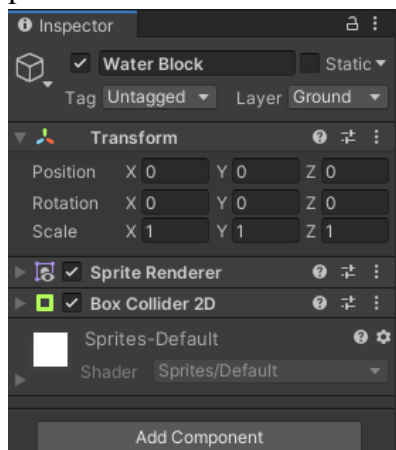
- 2 Pada hierarchy buat Empty GameObject dengan nama water-block-container yang memiliki posisi (2.5, -4, 0). GameObject ini untuk membuat animasi Water Block.



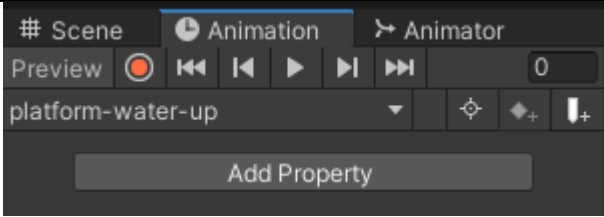
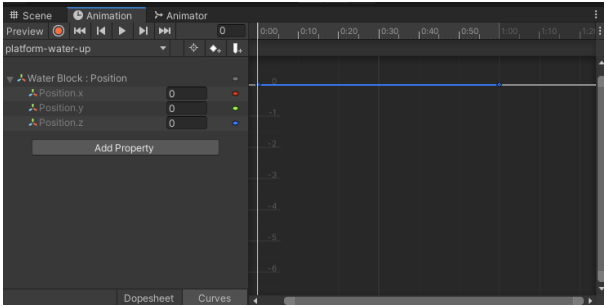
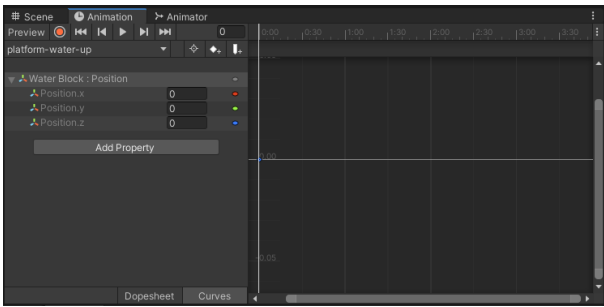
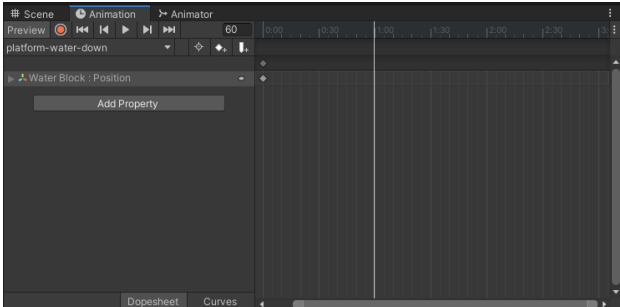
- 3 Drag sprite Water Block dari folder PotatoMan2DAssets | Sprites ke scene dan ditempat sebagai anak/child dari GameObject water-block-container. Pastikan posisi GameObject Water Block pada posisi (0, 0, 0), sehingga menempel pada wall block platform.

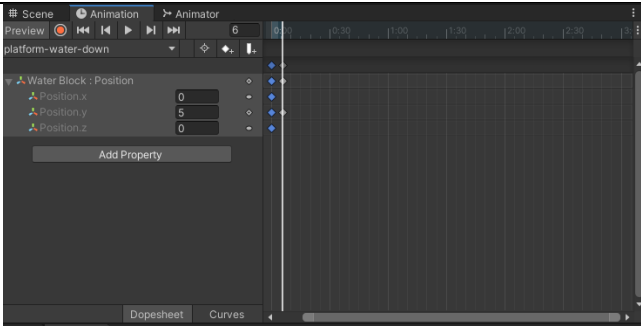
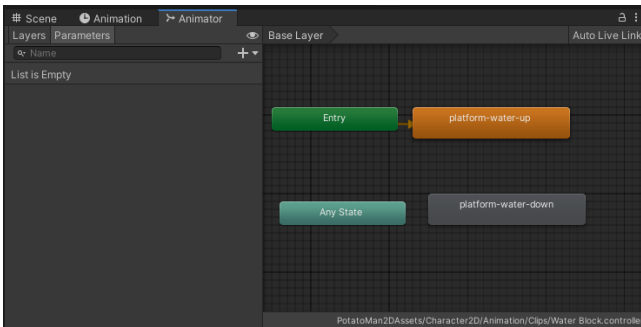
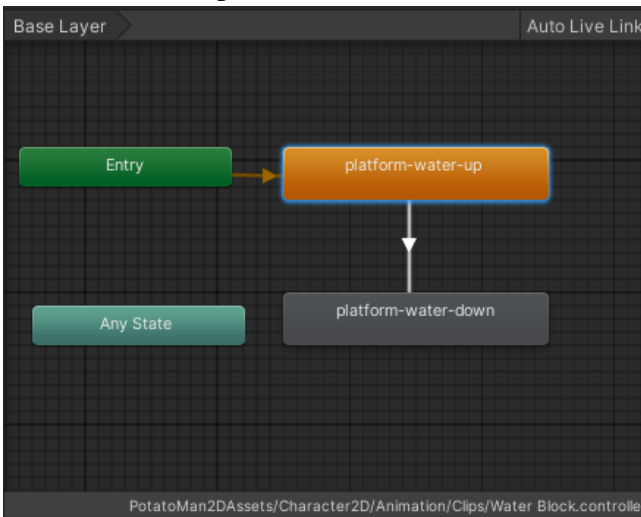


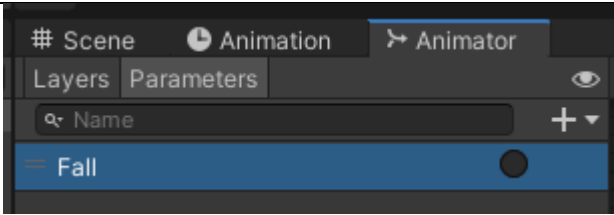
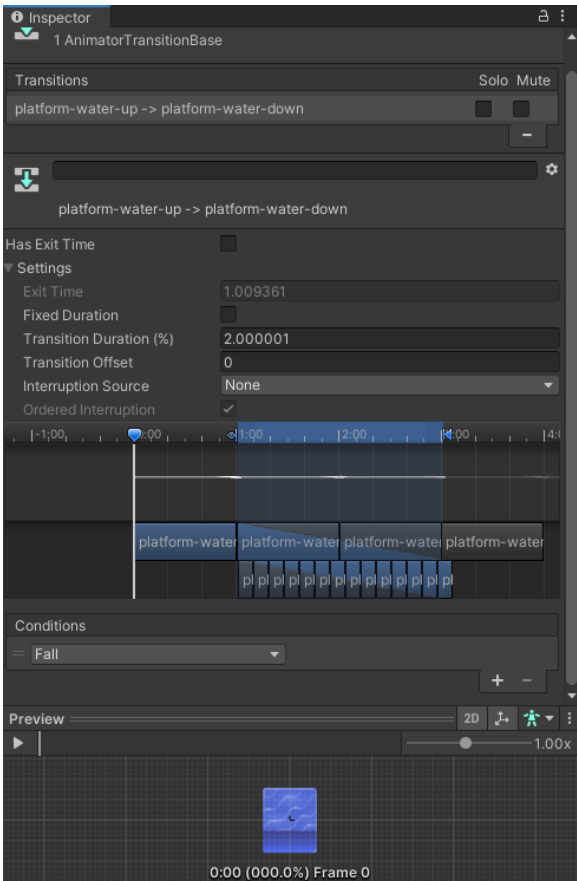
- 4 Tambahkan Box Collider 2D pada GameObject Water Block, dan ubah layer GameObject menjadi Ground, sehingga character2D dapat melompat diatas water block platform.

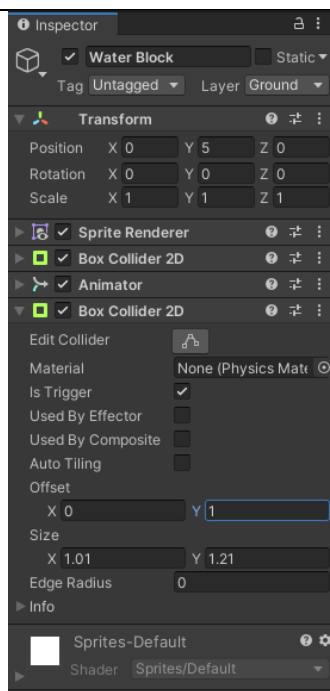


- 5 Pastikan GameObject Water Block dipilih pada Hierarchy, kemudian buka Animation Panel dan buat clip dengan nama named platform-water-up. Simpan pada folder Animations.

| | |
|---|---|
| |  |
| 6 | <p>Klik Curves dan add property kemudian pilih Transform Position.</p>  |
| 7 | <p>Hapus keyframe kedua yang terletak pada timeline 1:00. Pada tahap ini proses membuat animasi water block keatas/naik sudah selesai.</p>  |
| 8 | <p>Buat animation clip kedua dengan nama platform-water-down. Kemudian add property dan pilih Transform Position. Hapus keyframe kedua yang terletak pada timeline ke 1:00.</p>  |
| 9 | <p>Masih pada clip platform-water-down, pilih timeline ke 0:00 kemudian ubah nilai Y=- 5. Pada tahap ini proses membuat animasi water block kebawah/turun sudah selesai maka proses merekam dengan cara menekan record button sekali.</p> |

| | |
|----|--|
| |  |
| 10 | <p>Bukan Animator panel maka anda akan melihat tampilan.</p>  |
| 11 | <p>Walaupun sudah membuat 2 animation clip (state), hanya Up state yang aktif. Hal ini disebabkan ketika scene mulai (Entry state) object akan menuju state platform- water-up, akan tetapi tidak ada transition ke state platform-water-down, sehingga GameObject Water Block akan selalu pada Up state.</p> |
| 12 | <p>Untuk menangani kasus diatas maka perlu dibuat transition ke state platform-water-down dengan cara klik kanan platform-water-up kemudian pilih Make Transition dan arahkan ke state platform-water-down.</p>  |
| 13 | <p>Coba run scene yang telah dibuat, maka setelah 0.9 detik Water Block transition ke Down state. Kita akan merubah hal ini, jika player jalan diatas Water Block maka animasi downward/kebawah aktif. Sehingga akan dibuat Trigger dengan nama Fall, dengan cara memilih Parameter tab pada Animator Panel, kemudian klik button '+' dan pilih Trigger kemudian pilih Fall.</p> |

| | |
|----|--|
| |  |
| 14 | <p>Lakukan langkah berikut untuk menset Trigger:</p> <ol style="list-style-type: none"> Pada Animator panel pilih Transition. Has Exit Time: uncheck. Pada Inspector panel drag Transition end time ke 2:00 seconds (sehingga Water Block akan Transition secara perlahan ke Down state selama 2 detik). Conditions: Add Fall (click plus-sign '+' button at bottom).  |
| 15 | <p>Kita perlu untuk menambahkan collider trigger di atas Water block, dan menambahkan C# script untuk mengirim Animator Controller Trigger ketika player ketika memasuki collider. Pastikan GameObject Water Block dipilih, tambahkan 2D Box Collider yang kedua, dengan Y-Offset=1 dan tick Trigger checkbox.</p> |



- 16 Buat C# script dengan nama WaterBlock seperti berikut kemudian tambahkan ke GameObject Water Block.

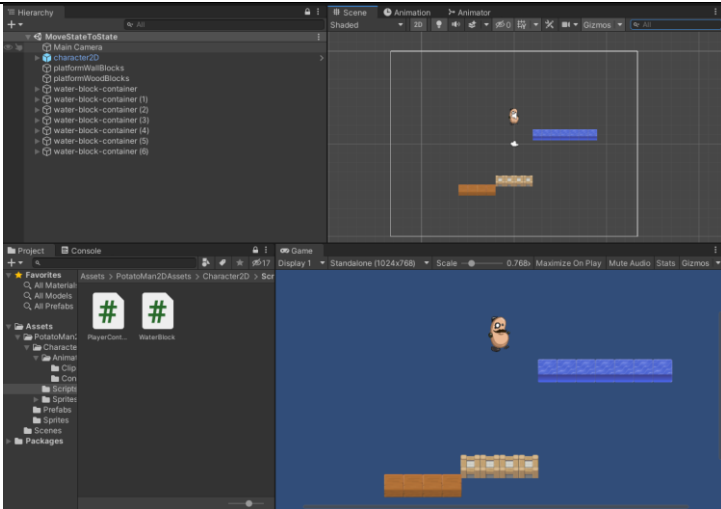
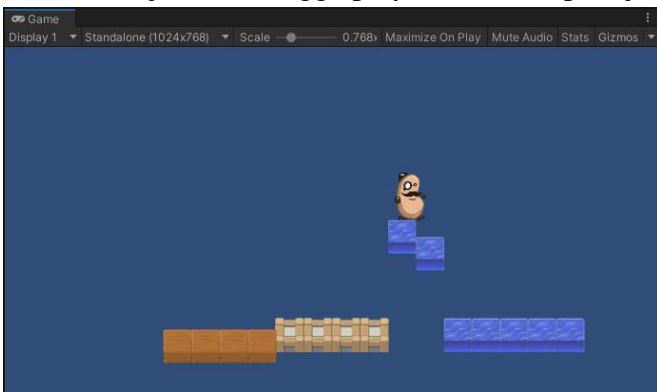
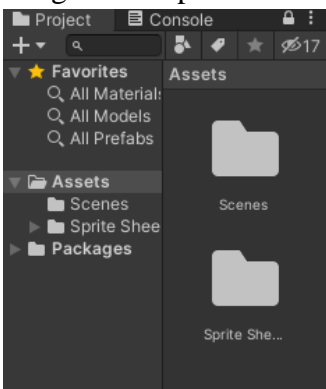
WaterBlock.cs

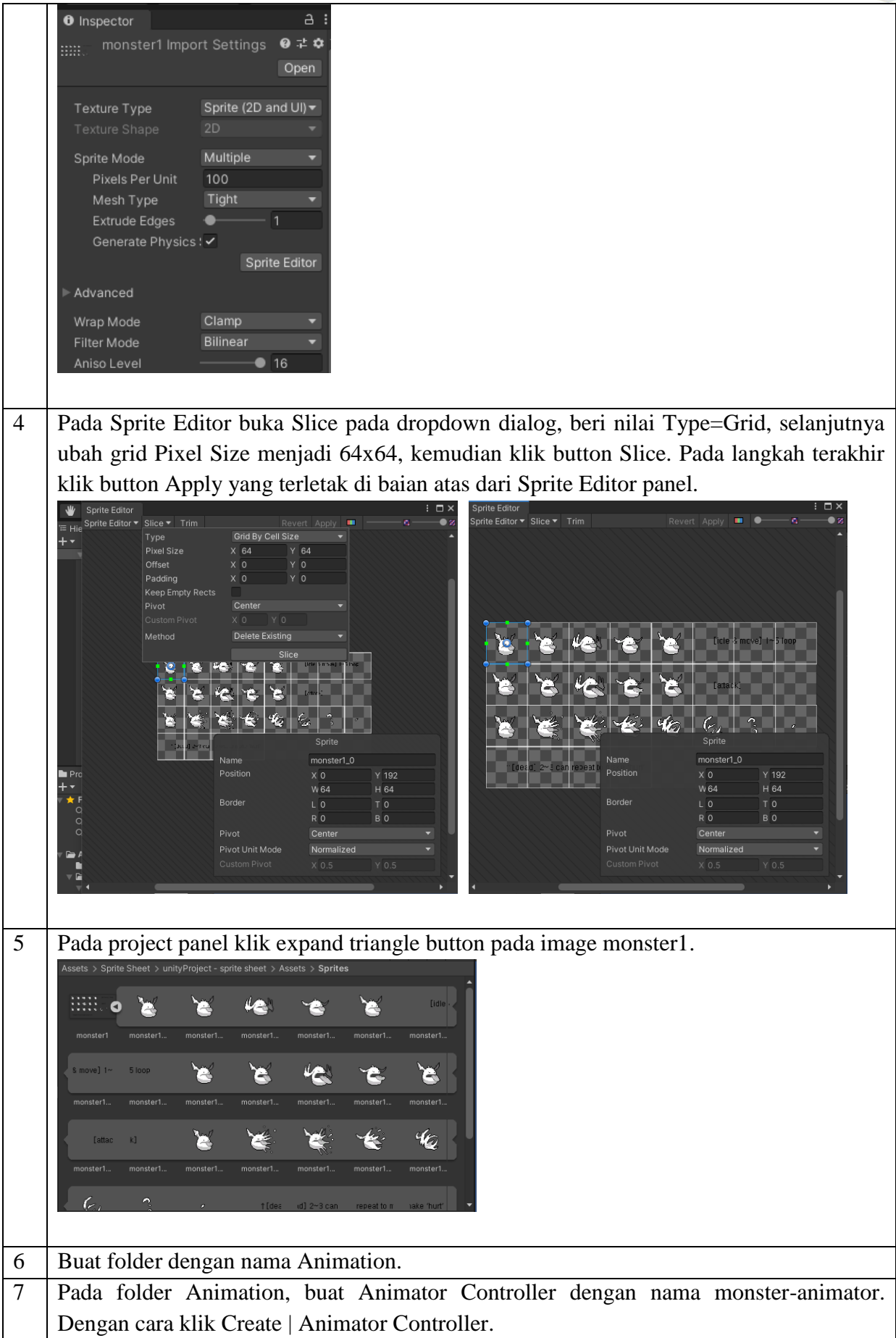
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class WaterBlock : MonoBehaviour
{
    private Animator animatorController;
    // Start is called before the first frame update
    void Start()
    {
        animatorController = GetComponent<Animator>();
    }

    void OnTriggerEnter2D(Collider2D hit)
    {
        if (hit.CompareTag("Player"))
        {
            animatorController.SetTrigger("Fall");
        }
    }
}
```

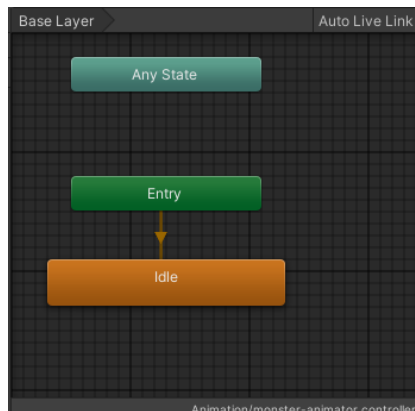
- 17 Copy GameObject water-block-container sebanyak 6 kali, dengan posisi X bertambah 1 untuk setiap GameObject, misalnya 3.5, 4.5, 5.5 dan seterusnya.

| | |
|---|---|
| |  |
| 18 | <p>Run scene yang telah dibuat, dan amati ketika player melewati water block maka water block akan jatuh, sehingga player harus tetap maju ke depan.</p>  |
| Membuat Animasi Clip dari Sprite Sheet Sequences | |
| 1 | Buat project Unity 2D. |
| 2 | <p>Drag folder SpritesheetSequences ke dalam asset.</p>  |
| 3 | <p>Pilih image monster1 pada Project panel, ubah sprite mode menjadi Multiple di inspector, selanjutnya buka Sprite Editor panel dengan cara klik button Sprite Editor.</p> |

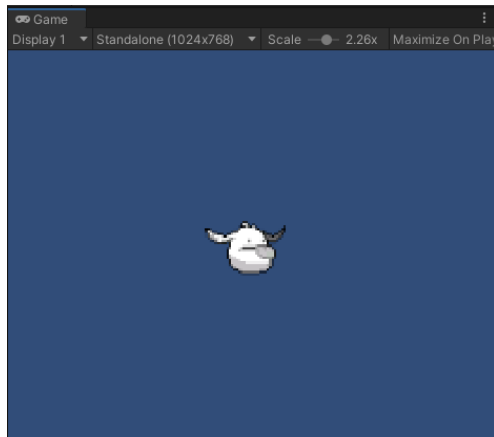


| | |
|----|--|
| | |
| 8 | <p>Pada scene buat Empty GameObject dengan nama monster1 dan set posisi (X, Y, Z) pada posisi (0, 0, 0) dan drag monster-animator pada GameObject.</p> |
| 9 | <p>Pada Hierarchy pilih GameObject monster1, kemudian buka Animation panel (terletak pada Window Animation), dan buat Animation Clip dengan nama Idle (dengan cara klik button Create pada Animation panel).</p> |
| 10 | <p>Pada Project Panel pilih monster1 kemudian drag 5 frame pertama (frame ke 0-4) satu per satu ke Animation Panel. Ubah nilai sample menjadi 12 yang artinya animasi ini digunakan untuk menjalankan 12 frame per second.</p> |

- 11 Amati state-chart maka anda melihat default size dengan nama idle, seperti pada gambar berikut.



- 12 Langkah terakhir run scene yang telah dibuat, maka anda akan melihat GameObject monster1 berjalan pada idle state. Karena sprite yang berukuran kecil maka anda dapat memperkecil ukuran Camera.



KODE PEMROGRAMAN

BugFlip.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BugFlip : MonoBehaviour
{
    private bool facingRight = true;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.LeftArrow) && facingRight)
  
```

```
Flip();
    if (Input.GetKeyDown(KeyCode.RightArrow) && !facingRight)
Flip();
}

void Flip()
{
    // switch the way the player is labelled as facing.
    facingRight = !facingRight;
    // multiply the player's x local scale by -1.
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}
}
```

PlayerControl.cs

```
using UnityEngine;
using System.Collections;

public class PlayerControl : MonoBehaviour
{
    [HideInInspector]
    public bool facingRight = true;           // For
    determining which way the player is currently facing.
    [HideInInspector]
    public bool jump = false;                 // Condition for
    whether the player should jump.

    public float moveForce = 365f;            // Amount of
    force added to move the player left and right.
    public float maxSpeed = 5f;               // The fastest
    the player can travel in the x axis.
    public AudioClip[] jumpClips;             // Array of clips for
    when the player jumps.
    public float jumpForce = 1000f;           // Amount of
    force added when the player jumps.

    private Transform groundCheck;            // A position
    marking where to check if the player is grounded.
    private bool grounded = false;            // Whether or
    not the player is grounded.
    private Animator anim;                    // Reference to
    the player's animator component.

    void Awake()
    {
        // Setting up references.
        groundCheck = transform.Find("groundCheck");
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        // The player is grounded if a linecast to the groundcheck
```

```

position hits anything on the ground layer.
    grounded = Physics2D.Linecast(transform.position,
groundCheck.position, 1 << LayerMask.NameToLayer("Ground"));

    // If the jump button is pressed and the player is
grounded then the player should jump.
    if(Input.GetButtonDown("Jump") && grounded)
        jump = true;
}

void FixedUpdate ()
{
    // Cache the horizontal input.
    float h = Input.GetAxis("Horizontal");

    // The Speed animator parameter is set to the absolute
value of the horizontal input.
    anim.SetFloat("Speed", Mathf.Abs(h));

    // If the player is changing direction (h has a different
sign to velocity.x) or hasn't reached maxSpeed yet...
    if(h * GetComponent<Rigidbody2D>().velocity.x < maxSpeed)
        // ... add a force to the player.
        GetComponent<Rigidbody2D>().AddForce(Vector2.right *
h * moveForce);

    // If the player's horizontal velocity is greater than the
maxSpeed...
    if(Mathf.Abs(GetComponent<Rigidbody2D>().velocity.x) >
maxSpeed)
        // ... set the player's velocity to the maxSpeed in
the x axis.
        GetComponent<Rigidbody2D>().velocity = new
Vector2(Mathf.Sign(GetComponent<Rigidbody2D>().velocity.x) * maxSpeed,
GetComponent<Rigidbody2D>().velocity.y);

    // If the input is moving the player right and the player
is facing left...
    if(h > 0 && !facingRight)
        // ... flip the player.
        Flip();
    // Otherwise if the input is moving the player left and
the player is facing right...
    else if(h < 0 && facingRight)
        // ... flip the player.
        Flip();

    // If the player should jump...
    if(jump)
    {
        // Set the Jump animator trigger parameter.
        anim.SetTrigger("Jump");

        // Add a vertical force to the player.
        GetComponent<Rigidbody2D>().AddForce(new Vector2(0f,
jumpForce));

        // Make sure the player can't jump again until the
jump conditions from Update are satisfied.
        jump = false;
    }
}

```

```

    }

    void Flip ()
    {
        // Switch the way the player is labelled as facing.
        facingRight = !facingRight;

        // Multiply the player's x local scale by -1.
        Vector3 theScale = transform.localScale;
        theScale.x *= -1;
        transform.localScale = theScale;
    }
}

```

WaterBlock.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class WaterBlock : MonoBehaviour
{
    private Animator animatorController;
    // Start is called before the first frame update
    void Start()
    {
        animatorController = GetComponent<Animator>();
    }

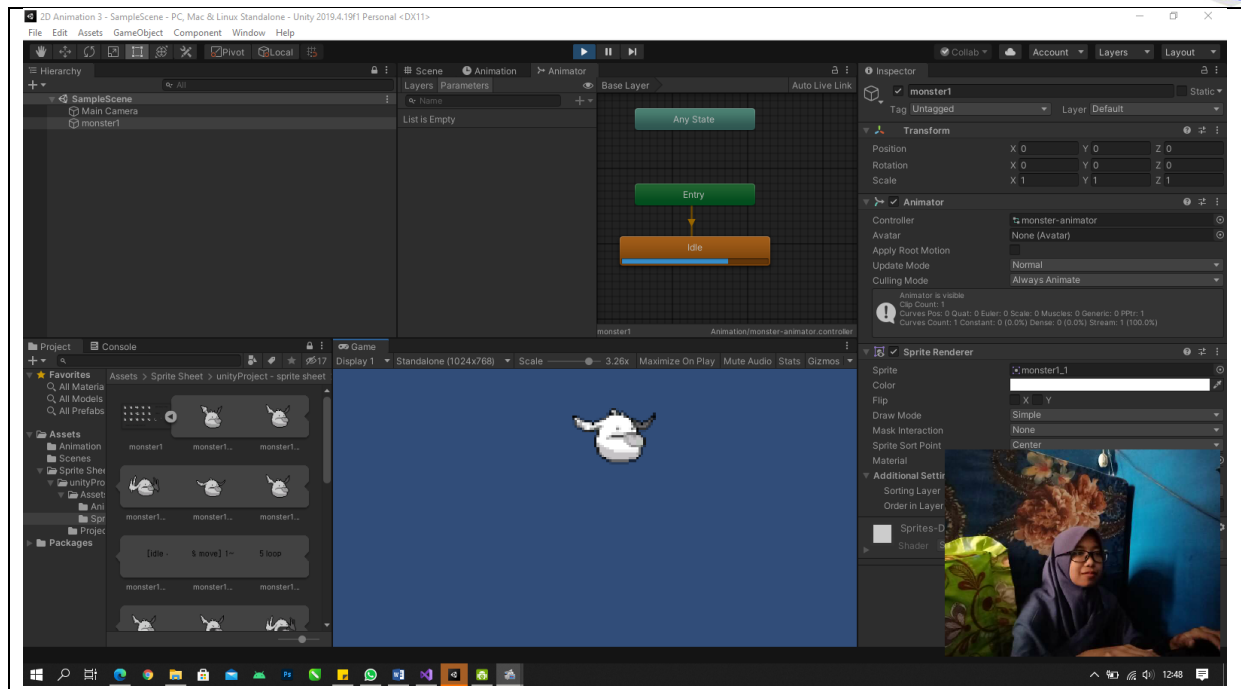
    void OnTriggerEnter2D(Collider2D hit)
    {
        if (hit.CompareTag("Player"))
        {
            animatorController.SetTrigger("Fall");
        }
    }
}

```

KESIMPULAN

Pada jobsheet kali ini, kita banyak mempelajari bagaimana sebuah game object dapat melakukan flip, dapat melakukan lompatan dan memberikan sebuah efek atau animasi pada topi (beanman). Dan pada jobsheet ini juga kita mempelajari bagaimana membuat sebuah animasi pada sebuah game object dengan cara men-slicing kemudian menambahkan gambar per frame.

SWAFOTO MAHASISWA+PROJECT



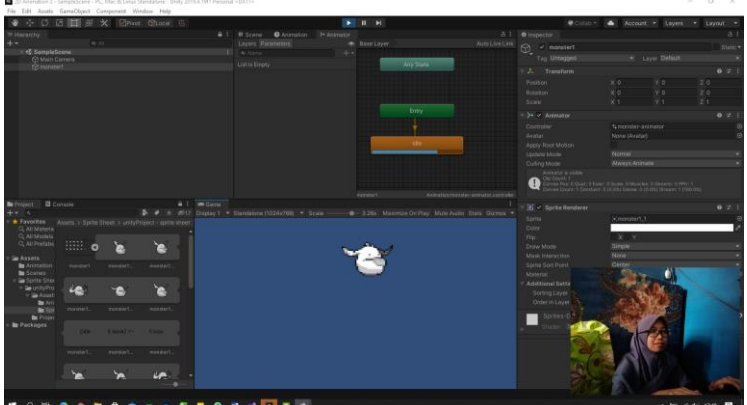
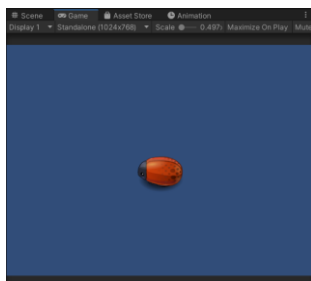
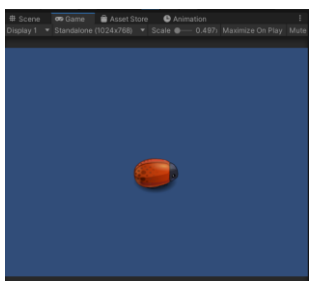
Tugas No.2 :

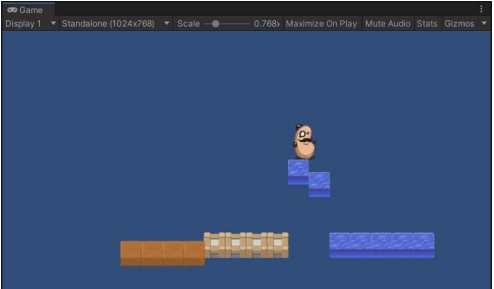
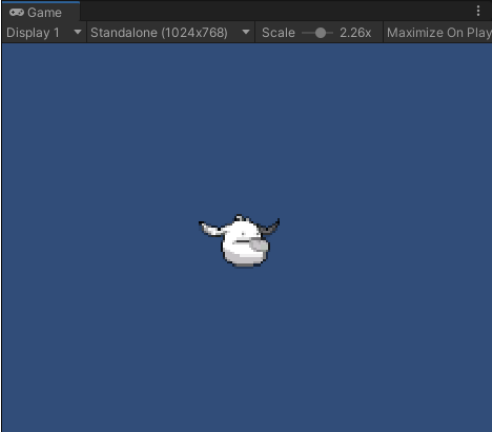

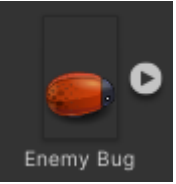
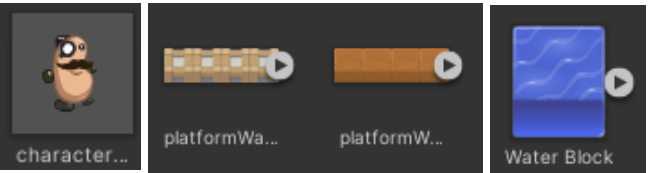
Download game pada link berikut (<https://www.assetstore.unity3d.com/en/#!/content/11228>), pelajari dan berikan review komponen apa saja yang diperlukan untuk membuat game tersebut.



Karena tidak ada, sehingga tidak bisa me-review komponen yang ada pada game yang dimaksud.

Format laporan:

| | | |
|--|---|---|
| NIM | : | 1841720004 |
| Nama | : | Bella Setyowati |
| Kelas | : | TI-3H |
| Swafoto | : |  |
| Deskripsi Game | : | |
| a. Nama Game | : | Game 1: Bug Flip Game 2: Potato Man Game 3: Monster |
| a. Alur Game (print screen dan penjelasan) | : | Game 1: - Membalikkan kumbung dengan cara klik arrow ke kanan atau ke kiri. <div style="display: flex; justify-content: space-around; align-items: center;">   </div> Game 2: |

| | |
|--|---|
| | <ul style="list-style-type: none"> - Lompat ke arah Water Block dan tetap maju agar tidak jatuh.  <p>Game 3:</p> <ul style="list-style-type: none"> - Meng-animasi-kan object secara terus-menerus.  |
| <p>b. Komponen materi yang dipakai (disertakan screenshoot)</p> | <p>: Game Object – Empty</p>  |
| <p>c. Asset yang dipakai (sertakan screenshoot, jika terdapat asset yang digunakan berasal dari internet atau sumber lain, cantumkan link)</p> | <p>: Semua assets yang dipakai berasal dari assets yang sudah disediakan oleh dosen pembimbing.</p> <p>Game 1:</p>  <p>Game 2:</p>  <p>Game 3:</p> |

