

## MODUL 4 – 2D Animation

---

### A. TUJUAN

- Mahasiswa dapat membuat animasi dengan menggunakan objek 2D.
- Mahasiswa dapat membuat animasi pada bagian tertentu pada GameObject
- Mahasiswa dapat memahami fungsi state pada pembuatan game.
- Mahasiswa dapat memahami cara membuat dan menggunakan trigger dalam pembuatan game
- Mahasiswa dapat membuat animasi dari sprite sheet sequence.

### B. PETUNJUK

1. Awali setiap kegiatan praktikum dengan berdoa
2. Baca dan pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik
3. Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur
4. Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas

### C. ALOKASI WAKTU: 6 jam pelajaran

### D. DASAR TEORI

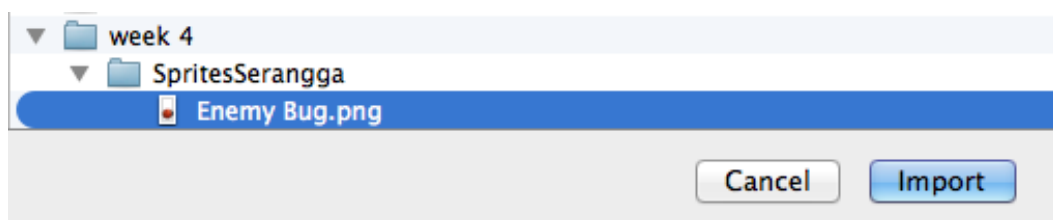
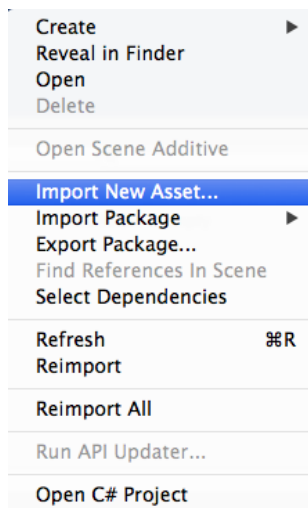
Animasi adalah hal yang sangat penting dalam pembuatan game. Karena Game tidak dapat dibuat tanpa animasi. Pada minggu ke 4 akan mempelajari beberapa hal untuk membuat animasi, antara lain:

- State adalah langkah-langkah atau alur untuk menyelesaikan permasalahan. Sedangkan untuk pembuatan animasi pada game, state digunakan untuk mengatur alur animasi.
- Trigger juga tidak kalah penting untuk membuat animasi pada game sehingga game yang dibuat lebih menarik. Pada modul ini akan dibuat animasi yang juga terdapat trigger pada project tersebut, contohnya untuk komponen yang jatuh ketika diinjak oleh Player.

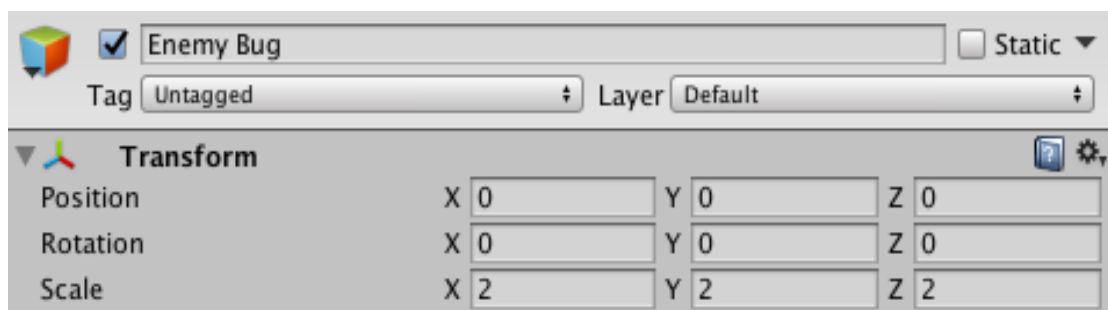
### E. LATIHAN PRAKTIKUM

#### 1. Flipping Sprite secara Horizontal

1. Buat project unity 2D.
2. Import gambar EnemyBug.png yang ada di folder SpritesSerangga kedalam project anda.



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.

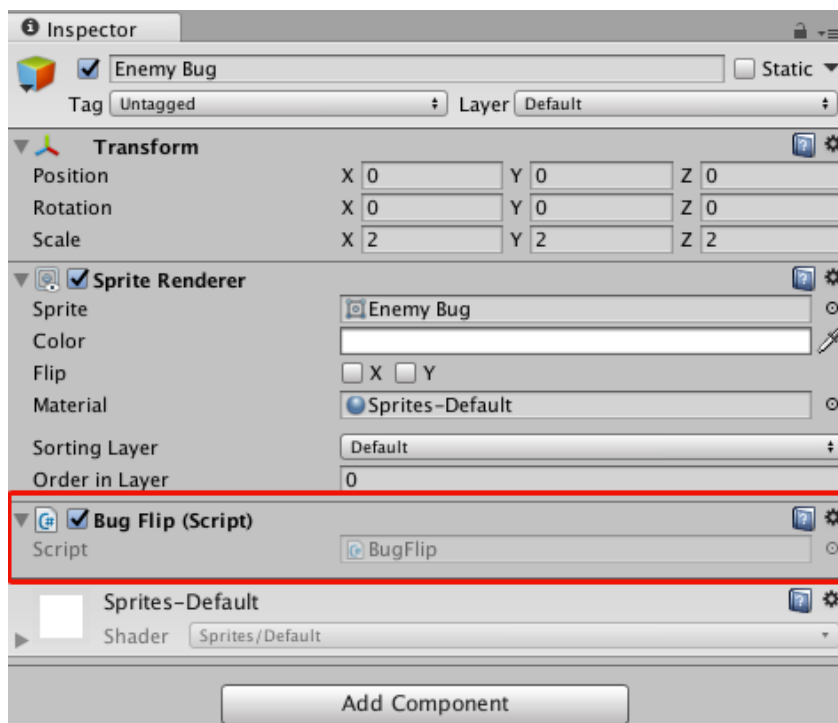
```
using System.Collections;
using UnityEngine;

public class BugFlip : MonoBehaviour {
    private bool facingRight = true;

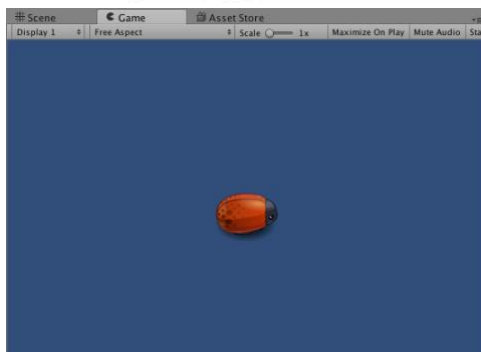
    // Update is called once per frame
    void Update() {
        if (Input.GetKeyDown(KeyCode.LeftArrow) && facingRight)
            Flip ();
        if (Input.GetKeyDown(KeyCode.RightArrow) && !facingRight)
            Flip();
    }

    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;
    }
}
```

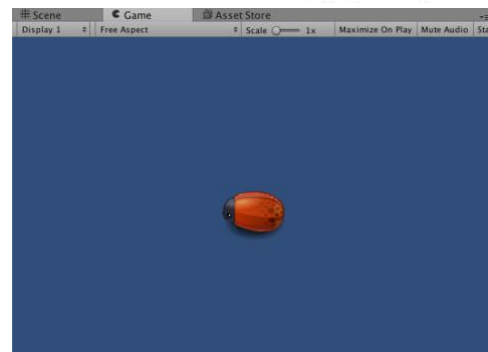
Jika script C# berhasil ditambah pada GameObject Enemy Bug hasilnya adalah seperti berikut:



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



Hasil ketika di tekan button arah ke kanan



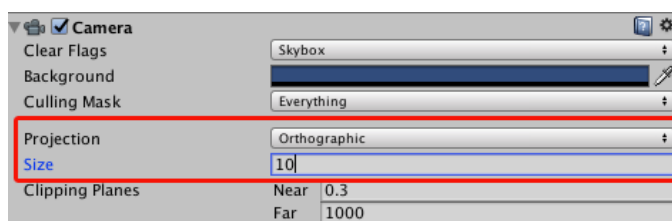
Hasil ketika di tekan button arah ke kiri

6. Langkah terakhir simpan scene yang telah dibuat.

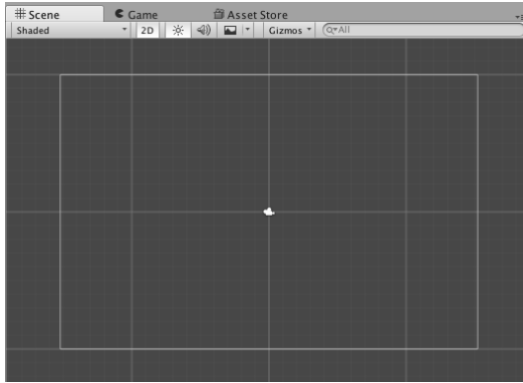
## 2. Membuat animasi bagian tubuh ketika karakter bergerak

Pada praktikum kedua ini akan membuat animasi bagian tubuh ketika karakter bergerak. Langkah-langkahnya adalah sebagai berikut :

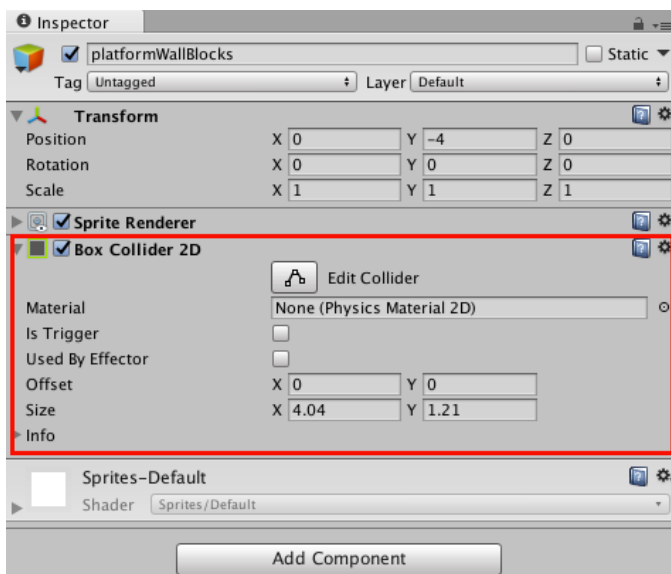
1. Buat project unity 2D.
2. Import asset yang ada di folder PotatoMan2DAssets.
3. Tambah size **Main Camera** menjadi 10.



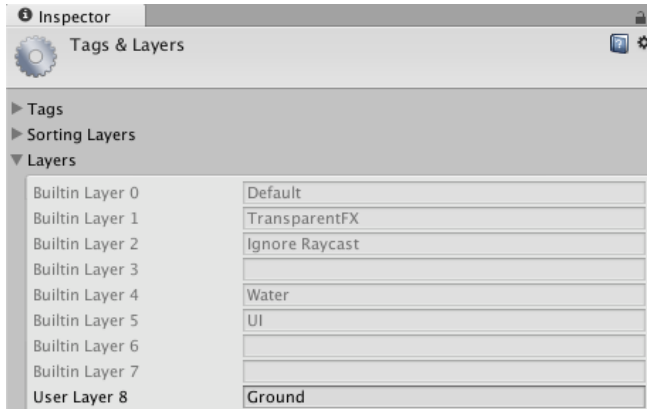
Dampaknya dapat dilihat scenenya menjadi lebih besar, seperti gambar berikut.



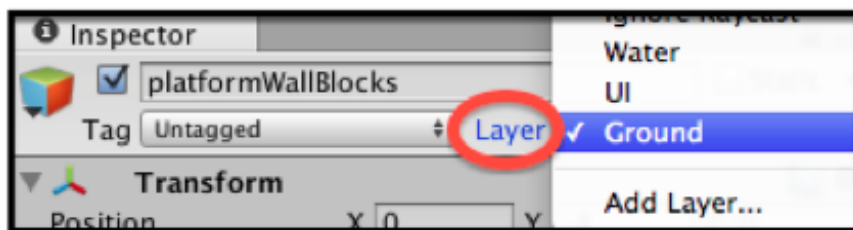
4. Mengatur 2D gravity dengan cara pilih **Edit | Project Settings | Physics 2D**, kemudian ubah nilai Y pada bagian paling atas menjadi -30.
5. Drag **character2D** dari folder **Prefabs** ke scene. Atur position (x,y,z) GameObject menjadi (0, 3, 0).
6. Drag sprite platformWallBlocks dari folder **Project | Sprites** ke scene. Atur position GameObject menjadi (0, -4, 0).
7. Tambahkan Box Collider 2D component pada GameObject platformWallBlocks dengan cara pilih **Add Component | Physics 2D | Box Collider 2D**.



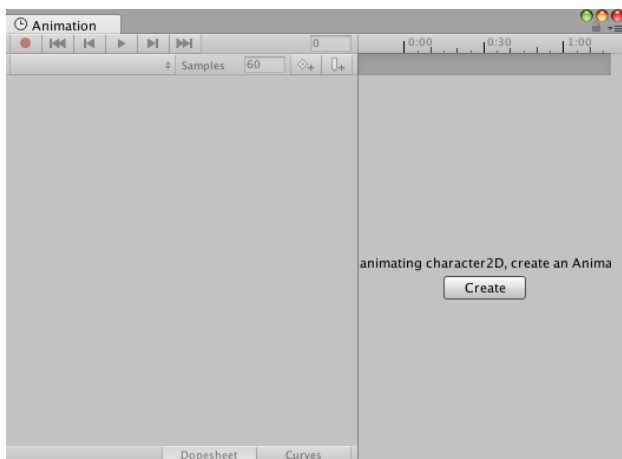
8. Ubah layer GameObject platformWallBlocks dengan cara pilih **Add layer** kemudian ketik **Ground**.



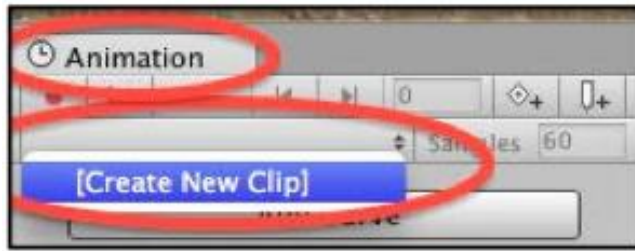
Pilih kembali platformWallBlocks pada hierarchy kemudian pilih layer Ground seperti pada gambar di bawah ini.



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**.

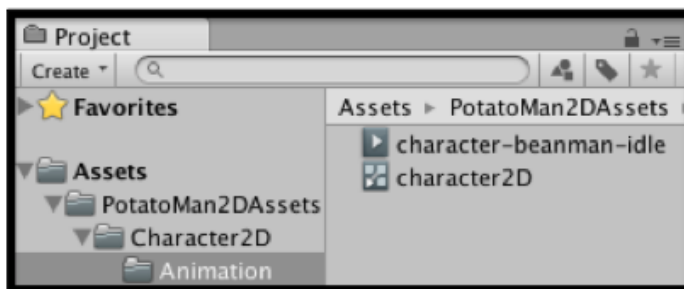


10. Klik dropdown menu pada panel Animation kemudian pilih menu item [Create New Clip].



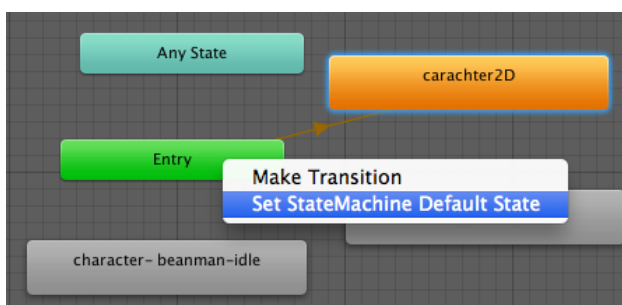
11. Simpan new clip di folder **Character2D | Animation**, beri nama character- beanman- idle. Anda pada tahap ini telah membuat clip Animation untuk 'idle' character state (tidak bergerak).
12. Amati pada folder **Character2D | Animation** yang ada di Project panel , seharusnya dapat dilihat terdapat 2 file yaitu Animation clip yang telah dibuat dengan nama (character-beanman-idle) dan Animator controller dari GameObject character2D dengan nama character2D.

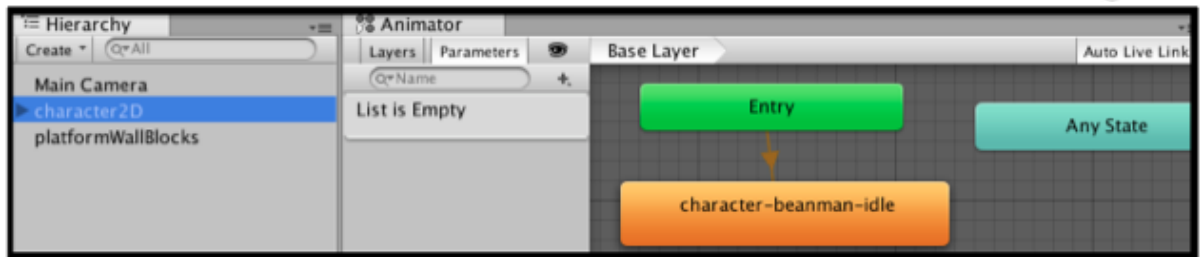
Note : Jika ada clip character2D maka delete saja karena tidak memerlukan file tersebut.



13. Pilih GameObject character2D di Hierarchy, kemudian buka Animator panel (**Window | 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.

Dengan cara klik kanan pada Entry kemudian pilih Set StateMachine Default State arahkan ke clip character-beanman-idle.

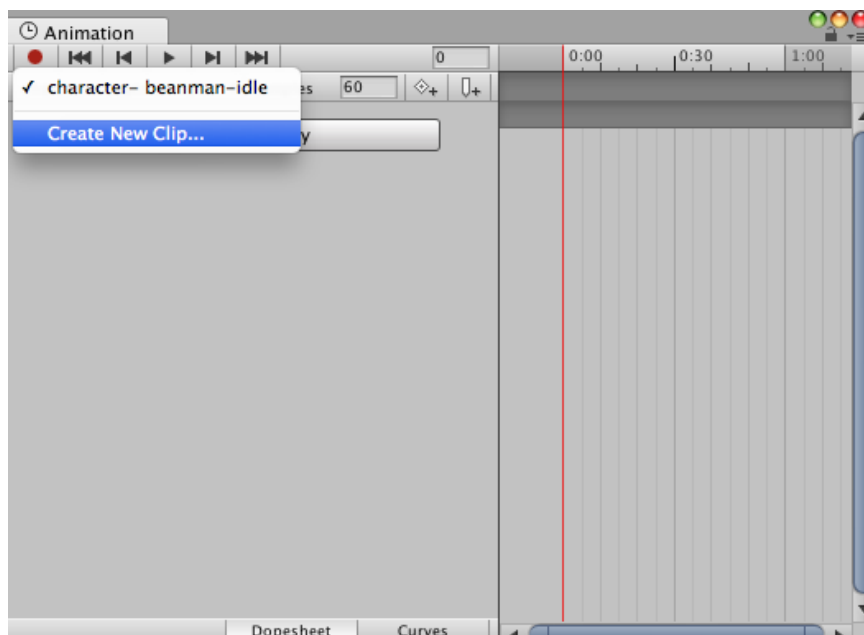




14. Play scene – maka anda akan melihat character selalu pada 'idle' state, untuk sementara ini anda tidak dapat menggerakkan character tersebut sampai membuat clip yang lain.

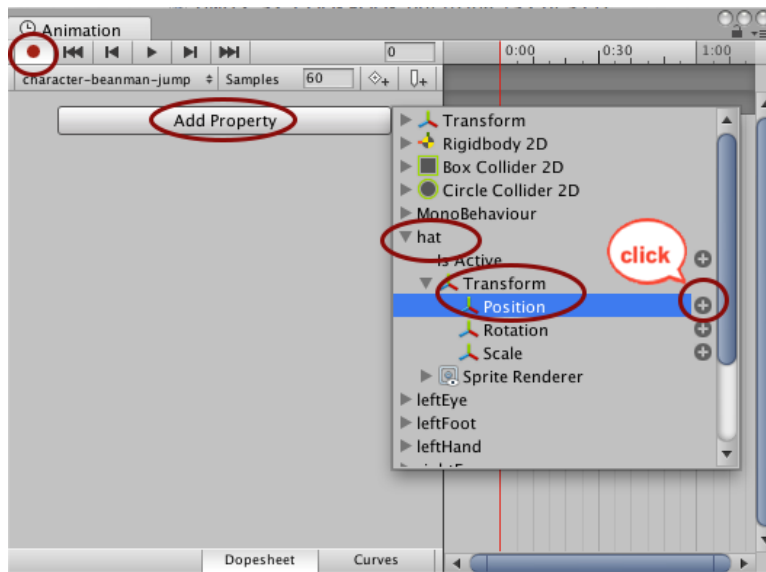
Note: karakter bergerak turun disebabkan gravity bernilai 1 pada Rigidbody 2D. Dan character berhenti pada platformWoodBlocks karena sama-sama memiliki Box Collider 2D.

15. 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 **character-beanman-jump** dan simpan pada folder Animation folder.

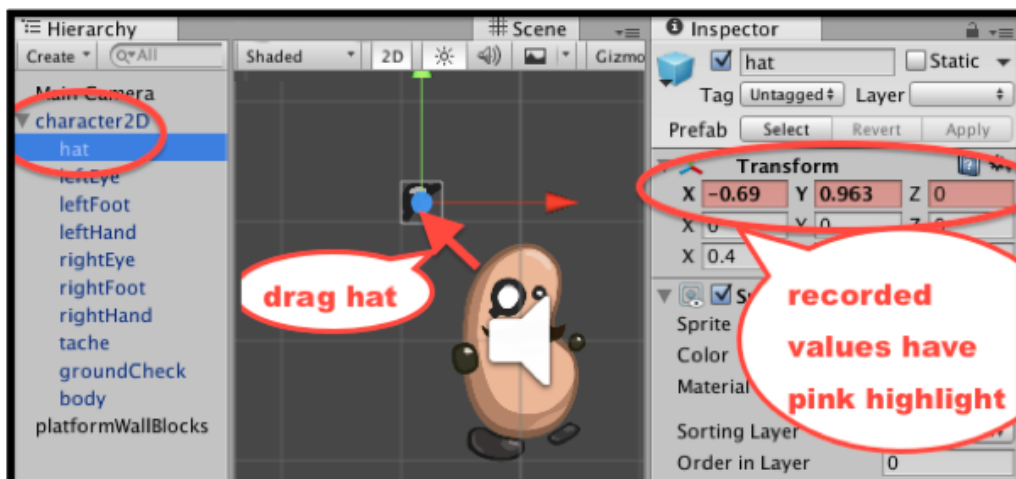


16. Klik button Add Property, dan pilih **Transform | Position** yang merupakan hat child object, dengan cara klik '+' plus-sign button. Pada tahap ini kita melakukan recording posisi (X, Y, Z) pada GameObject hat pada clip animation.

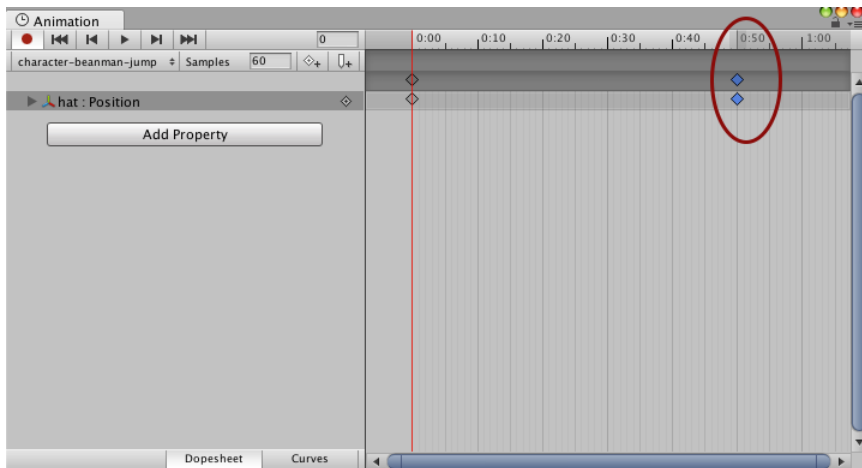




17. Dapat anda amati terdapat 2 'keyframes' pada 0.0 dan at 1.0. Hal ini direpresentasikan dengan diamonds pada Timeline area di sisi sebelah kanan Animation panel.
18. Pilih keyframe pertama pada waktu (0,0) kemudian pada **Scene panel** ubah posisi hat ke atas dan ke kiri, menjauhi kepala. Amati nilai X,Y,Z pada inspector memiliki background merah hal ini menunjukkan bahwa perubahab pada Transform component di record / rekam pada animation clip.



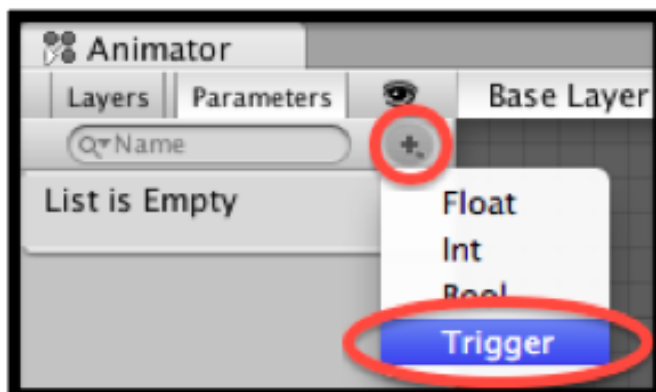
19. Karena 1 detik terlalu lama untuk jump animation, drag keyframe kedua ke kiri pada waktu ke 0.5.



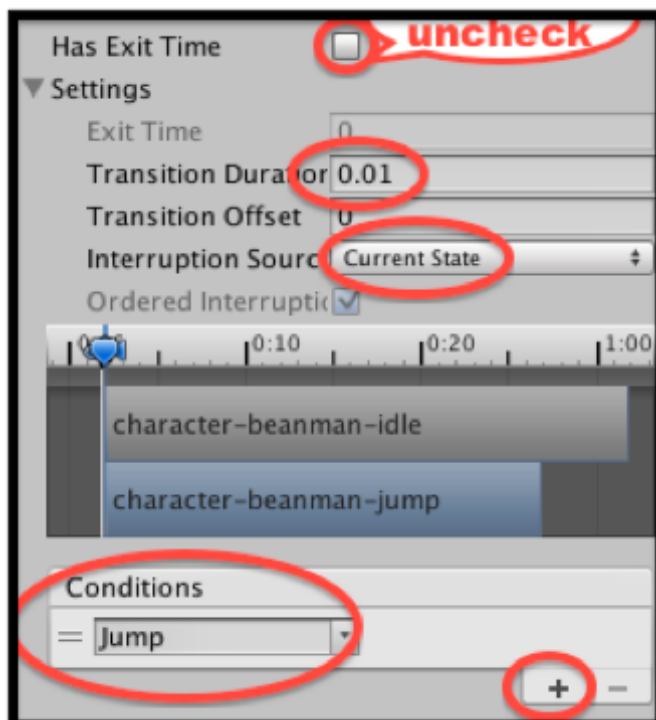
20. Pada langkah selanjutnya mengatur transisi dari 'idle' state ke 'jump' state. Pada Animator panel pilih character-beanman-idle selanjutnya buat transisi ke state character-beanman-jump. Dengan cara klik kanan dan pilih menu Make Transition, kemudian drag transition arrow ke state character- beanman-jump.



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. 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:
- Has Exit Time: uncheck
  - Transition Duration: 0.01
  - Interruption State: Current State
  - Conditions: Add Jump (click plus-sign '+' button at bottom)



24. 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.
25. Di Animator panel pilih state character-beanman-jump dan tambahkan Transition untuk kembali ke state character-beanman-idle. Pilih Transition arrow pada Inspector panel ubah propertiesnya seperti berikut
- Has Exit Time: check
  - Exit time :0.5 (nilai ini harus sama dengan nilai pada second keyframe pada clip Jump animation)
  - Transition Duration: 0.01
  - Interruption State: Current State

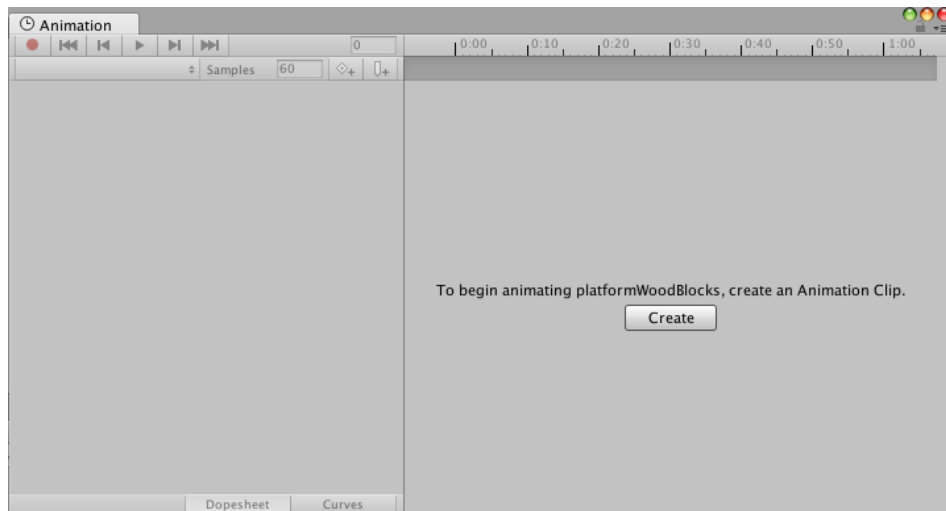
26. Simpan dan run scene. Sekarang ketika lompat topinya hanya menjauh dari kepalanya sekali, setelah itu character kembali pada Idle state.

### 3. Membuat 3-Frame Animation Clip Sehingga Animasi Bergerak Secara Terus Menerus

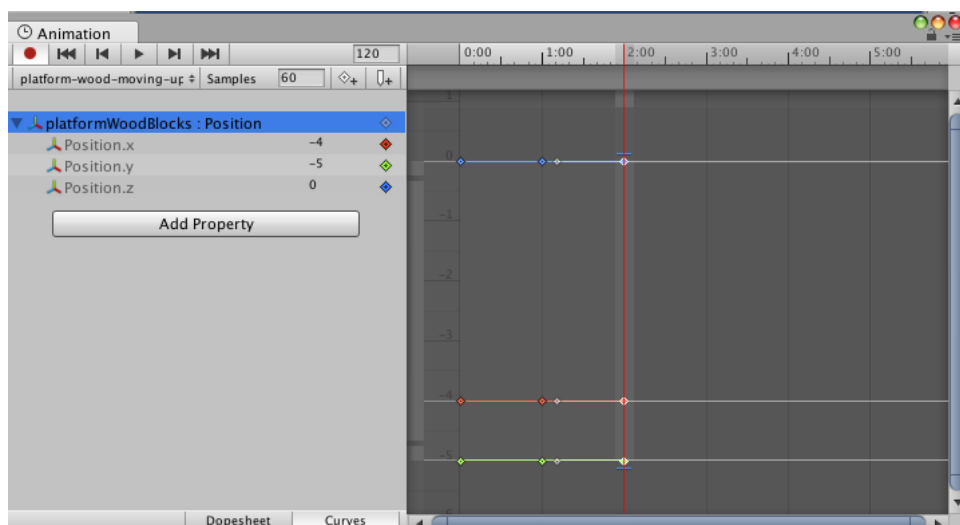
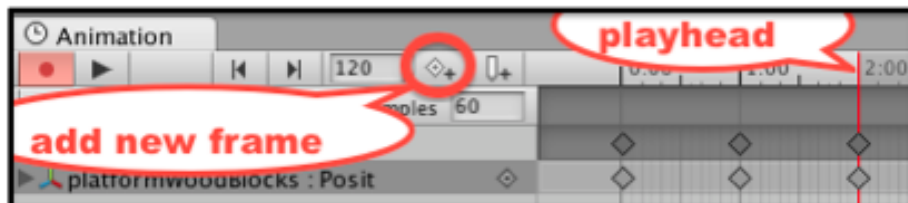
1. Melanjutkan praktikum pada sub bab kedua, langkah pertama buat scene dengan nama yang berbeda dengan scene sebelumnya, dengan cara tekan **File | save scene as ...** beri nama misalnya 3FrameAnimation.
2. Drag sprite platformWoodBlocks dari folder **Project | Sprites** ke dalam scene. Ubah posisi GameObject menjadi (-4, -5, 0).



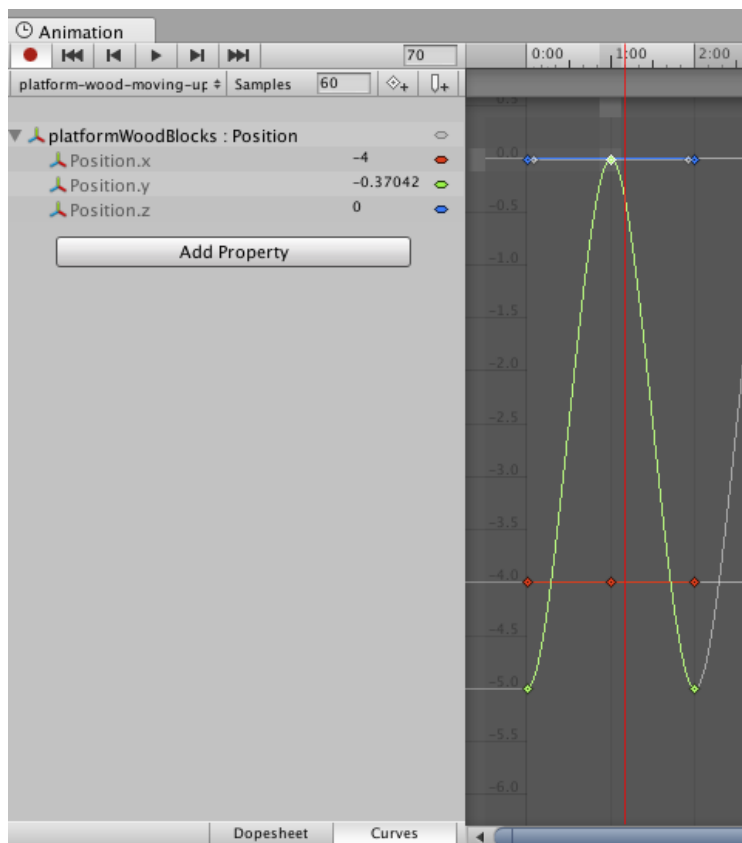
3. Tambahkan Box Collider 2D component pada GameObject platformWoodBlocks sehingga character2D juga dapat berdiri pada GameObject ini. Dengan cara **Add Component | Physics 2D | Box Collider 2D**.
4. Buat folder baru dengan nama Animations yang akan digunakan untuk menyimpan animation clip dan controller.
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. 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.
8. Anda akan melihat 2 'keyframes' pada posisi 0.0 dan 1.0. Keyframe direpresentasikan oleh diamonds.
9. 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



10. 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).
11. 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.

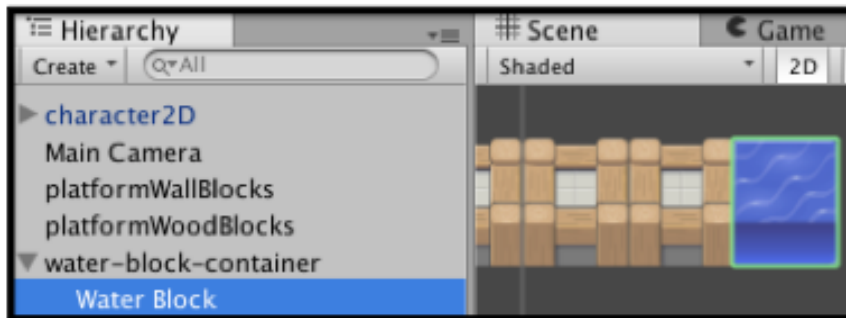


12. Pada langkah terakhir simpan dan run scene yang telah dibuat. Anda dapat amati wood platform bergerak perlahan naik turun secara terus menerus.

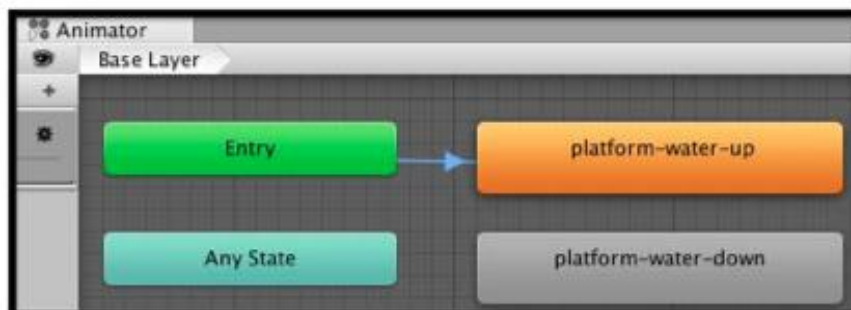
#### 4. Membuat Animasi yang dari Satu State ke State yang lain

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, seperti screenshot berikut:



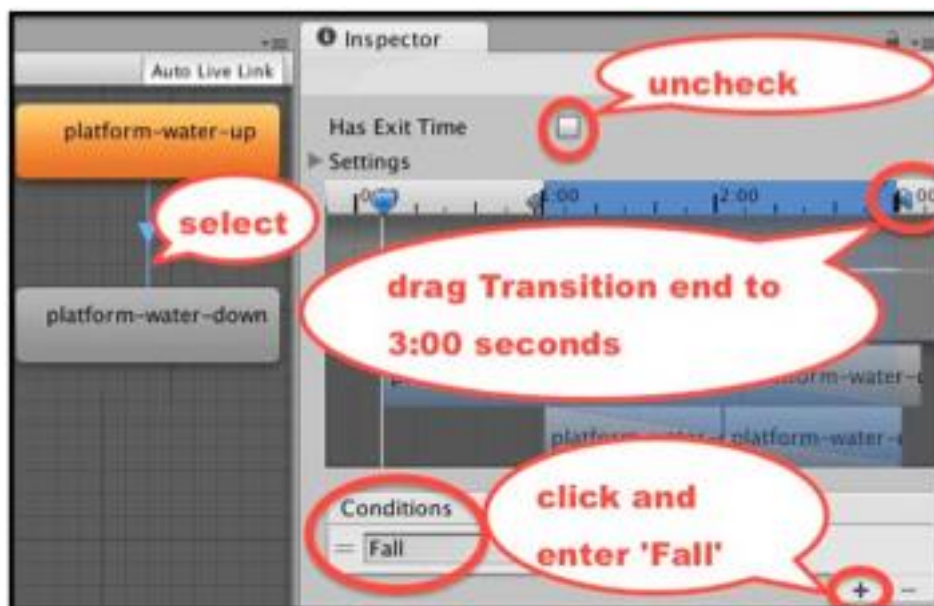
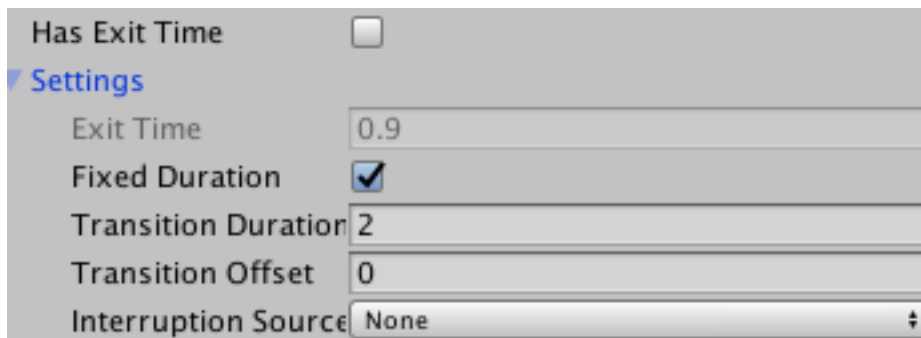
4. Tambahkan Box Collider 2D pada GameObject Water Block, dan ubah layer GameObject menjadi Ground, sehingga character2D dapat melompat diatas water block platform.
5. Pastikan GameObject Water Block dipilih pada Hierarchy, kemudian buka Animation Panel dan buat clip dengan nama **named platform-water-up**. Simpan pada folder Animations.
6. Klik Curves dan add property kemudian pilih Transform | Position.
7. Hapus keyframe kedua yang terletak pada timeline 1:00. Pada tahap ini proses membuat animasi water block keatas/naik sudah selesai.
8. 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.
9. 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.
10. Bukan Animator panel maka anda akan melihat tampilan seperti berikut:



11. 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-**

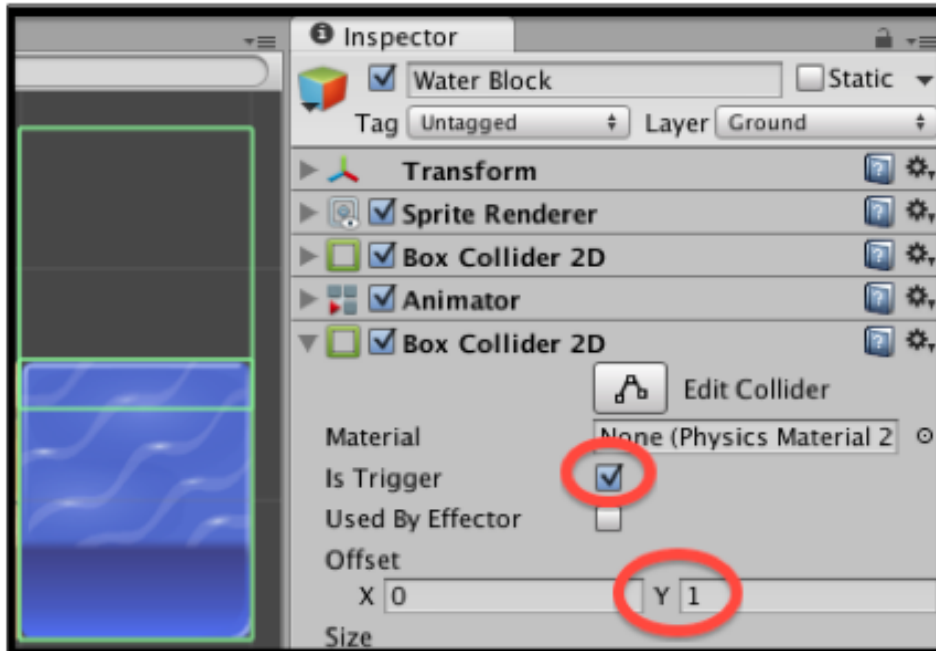
**water-up**, akan tetapi tidak ada transition ke state **platform-water-down**, sehingga GameObject Water Block akan selalu pada Up state.

12. 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.
13. 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.
14. Lakukan langkah berikut untuk menset Trigger:
  - a. Pada Animator panel pilih Transition
  - b. Has Exit Time: uncheck
  - c. Pada Inspector panel drag Transition end time ke 2:00 seconds (sehingga Water Block akan Transition secara perlahan ke Down state selama 2 detik)
  - d. Conditions: Add Fall (click plus-sign '+' button at bottom)





15. 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.



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

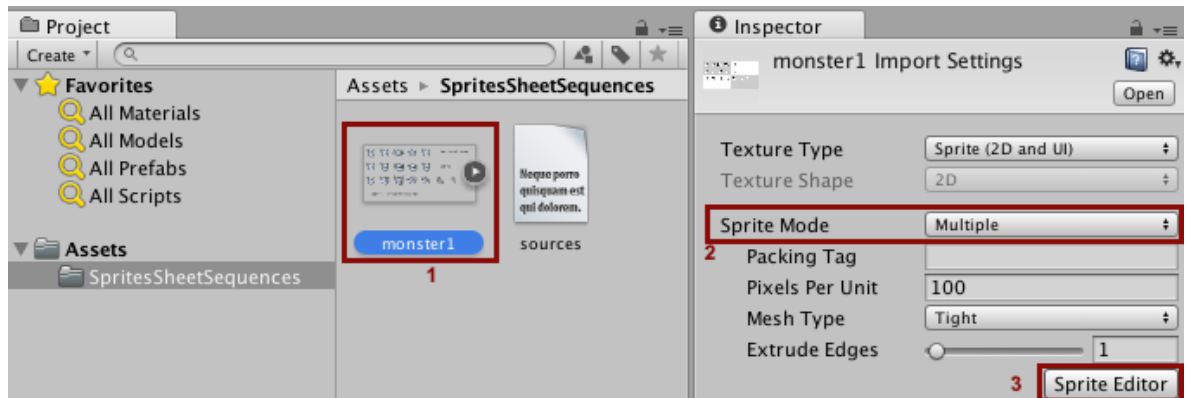
```
using UnityEngine;
using System.Collections;

public class WaterBlock : MonoBehaviour {
    private Animator animatorController;
    void Start(){
        animatorController = GetComponent<Animator>();
    }
    void OnTriggerEnter2D(Collider2D hit){
        if(hit.CompareTag("Player")){
            animatorController.SetTrigger("Fall");
        }
    }
}
```

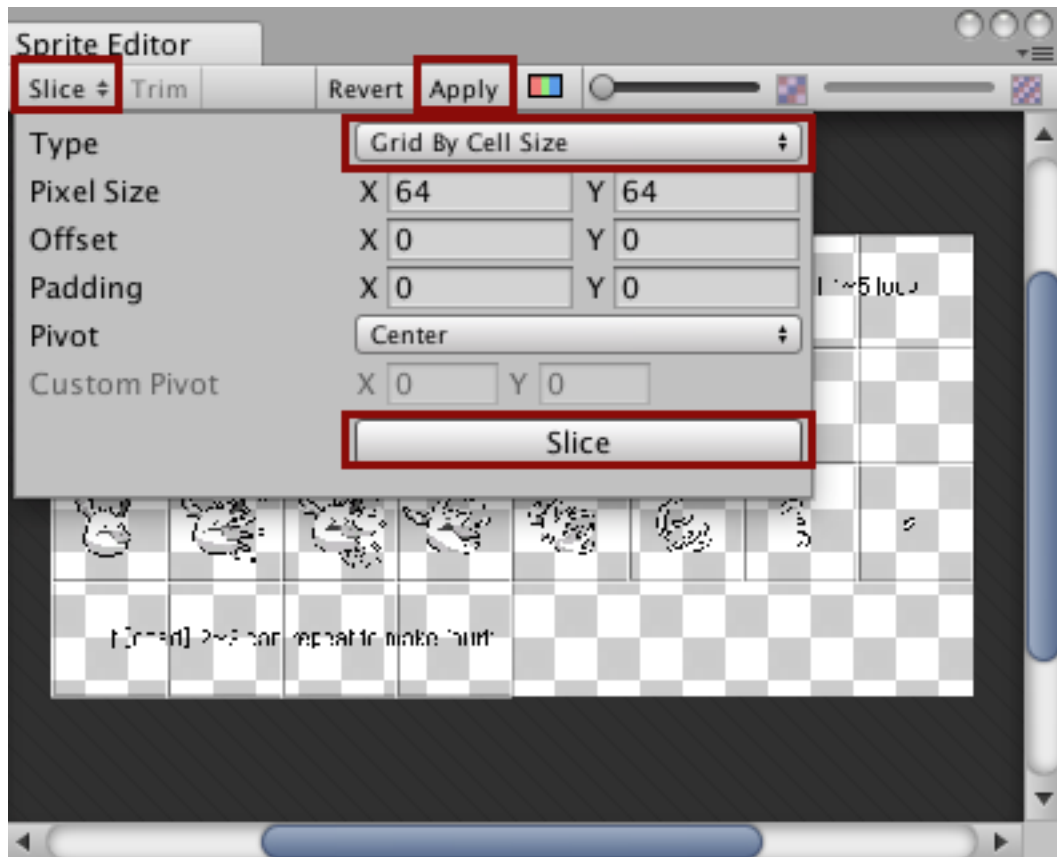
17. Copy GameObject water-block-container sebanyak 6 kali, dengan posisi X bertambah 1 untuk setiap GameObject, misalnya 3.5, 4.5, 5.5 dan seterusnya.
18. Run scene yang telah dibuat, dan amati ketika player melewati water block maka water block akan jatuh, sehingga player harus tetap maju ke depan.

## 5. Membuat Animasi Clip dari Sprite Sheet Sequences

1. Buat project Unity 2D
2. Drag folder SpritesSheetSequences ke dalam asset.
3. Pilih image monster1 pada Project panel, ubah sprite mode menjadi Multiple di inspector, selanjutnya buka Sprite Editor panel dengan cara klik button Sprite Editor.



4. Pada **Sprite Editor** buka **Slice** pada dropdown dialog, beri nilai Type=**Grid**, selanjutnya ubah grid **Pixel Size** menjadi 64x64, kemudian klik button Slice. Pada langkah terakhir klik button **Apply** yang terletak di bagian atas dari **Sprite Editor** panel.



5. Pada project panel klik expand triangle button pada image monster1, seperti pada gambar berikut:



6. Buat folder dengan nama Animation.
7. Pada folder Animation, buat **Animator Controller** dengan nama **monster-animator**. Dengan cara klik Create | Animator Controller.
8. 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.

9. 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).
10. 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.



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.

#### A. TUGAS PRAKTIKUM

1. Buat laporan hasil praktikum week4.
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.



--- SELAMAT BELAJAR ---