



Final Project Report

Combat Game

Group 2



Jobs

Student ID	Name	Jobs
109703006	王世揚	遊戲設計、上台報告
109703010	李承恩	動作偵測、簡報製作
109703013	周彥綸	動作錄製、深度辨識
109703014	賴光禹	遊戲設計、簡報美化

Tools



Kinect SDK 2.0

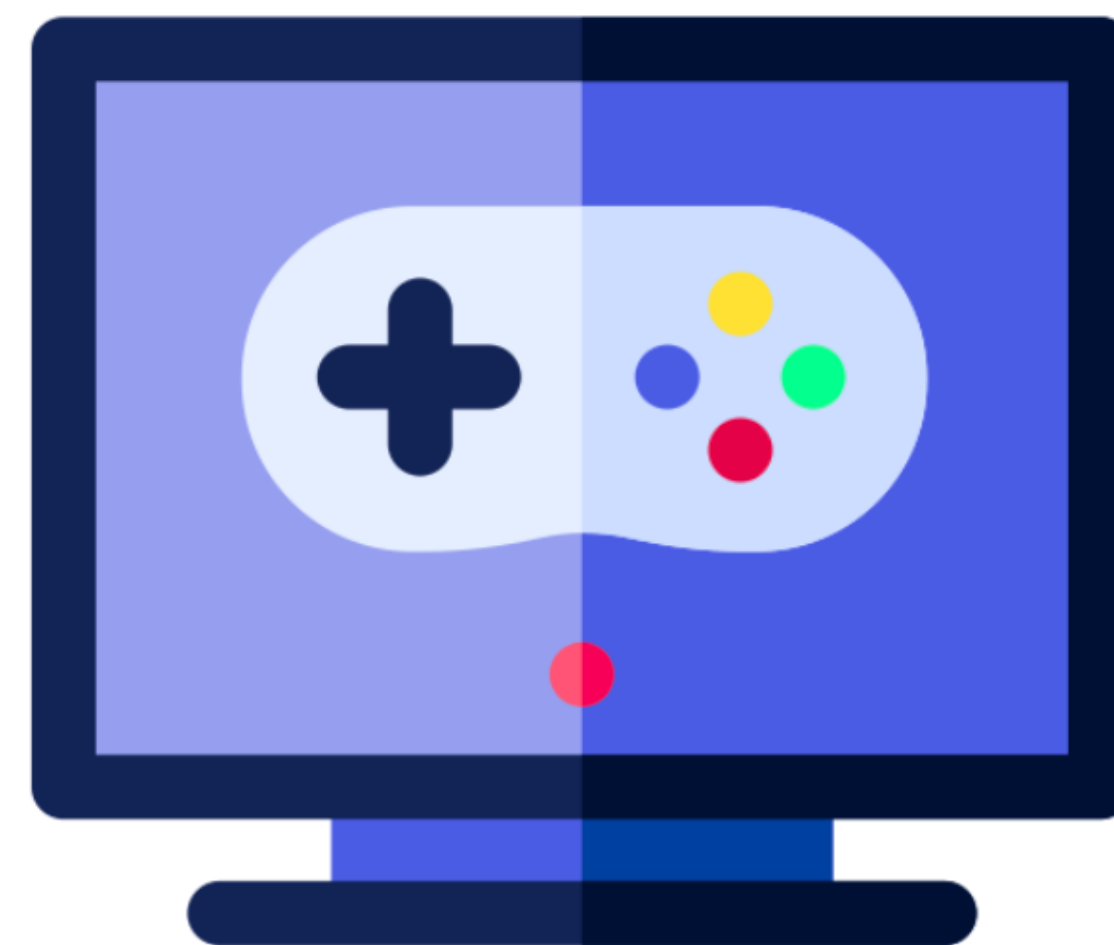
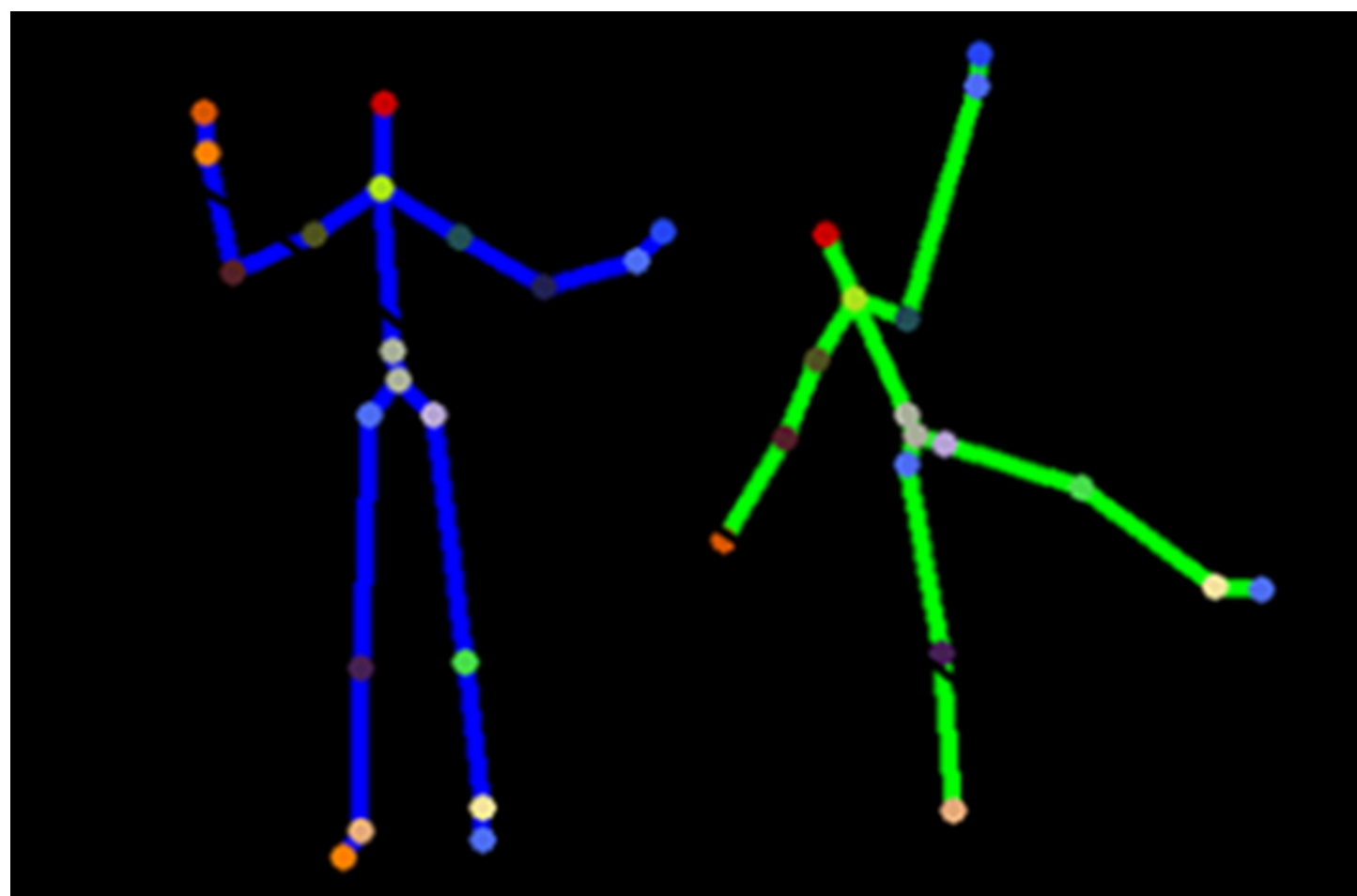
2.0.1410.19000



Unity

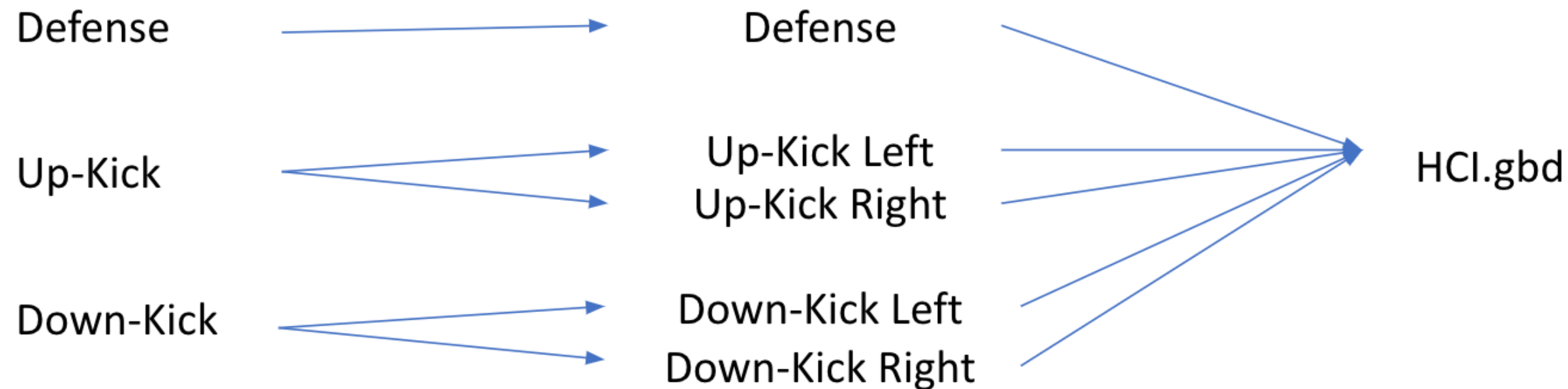
2021.3.9f1

Motivation



Approach – Gesture Recognition

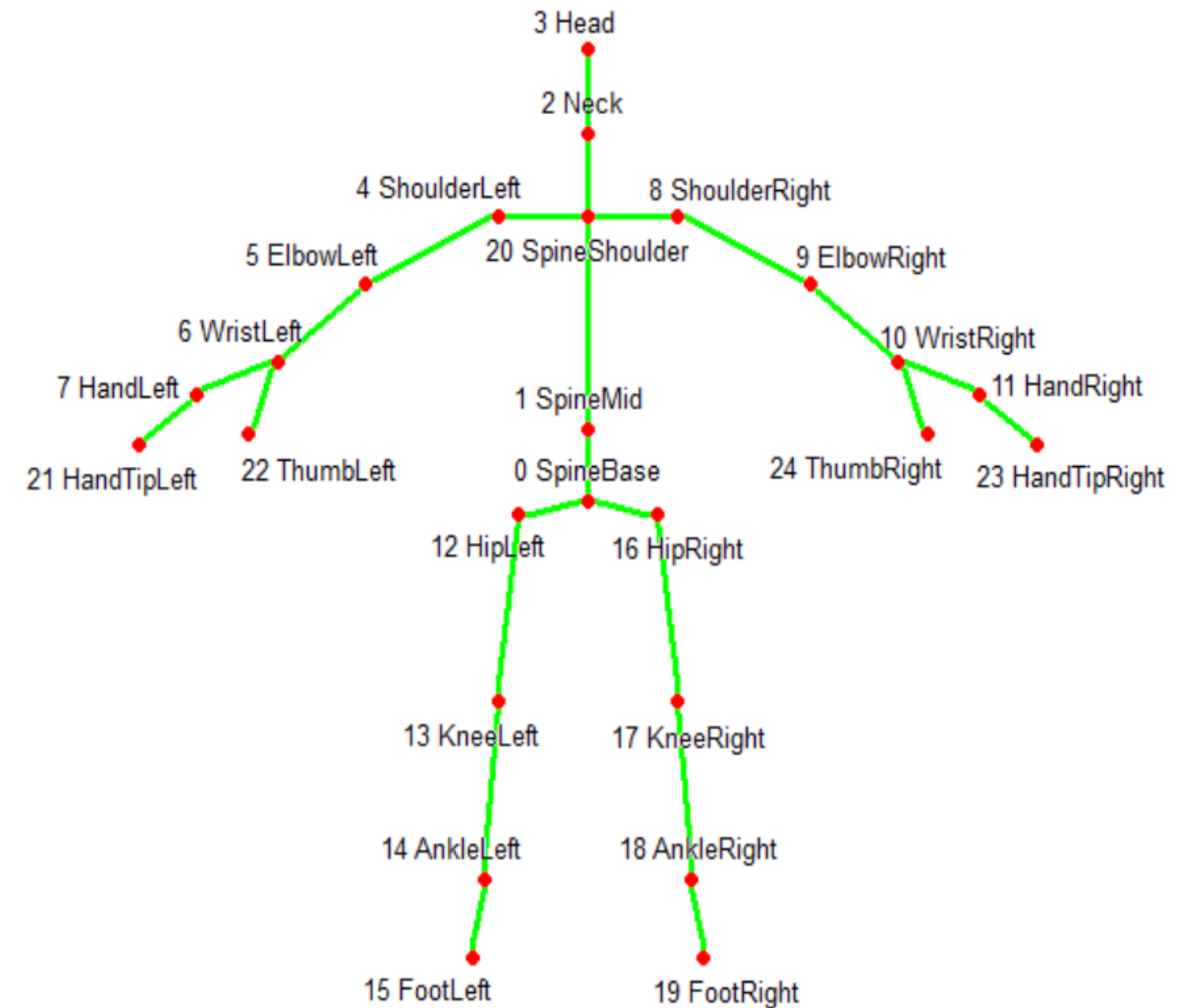
- Kinect Studio
Gesture Record
 - Visual Gesture Builder (VGB) – Discrete Gesture
(.XEP file)
- Select Valid Section
- Train model



<https://youtu.be/CH1zGg677IE>

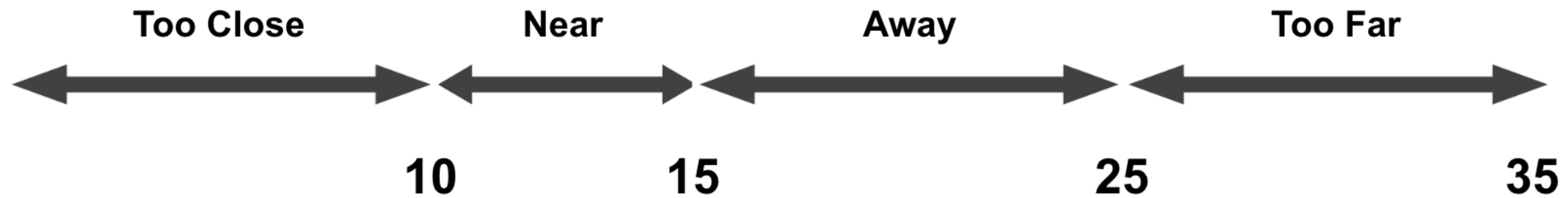
Approach – Moving

- Kinect 2.0 – Depth Camera
- Kinect 2.0 – IBody.GetJoints()
- Pass to Unity's Manager



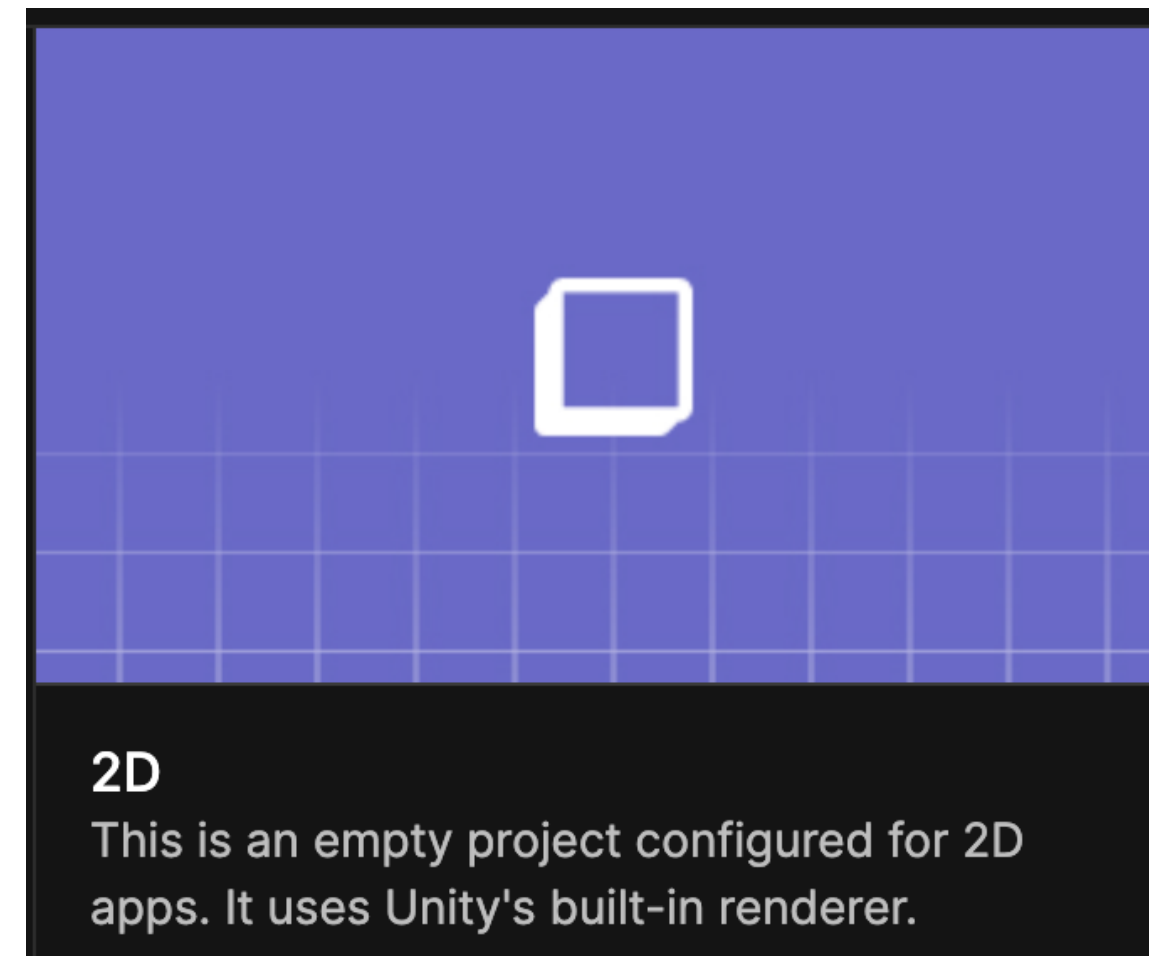
Approach – Moving

- Kinect 2.0 Distance Determination Table



Approach – Game Design

- Unity2D template





Approach – Game Design

- 鍵盤操作實現攻擊和防禦

```
[Header("KeyBinds")]  
public KeyCode downKick = KeyCode.F;  
public KeyCode upKick = KeyCode.G;  
public KeyCode defense = KeyCode.R;
```

Approach – Game Design

- 深度攝影機實現移動

```
_bsv = GameObject.Find("GameManager").GetComponent<BodySourceView>();
```

```
distanceState = (playerId == 1) ? _bsv.user1 : _bsv.user2;
```

```
if (distanceState == 2 && this.transform.position != attackPos) {  
    curstate = playerState.move;  
    this.transform.position += forward * speed;  
}  
else if (distanceState == 3 && this.transform.position == attackPos) {  
    curstate = playerState.move;  
    this.transform.position += -forward * speed;  
}
```

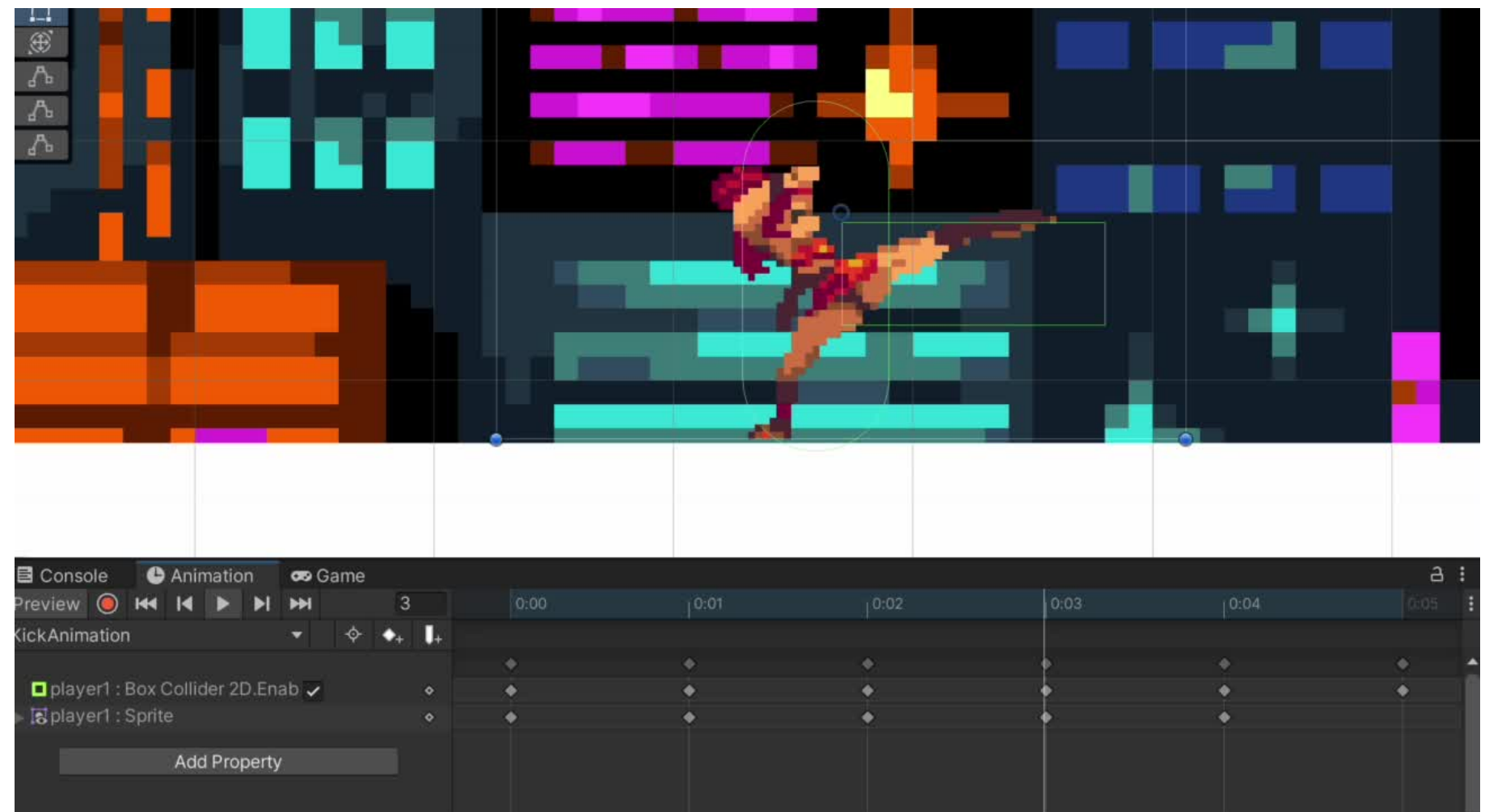
Approach – Game Design

- 攻擊和防禦按鍵觸發動畫

```
else if (Input.GetKeyDown(downKick)) {  
    curstate = playerState.kick;  
    animator.SetInteger("state", 1);  
    animator.SetTrigger("Kick-Trigger");  
    delay = 0.8f;  
}  
else if (Input.GetKeyDown(upKick)) {  
    curstate = playerState.ukick;  
    animator.SetInteger("state", 2);  
    animator.SetTrigger("Upkick-Trigger");  
    delay = 0.8f;  
}  
else if (Input.GetKey(defense)) {  
    curstate = playerState.defence;  
    animator.SetInteger("state", 3);  
    animator.SetTrigger("Defense-Trigger");  
    delay = 0.8f;  
}  
else if (Input.GetKeyDown(KeyCode.T)) {  
    curstate = playerState.hurt;  
    animator.SetInteger("state", 4);  
}
```

Approach – Game Design

- 動畫搭配碰撞框



Approach – Game Design

- 發生碰撞時判斷角色狀態

```
private void OnTriggerEnter2D(Collider2D collision) {  
    if (collision.gameObject.tag == "Player") {  
        PlayerController other = collision.GetComponent<PlayerController>();  
        playerState otherState = other.getCurState();  
        if (curstate == playerState.kick || curstate == playerState.ukick) {  
            if (otherState != playerState.defence) {  
                other.GetHurt(0.8f);  
            }  
            else {  
                delay = 2f;  
                curstate = playerState.idle;  
                animator.SetInteger("state", 0);  
            }  
        }  
    }  
}
```

Demo Time!



<https://youtu.be/vDAVfbOmlJw>