3D콘텐츠 이론 및 활용

5주. 클래스

- 메서드와 클래스
- 오브젝트 이동처리

학습개요

학습목표

- 메서드의 의미를 이해한다.
- 클래스의 의미를 이해하고 활용할 수 있다.
- 객체의 이동처리에 사용되는 메서드를 활용할 수 있다.

학습내용

- 메서드와 클래스
- 오브젝트 움직이기
- 객체 위치, 회전 처리

1. 메서드

- Method - 자주 사용하는 기능이나 동작을 한 덩어리로 묶어서 작성된 프로그램 코드

  - 위치는 start() 앞뒤 상관 없으며 Update()와 start()사이에 있어도 가능.

  - 병렬만 맞춰서 작성하면 위치는 상관없음.

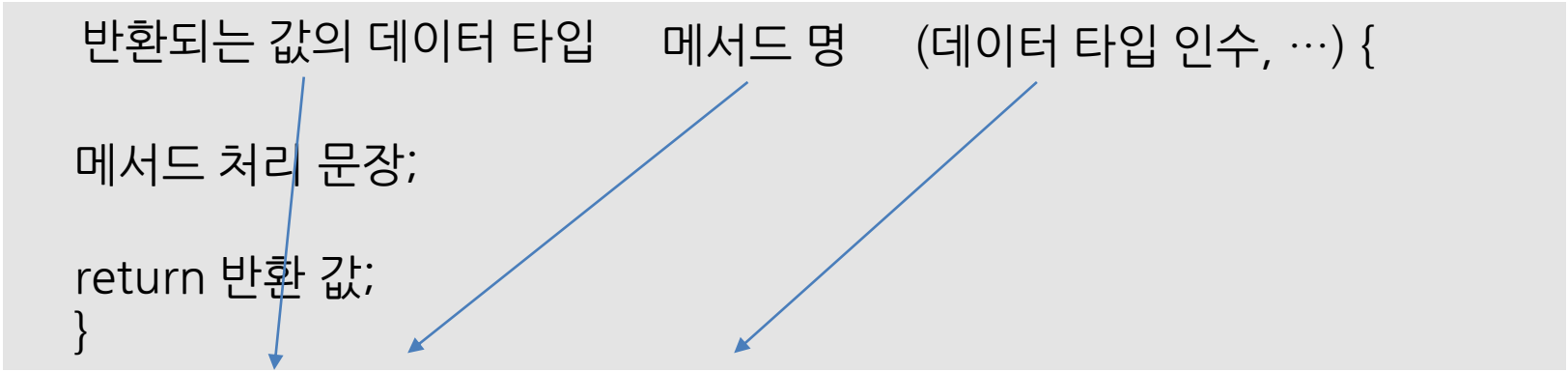
  - Start() 메서드는 플레이 될 때 최초 1회 실행할 기능들을 코딩

  - Update() 메서드는 실시간 매 프레임 마다 실행할 기능들을 코딩

```
void Start () {

}

// Update is called once per frame
void Update () {

}
```

1. 메서드

- 인수와 반환되는 값이 있는 메서드




```
int Add(int a, int b){
    int c = a + b;
    return c;
}

void Start () {
    int answer;
    answer = Add(3, 4);
    Debug.Log (answer);
}
```

1. 메서드

- 인수도 반환 값도 없는 메서드

```
void Start () {
    CoffeeOrder ();
}

void CoffeeOrder(){
    print ("커피가 완성 되었습니다");
}
```

1. 메서드

- 인수가 있는 메서드

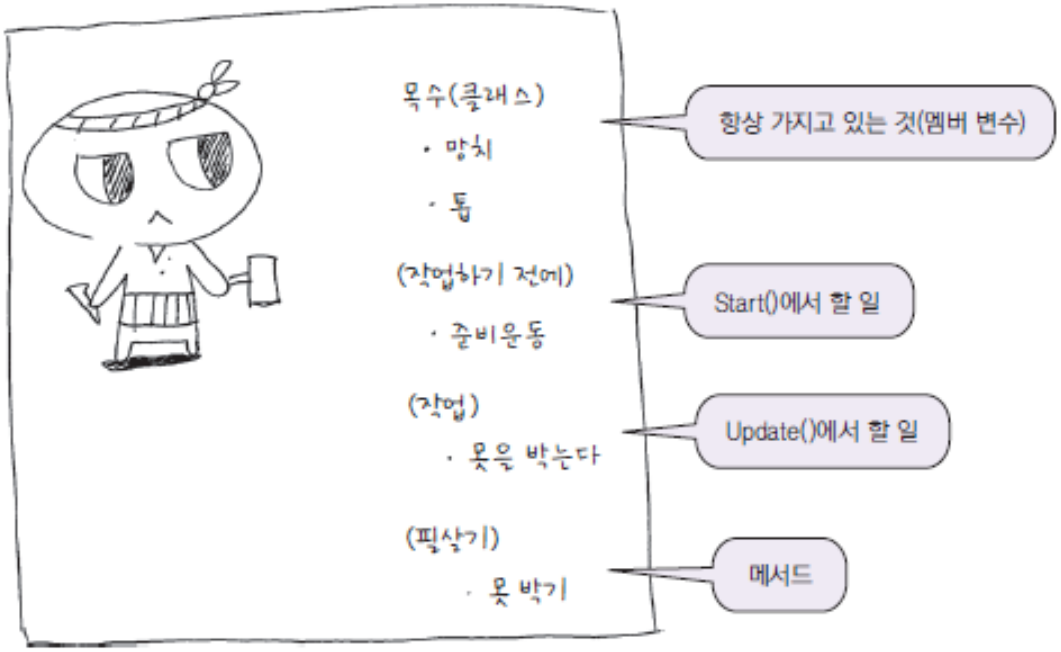
```
void CallName(string name) {
    Debug.Log ("Hello " + name);
}

void Start () {
        CallName("Tom");
}
```

2. 클래스(Class)

1) 클래스의 정의

- 메서드가 어떤 일의 처리과정을 모아 둔 것이라면, 클래스는 1개 이상의 메서드와 변수 등을 모아 둔 것
- 유니티에서 만들어 놓은 클래스, 사용자가 만들어 놓은 클래스 가 있음
- 클래스는 스크립트 프로그램 뿐만 아니라, 모든 오브젝트를 포함함

〈목수클래스 예〉



2. 클래스(Class)

2) 클래스의 활용

- 객체 인스턴스 생성
  - int number ;
  - NavMeshAgent agent;
  - Animator animator;
  - public GameObject target;

- 특정 객체의 메서드 실행
  - mPlayer.Attack();

- 작성한 클래스를 통해 인스턴스 객체를 만들어 사용하는 방법

```
Player myPlayer = new Player();
```

3. 클래스의 로컬 변수, 멤버 변수

- 선언되는 위치에 따라 로컬변수, 멤버변수로 나눌 수 있다.
- 스크립트 이름과 같은 클래스는 꼭 존재하여야 한다.

(=전역변수) 클래스가 사라지지 않는 이상
계속 사용할 수 있는 변수
즉, 클래스 전역에 선언되었으니 update 뿐 아니라
다른 메서드에서도 사용할 수 있다.

이 지역에서만 사용이 한정되어 있으면 로컬 변수!

studyScript.cs

```
public class studyScript: MonoBehaviour {
    void Start() {
    }

    int score = 1234;              // 멤버 변수.

    void Update() {
        int memo1 = 567;           // Update() 안에서만 유효한 로컬 변수.

        score++;
        memo1++;

        if(memo1 > 100) {          // memo1이 100보다 클 때만.
            int memo2 = 89;        // 이 if문 안에서만 유효한 로컬 변수.
            memo2++;
        } ——————————————— 프로그램 처리가 여기에 도달한 시점에서 memo2는 사라진다

        Debug.Log(score);
        Debug.Log(memo1);          이곳은 if문 밖이므로 이미 memo2는 없다. 주석처리를 벗기면
        // Debug.Log(memo2); —— 존재하지 않는 변수를 참조하려 하므로 오류가 생긴다
    } ——————————————— 프로그램 처리가 여기에 도달한 시점에서 memo1도 지워진다. score는 지워지지 않는다
}
```

4. 클래스 활용실습

- 다른 클래스 파일 활용
  - 다른 클래스의 메서드, 변수 활용
  - 다른 클래스에서 변수를 활용할 수 있도록 지정

```
public class oyaji : MonoBehaviour {
    int hungry = 0;

    public void akubi(){
        Debug.Log ("하품을 한다" + hungry);
    }
}
```

```
public class classStudy : MonoBehaviour
{
    private oyaji oyj = new oyaji ();

    void Start ()
    {
        oyj.akubi ();

        Debug.Log ("준비운동");
    }

    void Update ()
    {
        begin ();
    }

    void begin ()
    {
        Debug.Log ("못을 박는다");
    }

}
```

5. 객체 이동 실습

- Vector3.forward
- Vector3.back
- Vector3.left
- Vector3.right
- this 키워드

```
void Update ()
{
    if (Input.GetKey (KeyCode.W)) {
        this.transform.Translate (Vector3.forward);//(0,0,1)
    }
    if (Input.GetKey (KeyCode.S)) {
        this.transform.Translate (Vector3.back);
    }
    if (Input.GetKey (KeyCode.A)) {
        this.transform.Translate (Vector3.left);
    }
    if (Input.GetKey (KeyCode.D)) {
        this.transform.Translate (Vector3.right);
    }

}
```

5. 객체 이동 실습

- Position 초기화

```
    Vector3 init;

    // Use this for initialization
    void Start ()
    {
        init = this.transform.position;
    }

    void Update ()
    {
//
//
        if (Input.GetMouseButtonDown(0)) {
            this.transform.position = init;
            this.transform.rotation = Quaternion.Euler(init);
        }

    }
```

5. 객체 이동 실습

- Rotate 메서드

```
    void Update ()
    {

//
//
//
        if (Input.GetKey (KeyCode.Q)) {
            this.transform.Rotate (0, -90, 0);
        }
        if (Input.GetKey (KeyCode.E)) {
            this.transform.Rotate (0, 90, 0);
        }
    }
```

5. 객체 이동 실습(속도보정)

- Time.deltaTime 메서드
  다양한 기기의 속도차를 동일하게 보정해 주는 메서드

```
    void Update ()
    {

//
//
//
        if (Input.GetKey (KeyCode.Q)) {
            this.transform.Rotate (0, -90 * Time.deltaTime, 0);
        }
        if (Input.GetKey (KeyCode.E)) {
            this.transform.Rotate (0, 90 * Time.deltaTime, 0);
        }
    }
```

6. 객체 이동 실습

- X축 우측으로 계속 이동하는 객체 만들기

```
using UnityEngine;
using System.Collections;

public class BoxMove : MonoBehaviour {
    public float speed;

    void Start () {
    }

    void Update () {
        this.transform.Translate (speed * Time.deltaTime, 0, 0);
    }
}
```

학습정리

1. 메서드는 게임에서 특정 동작(행동)이나 기능을 하나로 등록해 놓은 것이다.

2. 클래스는 다양한 메서드와 변수들의 묶음이다.

참고자료

- 초보자를 위한 유니티 5, 아라카와 다쿠야, 아사노 유이치 저 윤준 역 한빛미디어 2016.01.20
- 길벗, “반다이 남코 현역 디렉터가 알려주는 유니티 게임제작입문”