



Unity

# Cheat Sheet



MOKWON GAME

ICE since 2023

## 단축키

용도	단축키
툴박스	Q(view), W(move), E(rot), R(scale), T(rect), Y(trans)
flythrough	마우+AD(좌우), +QE(상하), +WS(전후)
궤도(orbit)	Alt+마좌
스냅(snap)	Ctrl+변형
줌(zoom)	마휠, Alt+마우
전체화면	Shift+Space
플레이	Ctrl+P
빈오브젝트	Ctrl+Shift+N
복제	선택+Ctrl+D
삭제	선택+Delete
초점(focus)	선택+F, 선택+마더블

## CG

개념	설명
좌표축	X(R:좌우), Y(G:상하), Z(B:전후)
카메라	씬(개발자), 게임(메인:플레이어)
오버레이	Cameras, Orientation, Tools
씬이름	Project > Assets > Scenes
크기(scale)	기준 유닛: 큐브=1, 구=1, 평면=10

## C# 스크립트 - 메소드

### MonoBehaviour 메소드(함수)

**void** Start()

Component가 생성되고 Awake() 메소드 후, Update() 메소드 전에 한 번만 호출되는 메소드; Awake() → Start() → Update() 순으로 호출

**void** Update()

매 프레임(장면)마다 호출되는 메소드; Awake() → Start() → Update() 순으로 호출

**void** Invoke(**string** methodName, **float** time)

메소드명 methodName을 가진 메소드를 time 초 후에 호출

**void** InvokeRepeating(**string** methodName, **float** time, **float** repeatRate)

메소드명 methodName을 가진 메소드를 repeatRate 초 만큼 반복 호출; 이 메소드는 time 초 후에 최초 호출

**static void** print(**Object** object)

오브젝트인 object의 값을 확인하기 위해 콘솔에 출력; Debug.Log()와 동일

**void** OnCollisionEnter(**Collision** collision)

Collider/Rigidbody가 다른 Collider/Rigidbody를 최초로 접촉할 때 호출되는 이벤트 처리기; 충돌에 대한 정보는 Collision에 담겨 전달됨; 물리학이 작동하는 충돌

**void** OnCollisionExit(**Collision** collision)

Collider/Rigidbody가 다른 Collider/Rigidbody를 더 이상 접촉하지 않을 때 호출되는 이벤트 처리기; 충돌에 대한 정보는 Collision에 담겨 전달됨

**void** OnTriggerEnter(**Collider** other)

두 게임 오브젝트가 충돌할 때 호출되는 이벤트 처리기; 충돌에 대한 정보는 Collider에 담겨 전달됨; OnCollisionEnter()과 다르게 이 함수는 물리학이 작동하지 않음; 이 함수가 호출되려면, 두 오브젝트 중 하나는 트리거 설정이 되어야 함

**void** OnTriggerExit(**Collider** other)

두 게임 오브젝트의 충돌이 끝날 때 호출되는 이벤트 처리기; 충돌에 대한 정보는 Collider에 담겨 전달됨

### Object 메소드(함수)

**static Object** Instantiate(**Object** object, **Vector3** position, **Quaternion** rotation)

오브젝트인 object가 position, rotation인 Transform을 가지도록 생성

**static void** Destroy(**Object** object, **float** time)

오브젝트인 object를 time 초 후에 파괴; time을 쓰지 않으면 당장 파괴

**static T** FindAnyObjectByType();

현재 실행 중인 오브젝트 중에서 자료형이 일치하는 오브젝트 중 하나를 빠르게 반환

**void** DontDestroyOnLoad(**Object** target);

새로운 씬으로 전환되더라도 target 오브젝트를 파괴하지 않음



Unity

# Cheat Sheet



MOKWON GAME

ICE Since 2023

## Input 메소드(함수)

`static bool GetKeyDown(KeyCode key)`

최초 프레임에서 key인 키를 사용자가 눌렀는지 확인; 키 코드는 KeyCode로 선택; KeyCode 대신 string도 가능

`static bool GetKey(KeyCode key)`

모든 프레임에서 key인 키를 눌렀는지 확인; GetKeyDown()은 최초 프레임 확인, GetKey()는 모든 프레임 확인

`static float.GetAxis(string axisName)`

axisName으로 설정된 가상축의 입력을 -1~+1 사이로 얻기; axisName: 키보드("Horizontal", "Vertical"), 마우스 이동("Mouse X", "Mouse Y": -1~+1 범위 넘기 가능), 마우스 클릭("Fire1", "Fire2", "Fire3": 좌클, 우클, 휠클; 0, 1만 가능)

`static float.GetAxisRaw(string axisName)`

axisName으로 설정된 가상축의 입력을 -1, 0, +1 중에서 하나 얻기

## 기타 메소드(함수)

`static void Application.Quit()`

현재 앱을 바로 종료; namespace UnityEngine에 Application이 정의됨

## C# 스크립트 – 클래스와 멤버

### 메인 클래스

`class MonoBehaviour`

모든 유니티 스크립트의 기반 클래스; Object > Component의 자식 클래스

`class GameObject`

유니티에서 사용하는 게임 오브젝트; Object의 자식 클래스

- T GetComponent(): GameObject에 부착된 Component 얻기; 편의를 위해 Component에도 정의됨
- string tag: 게임 오브젝트의 태그 정보

`class Transform`

게임 오브젝트의 position(Vector3), rotation(Quaternion), lossyScale(Vector3)을 바꾸는 클래스; Component의 자식 클래스

- void Rotate(Vector3 eulers): 오일러 각 eulers만큼 회전 운동
- void Translate(Vector3 move): move만큼 그 방향으로 병진 운동
- Vector3 rotation.eulerAngles: 회전된 결과(사원수)를 오일러 각으로 접근

`class Renderer`

게임 오브젝트의 렌더링 객체; GameObject.GetComponent<Renderer>()로 획득; 유용한 멤버: material

`class Rigidbody`

게임 오브젝트에 물리 엔진(힘) 적용; GameObject.GetComponent<Rigidbody>()로 획득

- void AddForce(Vector3 force): force 방향과 크기로 힘을 추가
- bool isKinematic: true로 설정하면 스크립트로 운동 제어

`class Collider`

게임 오브젝트에 물리학 효과를 추가하는 Component

`class Collision`

충돌 정보를 담고 있는 클래스

- GameObject gameObject: 충돌한 게임 오브젝트
- Transform transform: 충돌한 gameObject의 transform

`class Material`

오브젝트의 재질(색깔, 반사 등) 결정; 유용한 멤버: color

`class SceneManager`

씬을 제어하는 클래스; using UnityEngine.SceneManagement 선언 필요

- static void LoadScene(string scene): 이름이 scene인 씬을 가져오기



## 보조 클래스

### class Input

사용자 입력(키보드, 마우스, 조이스틱 등)을 처리하는 클래스

### class Time

유니티가 사용하는 시간 정보를 제공하는 클래스; 유용한 멤버: deltaTime

### class Random

게임용 난수 발생기

- static float Range(float minIn, float maxIn): [minIn, maxIn] 범위의 난수 발생

### class AudioSource

오디오를 재생하는 클래스

- void Play(): clip에 미리 설정된 AudioClip을 재생
- void PlayOneShot(AudioClip clip, float volume): clip을 volume 크기로 한 번 재생
- void Stop(): AudioClip을 중지
- AudioClip clip: Play()가 재생할 AudioClip

### class AudioClip

오디오 파일을 저장하는 저장소

- float length: 오디오 파일의 재생 시간(초)

## 자료 클래스

### class Vector3

3차원 벡터

- static Vector3 right, up, forward: 각각 x, y, z의 단위 벡터; left, down, back도 가능

## UI 클래스

### class TMP\_Text

UI Text를 관리; 사용하려면 먼저 using TMPro를 선언

- string text: Text의 글자를 변경

### class TMP\_InputField

사용자가 입력하는 UI Text를 관리; 사용하려면 먼저 using TMPro를 선언

- string text: 사용자가 입력하는 Text가 저장되는 위치

## 자체 클래스

### class GameManager

우리가 만든 게임 관제소

### class PlayerPrefs

사용자 설정을 불러오거나 저장하는 클래스

- int GetInt(string key), int SetInt(string key, int value): int를 처리
- GetFloat(), SetFloat(): float을 처리
- GetString(), SetString(): string을 처리

## 구조체와 열거

### struct Color

Material의 색깔을 RGBA로 정하는 구조체: Color(1.0f, 1.0f, 1.0f, 1.0f); static Color: red, green, blue, white, black, yellow, cyan, magenta, gray, clear

### struct Color32

Color를 32비트로 설정하는 구조체; 사용 범위는 Color32(255, 255, 255, 255)

### enum KeyCode

사용자가 입력하는 키 코드의 정보를 제공하는 열거 클래스



Unity

# Cheat Sheet



**MOKWON GAME**

ICE Since 2023

## 프로퍼티(변수)

**GameObject** Component.gameObject

현재 Component가 부착된 GameObject의 인스턴스

**Transform** Component.transform

현재 Component의 gameObject에 부착된 Transform의 인스턴스; right, up, forward는 gameObject의 변환 결과를 기준으로 설정됨

**float** Time.deltaTime

각 프레임 사이의 시간 간격; 프레임이 표시되는 주기; 미리 설정한 speed와 speed\*Time.deltaTime를 사용해 오브젝트의 이동 거리를 일정하게 처리