



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 InvokeRepeating(string methodName, float time, float repeatRate)

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

void print(Object object)

오브젝트인 object의 값을 확인하기 위해 콘솔에 출력

Object 메소드(함수)

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

오브젝트인 object가 position, rotation인 Transform을 가지도록 생성

void Destroy(Object object, float time)

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

Input 메소드(함수)

bool Input.GetKeyDown(KeyCode key)

최초 프레임에서 key인 키를 사용자가 눌렀는지 확인; 키 코드는 KeyCode로 선택

bool Input.GetKey(KeyCode key)

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

기타 메소드(함수)

static void Application.Quit()

현재 앱을 바로 종료



Unity

Cheat Sheet



MOKWON GAME

ICE Since 2023

C# 스크립트 - 클래스

클래스와 열거

class MonoBehaviour

모든 유니티 스크립트의 기반 클래스; Object > Component의 자식 클래스

class GameObject

유니티에서 사용하는 게임 오브젝트; Object의 자식 클래스

class Transform

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

class Input

사용자 입력(키보드, 마우스, 조이스틱 등)을 처리하는 클래스

class Time

유니티가 사용하는 시간 정보를 제공하는 클래스

enum KeyCode

사용자가 입력하는 키 코드의 정보를 제공하는 열거 클래스

프로퍼티(변수)

GameObject Component.gameObject

현재 Component가 부착된 GameObject의 인스턴스

Transform Component.transform

현재 Component의 gameObject에 부착된 Transform의 인스턴스

float Time.deltaTime

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