SAPIENZA
UNIVERSITÀ DI ROMA

KARATE VR

A VR game for Karate training

START

Authors: Federico Barreca, Ilaria De Sio, Giuseppe Bello

MAIN IDEA

Our project aims to revolutionize karate training through a virtual reality (VR) platform. The game allows users to practice kumite defensive techniques and blocks with real-time feedbacks of their movements.

TABLE OF CONTENTS

01.

ABOUT KARATE

Different components of karate

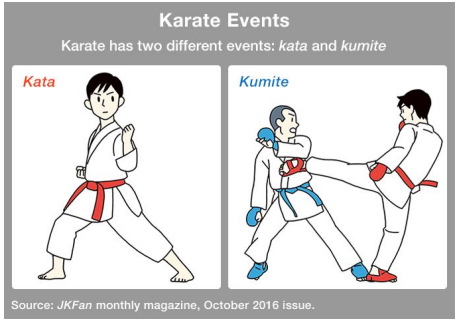
KARATE AND ITS COMPONENTS

KATA

It refers to a detailed choreographed pattern of martial arts movements

KIHON

Set of basic karate techniques practiced frequently, often during each training session. Mostly related to kata

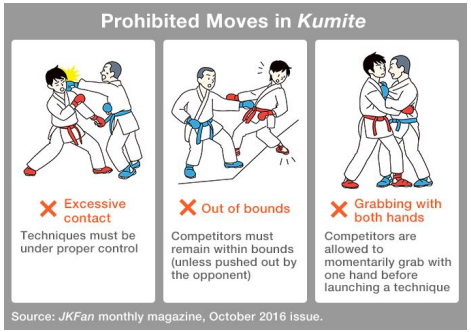


KUMITE

Dynamic combat that includes protective gear and a time- and point-based rule structure

KATA AND KUMITE

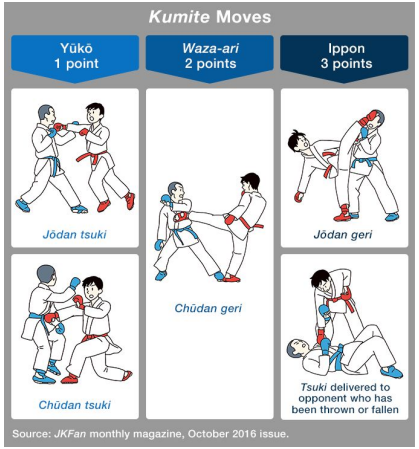
Kata and kumite feature different sets of techniques, although both separate offensive techniques (Tori) and defensive techniques (Uke)



Kata and kumite have different technique sets. Some of kata techniques may be illegal in kumite!

KUMITE TECHNIQUES

Every offensive technique corresponds to a defensive technique, whether it is a block or a movement. The latter is subject to all the execution quality criteria that apply to offensive techniques



KUMITE DEFENSE TECHNIQUES

In particular, in our project, the player must use all the blocking techniques required in kumite, such as:

Soto Uke, a forearm technique used to deflect a punch aimed at the abdomen

Teisho Uke, an open-hand technique used to deflect a kick aimed at the face

Gedan Barai, a forearm technique used to deflect a kick aimed at the abdomen or back

Te Nagashi Uke, an open-hand technique used to deflect a punch aimed at the face

SPORTS AND VR

Although some martial arts and boxing VR applications have been developed, they are usually oriented towards engagement and gaming, ignoring the technical and competitive aspects of the sport in question

This is where our solution comes in!

02.

WHY META QUEST 3?

Features that make it suitable for our project

IDEA

GAMEPLAY

SOFTWARE DESIGN

LIMITS

FUTURE WORKS

ABOUT THE PRODUCT

Meta Quest 3 provides an unmatched VR experience with its powerful performance and high-resolution display. As a standalone device, it offers complete freedom of movement without the need for a PC or cables. The advanced tracking system ensures precise and realistic movements, essential for an immersive karate game.

03.

GAMEPLAY

Game design and gamification components

IDEA

GAMEPLAY

SOFTWARE DESIGN

LIMITS

FUTURE WORKS

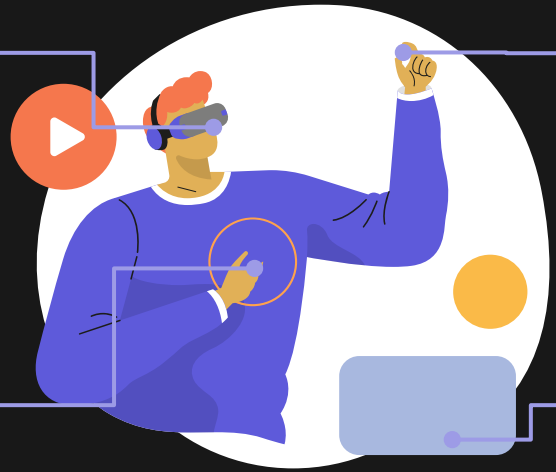
GAMEPLAY COMPONENTS

IMMERSIVITY AND SCALING

Adaptive Experience

HUD

Life , techniques counter and difficulty settings



ACOUSTIC FEEDBACK

Realistic sound effects and responses

MECHANICS

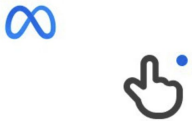
Ruleset and moveset

04.

SOFTWARE DESIGN

Unity Engine and combat logic

LIBRARIES



META XR INTERACTION SDK

Hand Tracking, Ray Interactor

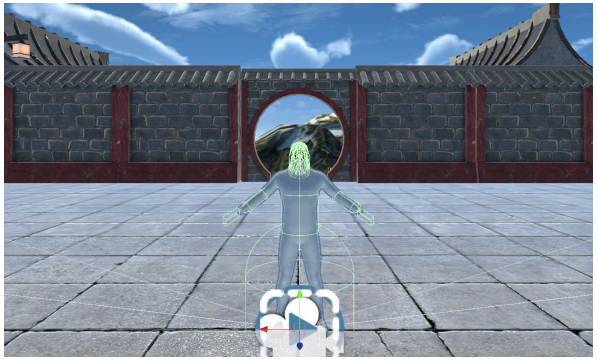
META SDK ALL-IN-ONE

All others XR functionalities

UNITY ENGINE

Collision detection, animation, sound and environment design

COLLISION DETECTION



PLAYER

FIGHTER

OUR STRUCTURE

```
List<(string, string, string, Dictionary<string, bool>, string)> techniques = new List<(string, string, string, Dictionary<string, bool>, string)>
{
    ("doJabL", "Volume1_jab_L", "LeftHandCollider", new Dictionary<string, bool>{ {"HeadCollider", false}, {"LeftHandCollider", true}, {"RightHandCollider", true} }, "punch_short_whoosh_16"),
    ("doJabR", "Volume1_jab_R", "RightHandCollider", new Dictionary<string, bool>{ {"HeadCollider", false}, {"LeftHandCollider", true}, {"RightHandCollider", true} }, "punch_short_whoosh_16")
    ("doUshiro", "Ushirouramawashigeri", "RightFootCollider", new Dictionary<string, bool>{ {"HeadCollider", false}, {"LeftHandCollider", true} }, "kick_long_whoosh_19"),
    ("doRoundKick", "Round_kick", "LeftCalfTwistCollider", new Dictionary<string, bool>{ {"BodyCollider", false}, {"LeftArmLowerCollider", true} }, "kick_long_whoosh_19"),
    ("doHigherKickR", "Higher_Kick_R", "RightFootCollider", new Dictionary<string, bool>{ {"HeadCollider", false}, {"LeftHandCollider", true} }, "punch_long_whoosh_21"),
    ("doLowerPunch", "lower_punch", "LeftHandCollider", new Dictionary<string, bool>{ {"BodyRadioCollider", false}, {"LeftArmLowerCollider", true} }, "punch_long_whoosh_30"),

};
```

Structure to manage combat logic and collision rules, audio feedback and animations

< Event Trigger, Technique, Collider fighter, < Collider target player, Collider defense >, Sound associated >

SCALING AND HUD

SCALING

```
public class FighterScaler : MonoBehaviour
{
    public GameObject vrCamera;
    public GameObject fighterHead;
    private GameObject fighter;

    void Start()
    {
        fighter = GameObject.FindWithTag("Fighter");
        AdjustFighterHeight();
    }

    private float time = 0.0f;
    public float interpolationPeriod = 0.5f;

    void Update()
    {
        time += Time.deltaTime;

        Vector3 fighterFocusPoint = fighter.transform.position + Vector3.up * 1.5f;
        vrCamera.transform.LookAt(fighterFocusPoint);

        AdjustFighterPosition();

        if (time < 2)
            AdjustFighterHeight();
    }

    private void AdjustFighterPosition()
    {
        fighter.transform.position = new Vector3(
            vrCamera.transform.position.x,
            fighter.transform.position.y,
            vrCamera.transform.position.z + 0.92f);
    }

    private void AdjustFighterHeight()
    {
        float targetHeight = vrCamera.transform.position.y;
        float currentHeight = fighterHead.transform.position.y;
        Debug.Log($"Camera Height: {targetHeight}, Fighter Head Height: {currentHeight}");
        if (targetHeight > 0 && currentHeight > 0)
        {
            float scaleFactor = targetHeight / currentHeight;

            fighter.transform.localScale = new Vector3(
                fighter.transform.localScale.x,
                fighter.transform.localScale.y * scaleFactor,
                fighter.transform.localScale.z);
        }
    }
```

HUD (EASY)

SCREEN IN-GAME

MAIN MENU

OPTIONS

05.

LIMITS

Challenges faced during development

LIMITATIONS AND CHALLENGES

LIBRARIES INSTABILITY

Libraries showed fragile compatibility between Unity Engine and Meta Quest 3 major functionalities

LIMITED CAMERA ANGLE

Meta Quest 3 cameras cannot see the environment at 360°. Depending on the orientation of the headset, some objects go out of sight

06.

FUTURE WORKS

Potential improvements and new features

POSSIBLE NEW FEATURES

ATTACK AND DODGE TECHNIQUES

Being able to attack the opposing fighter and dodge with whole-body movements within the game space

COOP MODALITY

The possibility to create a training virtual room with a friend or take a lesson with a trainer from your home

LEARNING MODALITY

Enabling even people who do not practise karate to learn through specific virtual training sessions with real-time correction of movements

IDEA

GAMEPLAY

SOFTWARE DESIGN

LIMITS

FUTURE WORKS

THANKS FOR THE ATTENTION!

Do you have any questions?

Now let's see a Live-Demo!