Я хочу сделать игру используя **JavaScript (Phaser/Kaboom.js)**

Игра называется

jiu jitsu 2: переполох в Гонконге

Я работаю на макбуке

Игра должна быть вертикальная под мобильные устройства

Когда я увижу что все заработало я хочу перенести её в телеграм миниапп.

Документацию ты получил отдельным файлом.

Суть игры:

Платформер с рандомно генерируемыми уровнями — бесконечное их число. Примерно как в flopping bird

Главное меню:

Сохраняемая статистика

Фон который я сделаю в svg

Табличка на которой отображается рекорд: текст РЕКОРД 0000000 где вместо нулей – лучший набранный результат

Музыка на фоне меню

Игровой процесс:

Музыка на фоне уровня

Вверху справа кнопка крест которая закрывает уровень

Персонаж, анимацию движения и прыжков которого я нарисую с помощью покадровой анимации

Персонаж автоматически двигается слева направо с небольшой скоростью

Одиночное нажатие на любую точку экрана кроме кнопки закрыть заставит его прыгнуть

Второе нажатие заставит прыгнуть выше

Вверху экрана отображаются три сердечка-жизни

Вверху экрана отображается Уровень (номер)

Вверху экрана отображается счет – по 100 очков за монетку

Каждый уровень занимает 30-40 секунд

Каждый уровень имеет случайное расположение прямоугольных препятствий

Каждый уровень имеет случайное расположение прямоугольных пропастей

Каждый уровень имеет случайно сгенерированный задний фон – красивый цветной градиент

Есть препятствия через которые надо перепрыгивать, сквозь них не пройти

Есть пропасти через которые надо перепрыгивать, в них можно упасть, тогда отнимается жизнь и герой возрождается перед пропастью

Есть монетки которые надо собирать

Есть здание в которое в конце надо попасть, как в марио

Есть анимация салют когда герой попадает в здание

Есть звук салюта по завершении уровня

Если все сердечки закончились — появляется экран ВЫ ПРОИГРАЛИ Финальный счет: 00000

Где вместо нулей указан финальный счет.

Если счет больше предыдущего рекорда, он отображается как рекорд на табличке в главном меню.

Я хочу чтобы ты мне переписал:

Код для каждого файла, чтобы оно работало: index.html, style.css, game.js

Дал таблицу — формат файлов и размер файлов.

И дал инструкцию пошаговую очень простую для совершенно неопытного человека инструкцию: как собрать проект, как проверить, как опубликовать в github pages, как превратить в мини-апп

ВАЖНО: если ты чего-то не знаешь и не умеешь — никогда не придумывай. Предлагай альтернативы в которых ты полностью уверен.

Дерево проекта:

/jiujitsu-rampage-2/

│

├── index.html # Главный HTML-файл

├── style.css # Стили интерфейса

├── game.js # Логика игры (Kaboom.js)

│

├── assets/

│ ├── images/

│ │ ├── characters/

│ │ │ ├── player-run1.png # Кадр 1 анимации бега (64x64)

│ │ │ ├── player-run2.png # Кадр 2

│ │ │ ├── player-run3.png # Кадр 3

│ │ │ └── player-jump.png # Прыжок (64x64)

│ │ │

│ │ ├── objects/

│ │ │ ├── coin.png # Монетка (32x32)

│ │ │ ├── obstacle.png # Препятствие (64x32)

│ │ │ └── building.png # Здание-финиш (128x128)

│ │ │

│ │ ├── ui/

│ │ │ ├── heart.png # Сердечко жизни (32x32)

│ │ │ ├── button-bg.png # Фон кнопок (300x100)

│ │ │ └── game-over.png # Надпись Game Over (256x64)

│ │ │

│ │ └── backgrounds/

│ │ ├── hongkong-bg.jpg # Фон меню (1920x1080)

│ │ ├── level1-bg.png # Градиент уровня 1 (360x640)

│ │ └── level2-bg.png # Градиент уровня 2 (360x640)

│ │

│ ├── sounds/

│ │ ├── music/

│ │ │ ├── menu-theme.mp3 # Музыка меню

│ │ │ └── level-theme.mp3 # Музыка уровня

│ │ │

│ │ └── sfx/

│ │ ├── jump.mp3 # Прыжок (0.5 сек)

│ │ ├── fall.mp3 # Падение (1 сек)

│ │ ├── coin.mp3 # Монетка (0.3 сек)

│ │ └── victory.mp3 # Победа (2 сек)

│ │

│ └── fonts/ # Локальные шрифты (если нужно)

│ ├── oswald-bold.woff2

│ └── roboto-regular.woff2

│

├── docs/ # Документация (опционально)

│ └── how-to-play.md

│

└── README.md # Инструкции по запуску

Код файлов:

Index.html

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">

<title>Jiujitsu Rampage 2</title>

<link rel="stylesheet" href="style.css">

<script src="https://kaboomjs.com/lib/0.6.0/kaboom.js"></script>

<link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap" rel="stylesheet">

</head>

<body>

<div id="game-container"></div>

<script src="game.js"></script>

</body>

</html>

Game.js

kaboom({

width: 360,

height: 640,

canvas: document.getElementById("game-container"),

background: [0, 0, 0],

scale: 1,

});

// Загрузка ресурсов

loadSprite("player", "assets/images/characters/player.png");

loadSprite("coin", "assets/images/objects/coin.png");

loadSprite("obstacle", "assets/images/objects/obstacle.png");

loadSprite("building", "assets/images/objects/building.png");

loadSprite("heart", "assets/images/ui/heart.png");

loadSound("jump", "assets/sounds/sfx/jump.mp3");

loadSound("fall", "assets/sounds/sfx/fall.mp3");

loadSound("coin", "assets/sounds/sfx/coin.mp3");

loadSound("victory", "assets/sounds/sfx/victory.mp3");

// Глобальные переменные

let score = 0;

let lives = 3;

let level = 1;

let highScore = localStorage.getItem("highScore") || 0;

let playerSpeed = 120;

// Главное меню

scene("menu", () => {

add([

text("Jiujitsu Rampage 2", { size: 32 }),

pos(center().x, 100),

origin("center"),

color: rgb(255, 50, 50),

]);

add([

text(`Рекорд: ${highScore}`, { size: 24 }),

pos(center().x, 180),

origin("center"),

]);

// Создаем кнопку в DOM

const startBtn = document.createElement("button");

startBtn.id = "start-button";

startBtn.textContent = "НАЧАТЬ ИГРУ";

document.body.appendChild(startBtn);

startBtn.addEventListener("click", () => {

startBtn.remove();

go("game");

});

});

// Генератор уровня

function generateLevel() {

// Очищаем сцену

destroyAll();

// Случайный градиент фона

add([

rect(width(), height()),

color(

rand(0, 255),

rand(0, 255),

rand(0, 255)

),

pos(0, 0),

]);

// Платформы

for (let i = 0; i < 10 + level; i++) {

add([

sprite("obstacle"),

pos(rand(0, width() - 100), rand(100, height() - 100)),

area(),

solid(),

"platform",

]);

}

// Монетки

for (let i = 0; i < 5 + level; i++) {

add([

sprite("coin"),

pos(rand(50, width() - 50), rand(50, height() - 50)),

area(),

"coin",

]);

}

// Финишное здание

add([

sprite("building"),

pos(width() - 100, height() - 150),

area(),

"building",

]);

}

// Игровая сцена

scene("game", () => {

generateLevel();

// Интерфейс

const scoreText = add([

text(`Счет: ${score}`, { size: 24 }),

pos(20, 20),

]);

const levelText = add([

text(`Уровень: ${level}`, { size: 24 }),

pos(20, 50),

]);

// Жизни

for (let i = 0; i < lives; i++) {

add([

sprite("heart"),

pos(width() - 40 - (i \* 40), 20),

scale(0.5),

]);

}

// Персонаж

const player = add([

sprite("player"),

pos(50, height() / 2),

area(),

body(),

]);

// Автодвижение

player.onUpdate(() => {

player.move(playerSpeed, 0);

});

// Прыжок

onTouchStart(() => {

if (player.isGrounded()) {

player.jump(600);

play("jump");

}

});

// Сбор монеток

player.onCollide("coin", (coin) => {

destroy(coin);

score += 100;

scoreText.text = `Счет: ${score}`;

play("coin");

});

// Финиш

player.onCollide("building", () => {

play("victory");

level++;

playerSpeed += 10;

generateLevel();

});

// Падение

player.onUpdate(() => {

if (player.pos.y > height()) {

player.pos = vec2(50, height() / 2);

lives--;

play("fall");

if (lives <= 0) {

if (score > highScore) {

highScore = score;

localStorage.setItem("highScore", highScore);

}

go("menu");

}

}

});

});

// Старт

go("menu");

style.css

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

-webkit-tap-highlight-color: transparent;

}

body {

overflow: hidden;

touch-action: manipulation;

background: #000;

font-family: 'Press Start 2P', cursive;

}

#game-container {

position: fixed;

top: 0;

left: 0;

width: 100vw;

height: 100vh;

}

#start-button {

position: absolute;

bottom: 20%;

left: 50%;

transform: translateX(-50%);

padding: 15px 30px;

background: #ff3355;

color: white;

border: none;

border-radius: 10px;

font-family: 'Press Start 2P', cursive;

font-size: 1.2rem;

text-transform: uppercase;

cursor: pointer;

z-index: 100;

box-shadow: 0 5px 0 #cc0022;

}

#start-button:active {

transform: translateX(-50%) translateY(2px);

box-shadow: 0 3px 0 #cc0022;

}