const player = {

    health: 100,

    defense: 0,

    deck: []

};

const enemy = {

    health: 100,

    stunned: false

};

let currentNode = null;

const storyNodes = {

    start1: {

        text: "Je start je shift in de innovatieve onderzoeksfaciliteit ‘B.I.T’. Alles lijkt routine.",

        choices: [

            { text: "Controleer de storingslogboeken", next: "start2" },

            { text: "Ga direct naar het lab", next: "start3" }

        ]

    },

    start2: {

        text: "Er staat een foutmelding: ‘Onbekende subsystemen gedetecteerd’.",

        choices: [

            { text: "Stel een intern onderzoek in", next: "start4" },

            { text: "Negeer het en ga door met je werk", next: "start3" }

        ]

    },

    start3: {

        text: "In het lab merk je vreemde trillingen in de muur. Iets klopt niet.",

        choices: [{ text: "Ga terug naar je station", next: "start4" }]

    },

    start4: {

        text: "Een AI-systeem raakt geïnfecteerd. Bereid je voor op een gevecht!",

        choices: [{ text: "Start Battle", next: "battle" }]

    },

    afterBattle: {

        text: "De vijand is verslagen, maar dit was slechts het begin...",

        choices: [

            { text: "Volg het signaal terug naar de bron", next: "start1" },

            { text: "Rapporteer het incident aan de centrale", next: "start2" }

        ]

    }

};

const allCards = [

    {

        name: "EMP Surge",

        description: "Valt aan met 15 schade aan vijand.",

        type: "attack",

        effect: (player, enemy) => {

            enemy.health -= 15;

            logAction("EMP Surge deals 15 damage.");

            updateUI();

        }

    },

    {

        name: "Overdrive Strike",

        description: "Sterke aanval: 25 schade, maar je verliest zelf 5 HP.",

        type: "attack",

        effect: (player, enemy) => {

            enemy.health -= 25;

            player.health -= 5;

            logAction("Overdrive Strike: 25 to enemy, 5 recoil.");

            updateUI();

        }

    },

    {

        name: "Shock Pulse",

        description: "10 schade en verdooft de vijand (vijand slaat over).",

        type: "attack",

        effect: (player, enemy) => {

            enemy.health -= 10;

            enemy.stunned = true;

            logAction("Shock Pulse: 10 damage, enemy stunned.");

            updateUI();

        }

    },

    {

        name: "Auto-Stim Injector",

        description: "Geneest 20 HP.",

        type: "health",

        effect: (player) => {

            player.health += 20;

            logAction("Healed 20 HP with Auto-Stim Injector.");

            updateUI();

        }

    }

];

function showStoryNode(nodeKey) {

    currentNode = nodeKey;

    const node = storyNodes[nodeKey];

    document.getElementById("startScreen").style.display = "none";

    document.getElementById("gameContainer").style.display = "none";

    document.getElementById("storyContainer").style.display = "flex";

    document.getElementById("storyText").textContent = node.text;

    const choicesDiv = document.getElementById("choiceButtons");

    choicesDiv.innerHTML = "";

    node.choices.forEach(choice => {

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

        btn.textContent = choice.text;

        btn.onclick = () => {

            if (choice.next === "battle") {

                startBattle();

            } else {

                showStoryNode(choice.next);

            }

        };

        choicesDiv.appendChild(btn);

    });

}

function startBattle() {

    enemy.health = 100;

    document.getElementById("storyContainer").style.display = "none";

    document.getElementById("gameContainer").style.display = "flex";

    updateUI();

    drawCards();

}

function updateUI() {

    document.getElementById("playerHealth").textContent = "Health: " + player.health;

    document.getElementById("playerDefense").textContent = "Defense: " + player.defense;

    document.getElementById("enemyHealth").textContent = "Health: " + enemy.health;

}

function drawCards() {

    const container = document.getElementById("cardContainer");

    container.innerHTML = "";

    const hand = allCards.sort(() => 0.5 - Math.random()).slice(0, 3);

    hand.forEach(card => {

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

        btn.textContent = card.name;

        btn.title = card.description; // Tooltip hier

        btn.onclick = () => {

            card.effect(player, enemy);

            enemyTurn();

            checkGameOver();

            drawCards();

        };

        container.appendChild(btn);

    });

}

function enemyTurn() {

    if (enemy.stunned) {

        logAction("Enemy is stunned!");

        enemy.stunned = false;

        return;

    }

    const damage = 15 - (player.defense || 0);

    const dealt = Math.max(0, damage);

    player.health -= dealt;

    logAction("Enemy attacks for " + dealt + " damage.");

    updateUI();

}

function checkGameOver() {

    if (player.health <= 0) {

        alert("Game Over!");

        location.reload();

    } else if (enemy.health <= 0) {

        logAction("Enemy defeated!");

        // Heal 20% after battle

        const healed = Math.floor(player.health \* 0.2);

        player.health += healed;

        alert("Victory! You healed " + healed + " HP.");

        showStoryNode("afterBattle");

    }

}

function logAction(msg) {

    const log = document.getElementById("battleLog");

    const entry = document.createElement("p");

    entry.textContent = msg;

    log.appendChild(entry);

    log.scrollTop = log.scrollHeight;

}

document.getElementById("startGameBtn").addEventListener("click", () => {

    showStoryNode("start1");

});