1. Felhasználói dokumentáció
   1. Szoftveralkalmazás feladata
      1. Asztali alkalmazás

Az asztali felületet kifejezetten magasabb hozzáféréssel rendelkező személyeknek (pl.: adminok, stb) lett kifejlesztve a szoftver és annak egységeinek karbantartására. Az admin fiók beléptetését követően, az alkalmazásban hozzáféréshez jut a szoftver által használt adatbázis rekordjaihoz és azokon módosítani.

Az admin felhasználók jogosultak az adatbázisban rekordok létrehozására, keresésére, módosítására és törlésére. Az adatbázisban külön, táblázatokba vannak rendezve bizonyos típusú adat egységek, rekordok: rendelések, felhasználók stb. ezeket a táblákat más-más alkalmazási felületen tudják átalakítani.

* + 1. Webes alkalmazás

Webalkalmazás publikus felhasználói használatra lett tervezve, ahol különböző termékeket, szolgáltatásokat rendelhetnek, vásárolhatnak meg valamint nézhetnek utána.

A web alkalmazás célja, hogy hivatalos webáruházként álljon rendelkezésre a kliensek számára, ebből fakadóan az ennek megfelelő funkciókkal rendelkezik: felhasználói adatok kezelése, védelme; termékek és/vagy szolgáltatások vásárlásának, rendeléseinek ellenőrzése; igényes felhasználóbarát felület; szoftverfutási hibák elhárítása.

* 1. Futási környezet

Mindkét alkalmazási felületre - mobil, asztali eszközre - lett tervezve:

* naprakész technológiák
* technikák beépítése
* futásbéli szoftver hibák lekezelése
  + 1. Minimális rendszerkövetelmények

|  |  |
| --- | --- |
| **Asztali alkalmazás** | |
| **Operációs rendszer** | **Hardver követelmények** |
| Windows 10 | * CPU: ARM64 vagy x64 processzor; Négymagos vagy jobb ajánlott. Az ARM 32 processzorok nem támogatottak. 64 bites operációs rendszerben futtatható kizárólag * RAM: 8 GB * SSD: 850 MB szabad tárhely. * Videókártya: 1366 x 768 felbontás * Stabil internet kapcsolat |
| Android 21.0 | * CPU: x86\_64 * RAM: 4GB * Képernyő felbontás: 320 x 480 pixel vagy nagyobb * Stabil internet kapcsolat |
| IOS 11.0 | * CPU: Apple A7 processzor vagy egyéb fejlettebb 64 bites processzorok. * RAM: 8 GB * Képernyő felbontás: 320 x 480 pixel vagy nagyobb * Stabil internet kapcsolat |

|  |  |
| --- | --- |
| **Webes alkalmazás** | |
| **Operációs rendszer** | **Hardver követelmények** |
| Windows 10 | * CPU: Dual-core Intel/AMD vagy jobb 64 bites processzorral és operációs rendszerrel futtatható kizárólag * Képernyő felbontás: 400 x 400 pixel feletti képernyő arány * RAM: 4 GB * Stabil internet kapcsolat |
| Android 21.0 | * CPU: x86\_64 * RAM: 4GB * Képernyő felbontás: 400 x 400 pixel feletti képernyő arány * Stabil internet kapcsolat |
| IOS 11.0 | * CPU: ARM64 vagy x64 processzor; Négymagos vagy jobb ajánlott. Az ARM 32 processzorok nem támogatottak * RAM: 8 GB * Képernyő felbontás: 400 x 400 pixel feletti képernyő arány   Stabil internet kapcsolat |

* + 1. Böngészők

Webalkalmazás használatához alkalmas böngészők, az alábbi listában találhatók:

|  |
| --- |
| **Böngésző** |
| Chrome |
| Firefox |
| OperaGx |

A webalkalmazás működéséhez az alábbi böngészők legújabb verzióit szükséges feltelepíteni.

* 1. Használat
     1. Az alkalmazás indítása és használata

Az asztali alkalmazás elindítása esetén az eszköz rendszer mappáiban válassza ki azt a mappát, ami tartalmazza az projekt fájlt. Az alkalmazás projekt mappáján belül bin\Debug mappában válasza ki azt, a mappát és .exe fájlt amilyen OS rendszer fut az eszközön.

Mind asztali és webes részen több oldal vagy felület között lehet váltani. Asztali applikáció nem lehet publikusan letölteni, hanem egyedileg kapnak a felhatalmazót dolgozok lehetőséget a letöltésre és így az alábbi kezelő felületekhez férnek hozzá: Kezdőlap, Felhasználói oldal, Rendelési oldal, Értékelési oldal, Termék oldal.

Mielőtt a használhatná az alkalmazást, igazolnia kell admin fióknévvel és jelszóval a személy azonoságát. Két bemeneti mezőbe kell megadnia a fent megemlített adatokat, majd utána a belépés gombbal tovább mehet a kezdőlapra.

A felhasználói oldalon a digitálisan eltárolt felhasználókat és azokat alkotó adatokat tudja elérni. Kereshet közöttük, hozzáadhat új felhasználókat, módosíthat az adataikon, vagy kitörölheti azokat.

A rendelési oldalon egy adott szolgáltatásra, termékre le adót digitális rendelést lehet megtekinteni. Kereshet közöttük, hozzáadhat új rendeléseket, módosíthat az adataikon, vagy kitörölheti azokat.

Az értékelési oldalon a felhasználóktól származó értékelésekhez, visszajelzésekhez férhet hozzá. Kereshet közöttük, hozzáadhat új visszajelzéseket mint admin bizonyos kérdésekre, válaszokra, módosíthat az adataikon, vagy kitörölheti azokat.

A termék oldalon az áruházban található szolgáltatásokat, termékeket ki listázhatja. Kereshet közöttük, hozzáadhat új termékeket, módosíthat az adataikon, vagy kitörölheti azokat.

A felületek tetején található bemeneti mezőknél tudjuk megadni azokat az adatokat melyeket az adatbázisban létrehozásra vagy módosításra kerülnek, kivétel a törlendő adat sorok melyeket a felület listáján ki lehet választania egérrel, vagy érintőképernyő esetén érintés által. A módosítások elvégzéséhez a felületek alján található gombok használatával lehet az adatbázist megváltoztatni.

A webes applikáció használatakor ügyeljen a dokumentáció megemlített böngészők használatára, hibák elkerülése érdekében.

Az alkalmazás elindításakor

Az áruházban, kivétel nélkül regisztrálás szükséges kifejezet funkciók, tevékenységek elérhetővé válásához pl.: rendelés, fiók kezelés stb. Ezt, a regisztráció fülnél vagy ha fiókot igénylő funkciókat próbál használása esetén tudja, elérni a regisztrálás megkezdését.

Fiókkal rendelkező vevőink továbbá kosaruk, fiókjuk adatainak tartalmán könnyen és rugalmasan tudnak módosítani, a szempontnak megfelelő fülön, ráadásul még véleményt is tudnak a termékekhez csatolni, így az általuk megvásárolt árukkal kapcsolatos tapasztalataikat meg tudják osztani komment formájában szöveggel és 1-10 csillagos értékeléssel. Ezeket, a változtatásokat a módosítások beküldése után - eltároljuk, felül írjuk vagy töröljük az adatokat – az adatbázisunkban.

A fenti fiókkal rendelkezőkkel ellentétben az árukészlet termékei között, keresni és ahhoz kapcsolatos információkat megtekinteni minden az áruházat látogató személy számára biztosítót.

* + 1. Alkalmazás bemenete

A szoftver program elemeiben az adatokat a kattintás, érintés és fizikai vagy digitális billentyűzetről olvassa be.

* + 1. Alkalmazás kimenete

Webalkalmazás esetén a kimeneteket az adatbázisban vagy a weboldalon láthatjuk, mint visszajelzés, pl.: új rendelés, fiókba való beléptetési hiba.

**Simon Soma Benedek – Flappybird dokumentáció**

1. Felhasználóbarát környezet:
   1. Reszponzív megjelenítés: mobil, tablet, asztali gép, stb.
   2. A játék élményhez méretezés különböző eszközökhöz.
   3. Zoomnál és képernyő méret változás esetén külön méretezés.
   4. A felhasználó informálása a játék használatáról.
2. Design: a játék kinézetének, minőségbeli feljavítása.
   1. Weboldal háttérszíne
   2. A játékablak hátterének látsz hatósága, méretezése
   3. Játék elemek: madár, oszlopok stb. - eszköz a méretre igazításra
   4. Ugráson való igazítás, oszlopok sebességének képernyő mérethez mért sebesség.
3. Megjelenése: A webáruházból való, átirányítás internet megszakadása esetén.
4. Felhasználói leírás: A felhasználó számára a játék kezelése
   1. Játék indítása - Enter (vagy koppintás, kattintás) bevitel esetén indul el a játék.
   2. A madár ugrásához: a Szóköz, X, és a felnyíl billentyű valamint az érintés event mobil alkalmazására
   3. Game Over kimenetel esetén: X, fel nyíl, Szóköz és a Restart gomb az újrakezdés
5. Játék működése: A játék müködésében három fájl játszik szerepet.
   1. flappybird.html: A Html fájl leíró tartalmat tárol a böngészőnek, ami játék megjelenítésért felel, pl.: canvas, képek, méretek stb.
   2. flappbird.css: A CSS tartalmában a weboldal tartalmának kinézetében járul hozzá a programhoz, pl.: színes háttér, háttérkép, méretek stb.
   3. flappbird.js: A Javascript fájl a böngésző interakció részével kezel le eseményeket pl.: Felhasználói billentyű bevitel.

5.1 Mi ez a játék és hogyan működik?

A flappybird játék a .Gears programfejlesztő csapat készített és a játék ötlete Dong Nguyen eszmei tulajdona, ami 2013-ban került ki a játékpiacra, és 2014-ben lett a világpiacon kellendő mint mobiljáték.

A játék side-scroller technikát használ, ami azt, jelenti, hogy a képernyő követi a játékos mozgását a digitális világban, pl.: a madárt követi képernyő és úgy változik a pálya. Ez a módszer a régebbi játékteremben használt játékokra volt, jellemző.

Hogyan müködik? A játékban „flappy” a madár célja minél több pont összegyűjtése, amihez a játékosnak minél több cső között át kell „repülni”, ahhoz hogy pontot kapjon.

Ezért logikai, kordinációs és koncentrációs készséget is igényel a játék ahhoz, hogy minél több ponttal zárja a játékos egy-egy játék menetét.

A madár alapból telefonon felugrani érintésre tud, a mi változatunk ezt kiegészíti billentyűzetről, egérről megadott bemenetekre.

Hogyan jelenik meg ez a mi projektünkben?

…

Az internetkapcsolat megszakadása esetén a játék fájljai a böngészőben betöltésre kerülnek, és az internet visszatérése alkalmával annak az oldalnak visszahozása ahol a felhasználó utoljára tartózkodót.

A főbb böngészők nagy része, nem tesz nagy különbséget a kattintás és az érintés esemény érzékelésében, ezért kód és futásidő szempontjából előnyős ezt a fajta eseményt használni több helyen, viszont ez szűkíti a használhatóságot böngészőkre.

A bemenetek lekezelésénél: Érintés, kattintás kisebb gravitációt, ugrást eredményez, mint a billentyűzeti bemenet. Ennek oka felhasználói minőség javítása, hiszen egéren és billentyűzeten eltérő játék kézbe vehetősége, kézre állása irányítás alapján.

A játékfelület: Egy kisebb ablak az oldalon, ami megjelenítésért felel a játék elemeinek.

A játékablak automatikusan felvesz egy méretet: mobil és asztali eszközök képernyőihez. A 547x640 pixeles böngésző oldalak,ablakok esetén a képernyő 72%-át kiteszi ha a fent meghatározott ablak arány nem kisebb vagy egyenlő a fenti feltétellel pixelben.

Ez a meghatározás lefedi így, az asztali és a mobil eszközöket kép mérethez igazítva. Pl.: ki tölti a képernyőt az ennél kisebb vagy megegyező eszközök, böngésző oldal esetén.

Futtásidő - a madár ugrása és a csövek sebessége – függ, a képernyő mérettől de nem csak, ettől függ ha nem attól is hogy a bevitel típusa kattintás, érintés vagy billentyű volt-e, ez ugyan is más-más ugrást add a madárnak a kezelhetőség érdekében.

Kimenetek: A madár ütközése a csővel, vagy kiesése a pályáról a játék menet végét jelenti. Ez után a megszerzett pontszámok megmaradnak, de a játékot az elejéről kell kezdeni.

Hogyan megy ez végbe?

A játék folyamán egy vászon Html Objektum Elemet használunk fel a megjelenítésre, ami azért különleges hiszen ezt, az elemet kitölteni, „festeni” Javascript kóddal. Ennek, a játéknak a motorja, logikai komponensei a javascript kódban valósulnak meg és a Html vásznon jelenik meg.

Milyen komponensek, kód részletek?

A játék alapváltozói: //browser sizes:

let windowWidth = window.innerWidth;

let windowHeight = window.innerHeight;

//board

let board;

let boardWidth = 457;//image width

let boardHeight = 640;//image height

let context;//drawing on the canvas.

let zoom;

//bird

let birdWidth = 170;// width/height = ratio --> Actual bird image size.

let birdHeight = 119;//Same just in smaller version. The image is not "deformed".

let birdX = boardWidth/8;

let birdY = boardHeight/2;

let birdImg;

let bird = {

    x: birdX,

    y: birdY,

    width: birdWidth,

    height: birdHeight

}

//pipes

let pipeArray = [];

let pipeWidth = 64; //width/height ratio = 384/3072 = 1/8

let pipeHeight = 512;

let pipeX = boardWidth;

let pipeY = 0;

let topPipeImg;

let bottomPipeImg;

//physics:

let velocityX = -2.5;//pipes left moving speed

let velocityY = 0;//bird jump speed

let gravity = 0.4;

let gameStart = false;//the game started

let gameOver = false;//the game ended

let restartButton = document.getElementById("restart");

let score = 0;

let maxScore = score;

let time = 2500;

let speed = 4.5/(time/1000); //calculates the pipes moving speed based on the time and velocity ratio;

Ezek a változók felelnek a játék működésében az elemek méreteiért, helyezeteiért – cső szélessége, magassága, X,Y koordináttái; tárolásáért – a csövek tömb, karakterek, elemek kinézete; a játék fizikájáért- gravitáció, pontszám, sebesség…

A játék betöltése az ablak (window) objektum betöltése során valósul meg amikor is több alapvető beállítás lesz megszabva. A képek, elemek megrajzolása betöltése, a canvasResize függvény a böngésző méreteihez számítva képez le egy bizonyos vásznat méretben.

A játék elindítása:

window.onload = function(){

    board = document.getElementById("board");

    board.height = boardHeight;

    board.width = boardWidth;

    context = board.getContext("2d"); // used drawing on the canvas

    console.log(boardWidth);

    canvasSize();

     //draw flappy bird

     //context.fillStyle = "green";

     //context.fillRect(bird.x, bird.y, bird.width, bird.height);

    //load images

    birdImg = new Image();

    birdImg.src = "./flappybird6.png";

    birdImg.onload = function(){

        context.drawImage(birdImg, bird.x, bird.y, bird.width, bird.height);

    }

    topPipeImg = new Image();

    topPipeImg.src = "./toppipe.png";

    bottomPipeImg = new Image();

    bottomPipeImg.src = "./bottompipe.png";

    context.fillStyle = "white";

    if(gameStart === false)

    {

        document.addEventListener("keypress", startGame);

        document.addEventListener("touchend", startGame);

    }

}

function startGame(e){

    if(gameStart === false){//ToDo: If there are wrong inputs it could alert the user about it.

        console.log(e.type);

        if(e.code === "Enter" || e.type === "touchend"){

            gameStart = true;

            requestAnimationFrame(update);

            setInterval(placePipes, time);//1500 ms -- every 1.5 seconds

            document.addEventListener("keydown", moveBird);

            board.addEventListener("click", moveBird);

        }

        else{

            alert("Wrong input! Correct: Enter or tap the screen");

            return; //Throws out the user if it's the incorrect input.

        }

    }

}

A játék kezelhetősége, vezérelhetősége a felhasználó által a játék elindítása is mint funkció, ezt billentyű vagy egér kattintás, érintés alapján veszi be mint interakciót, meggyőződve arról, hogy tényleg játszani kíván a játékos, a document.addEventListener függvény segítségével hívódik meg az a függvény mely a kattintást, billentyű, érintés eseményt várja, amikor az eventListener megkapja elindítja a fenti függvényt ami a start beállításokat, paramétereket teszi helyre.

Ez a függvény hozzátartozik szorosan a placePipes segéd fügvényhez, ami a setInterval függvény által bizonyos időnként leképez egy pár csőt, amit hozzáad a pipes tömbhőz amit ki is űrit akkor ha, egy cső bizonyos hogy már nincs benne a játék ablak koordinátái között

A moveBird függvény is ide tartozik ugyan is ez által, indul el a madár mozgása mint definiált adat az eventListener függvény most már meghívja és mozgatja ezzel a madarat, akkor ha a listener megkapja az adót interakciót.

A játék futtatása:

function update(){

    requestAnimationFrame(update);

    if(gameOver)

    {

        return;

        //Ends the loop of calling itself if the game is over.

    }

    context.clearRect(0,0, board.width, board.height);

    //Clears the last frame to load the new.

    //bird

    velocityY += gravity;

    bird.y = Math.max(bird.y + velocityY, 0);//apply gravity to current bird.y, limit the bird.y to the top of the canvas

    context.drawImage(birdImg, bird.x, bird.y, bird.width, bird.height);

    if(bird.y > board.height)//the game met the bird has fallen condition

    {

        gameOver = true;

        restartButton.style.display = "block";

        //Make the restart button visible.

    }

    //pipes

    for(let i = 0; i < pipeArray.length; i++ ){

        let pipe = pipeArray[i];

        pipe.x += velocityX;

        context.drawImage(pipe.img, pipe.x, pipe.y, pipe.width, pipe.height);

        if(!pipe.passed && bird.x > pipe.x + pipe.width)

        {

            score+=0.5;//because there are two pipes: top and bottom! 0.5\*2 = 1, 1 for each set of pipes.

            pipe.passed = true;

        }

        if(detectCollision(bird, pipe))//the game met the bird collided with pipe condition

        {

            gameOver = true;

            restartButton.style.display = "block";

            //Make the restart button visible.

        }

    }

    //clear pipes

    while(pipeArray.length > 0 && pipeArray[0].x < -pipeWidth)

    {

        pipeArray.shift();//removes first element of the game

        //memory overflow prevention.

    }

    //score

    context.fillStyle = "white";

    context.font = windowWidth < 1500 ? "3rem sans-serif" : '45px sans-serif';

    context.fillText(score, 5, 45);

    if(gameOver){

        const textWidth = context.measureText("GAME OVER").width;

        context.font = windowWidth < 1500 ? "3rem sans-serif" : '45px sans-serif';

        context.fillText("GAME OVER", (boardWidth-textWidth)/2 , boardHeight/2 - 20);

        /\*context.fillText("GAME OVER", boardWidth/2, boardHeight/2 - 20);\*/

    }

}

A start beállítások után rekurziós folyamat megy végbe egy update függvényben ami addig fut ameddig a kimenetek egyik esete nem megy végbe (kiesik a pályáról, csövekkel ütközik).

A requestAnimationFrame függvény segít megtartani a változó folyamatok esetén képeket, és rajzokat minősége szerint pixelekben megtartja - új elemeket úgy megrajzolja mintha az animációs pixelek csere által nem romlanak le, hanem megőrzi azt amilyen volt.

Az update-on belül három segédfüggvény található: moveBird, detectCollison, placePipes. Ezen függvényben segítenek, a játék állapotának lekezelésében. A moveBird segédfüggvény által a madár frissített pozíciója kirajzolásra kerül és itt változik maga az X, Y koordintái a madárnak és a csöveknek.

A koordináták felülírása miatt, itt vizsgáljuk a kimeneteleket is: a madár kieset a pályáról vagy ütközött egy csővel. A placePipes függvény és a moveBird nincsenek fizikailag magában az update-ban de logikai alapon jelen vannak a függvény futtatása során mint addEventListener vagy setInterval által hívót függvények, kivétel a detectCollison ami valóban bele van írva a függvénybe, és vizsgálja a madár és legközelebbi cső koordinátáit, hogy azok megegyeznek-e valahol.

Memória túltelítettség miatt, ha egy cső X koordinátája egyenlő a cső mínusz szélességének a vásznon akkor azt az elemet törli a csövek tömbből.

Összefoglalva update feladatai:

* játék elemek koordinátáinak frissítése, kirajzolása
* pontszámok kiírása
* a játék menet megszakítása, kiírása

A játék vége, újraindítása:

A játékmenet véget ér a játékos számára, ha teljesíti az egyik kimenetel feltételét. Az eddig összegyűjtőt pontszám ekkor kiírásra kerül és eltárolva lesz addig amíg a játék oldalát el nem hagyja a felhasználó.

A játékos több fajta módon újra indíthatja a játékot, ha a játéknak vége van:

* Billentyűzet: X, Szóköz, Fel-nyíl;
* Egér: Kattintás.
* Érintőképernyő: érintés.
* Újraindítás gomb megnyomásával.

Mindegyik esetben a játék visszaállítja a kezdeti állapothoz a játék paramétereit: pontszámát, a madár eredeti pozícióját, az eltárolt csövek számát, stb.

A játék újraindításáért az újraindítás gomb esetén egy külön restart függvény felel, még a billentyűzeti, egér és érintéses bemenetek esetén a moveBird függvény kezeli ezt le.

function restart(){

    bird.y = birdY;

    pipeArray = [];

    score = 0;

    restartButton.style.display = "none";

    gameOver = false;

}

function moveBird(e){

    if(e.code == "Space" || e.code == "ArrowUp" || e.code == "KeyX" || e.type == "click")

    {

        velocityY -= e.type === 'click' ? 4 : 10;//jump: 3.5 - 7

        gravity = e.type == 'click' ? 0.245 : 0.4;//reset jump grav

        //reset game

        if(gameOver){

            bird.y = birdY;

            pipeArray = [];

            score = 0;

            //max

            restartButton.style.display = "none";

            gameOver = false;

        }

    }

}

Segéd függvények:

moveBird: A madár ugrását kezeli le billentyű, kattintás, érintés esetén valamint az ahhoz tartozó gravitációt és a játék újraindítását ha, az bekövetkezet.

Feladata: Billentyű vagy kattintás, érintés esetén megvizsgálni melyik volt az és ez alapján ugrást és gravitációt generál. Ha, a játénak vége akkor üressé teszi a csövek tömböt, a madár alap helyiértéke beállítása (X,Y), az újraindítás gomb eltűntetése.

detectCollison: Ez a függvény meghatározza a madár pózíciója és a legközelebbi cső elhelyezkedése által hogy a madár neki ment a csőnek vagy nem.

Feladata: Két paraméter alapján - a cső és a madár – megmondja, hogy a X,Y koordináták és a magasság, szélességüket összevetve metszik egymást.

placePipes: Ez cső elemeket készít mint felső-alsó változatából, ezeket pedig a pipes tömbhőz adja valamint vizsgálja a csövek tehetőségét a szerint hogy nem helyezi-e rá egy másik csőre és hogy van-e elég hely köztük hogy a madár átférjen majd. Addig fut amíg a játék egyik kimenetele által a „vége a játéknak” változó nem igaz.

Feladata: Két cső elem párt készít el – lenti és fenti – és add hozzá a csövek tömbhöz. Figyeli a lehelyezett és a lerakandó csövek helyzeténél, hogy ne rajzolja egymásra a csöveket és a távolság köztük legyen elég a madárnak.

canvasResize: A függvény két alkalommal kerül használtra - a játék elindulásakor és ha a böngésző méretében változás megy végbe, pl.: zoom…

Feladatai:

* beállítja a méreteit a vászonnak, csöveknek, madárnak
* Megjelenít más-más tájékoztató szöveget: pontszám, utasítások valamint az újraindítás gombot, ha újra méretezés történik - zoom, stb.

birdResize: Újra méretezi a madarat vászon méretének és madár képének arányai alapján.

Feladata: A madár méretének átállítása az eredeti vászon és madár kép közötti arányok alapján szélesség és magasság szerint.

pipeResize: Ugyan úgy, újra méretezi a csövek méretét a csőnek képe és a vászon mérete arányaiból

Feladata: A jelenlegi cső méretei és a lukak közötti helyből kiszámítani a felső és az alsó cső magasságát, szélességét hogy tudva az arányokat (cső – max.2/3 magasság, szélesség 1/12 a vászónak) mennyit kell változtatni rajta. Valamint a játék menet közbeni képernyőméret változás (zoom, telefon eldöntése…) esetén azonnal átállítsa a csövek méretét az arányoknak megfelelően.

Kompatibilis böngészők:

A jelenlegi kódbéli megvalósítás alapján az alábbi böngészők fejletek technólogiában, könyvtárakban a szoftver futtatásához:

* Firefox,
* Google Chrome
* Microsoft Edge

1. Hiba lehetőségek:
   1. Ha, nem megfelelő a böngészővel, a javascript vagy annak verziója: nem indul el a játék vagy nem fut rendesen.
   2. Kinézeti próblémák mobil, zoom, képernyőméretezés esetén: Press Enter to Start, stb.
   3. A csövek sebességének gyorsasága játékbarát-e: elfogadható vagy nem ideális.
2. Fejlesztési lehetőségek:
3. A csövek sebessége, a madár ugrása állítható legyen felhasználó számára.
4. Különböző nehézségi szintek, pályák, akadályok, kinézetek a játék elemeknek (játékablak, madár, stb.)
5. Rank számítás pontszám alapján, pl.: 580 pont – Metal, 850 pont – Über…

Dátum: 2025.04.14 Simon Soma Benedek

**Flappy Bird – Projekt Összefoglalás (rövidített verzió)**

1. **Áttekintés és fájlok**
   * **HTML (flappybird.html)**: Beállítja a játékvásznat és az elrendezést.
   * **CSS (flappbird.css)**: Stílusolja az oldal hátterét és a vásznat.
   * **JavaScript (flappbird.js)**: Megvalósítja a játék logikáját, a megjelenítést és a felhasználói bemeneteket.
2. **Reszponzív kialakítás**
   * A vászon automatikusan méreteződik mobilra, tabletre és asztali gépre.
   * A canvasResize függvény kezeli a nagyítást és az ablakméret változását.
   * Az érintés/kattintás és a billentyűzet (Space, ↑, X) különböző ugrásfizikát eredményez a jobb élményért.
3. **Játékmenet**
   * Oldalra scrollozó “Flappy Bird” klón: koppints vagy nyomj egy gombot, hogy a madár átrepüljön a csőrések között.
   * A csövek időközönként megjelennek; ütközés vagy lezuhanás a játék végét jelenti.
   * Minden átrepült csőpárral egy pont jár.
4. **Főbb függvények**
   * **startGame**: Enter vagy érintés kezdi el a játékot.
   * **update**: Folyamatos ciklus, ami törli a vásznat, alkalmazza a gravitációt, mozgatja a madarat és a csöveket, ellenőrzi az ütközést, és kirajzolja a pontszámot vagy a „GAME OVER” feliratot.
   * **placePipes**: Időközönként új csőpárokat generál és helyez el.
   * **moveBird**: Kattintás/érintés vagy gombnyomás hatására ugratja a madarat (illetve újraindít, ha vége a játéknak).
   * **detectCollision**: Ellenőrzi a madár és a csövek találkozását a koordináták alapján.
   * **restart**: Visszaállítja a madár pozícióját, a pontszámot és a csövek tömbjét.
5. **Felhasználói vezérlés**
   * **Indítás**: Enter, képernyőkoppintás vagy kattintás.
   * **Ugrás**: Space, ↑, X, koppintás vagy kattintás.
   * **Újraindítás**: X, Space, ↑, koppintás/kattintás vagy az újraindítás gomb.
6. **Jövőbeli fejlesztések**
   * Állítható nehézség: csősebesség és madárugrás erőssége.
   * Több pálya, kinézet és akadálytípus.
   * Pontszám szerinti ranglista.

*Dátum: 2025. április 14.*  
*Szerző: Simon Soma Benedek*