



Szkubisznekk
csapat

Esemény vezérelt programozás



Szukibisznekk csapat
[Mario Kópia]

Készítette:

Kunovics Dávid

Németh Ádám

Ujláb Bálint



Tartalom

A programról:	3
Követelmény elemzés:	3
Funkcionális követelmények:	3
Nem funkcionális követelmények:	3
Jegyzőkönyv az alkalmazásról	4
Utólagos kiegészítés a jegyzőkönyvbe:	5
Java	6
Lightweight Java Game Library	6
JOML – Java OpenGL Math Library	6
Dominion - Entity Component System (ECS)	6
Fejlesztői környezet	6
Tiled pályakészítő	7
A program felépítése	8
A program fejlesztési terve	8
A játék	9
A játékról képek	9
A menü	9
Hangrendszer	9
Karakterkinézetek	10
A pályán	10
Pontrendszer	11
Pálya vége	11
GitHub oldala a programnak:	12
Kapcsolat	12



A programról:

A "Mario Kópia" egy 2D-s platformer játék, amelyben Dobozi Lacikát irányítva különböző pályákon követhetjük végig kalandjait. Közben pontokat gyűjtünk, hogy elérhessük a bámulatos 1 eurót. A program az Eötvös Lóránd Tudományegyetem Informatika karának Esemény vezérelt programozás órájára a Szkubisznekk csapat beadandó feladata. Könnyen bárki által érthető és ismert játékot szerettünk volna készíteni, ami az óra követelményeinek megfelel. Az alkalmazás hosszan tartó támogatással garantálja a rendszeres hibajavításokat és bővítéseket, amik meghosszabbítják a játék élvezetes játszhatóságát.

A projekten dolgozó csapat folyamatos fejlődés mellett írta programot ami a saját tudásuk bővítésére is szolgált. A projekthez tartozó GitHub repository, weboldal, Gantt diagram, UML diagram folyamatos frissítéssel rendelkezett.

Követelmény elemzés:

Nem támasztunk nagy elvárást a célszámítógépek specifikációjától, mivel a program grafikai megjelenése hasonlítani fog a platformer játékok nagyrészt, viszont ésszerűen kell tartanunk, hogy a java nyelv játékok fejlesztésére csak korlátozottan alkalmas.

A megbízhatóságot szem előtt tartva minden pálya után menti a program az előrehaladást és az összegyűjtött pontokat.

Funkcionális követelmények:

A program támogatni fogja billentyűzetről és a kontrollerekről érkező irányítást valós időben. Képes lesz tárolni a felhasználó előrehaladását a játékban. Szükséges a Java Runtime Environment.

Nem funkcionális követelmények:

Fontos szempont a fejlesztés során a játék egyértelműsége és mindenki számára izgalmas és szórakoztató játékmenete.



Jegyzőkönyv az alkalmazásról

Miről szóljon a projektünk?

Elsődlegesen egy játékot képzelünk el, amely 2 dimenziós térben játszódik. Igazából a jelenleg széles körökben ismert Mario játéknak egy kezdetleges megalkotása lenne. Ez reméljük belefér a tudásunkba és az egyetem mellett sikerülni fog összerakni.

Ugyan úgy a Mario lesz a főszereplője?

Nem, szeretnénk egy egyedi karaktert belevinni a játékba. Több ötlet is felmerült de még nem sikerült döntést hozni, úgyis ráér a kinézet legyártásakor.

Milyen lesz a játékmenet?

Pályarendszert képzelünk el, amelyben a Mario alapjaira épülve, easter-eggekkel kiegészítve.

Grafikát, hogy oldjuk meg?

Előre elkészített textúrákat használnánk, amelyeket saját magunk rajzolunk meg és tölti be a program.

Hány pályát próbáljunk meg megírni?

Egyenlőre nem szeretnénk olyan célt meglőni amit nem tudunk teljesíteni szóval nagyjából 4 darab pályával számolunk egyenlőre, de az a cél, hogy minél többet összerakjunk.

A pályákat milyen formába tároljuk?

Könnyebb kódolás és bővíthetőség miatt arra gondoltunk, hogy a pályákat nem közvetlen a programba íránk bele hanem azt is, mint a textúrákat egy előre elkészített fájlból töltenénk be, amelyben karakterekkel ábrázolnánk a különböző objektumokat és alkotóelemeket a pályákon.

Pont gyűjtési rendszer is ugyan úgy lesz bent?

Az volt az első ötlet, hogy legyen szkubisznekk a gyűjthető pont, ahogy a nevünk is magába foglalja, de inkább forint mellett döntöttünk. Minden pályán lehet gyűjteni valamennyi forintot, amit a pálya végén ír jóvá a játék, tehát amit összegyűjtöttünk a pályán azokat elveszítjük, amikor leesik a játékos.

Lesz benne valamiféle játékhoz tartozó útmutató a felhasználónak?

Mivel egy gyakorlatilag mindenki által ismert játékról van szó, nem terveztünk konkrétan útmutató pályát létrehozni.



Szkubisznekk
csapat

Esetleg valami extra?

Ezeket a játék fejlesztése alatt szeretnénk kitalálni, de gyűjthető tárgyakat lehet be fogunk vezetni, amelyeket ha mindegyiket összegyűjtünk a végén egy nagyon különleges kinézet fog megnyílni játékosunk számára.

Utólagos kiegészítés a jegyzőkönyvbe:

A kitűzött célokat sikerült megvalósítani, és tökéletesíteni. Dobozi Lacika lett a főhőse a játékunknak és utólag bekerült a háttérzene gondolata amelyet meg is sikerült valósítanunk.



Java

A Java általános célú, objektumorientált programozási nyelv, amelyet a Sun Microsystems fejlesztett, majd a céget felvásárolta az Oracle. Bár a nyelv neve kezdetben Oak (tölgyfa) volt, (James Gosling, a nyelv atyja nevezte így az irodája előtt növvő tölgyfáról), később kiderült, hogy ilyen elnevezésű nyelv már létezik, ezért végül Java néven vált ismertté. A Java szó a Oracle védjegye. Ennélfogva engedélye nélkül nem használható mások által kifejlesztett termékek megjelölésére; még például Java-szerű... stb. összetételben sem, mert ez a védjegyjogosult jogaiba ütközik.

A Java alkalmazásokat jellemzően bájt kód formátumra alakítják, de közvetlenül natív (gépi) kód is készíthető Java forráskódból. A bájt kód futtatása a Java virtuális géppel történik, ami vagy interpretálja a bájt kódot, vagy natív gépi kódot készít belőle, és azt futtatja az adott operációs rendszeren.

A Java nyelv a szintaxisát főleg a C és a C++ nyelvektől örökölte, viszont sokkal egyszerűbb objektummodellel rendelkezik, mint a C++.

Lightweight Java Game Library

Az LWJGL egy Java könyvtár, amely lehetővé teszi a többplatformos hozzáférést a népszerű natív API-khoz, amelyek hasznosak a grafikus (OpenGL, audio (OpenAL) alkalmazások fejlesztésében. Ez a hozzáférés közvetlen és nagy teljesítményű, ugyanakkor egy típusbiztonságos és felhasználóbarát rétegbe van csomagolva, amely megfelel a Java ökoszisztémának. Az LWJGL egy nyílt forráskódú szoftver, amely ingyenesen elérhető. A legfrissebb változata a 3-as ami Java 8 vagy későbbi verziót igényel. C könyvtárakat enged használni javában.

JOML – Java OpenGL Math Library

Az JOML egy Java matematikai könyvtár amely OpenGL rendereléshez szükséges kalkulációkat képes elvégezni. Githubon található meg és több fejlesztő is dolgozik rajta. Célja, hogy könnyen használható funkciókban gazdag és könnyen használható lineáris algebrai műveleteket biztosítson, amelyekre bármely 3D-s alkalmazásnak szüksége van. Továbbá megpróbál alacsony követelményekkel dolgozni a végrehajtási környezettel szemben mivel Java 1.4-gyel is kompatibilis és nem használja a JNI-t. 3D-s és 2D-s grafikus környezetet is létre tud hozni különböző kameraszögekkel. Ebből nekünk a 2D-s volt szükséges.

Dominion – Entity Component System (ECS)

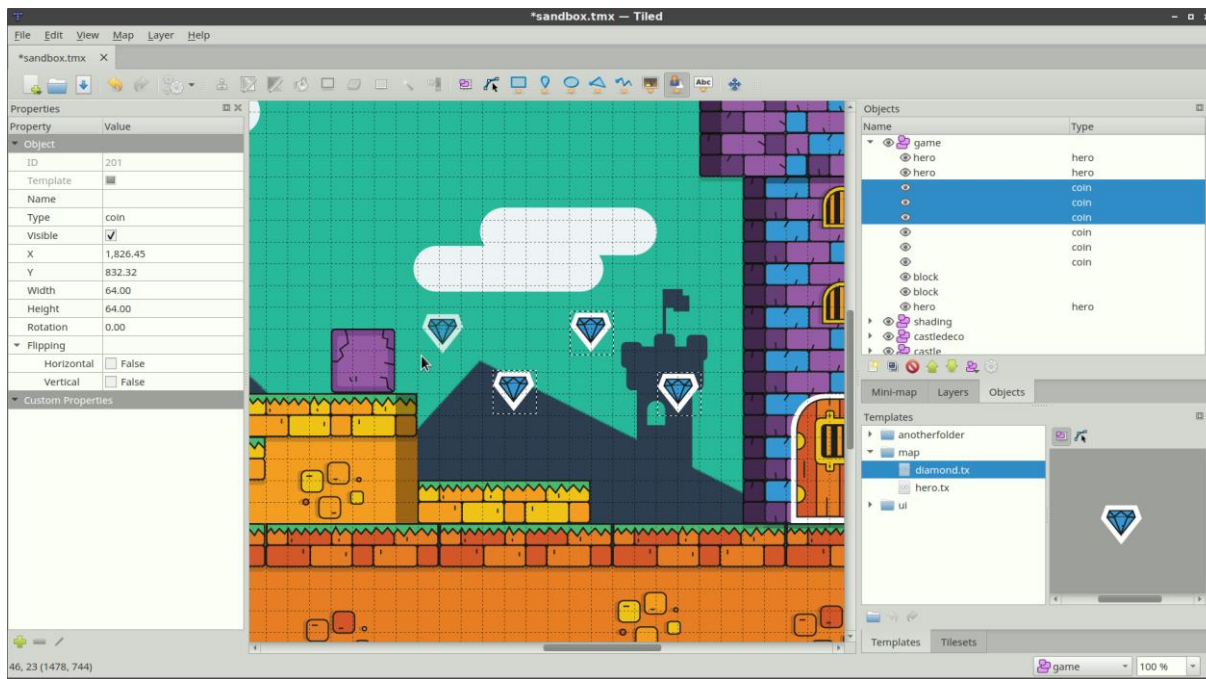
A Dominion egy entitáskomponens könyvtár Java számára. Az Entity Component System (ECS) architektúra támogatja az adatorientált programozást. Minden az adatokról (összetevőkről) és az adatokon működő első osztályú funkciókról (rendszerekről) szól. A program felgyorsítására szolgál elsősorban.

Fejlesztői környezet

A fejlesztés IntelliJ IDEA 2022.2-ben történt és a git rendszerével történt a verziókövetés. A projekt népszerűsítésére Wix weboldalt hoztunk létre.

Tiled pályakészítő

A Tiled egy 2D-s pálya szerkesztő. Elsődleges használata a „tile map” készítése különböző formákban akár kép texturák felhasználásával is, valamint különböző módokat ad extra információ elhelyezésére a pályán.



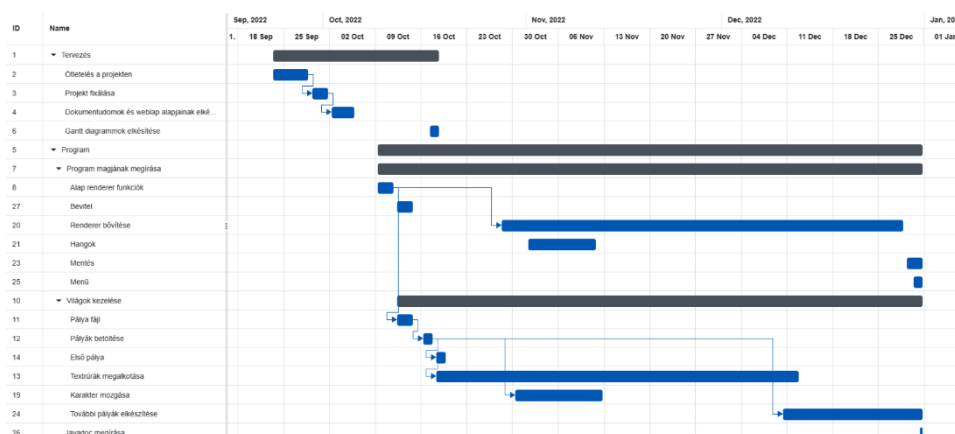
A program felépítése

Ezen UML osztálydiagrammos ábrázolása a program felépítésének egy IntelliJ bővítmény segítségével lett létrehozva.



A program fejlesztési terve

A program folyamatos fejlesztéssel hoztuk létre, előre elterveztük milyen ütemben szeretnénk az egyes lépésekkel haladni. Ezen előrehaladásokat Gantt diagrammon ábrázoltuk majd miután sikerült a megvalósítása az egyes részeknek frissítettük a diagrammot.



Tesztelési terv:

A programot több felhasználó (külsősök) tesztelte több számítógépen, a hibákat kijavítottuk, újra teszteltettük majd a hibátlan verziót publikáltuk.



Szkubisznekk
csapat

A játék

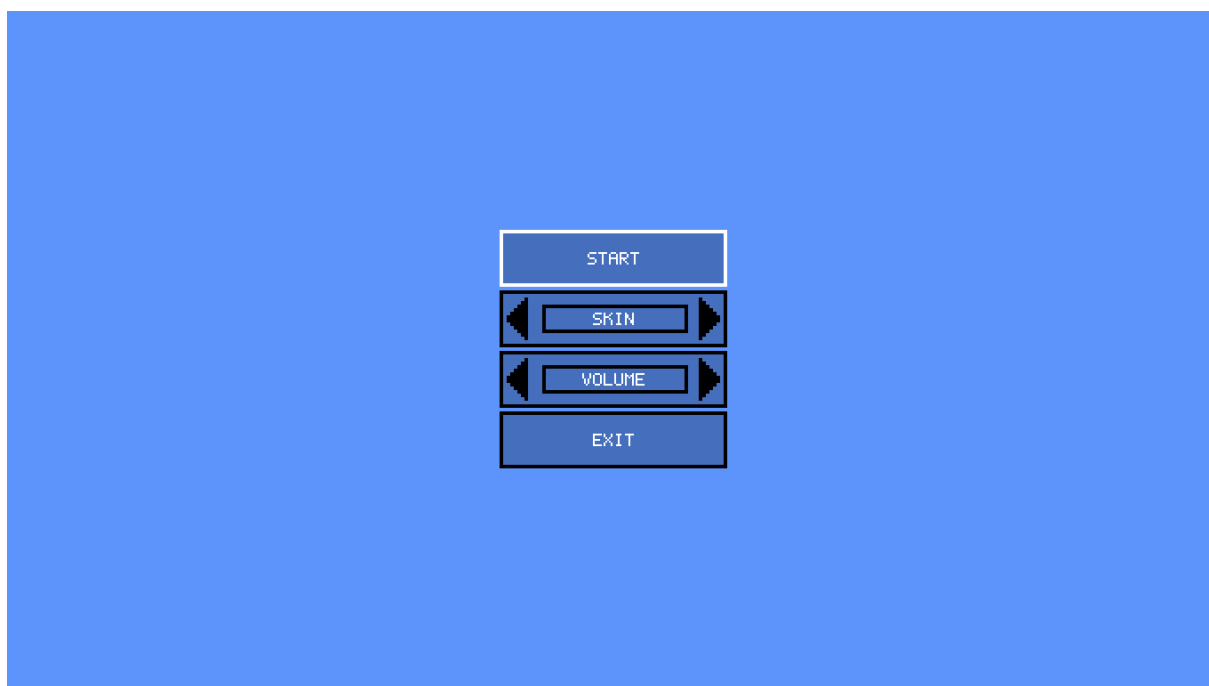
Játékunkban pályákon kell végig haladni sorban, ahol nem szabad lezuhanni (halál). Az akadályokat leküzdve kell pontokat gyűjteni és eljutni a pálya végére majd elkezdni a következőt. A pálya kezdésekor menti a játék hányadik pályán tartunk, valamint az aktuális pontegyenleget, tehát ha meghalunk vagy félbehagyjuk a játékot akkor az aktuális pálya elejétől folytathatjuk.

A játékról képek

(részletesebb leírás a felhasználói kézikönyvben)

A menü

A menüben el tudjuk indítani a játékot, valamint be tudjuk zárni. A főhősünk kinézetét is tudjuk változtatni, amely pályakezdekor frissül, továbbá tudjuk változtatni a hangerőt.



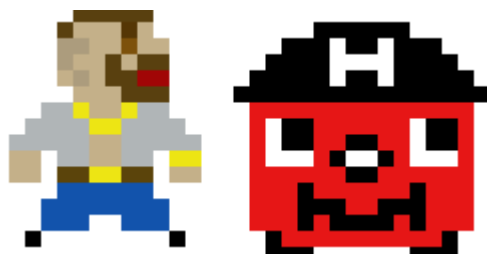
Hangrendszer

A játéknak van háttérzenéje. A hangerőt a menüben lehet állítani. A pontok (forint) gyűjtésénél és a főhős halálakor is van hangeffekt.

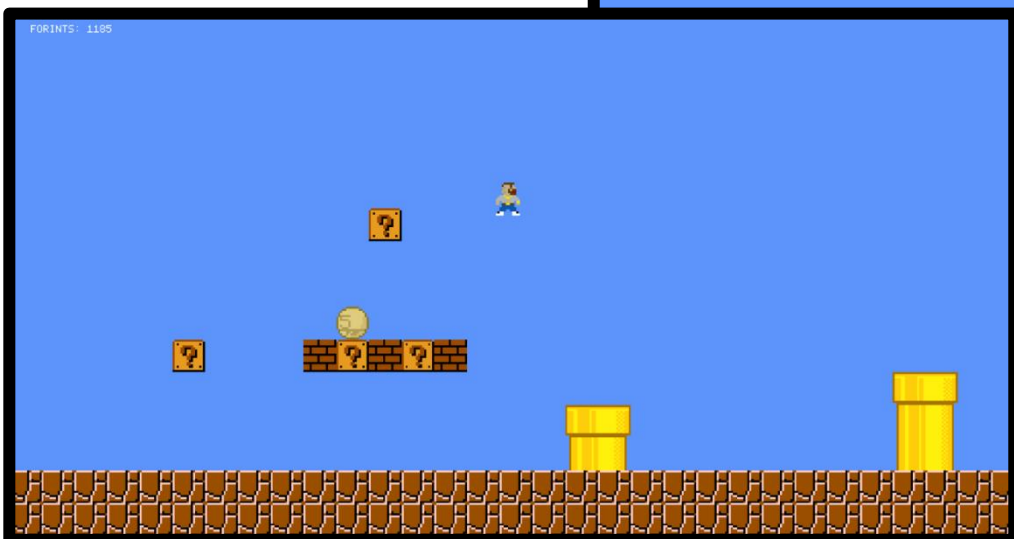
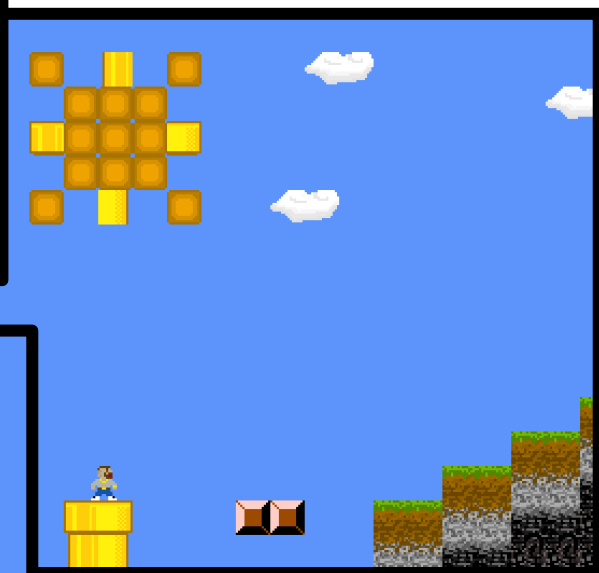


Karakterkinézetek

Kétféle kinézet található a játékban. Az elsődleges Dobozi Lacika, a főhős. Rajta egy különleges kinézetet is hozzáadtunk, amely nem más, mint Henry the Hoover!



A pályán





Szkubisznekk
csapat

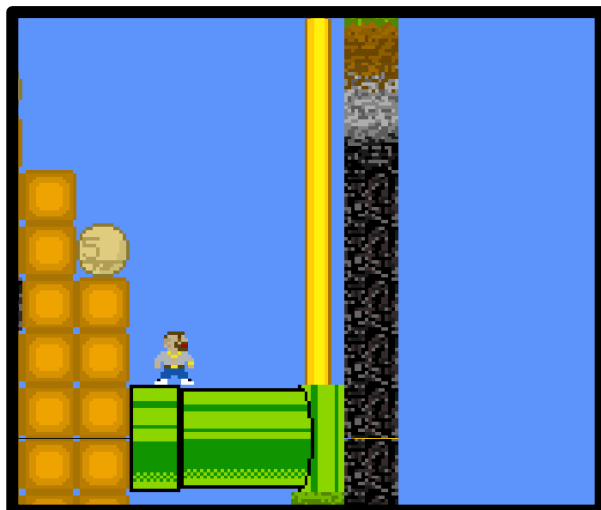
Pontrendszer

A pontokat a Forintban lehet gyűjteni.



Pálya vége

A pálya végét egy arany rúd jelzi, amelyhez ha hozzáér a karakter a következő pályára dob.



Fejlesztési tervek a jövőre nézve

Sok tervünk van a jövőre nézve. Elsődlegesen több kinézetet és további pályákat szeretnénk kreálni. Ez mind a felhasználói visszajelzésektől függ, mennyire győzi meg a közönséget a játék és az alapján döntenénk a továbbiakról.



GitHub oldala a programnak:

A programmal kapcsolatos minden anyag megtalálható ezen linken.

[GitHub link](#)

Kapcsolat

Bármilyen kérdése, kérése vagy észrevétele lenne kérem keresse fel a fejlesztőket!

Kunovics Dávid Zoltán: SFCJ7H@inf.elte.hu

Ujláb Bálint: AGJ88S@inf.elte.hu

Németh Ádám: CP48Q6@inf.elte.hu