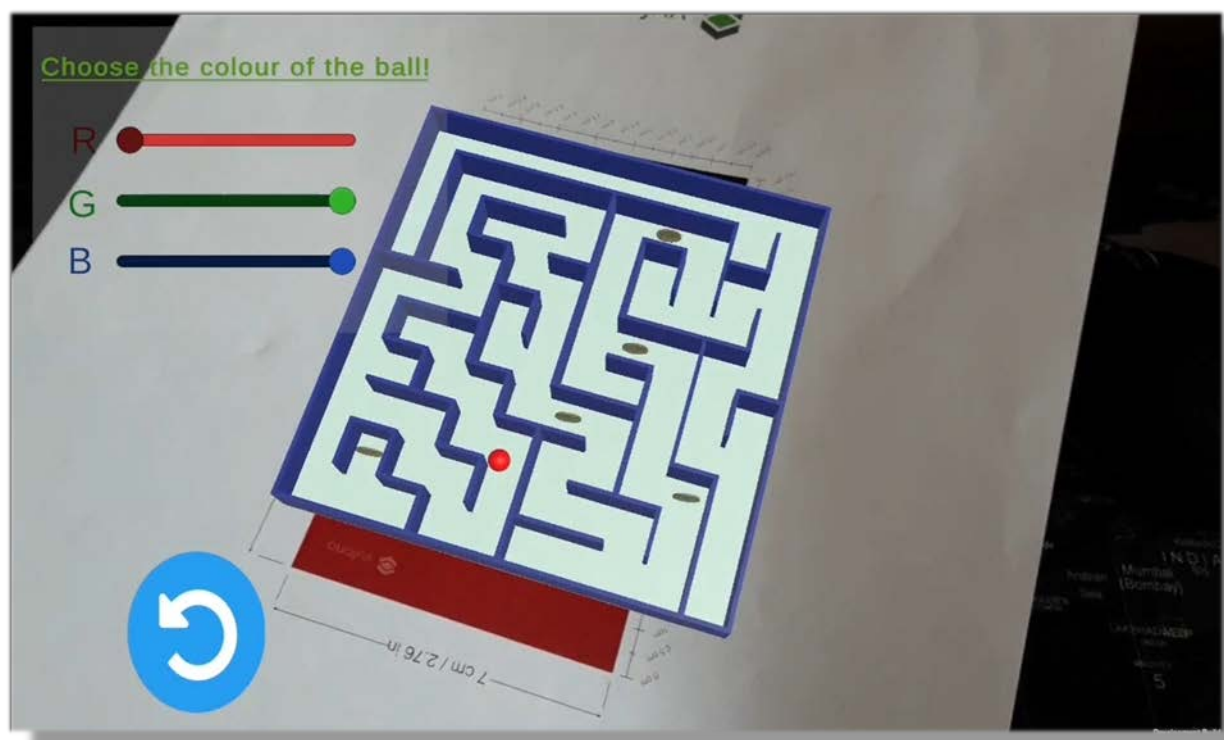


Projekt neve: VufMaze

Konzulens neve: Varnyú Dóra



1. ábra: Kép játék közben

Projektről röviden

E projekt a Budapesti Műszaki és Gazdaságtudományi Egyetemen folytatott ötödik féléves Témalabor c. tárgy keretein belül készült. A projekt célja új területeket megismerése az eddig megismert programozási nyelvek (környezetek, szabályok) felhasználásával.

Az elkészített programban lehetőségünk van egy labirintust kivetíteni magunk elé egy speciális (nyomtatott) kép segítségével. A labirintusban elhelyezett labdát ezen nyomtatott kép segítségével (annak döntésével) lehet mozgatni. A játék célja, hogy a labirintusban megjelenő pénzerméket összegyűjtsük a labdánkkal. Amennyiben ez sikerült, akkor egy „WIN” felirat hívja fel a felhasználó figyelmét a győzelemre.

A játék megnyerése során lehetőségünk van egy új játék indítására a bal alsó sarokban található ikon segítségével. Ekkor egy új labirintus generálódik. (Ennek legenerálása körülbelül 3 másodpercet is igénybe vehet. Ne ijedjünk meg elsőre!)

Továbbá még lehetőségünk van a labdánk színét RGB szinten állítani. Az első csúszka felel a piros, második a zöld és végül a harmadik a kék színért.

Ezen dokumentációban a fentebb leírtak vesszük át általánosan. A projekt során Vuforia-t és Unity-t használtam. Ezekről lesz bővebben szó.

Vuforia

1. Mi is a Vuforia?

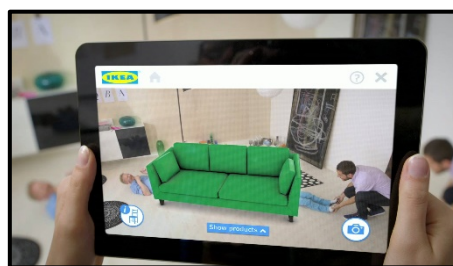
A Vuforia egy kiterjesztett valóságú (AR) szoftverfejlesztői készlet (SDK) mobil eszközökhöz, amely lehetővé teszi a AR alkalmazások létrehozását.

Kiterjesztett valóság (AR: Augmented Reality)

A valóság virtuális kibővítése. Lényege, hogy a világunk kontextusába virtuális tárgyakat helyezünk valós időben. Lásd ezen projekt keretében egy labirintust vetítünk a felhasználó elé, melyet android-os (mobil, tablet) eszközön keresztül követhet. Legnagyobb haszna jelenleg az orvostudományban van.



2. ábra: Snapchat
(<https://studiousguy.com/examples-augmented-reality/>)



3. ábra: IKEA place
(<https://studiousguy.com/examples-augmented-reality/>)

Ebbe a területbe tartoznak az alábbiak is, amelyekkel hétköznapi szinten is találkozhatunk:

- | | |
|---|---|
| <ul style="list-style-type: none"> • Snapchat • HoloLens • Google ARCore • Pokemon Go | <ul style="list-style-type: none"> • Belső dekorációs alkalmazások (ilyen például az IKEA place app) • Rendszerkarbantartás (ahogy a Vuforia vállalatoknak is nyújt különféle megoldásokat) |
|---|---|

Szoftverfejlesztői készlet (SDK: Softver Development Kit)

A fejlesztők számára készült, egy adott program vagy platform szolgáltatásainak használatát, azon futó programok készítését lehetővé tevő készlet.

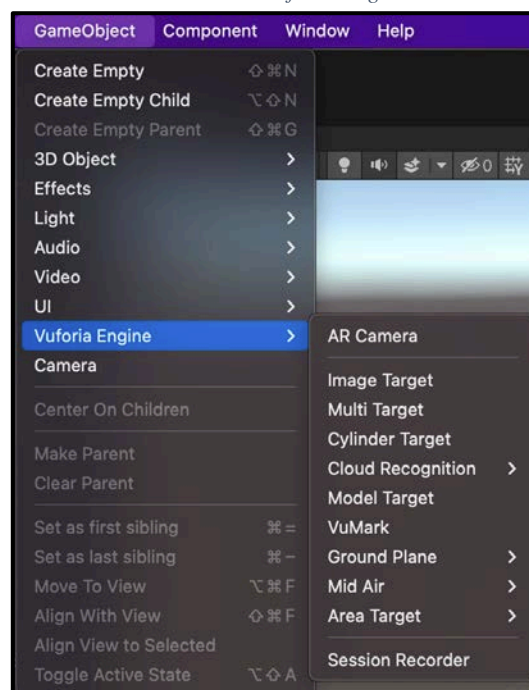
A fejlesztőkészletek a feltétlenül szükséges dokumentáció és könyvtárfájlok mellett sokszor segédprogramokat, vagy akár komplett fejlesztőeszközöket is magukban foglalhatnak. Például a Vuforia első beimportálásával Unity-ben rendelkezésünkre áll egy kész tutorial és egy 'exampleScene' is.

2. Vuforia beimportálása Unity-be

4. ábra: Vuforia Engine

A Vuforia-t Unity keresztül használom. Ennek beimportálása Unity-be a következő lépéseket vonja magába:

- Először is a [Vuforia főoldalán](#) hozzunk létre egy ahhoz tartozó felhasználói fiókot.
- [Töltsük le](#) az aktuális SDK-t '.unitypackage'-ként.
- Végül ezen csomagot importáljuk be a Unity-be: Assets → Import Package → Custom Package
- A Vuforia oldalán ezt követően létre kell hoznunk egy 'App License' kulcsot¹, melyet Unity-be egy általunk létrehozott AR Camera objektum 'Vuforia Configuration' beállításán keresztül illesszünk be. Ezzel aktiváltuk is a Vuforia motorját.



¹ E kulcs létrehozásához jelentkezünk be a Vuforia oldalán, ahol a Develop menüben a License Manager alatt érjük el az összes kulcsot, amelyet létrehozunk. Egy új előállításához kattintsunk a Get Basic gombra.

Csak ezeket követően érhetjük el a Vuforia Engine által nyújtott funkciókat. Ezen funkciók közül a projekt egy Image Target-et használ.

3. Képkövetés

A Vuforia 4 különböző lehetőséget nyújt képekre történő kivetítés során. Lehetőségünk van nyomtatott képekre (**Image Targets**), henger alakú tárgyakra (**Cylinder Targets**), összetett – esetleg több elemből álló tárgyat – (**Multi Targets**) és QR kódhoz hasonlóra (**VuMarks**) kivetíteni a virtuális objektumainkat. (képek forrása: [Vuforia Engine Overview](#) | [VuforiaLibrary](#))

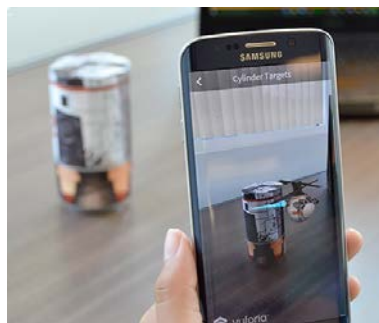
E projekt keretében egy egyszerű, a Vuforia által elkészített nyomtatott képet használok. Így az alkalmazás fejlesztése során Image Target objektummal dolgoztam Unity-ben.

Image Targets



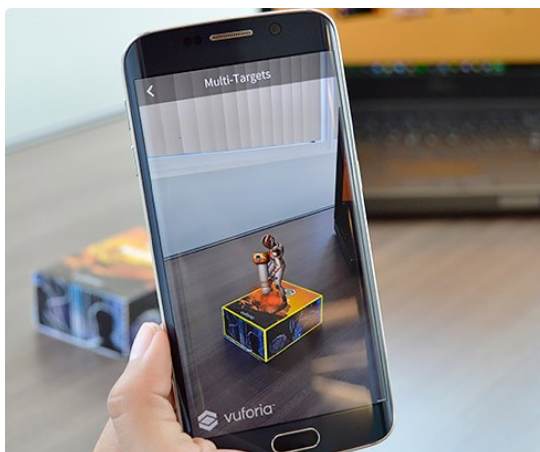
5. ábra: Image Targets

Cylinder Targets



6. ábra: Cylinder Targets

Multi Targets



7. ábra: Multi Targets

VuMarks



8. ábra: VuMarks

4. AR Camera

Minden olyan unity scene-hez (jelenethez), amely nyomon követi és megjeleníti a virtuális tartalmainkat egy AR Camera szükséges, a(z) unity alapkamerája itt felesleges.

5. Vuforia adatbázisok

Minden egyes 'target'-et, melyet telefonnal követünk adatbázisban kell tárolni. Ezen Vuforia adattároló neve a Target Manager. Itt tipikusan 3 különböző típusú adatokat tároló adatbázis közül választhatunk: Device DB, Cloud DB, VuMark DB.

Device Database²

Ezen adatbázisban az előző oldalon felsorolt négy 'target' típus közül mindegyiket képes tárolni, viszont csak lokálisan. Azaz az adatbázisba feltöltött elemeket úgy érhetjük el, ha az egész adattárolót letöltjük, és beimportáljuk a Unity projektünkbe.

Ezen projekt keretében mivel kizárólag egy darab Image Target-et használok, így tökéletesen elég egy ilyen lokális adatbázis, hiszen a mérete elenyésző.

Cloud Database

Nevéből adódóan ez egy felhős adattároló, ahol internetkapcsolat elengedhetetlen.

Javára szóljon cserébe több millió 'target' tárolására képes, ellentétben a Device DB-vel, melyet inkább csak maximum 1000 'target' tárolására használunk. Cloud 'target'-ek lekérdezése stabil netkapcsolat mellett http kéréssel kerülnek lekérdezésre automatikusan.

Hátránya viszont, hogy csak a négy képkövetés közül csak az Image Target-eket támogatja.

VuMark DB

Szintén egy lokális adatbázis, mely csak VuMark 'target'-eket képes tárolni.

Összegzés

Ha egyszerre nagyon sok 'target'-tel kell dolgoznunk, akkor célszerűbb Cloud DB-t használni, mivel ezen adatbázis rugalmasabb is a 'target'-ek kezelésében (új hozzáadása, régi törlése). Válasszunk Device DB-t, ha nem dolgozunk sok 'target'-tel, és tudjuk előre, hogy ezeket nem is fogjuk a jövőben (gyakran) megváltoztatni.

² Egy ilyen adatbázis létrehozásához jelentkezünk be a Vuforia weboldalán. Develop menü alatt a Target Manager fület kiválasztva hozhatunk létre egy új adatbázist. Add Database gombra kattintva és a Device típus bejelölésével létre is hoztunk egy Device DB-t.

6. Vuforia target

Minden 'target' 3 komponensből áll, és ezeken kívül még 2 fontos komponenst adhatunk hozzá.

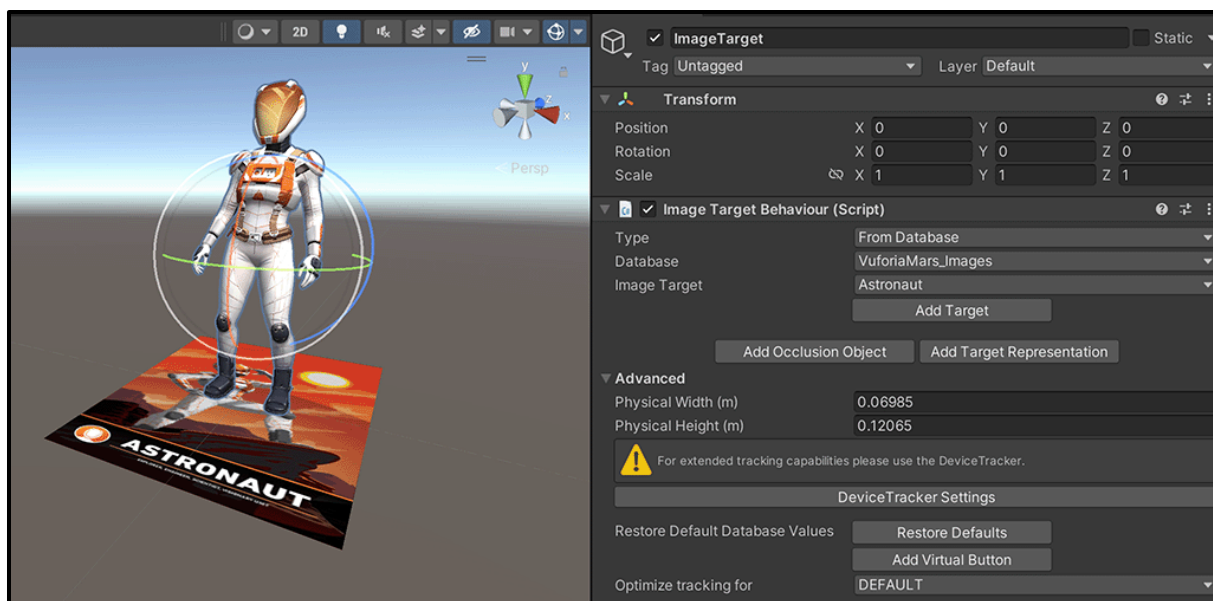
<u>Komponens neve</u>	<u>Komponens leírása</u>
Vuforia <target type> Behaviour	<i>Adatbázisból történő kiválasztásért és a 'target' alaptulajdonságainak beállításáért felel.</i>
<target> ObserverEventHandler	<i>Események és megjelenítési feltételek nyomkövetésért felel.</i>
Target preview	<i>'Target' előnézete, melyet unity scene-ben is láthatunk. Ki- és bekapcsolható.</i>
Occlusion Object	<i>Egy GameObject a modellről, mely vizuálisan is elzárja a digitális tartalmat a többi objektumtól.</i>
Target Representation	<i>Szintén egy GameObject a modellről, mely az eredeti kép árnyékainak, részeinek kiemelésére és ütköztetők alkalmazására szolgál.</i>

Esetünkben az utolsó két komponens a legfontosabb, mivel a projekt Astronaut 'target'-ére egy futási időben keletkező objektumot teszünk, mely azzal a nehézséggel jár, hogy az alkalmazás elindítása előtt teljesen pontosan nem meghatározható, hogyan fog megjelenni kivetítve, hiszen nem látjuk előre magunk előtt a kész labirintust.

Tehát, mivel az utolsó két komponens önmagában is egy GameObject, így méretezhető, viszont ezzel teljesen ellentétes és rossz működést fog maga után vonni a Vuforia képmegjelenítés.

Ezért a megjelenítendő tartalmunk jó képkövetéséhez – tehát a digitális tartalmunk nem túl nagy, és nem is túl kicsi megjelenítéséhez – az AR Camera-n keresztül a Vuforia konfigurációjában lehetőségünk van egy Virtual Scene Scale skála beállítására. Ezzel érjük el, ha kisebb vagy nagyobb digitális tartalmat szeretnénk, ne a komponensek méretezésével.

Fontos, hogyha ezt a konstanst értéket beállítottuk, akkor a 'target'-ünkhöz adjuk hozzá újra az Occlusion Object-et és Target Representation-t, mivel ezen GameObject modelleket nem méretezi át.



9. ábra: Image Target

Digitális tartalom

Amennyiben elkészültünk 'target'-ünkkel, akkor bármilyen digitális tartalmat rápakolhatunk, és így megjeleníthetünk kivetítve a valóságba. Ezen tartalom lehet Unity-ben bármilyen 3D Object vagy Video, sőt előre elkészített, tehát Asset Store-ból is elérhető tartalmakat is megjeleníthetünk.

Ilyen tartalmak 'target'-re való ráaggasztása a következő lépéseket vonja magába:

- Helyezzük el az elkészített GameObject-et a 'target' alatt, vagyis tegyük a gyerekévé.
- Igazítsuk a 'target'-en a digitális tartalmat úgy, ahogy azt a valóságban is szeretnénk, hogy megjelenjen.

Ezzel el is készültünk. E projektben szintén hasonló történt: Az Astronaut Image Target-nek megadtuk a labirintust gyerekként.

7. Összefoglalás

A Vuforiát ennyiben használtam ki projekt szinten, vagyis: Létrehoztam egy Device adatbázist, melyben egy Image Target-et eltároltam. Ezen adatbázist elérhetővé tettem a Unity projektemnek. Végül a Unity-be bepakoltam ezen Image Target GameObject-et. Átméreteztem, és gyerekének megadtam a labirintust – mely így több GameObject-ból álló digitális tartalom.

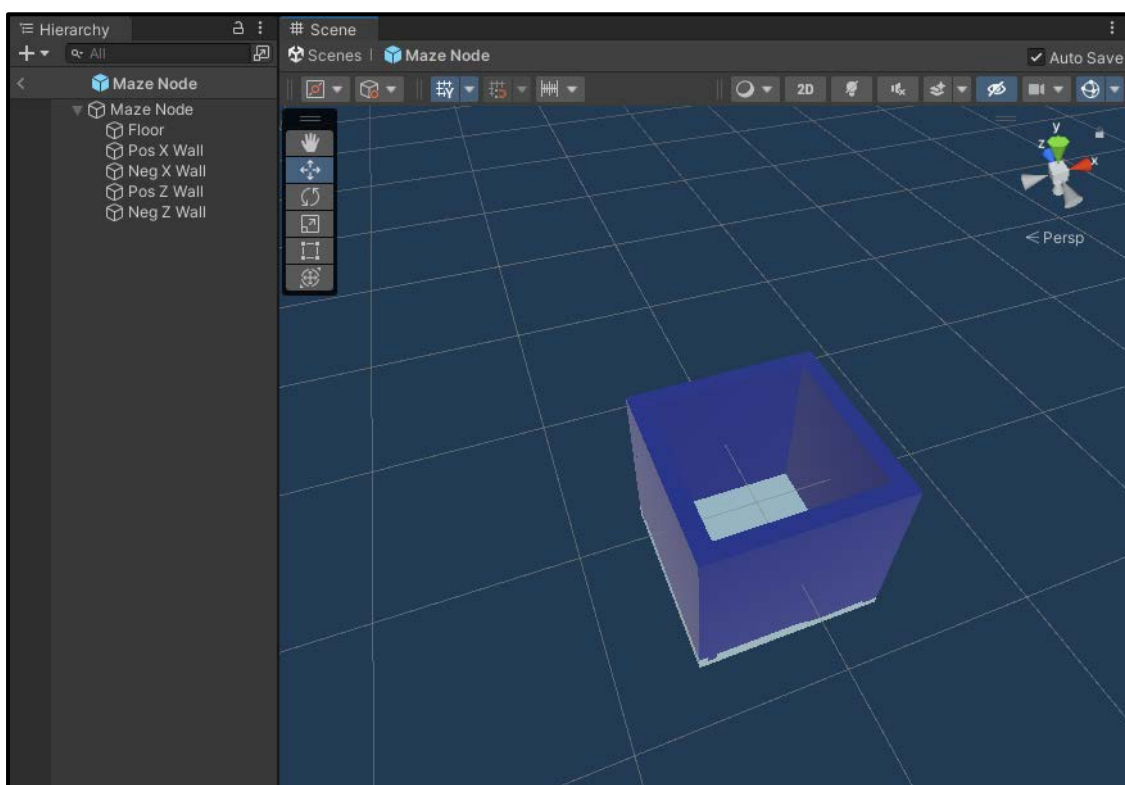
Alkalmazás elkészítése Unity-ben

1. Labirintus

A labirintus az következőkből épül fel: mezők, pénzérmék és a labda.

Mező

A labirintus mezőkből épül fel. Egy mezőnek négy fala és padlója van. Amikor a labirintus generálódik, akkor ezen mezők annyiban változnak, hogy a labirintus-generáló algoritmus kitöröl adott falakat az út meghatározásának érdekében.



10. ábra: Mező

Pénzérmék

A pénzérmék forrása az Asset Store, mely elemeknek forgási animációjuk van. Fontos, hogy annak érdekében, hogy a labda `onTriggerEnter` függvénye érzékelje, hogy most pénzérmébe ütközött – és majd azt felszedje –, ahhoz be kell állítani ezen GameObject Tag-jét valamire, amivel tudunk minden pénzérmére hivatkozni. Jelen esetben ennek a neve 'Coin'.

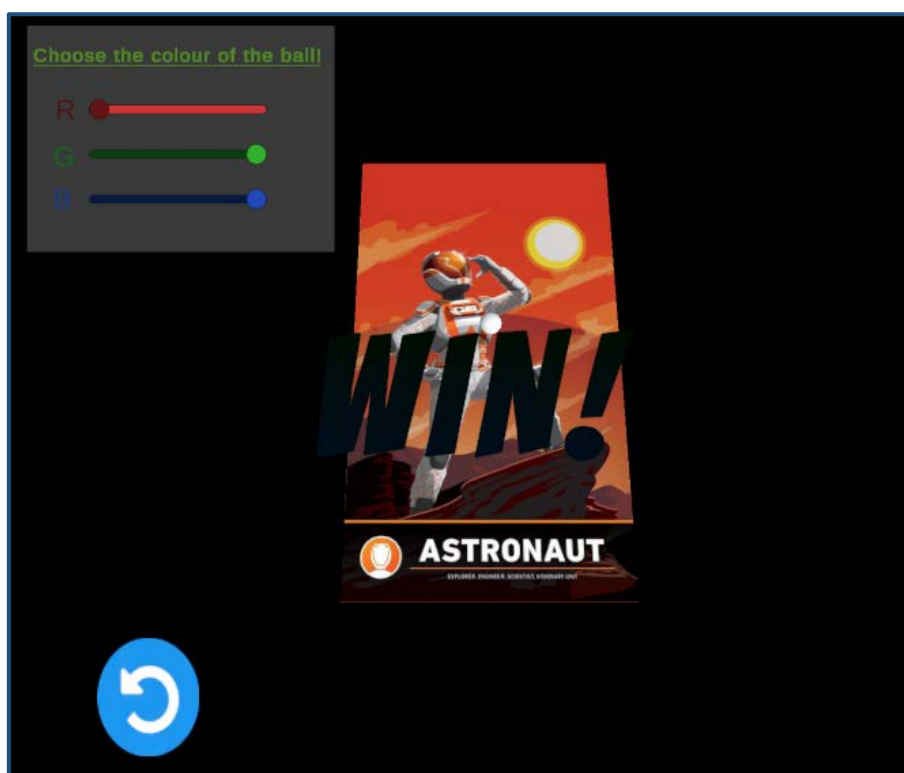
Labda

A labda mindig az első labirintus-generáló algoritmus által érintett mező helyére kerül. A labda mozgatásához egy RigidBody komponenst kell ráaggasztani, hogy érvényesüljön rá a gravitáció.

2. Felhasználói felület

A felhasználói felület a korábban is említett 3 darab csúszkából, egy „WIN” feliratból és a labirintust újra generálásáért felelős gombból áll.

Fontos, hogy ezek mindegyike egy UI GameObject, melyek létrehozásakor – ha eddig még nem létezett volna – létrejön egy Canvas (doboz). Ezen Canvas foglalja magába azon dolgokat, amelyeket a felhasználó képernyőjére kifeszítetten megjelenítünk. (Minden, ami AR kiterjesztett valóságban megjelenítendő azt az Image Target-re pakoltuk.)



11. ábra: Canvas

„WIN” felirat az összes pénzérme összegyűjtésekor jelenik meg.

Összefoglalás

A projekt során a Vuforia környezettel ismerkedhettem meg. Rengeteg lehetőséget még rejt magában, amit még nem érintettem, viszont remek lehetőséget nyújt kezdők számára az AR világ megismeréséhez. Nagyon sok Vuforia projekt készül Unity-ben, melyekről remek bemutató és tutorial videókat találunk az interneten. Lásd: Floppy birds, Snapchat filters, Room detection, Throw-games (Basketball) stb.

Felhasznált források

- AR apps - <https://studiousguy.com/examples-augmented-reality/>
- SDK - <https://pcforum.hu/szotar/Szoftverfejleszt%C5%91+K%C3%A9szlet>
- Vuforia tutorial - <https://library.vuforia.com/getting-started/getting-started-vuforia-engine-unity>
- Vuforia Engine - <https://library.vuforia.com/getting-started/vuforia-features>
- Unity – Vuforia importálás - https://docs.unity3d.com/2017.4/Documentation/Manual/vuforia_get_started_project_setup.html
- AR Maze - <https://youtu.be/7z8VgtDHIhg?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- Floppy Birds - <https://youtu.be/ZxF8ac3Moy0?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- AR in Unity - <https://youtu.be/gpaq5bAjsa8?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- Colouring - <https://youtu.be/9OknCymbTG0?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- AR Ball - <https://youtu.be/qkCnIYINTDo?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- Vuforia tutorial - <https://youtu.be/uXNjNcqW4kY?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>
- Maze generator - <https://youtu.be/OutlTTOm17M?list=PLqfyHlaeq6hO7fFPKfUhrky2nfSOQrYMD>

Tartalomjegyzék

Projekt neve: VufMaze	1
Konzulens neve: Varnyú Dóra	1
<i>Projektről röviden</i>	<i>2</i>
Vuforia.....	2
1. Mi is a Vuforia?.....	2
<i>Kiterjesztett valóság (AR: Augmented Reality)</i>	<i>2</i>
<i>Szoftverfejlesztői készlet (SDK: Software Development Kit)</i>	<i>3</i>
2. Vuforia beimportálása Unity-be	3
3. Képkövetés	4
4. AR Camera	5
5. Vuforia adatbázisok.....	5
<i>Device Database</i>	<i>5</i>
<i>Cloud Database</i>	<i>5</i>
<i>VuMark DB</i>	<i>5</i>
<i>Összegzés.....</i>	<i>5</i>
6. Vuforia target	6
<i>Digitális tartalom</i>	<i>7</i>
7. Összefoglalás.....	7
Alkalmazás elkészítése Unity-ben	8
1. Labirintus	8

<i>Mező</i>	8
<i>Pénzérték</i>	8
<i>Labda</i>	8
2. Felhasználói felület	9
Összefoglalás	9
Felhasznált források	10
Tartalomjegyzék	10