

Проект: №277
Направление: Приложни програми
ТЕМА: Обучение с обогатена реалност (AR LEARN)

<https://github.com/OgiJr/AR-LEARN-V3>

АВТОРИ:

Създатели:

Огнян Траянов, Американския Колеж, o.trajanov22@acsbg.org

Борис Радулов, Американския Колеж, b.radulov20@acsbg.org

РЪКОВОДИТЕЛ:

Д-р Паулина Иванова Тодорова p.todorova@acsbg.org,

Телефон: 089 619 4957

Информатика и Информационни Технологии, АКС

РЕЗЮМЕ:

Идея на проекта:

Технологиите в 21. век са важна част от живота на всеки един човек. Информационните технологии се развиват с бързи темпове във всяка една сфера на съвременното общество, като транспорт, производство, и телекомуникация.

За съжаление в образователната система информационните технологии са назад и не са интегрирани в класните стаи, а *това трябва да се промени*.

AR ще помогне с интеграцията на информационните технологии в класната стая, като допринесе възможността да се използва *обогатена реалност (Augmented Reality)* в часовете.

Обогатената реалност представлява комбинацията от данни от *истинския* свят с данни от *виртуалния*. Например, нашето приложение използва изображение от камерата в истинския свят и добавя виртуални 3D изображения.

Обогатената реалност дава възможност на учениците да виждат материала, по-добре илюстриран, както и допълнителна информация към него. Това помага на учениците в обучението им от класната стая и дома.

Един от аргументите срещу интеграцията на информационните технологии е необходимостта от много ресурси и средства. Но тук идват на помощ смартфоните на учениците - средство, което досега е използвано, като източник за разсейване и дори за измама.

Обаче смартфоните могат да се превърнат в източник за знание и учение. Почти всеки ученик има достъп до телефон и интернет, което му позволява да използва тази технология, която от своя страна не носи разход за държавата и МОН да екипира училищата с лаптопи, таблети или хромбуци.

Сканиране на изображението:

Обогатената реалност работи по следния начин: *първо*, сканираме с камерата на смартфона едно изображение и *второ*, излиза 3D модел върху изображението. Този 3D модел е паралелен към изображението, като и двете са активни едновременно. В приложение ни, след като се сканира дадено изображение, се подава заявка към сървъра. Взема се името на изображението и ако то бъде разпознато, се дава на приложението, за да зареди специфичния обект. Това позволява на потребителя да сканира изображението и да получава резултат към *обогатената реалност*.

Framework Дизайн:

Целта на проекта е потребителите сами да вкарват свои модели и изображения към базата данни. Това позволява представеното от нас приложение да не бъде подходящо само за един учебник, а да бъде framework за всеки учебник, което от своя страна, може да бъде използвано както от учители, така и от издателства.

3D Модели:

3D моделите се създават в някой 3D CAD софтуер или се вземат от някое друго лице. Те после се добавят към сървъра и базата данни през уебсайта. Те могат да бъдат *програмирани, анимирани*, имат *аудио и още много други характеристики*.

Лимитът на програмиране на потребителя е неограничен, защото формата Unity Asset Bundle поддържа модели, анимации и програми, както и всички видове компоненти в Unity, като го прави неограничен до една степен.

Изображения:

Изображенията, към които се прикачват 3D обектите, се избират от потребителя спрямо ситуацията. Ако обектът е предназначен за учебник, може да се сканира учебника, ако не е нужно да е учебник или книга, може просто да се качи всяко изображение.

Изображението ще бъде толкова по-лесно *разпознаваемо*, колкото повече ръбове има, за да могат да се слагат *markers*. Най-добри за сканиране са колажите, а облите обекти не помагат.

Качване на пакет към уебсайта:

Качването на материал се осъществява през нашия уебсайт: <https://arlearn.xyz/>.

Първо, избира се име на пакет, като пакет – това е група от различни обекти и техните изображения. *Второ*, избира се описание на пакета, за да се знае за какво става въпрос. Пример за пакет може да бъде „Учебник по литература 7 клас“ или „Учебници по физика Просвета“. След това към всеки обект се качат 3D модели, изображения и допълнителна информация към тях. За 3D моделите се избира Unity Asset Bundle. После се избира изображение. Накрая се подбира md файл, който служи за описание на обекта.

Употреба:

Приложението може да влезе в употреба за:

- учебници – като след сканиране на материала, излиза илюстриращ го 3D модел;
- книги при илюстриране на съдържанието;
- поддържане на видео игри ;
- поддържане на картини ;
- в час по даден предмет;
- други видове изкуство.

Инструкции:

1. Учителите/издателството качват снимки и модели в базата данни през уебсайта ни.
2. Учениците инсталират приложението през Google Play Store.
3. Учениците теглят пакета през приложението ни.
4. Учениците сканират изображението към модела.

Стъпки на създаване на проекта:

- Създаване на прототип в Unity (Огнян)
- Създаване на уебсайт (Борис)
- Създаване на примери (Огнян)
- Интегриране на сървъра с Unity (Борис)

Ниво на сложност:

Основни проблеми при реализацията на софтуера са основно архитектурни. С други думи това са трудности, възникнали при структуриране на необходима софтуерна архитектура за реализацията на проекта.

Трябва да бъде създадена *облачна система*, която да поддържа необходимите модели за *триизмерните визуализации* и да бъде достъпна до всякакви видове устройства. Тази система трябва да е достатъчно бърза и стабилна, за да може да бъде използвана от голям брой ученици и учители едновременно.

Друг голям приоритет пред приложението е поддържането на различни устройства. Тъй като приложението трябва да поддържа максимално количество устройства, необходимо е софтуерът да бъде тестван на колкото се може повече смартфони.

Сложността на проекта е на *високо ниво*, тъй като изисква комбиниране на много различни технологии и модули, чиято координация е ключова за успеха на приложението.

Логическо и функционално решение на проблема:

Приложението се нуждае от база данни, която да го снабдява с нужните 3D модели. Това ще бъде постигнато чрез базите данни на *Vuforia* за изображенията и база данни с 3D модели създадена от нас.

Мобилното приложение за *Android*, написано в Unity, ще се свързва с тази база данни чрез идентифициращи кодове. Оттам ще се свързва с моделите. От своя страна тези модели ще бъдат зареждани в *Unity модула* на приложението, който ще ги визуализира и ще контролира интерфейса на техните анимации.

Реализация:

За да реализираме този проект използваме *Unity, Visual Studio, Vuforia, PHP, Apache и MySQL*. Имплементацията на *Augmented Reality* е в *Unity* и визуализира 3D моделите чрез *Vuforia*.

Vuforia е библиотеката, с което се осъществява работата с AR системата. Тази програма разпознава изображенията, като тя съхранява всичките тях в своя база данни и поставя обектите във добавената реалност.

Лиценз:

This project is licensed under the terms of the GNU GPLv3 License. For more information, see LICENSE.

Библиография:

1. Hoagland, John. "Circulatory System", Sketchfab, Sketchfab, 1 Jan. 1968, sketchfab.com/3d-models/circulatory-system-4d9b279600264497958ac834eb1529a8.
2. Orbis Agenda. "Solar System." Unity Asset Store - The Best Assets for Game Making, assetstore.unity.com/packages/3d/environments/sci-fi/solar-system-24810.
3. MargetaCG. "Saturn V | 3D Model." CGTrader, CGTrader, www.cgtrader.com/free-3d-models/aircraft/historic/saturn-v-1479cb7d-b978-42b4-90d9-f3e844e6ec55
4. System-IntegraTech, "Biology Cell Package" Unity Asset Store, <https://assetstore.unity.com/packages/3d/biology-cells-pack-97118>

Chapter 3

Class Documentation

3.1 AnimatorManager Class Reference

Manages the animation for AR Models through Unity Swipe Controls.

Inheritance diagram for AnimatorManager:

3.2 ARVideoPlayer Class Reference

Inheritance diagram for ARVideoPlayer:

Collaboration diagram for ARVideoPlayer:

The documentation for this class was generated from the following file:

- ARVideoPlayer.cs

3.3 BackButtonManager Class Reference

Manages the back button on Android which basically does the same thing as the red cross button.

Inheritance diagram for BackButtonManager:

Collaboration diagram for BackButtonManager:

Public Member Functions

- void [Back](#) ()
Loads the original scene when pressed.

3.3.1 Detailed Description

Manages the back button on Android which basically does the same thing as the red cross button.

3.3.2 Member Function Documentation

3.3.2.1 Back()

```
void BackButtonManager.Back ( ) [inline]
```

Loads the original scene when pressed.

The documentation for this class was generated from the following file:

- BackButtonManager.cs

3.4 BackgroundManagerSearchEngine Class Reference

Inheritance diagram for BackgroundManagerSearchEngine:

Collaboration diagram for BackgroundManagerSearchEngine:

The documentation for this class was generated from the following file:

- BackgroundManagerSearchEngine.cs

3.5 BaseModelsFixed Class Reference

Inheritance diagram for BaseModelsFixed:

Collaboration diagram for BaseModelsFixed:

The documentation for this class was generated from the following file:

- BaseModelsFixed.cs

3.6 CheckNetwork Class Reference

Checks the network so the app can switch from and to online mode.

Inheritance diagram for CheckNetwork:

Collaboration diagram for CheckNetwork:

Public Member Functions

- void [OnlineMode](#) ()
If this button is pressed Unity loads the online mode scene.
- void [OfflineMode](#) ()
If this button is pressed Unity loads the offline mode scene.

3.6.1 Detailed Description

Checks the network so the app can switch from and to online mode.

3.6.2 Member Function Documentation

3.6.2.1 OfflineMode()

```
void CheckNetwork.OfflineMode ( ) [inline]
```

If this button is pressed Unity loads the offline mode scene.

3.6.2.2 OnlineMode()

```
void CheckNetwork.OnlineMode ( ) [inline]
```

If this button is pressed Unity loads the online mode scene.

The documentation for this class was generated from the following file:

- CheckNetwork.cs

3.7 CloudHandler Class Reference

Main vuforia handler. It takes information from the vuforia API and it augments it into the camera.

Inheritance diagram for CloudHandler:

Collaboration diagram for CloudHandler:

Public Member Functions

- void [OnNewSearchResult](#) (TargetFinder.TargetSearchResult targetSearchResult)
The main manager for augmenting objects.

3.7.1 Detailed Description

Main vuforia handler. It takes information from the vuforia API and it augments it into the camera.

3.7.2 Member Function Documentation

3.7.2.1 OnNewSearchResult()

```
void CloudHandler.OnNewSearchResult (
    TargetFinder.TargetSearchResult targetSearchResult ) [inline]
```

The main manager for augmeting objects.

Instatitates the object from the resources folder after detecting the GameObject. Then it changes the name so that we can find the game object in the scene.

Loads and adds all of the components to the instantiated object so that you can scale it, rotate it and change its animations.

The documentation for this class was generated from the following file:

- CloudHandler.cs

3.8 DictionaryManager Class Reference

Inheritance diagram for DictionaryManager:

Collaboration diagram for DictionaryManager:

The documentation for this class was generated from the following file:

- DictionaryManager.cs

3.9 DismissWarning Class Reference

Inheritance diagram for DismissWarning:

Collaboration diagram for DismissWarning:

The documentation for this class was generated from the following file:

- DismissWarning.cs

3.10 LanguageManager Class Reference

Inheritance diagram for LanguageManager:

Collaboration diagram for LanguageManager:

The documentation for this class was generated from the following file:

- LanguageManager.cs

3.11 MaintainRenderer Class Reference

Inheritance diagram for MaintainRenderer:

Collaboration diagram for MaintainRenderer:

The documentation for this class was generated from the following file:

- MaintainRenderer.cs

3.12 ServerDownloader.Package Class Reference

AR [Package](#) info class

3.12.1 Detailed Description

AR [Package](#) info class

The documentation for this class was generated from the following file:

- ServerDownloader.cs

3.13 SaveInfoSceneOne Class Reference

Save the info fom th TmPro input field.

Inheritance diagram for SaveInfoSceneOne:

Collaboration diagram for SaveInfoSceneOne:

3.13.1 Detailed Description

Save the info fom th TmPro input field.

The documentation for this class was generated from the following file:

- SaveInfoSceneOne.cs

3.14 SelectOfflineManager Class Reference

Inheritance diagram for SelectOfflineManager:

Collaboration diagram for SelectOfflineManager:

The documentation for this class was generated from the following file:

- SelectOfflineManager.cs

3.15 ServerDownloader Class Reference

Download AR Packages from the server at arlearn.xyz

Inheritance diagram for ServerDownloader:

Collaboration diagram for ServerDownloader:

Classes

- class [Package](#)
AR [Package](#) info class

Public Member Functions

- void [getInfo](#) (string id)
This function returns a serializable class with info for an AR package
- void [downloadModels](#) ()
Download the models and markdown files for a package

3.15.1 Detailed Description

Download AR Packages from the server at arlearn.xyz

3.15.2 Member Function Documentation

3.15.2.1 downloadModels()

```
void ServerDownloader.downloadModels ( ) [inline]
```

Download the models and markdown files for a package

Parameters

<i>p</i>	Package whose models to download
----------	--

3.15.2.2 getInfo()

```
void ServerDownloader.getInfo (
    string id ) [inline]
```

This function returns a serializable class with info for an AR package

Parameters

<i>id</i>	The unique id of the AR Package
-----------	---

Returns

The package class

The documentation for this class was generated from the following file:

- ServerDownloader.cs

3.16 SwipeControls Class Reference

Manages swipe controls for transformations of the AR Object.

Inheritance diagram for SwipeControls:

Collaboration diagram for SwipeControls:

3.16.1 Detailed Description

Manages swipe controls for transformations of the AR Object.

The documentation for this class was generated from the following file:

- SwipeControls.cs

3.17 UDTLoader Class Reference

Loads downloader AR models at arlearn.xyz

Inheritance diagram for UDTLoader:

Collaboration diagram for UDTLoader:

3.17.1 Detailed Description

Loads downloader AR models at arlearn.xyz

The documentation for this class was generated from the following file:

- UDTLoader.cs

3.18 UIEffects Class Reference

Handles effects and animations for the UI such as the buttons

Inheritance diagram for UIEffects:

Collaboration diagram for UIEffects:

Public Member Functions

- void [Exit](#) ()
Exits AR Mode.
- void [RevertExit](#) ()
Goes back into AR Mode
- void [Scan](#) ()
Activate Scan UI
- void [RevertScan](#) ()
Deactivate Scan UI

3.18.1 Detailed Description

Handles effects and animations for the UI such as the buttons

3.18.2 Member Function Documentation

3.18.2.1 Exit()

```
void UIEffects.Exit ( ) [inline]
```

Exits AR Mode.

3.18.2.2 RevertExit()

```
void UIEffects.RevertExit ( ) [inline]
```

Goes back into AR Mode

3.18.2.3 RevertScan()

```
void UIEffects.RevertScan ( ) [inline]
```

Deactivate Scan UI

3.18.2.4 Scan()

```
void UIEffects.Scan ( ) [inline]
```

Activate Scan UI

The documentation for this class was generated from the following file:

- UIEffects.cs

3.19 UIMenuManager Class Reference

Basically a ghetto PageViewer. It handles the menus.

Inheritance diagram for UIMenuManager:

Collaboration diagram for UIMenuManager:

Public Member Functions

- void [MinusOne](#) ()
Move to the left.
- void [PlusOne](#) ()
Move to the right.

3.19.1 Detailed Description

Basically a ghetto PageViewer. It handles the menus.

3.19.2 Member Function Documentation

3.19.2.1 MinusOne()

```
void UIMenuManager.MinusOne ( ) [inline]
```

Move to the left.

3.19.2.2 PlusOne()

```
void UIMenuManager.PlusOne ( ) [inline]
```

Move to the right.

The documentation for this class was generated from the following file:

- UIMenuManager.cs

3.20 UserDefinedMode Class Reference

Handles the user defined mode.

Inheritance diagram for UserDefinedMode:

Collaboration diagram for UserDefinedMode:

3.20.1 Detailed Description

Handles the user defined mode.

The documentation for this class was generated from the following file:

- UserDefinedMode.cs