ARLearn

По-добро образование чрез иновация

Борис Радулов
Огнян Траянов

Американски Колеж в София
30.03.2019г.

Въведение

Като всички други индустрии, образователната система се възползва от новите технологии все повече и повече. Много училища започват да използват системи за онлайн обучение като Quizlet, Google Classroom, Kahoot и други. Тази интеграция обаче оставя много място за подобрение. Съществуващите технологични решения в употреба в образователната система са подобрени версии на традиционни методи за учене, които не правят големи промени в името на иновацията. Например, Quizlet е сайт за електрони “флаш” карти, Kahoot за тестове, и Google Classroom за организация на информация. Въпреки че тези сайтове подобряват и надграждат над оригиналната концепция, тяхните педагогически основи не са променени. ARLearn е различен, защото представя един изцяло нов метод на учене. Това е постигнато чрез създаването на общодостъпни AR (Augmented Reality - Добавена Реалност) технологии за учебна употреба.

ARLearn е мобилно приложение чрез което ученици визуализират и управляват учебния си матерял по нови и иновативни начини чрез AR. Приложението сканира целеви снимки в учебници и показва интерактивни 3D модели и диаграми, които помагат на учениците да асимилират информацията по лесно и ефикасно. За разлика от традиционите AR приложения, които просто показват статични 3D модели върху екрана, ARLearn използва Unity Prefabs. Те позволяват на издателите на учебници да пишат код за моделите и да създават интерактивни елементи, които да са полезни за учене. Тъй като Unity Prefabs са част от платформата на Unity, те могат да съдържат код на C# и JavaScript. Това позволява на издателите на учебници за създават нови и по-ефикасни начини чрез които да помагат на учебния процес.

Функционално описание

Основен интерфейс

Мобилния интерфейс на ARLearn се състой от камера и от секция за допълнителна информация. Интерфейсът на камерата е основната част на приложението и първото нещо, което потребителя вижда. Показва какво задната камера на мобилното устройство вижда и има бутон в дъното, който активира засичането на целеви снимки в базата си данни. Когато целево изображение бъде намерено, приложението влиза в AR режим и бутона за сканиране е заменен с червен бутон за изключване на AR режима. Този бутон спира AR режима и подготвя устройството за ново сканиране на нов модел. В AR режим, приложението засича пропорциите и позицията на изображението и позиционира Unity Prefab-а асоцииран с изображението на подходящото място. Целият Prefab бъва създаден, включвайки елементи на интерфейс, модели, анимации, и звуци. Сега потребителят може да контролира Prefab-а. Ако потребителят плъзне екрана надясно, секцията за допълнителна информация се появява и покрива камерата. Информацията в тази секция се взима заедно с Prefab-а от издателя на учебника. Ако потребителят плъзне наляво, ще се върне обратно в AR режим. Потребителят може да натисне червеният бутон за да изключи сегашната визуализация и да махне всички ефекти. Преди да може да види дадена визуализация, нейният AR пакет трябва да бъде свален.

Достъп до AR пакети

AR пакетите са колекции от Unity Prefab-ове и Markdown файлове, които комбинирани създават AR преживявания. Те са контейнери, които подреждат различни AR преживявания в специфични логически групировки за по лесно дистрибуция.

Например, еди учебник по физика може да има един монолитен пакет с всичките си AR преживявания или да има отделен пакет за всяка секция. Всеки пакет има собствен уникален идентификационен код, който бива даден на учениците. Когато плъзнат екрана надолу, учениците виждат текстово поле, в което да въведат идентификационния код и да свалят приложението. По този начин, учениците могат да свалят само AR пакетите, които им трябват, и да пестят място на устройствата си.

Имплементация на AR пакетите

AR пакетите се качват на сайта на ARLearn -- https://arlearn.xyz. Всеки AR пакет може да съдържа практически неограничено голямо количество AR преживявания. Всяко AR преживяване съдържа един Unity Prefab и един Markdown файл. Целево изображение за всеки Prefab трябва да бъде качено. Ако издателите на учебник не искат да качат код в някои от преживяванията си, могат да качат само FBX файлове, които съдържат информация за 3D моделите. Когато създадат пакет, издателите получават тяхното уникално ID, което трябва да дадат на учениците. До момента в този документ, хората, които качват пакети са наречени “издатели на учебници”, но сайта е общодостъпен за всички учители, издатели и дори ученици. За момента, достъп до сайта е безплатен, но има планове за монетизация в бъдещето.

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 AnimatorManager Class Reference

Manages the animation for AR Models through Unity Swipe Controls.

Inherits MonoBehaviour.

2.1.1 Detailed Description

Manages the animation for AR Models through Unity Swipe Controls.

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

- AnimatorManager.cs

2.2 BackButtonManager Class Reference

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

Inherits MonoBehaviour.

Public Member Functions

- void Back ()

  Loads the original scene when pressed.

2.2.1 Detailed Description

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

2.2.2 Member Function Documentation

2.2.2.1 Back()

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

Loads the original scene when pressed.

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

- BackButtonManager.cs

2.3 CheckNetwork Class Reference

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

Inherits MonoBehaviour.

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.

2.3.1 Detailed Description

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

2.3.2 Member Function Documentation

2.3.2.1 OfflineMode()

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

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

2.3.2.2 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:

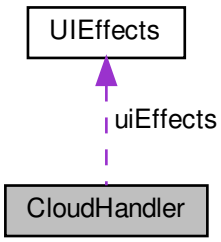
- CheckNetwork.cs

2.4 CloudHandler Class Reference

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

Inherits MonoBehaviour, and ICloudRecoEventHandler.

Collaboration diagram for CloudHandler:



Public Member Functions

- void OnNewSearchResult (TargetFinder.TargetSearchResult targetSearchResult)

  The main manager for augmeting objects.

2.4.1 Detailed Description

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

2.4.2 Member Function Documentation

2.4.2.1 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

2.5 Package Class Reference

AR Package info class

2.5.1 Detailed Description

AR Package info class

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

- ServerDownloader.cs

2.6 SaveInfoSceneOne Class Reference

Save the info fom th TMPro input field.

Inherits MonoBehaviour.

2.6.1 Detailed Description

Save the info fom th TMPro input field.

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

- SaveInfoSceneOne.cs

2.7 ServerDownloader Class Reference

Download AR Packages from the server at arlearn.xyz

Static Public Member Functions

- static Package getInfo (string id)

  This function returns a serializable class with info for an AR package
- static void downloadModels (Package p)

  Download the models and markdown files for a package

2.7.1 Detailed Description

Download AR Packages from the server at arlearn.xyz

2.7.2 Member Function Documentation

2.7.2.1 downloadModels()

```
static void ServerDownloader.downloadModels (
            Package p )  [inline], [static]
```

Download the models and markdown files for a package

Parameters

p	Package whose models to download

2.7.2.2 getInfo()

```
static Package ServerDownloader.getInfo (
            string id )  [inline], [static]
```

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

Parameters

id	The unique id of the AR Package

Returns

The package class

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

- ServerDownloader.cs

2.8 SwipeControls Class Reference

Manages swipe controls for transformations of the AR Object.

Inherits MonoBehaviour.

2.8.1 Detailed Description

Manages swipe controls for transformations of the AR Object.

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

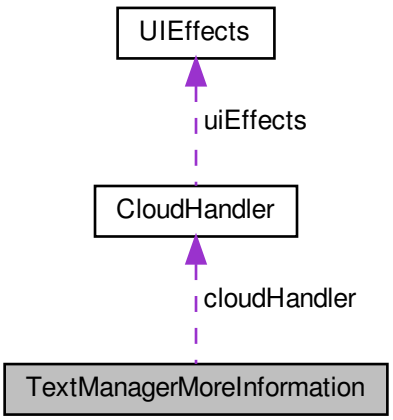
- SwipeControls.cs

2.9 TextManagerMoreInformation Class Reference

Manages the text on the more information section. It gets it from the asset bundles from ServerDownloader

Inherits MonoBehaviour.

Collaboration diagram for TextManagerMoreInformation:



2.9.1 Detailed Description

Manages the text on the more information section. It gets it from the asset bundles from ServerDownloader

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

- TextManagerMoreInformation.cs

2.10 UIEffects Class Reference

Handles effects and animations for the UI such as the buttons

Inherits MonoBehaviour.

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

2.10.1 Detailed Description

Handles effects and animations for the UI such as the buttons

2.10.2 Member Function Documentation

2.10.2.1 Exit()

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

Exits AR Mode.

2.10.2.2 RevertExit()

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

Goes back into AR Mode

2.10.2.3 RevertScan()

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

Deactivate Scan UI

2.10.2.4 Scan()

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

Activate Scan UI

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

- UIEffects.cs

2.11 UIMenuManager Class Reference

Basically a ghetto PageViewer. IIt handles the menus.

Inherits MonoBehaviour.

Public Member Functions

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

2.11.1 Detailed Description

Basically a ghetto PageViewer. IIt handles the menus.

2.11.2 Member Function Documentation

2.11.2.1 MinusOne()

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

Move to the left.

2.11.2.2 PlusOne()

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

Move to the right.

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

- UIMenuManager.cs

2.12 UserDefinedMode Class Reference

Handles the user defined mode.

Inherits MonoBehaviour.

2.12.1 Detailed Description

Handles the user defined mode.

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

- UserDefinedMode.cs