PROGETTO V4T

CORSO DI E-LEARNING

MODULO 3: PROGRAMMARE APPLICAZIONI PER DISPOSITIVI MOBILI

Questo Modulo inizia con una panoramica sulle tecnologie e sugli strumenti disponibili per la programmazione di applicazioni mobili e fornisce esempi indicativi che sono, o potrebbero essere, utilizzati con gli studenti delle scuole elementari. Segue una presentazione dettagliata dell'ambiente di programmazione di App Inventor. Viene descritto il processo di sviluppo di due giochi popolari in App Inventor. Infine, questo Modulo presenta la piattaforma web di Realtà Aumentata Metaverse che permette agli insegnanti e agli studenti di creare facilmente le proprie applicazioni interattive di Realtà Aumentata.

Capitolo 1 - Introduzione alla programmazione di APP per dispositivi mobili

1.1 - Tecnologie disponibili

Prima di poter iniziare a codificare la vostra prima app, dovete decidere su quale sistema software volete far funzionare la vostra App e quale linguaggio di programmazione volete utilizzare. I sistemi software più diffusi sono Android fornito da Google o iOS di Apple.

La programmazione per Android richiede la conoscenza del linguaggio di programmazione Java e dell'SDK Android.

iOS funziona solo con dispositivi Apple e così anche la piattaforma di programmazione XCode, quindi avrete bisogno di un computer Apple con Mac OSX. Per codificare un'applicazione per dispositivi iOS è necessario conoscere i linguaggi Objective-C o Swift.

Se si vuole iniziare a programmare si hanno diverse possibilità per iniziare. Nell'unità seguente ci concentriamo su alcune tecnologie disponibili.

Applicazioni native

Le applicazioni native sono scritte nei linguaggi per i sistemi software già citati (Objective-C, Cocoa).

Web Apps (HTML5-Apps)

Co-funded by the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Una tecnologia da cui partire se non si è sicuri di cosa codificare o da dove iniziare. Si inizia ad imparare Javascript e si inizia a programmare le web App, che possono essere facilmente convertite nelle versioni mobili utilizzate sui dispositivi mobili utilizzando emulatori. Un emulatore è un programma che clona un sistema operativo da un'architettura ad un'altra struttura, o sistema. Con un emulatore è possibile convertire una Web App in una utilizzabile per altri sistemi (ad es. Android).

Risorse online

Scopri alcune nozioni di base sulle App native -

https://searchsoftwarequality.techtarget.com/definition/native-application-native-app

1.2 - Ambiente di sviluppo e linguaggi di programmazione

Se si vuole iniziare a codificare un'applicazione per sistemi Android si può iniziare con Android Studio che fornisce un ambiente di sviluppo integrato IDE.

Android Studio offre:

- Sistema di costruzione progressivo
- Emulatore ricco di funzioni
- Ambiente per sviluppare per diversi dispositivi Android
- Esecuzione istantanea per apportare modifiche alla vostra applicazione in esecuzione senza costruire un nuovo APK
- Modelli di codice e integrazione con GitHub per aiutare a costruire caratteristiche comuni delle applicazioni e importare codice campione
- Strumenti e piattaforme di prova
- Strumenti per misurare le prestazioni, l'usabilità, la compatibilità delle versioni e altri problemi
- Supporto C++ e NDK

La guida per l'utente è scritta in un linguaggio semplice che motiva i principianti a iniziare a programmare le prime applicazioni.

Iniziare a creare la vostra prima App è possibile anche utilizzando la tecnologia a blocchi. Ciò è molto più facile di imparare un linguaggio in codice, ma limita anche le vostre possibilità. GoodBarber è un fornitore di strumenti di sviluppo per App. Si ottiene una prova di 30 giorni al momento della registrazione. Il sistema consente di sviluppare un'applicazione con caratteristiche interessanti per i futuri utenti (autenticazione degli utenti, notifiche push, monetizzazione e altro ancora). Goodbarber offre diverse opzioni di prezzo a partire da € 32,-- al mese.

Nel capitolo 2 ci concentriamo su App Inventor che è un ambiente di sviluppo per sistemi Android fornito da Google.

Co-funded by the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Risorse online

Android Studio - https://developer.android.com/studio

Scaricate Android studio e date un'occhiata alla guida utente per iniziare la programmazione.

GoodBarber - https://www.goodbarber.com/

Toolkit provider per costruire un'applicazione utilizzando i blocchi.

1.3 - Applicazioni e nuove tecnologie utilizzate nelle scuole elementari

Come preparare i nostri figli all'era digitale che già ci circonda? Dobbiamo trasferire contenuti complessi in modalità d'uso semplici. La comprensione del codice e dell'algoritmo sono competenze fondamentali in un mondo digitale.

Ci sono molti modi per insegnare ai bambini le competenze digitali o meglio per permettere loro di fare esperienze con gli ambienti digitali.

Imparare con le Applicazioni, i Giochi e le Piattaforme

Ci sono già molte App, Piattaforme online e Giochi che possono essere utilizzati per fare i primi passi nella codifica (vedi Risorse online). Queste App, piattaforme e giochi potrebbero essere utilizzati a scuola o a casa per le prime esperienze digitali.

App - Swift Playgrounds

In questa App per iPad fornita da Apple i programmatori alle prime armi risolvono enigmi e controllano i personaggi utilizzando il vero codice Swift. Nel libro 'Intro to App Development with Swift' impareranno come costruire una prima app dall'inizio alla fine - apprendendo le basi di Xcode. Inoltre Apple fornisce il programma "Everyone Can Code curriculum" che aiuta i docenti ad insegnare la programmazione dalla scuola elementare all'università.

Linguaggi di programmazione per bambini

Esistono speciali linguaggi di programmazione per bambini che si basano per lo più su codici visivi.

Scratch

Scratch è un linguaggio di programmazione visuale progettato al MIT Media Lab appositamente per i ragazzi di 8-16 anni per programmare storie interattive, giochi e animazioni che possono essere condivisi con altri. Effettivamente, gli studenti della scuola elementare e dell'università stanno imparando con Scratch discipline come la matematica, l'informatica, le materie linguistiche e gli studi sociali. Presso

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

l'Archivio della comunità online di ScratchEd gli educatori possono trovare risorse, forum di discussione e storie di molti altri educatori.

NEPO - Open Roberta Lab

L'iniziativa Open Roberta (del Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS) introduce nella programmazione dei robot. I bambini possono imparare il linguaggio di programmazione visiva NEPO e possono testare le loro capacità con un robot virtuale. La Roberta-Initiative fornisce corsi di formazione, materiali, programmi di studio e manuali di costruzione per gli insegnanti.

Workshop

I laboratori costituiscono una grande possibilità per i bambini di lavorare insieme e di imparare con un gruppo di altri bambini.

Maker Spaces

In Maker Spaces i bambini possono creare qualcosa di nuovo con diversi materiali e strumenti. I Maker Spaces utilizzano spesso strumenti come Arduino, Rasperry Pi, Calliope Mini, BBC micro:bit e Makey Makey o strumenti per la costruzione e la programmazione di robot. Smartphone, Apps, Tablet e stampante 3D possono essere utilizzati e provati. Qui non si acquisiscono solamente competenze di programmazione, il luogo dovrebbe costituire l'occasione dove i bambini possono essere creativi.

Risorse online

Video 'Quello che la maggior parte delle scuole non insegna' - https://youtu.be/nKIu9yen5nc:
Video motivazionale sulla costruzione di competenze informatiche/programmazione

Scoyo - Programma di apprendimento per bambini: in forma per il futuro con divertimento - https://www-de.scoyo.com/eltern/kinder-und-medien/programmieren-lernen-kinder-fit-fuer-die-zukunft
Rivista online per genitori sull'apprendimento, la scuola, la vita familiare e l'alfabetizzazione mediatica

Code.org - https://code.org/: Fornisce tutorial di un'ora progettati per tutte le età in oltre 45 lingue

CodeAcademy - https://www.codecademy.com/: Imparare codici come Python, PHP, jQuery, JavaScript,...

Lightbot - http://lightbot.com/index.html: App per bambini per imparare a programmare

Dash - https://uk.makewonder.com/dash/: Dash è un robot che può essere controllato tramite app adatto a diverse età

CoderDojo - https://zen.coderdojo.com/find: Una rete globale di club di programmazione gratuita, guidata da volontari e basata sulla comunità, per giovani dai 7 ai 17 anni.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Making - design digitale creativo e sperimentazione con i bambini. - https://de.slideshare.net/sandra_slideshare/making-kreatives-digitales-gestalten-und-experimentieren-mit-kindern-einfhrung-und-ausgewhlte-werkzeuge

Introduzione e strumenti selezionati

Arduino - https://www.arduino.cc/

Rasperry Pi - https://www.raspberrypi.org/

Makey Makey Makey - https://makeymakey.com/

Calliope Mini - https://calliope.cc/en

BBC micro:bit - https://microbit.org/

Capitolo 2: Costruire applicazioni con App Inventor

2.1 - Iniziare con App Inventor

App Inventor è un'applicazione web che permette di sviluppare applicazioni per dispositivi Android (smartphone o tablet) utilizzando un linguaggio di programmazione visuale. È un software libero e open source.

Per lavorare con App Inventor occorre avere:

- un account Google,
- un computer con accesso a Internet,
- un browser compatibile (Mozilla Firefox, Apple Safari, Google Chrome - Microsoft Internet Explorer non è supportato).

In caso di problemi di connessione a Internet o di registrazione con un account Google è possibile utilizzare App Inventor 2 Ultimate.

App Inventor salva i progetti nella cloud sotto il nome del tuo account Google. Non è necessario salvare i progetti nell'unità disco locale.

Un'applicazione di App Inventor è composta dal Component Designer e dal Blocks Editor. Presso l'App Inventor **Designer** è possibile creare le schermate dell'applicazione disponendo gli elementi (componenti) dell'applicazione. Presso l'App Inventor **Blocks Editor** è possibile programmare il comportamento dei componenti mediante il collegamento di blocchi di codice. I blocchi rappresentano comandi che specificano un'azione da eseguire.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Ci sono due tipi di componenti: visibili e non visibili. I componenti visibili (come i pulsanti, le etichette, i campi di testo) sono quelli che si possono vedere quando l'applicazione viene lanciata e che fanno parte dell'interfaccia utente dell'applicazione. I componenti non visibili (come Accelerometro, Suono, Sensore di orientamento) non si vedono e forniscono l'accesso alle funzionalità integrate del dispositivo. Non fanno parte dell'interfaccia utente.

È possibile creare più di una schermata per ogni applicazione, ma bisogna essere prudenti perché ogni schermata consuma risorse del computer. Ogni schermata ha il proprio nome e le sue proprietà. Come regola generale, non si dovrebbero avere più di 10 schermate per progetto.

Se non si dispone di un telefono o di un tablet Android da collegare al computer, è comunque possibile sviluppare e testare le applicazioni con App Inventor. App Inventor fornisce un emulatore Android, che installa un dispositivo virtuale sullo schermo del computer e che funziona proprio come un dispositivo Android.

Risorse online

Un tutorial per principianti di App Inventor: http://appinventor.mit.edu/explore/ai2/beginner-videos.html

Per iniziare con MIT App Inventor 2: https://appinventor.mit.edu/explore/get-started.html

Video su App Inventor realizzato da Google: https://www.youtube.com/watch?v=sGiaXOKqeKg

Una rapida panoramica del Designer e del Blocks Editor: https://appinventor.mit.edu/explore/designer-blocks.html

2.2 - L'ambiente di progettazione

Il **Designer** aiuta a personalizzare l'aspetto dell'applicazione. Si compone di cinque colonne - aree:

Tavolozza

Dalla Palette è possibile selezionare i componenti sia su schermo che fuori dallo schermo e trascinarli nell'area Viewer. La colonna Palette è organizzata in gruppi di componenti che sono chiamati **cassetti** (Interfaccia utente, Layout, Media, Disegno e Animazione, Mappe, Sensori, Social, Storage, Connettività, Sperimentale, Estensioni ecc.

I componenti dell'**interfaccia utente** sono: Pulsanti, Caselle di controllo, DatePickers, Immagini, Etichette, ListPickers, ListViews, Notifiers, PasswordTextBoxes, Sliders, Spinners, TimePickers e WebViewers.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Il layout consiste di elementi di formattazione. Utilizzando questi componenti è possibile determinare l'aspetto di ogni schermata (HorizontalArrangement, HorizontalScrollArrangement, TableArrangement, VerticalArrangement, VerticalArrangement, VerticalScrollArrangement).

Il cassetto per i **media** è composto da elementi come:

-Camcorder (un componente visibile per registrare un video utilizzando la videocamera del dispositivo)

-Telecamera (un componente non visibile per scattare una foto con la fotocamera del dispositivo)

-Player (un componente che riproduce l'audio e controlla le vibrazioni del telefono)

Il cassetto **Disegno e Animazione** permette di creare disegni e animazioni. Si compone di elementi come Ball, Canvas e ImageSprite.

-Canvas è un sotto-pannello rettangolare sensibile al tatto all'interno dell'app dove è possibile disegnare e spostare gli sprite.

-E' possibile programmare l'animazione inserendo i componenti Ball e ImageSprite all'interno di una tela. Questi componenti possono reagire ai tap e ai trascinamenti, interagire con altri sprite e con il bordo della tela e muoversi o trasformarsi secondo i loro valori di proprietà.

I componenti dei **sensori** forniscono l'accesso ai sensori del dispositivo (accelerometri, sensori di posizione, sensori di orientamento, giroscopi, scanner di codici a barre, sensori di prossimità, ecc.)

Storage è un componente non visibile per la scrittura e la lettura dei file nella directory privata dei dati associata all'applicazione.

Il cassetto della **connettività** è composto da ActivityStarter, BluetoothClient, BluetoothServer e componenti Web.

Visualizzatore

Nell'area **Viewer** è possibile vedere il contenuto di ogni schermata della propria applicazione.

Componenti

Nella colonna **Componenti** si possono vedere in una struttura ad albero tutti i componenti (visibili e non visibili) di una schermata. È possibile rinominare o cancellare un componente.

Media

Dalla **colonna Media** è possibile organizzare i file multimediali (file audio, immagini e video).

Proprietà

Nella colonna **Proprietà** è possibile determinare l'aspetto e le altre caratteristiche dei componenti.

Co-funded by the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Risorse online

Un rapido riferimento per il progettista
https://appinventor.mit.edu/explore/sites/all/files/Teach/media/MITAppInventorQuickReference.pdf

Tutorial su come utilizzare i componenti dell'interfaccia utente
http://www.appinventor.org/content/howDoYou/UIModules

Un elenco dei componenti di base di App Inventor
https://appinventor.mit.edu/explore/content/basic.html

2.3 - Programmazione con blocchi

Il **Blocks Editor** gestisce i blocchi di codice e programma il comportamento dell'applicazione. I blocchi sono forme simili a pezzi di puzzle che vengono utilizzati per creare un programma.
Le applicazioni create da App Inventor sono guidate dagli eventi. Le istruzioni non vengono eseguite in un ordine predefinito, ma rispondono agli eventi. Un evento è un'azione come cliccare un pulsante o toccare lo schermo. Anche scuotere il telefono è un evento. Il programmatore descrive come il telefono dovrebbe rispondere agli eventi utilizzando i blocchi di **gestione degli eventi**.

Blocchi incorporati

I blocchi incorporati (Controllo, Logica, Matematica, Testo, Liste, Colori, Variabili e Procedure) sono disponibili indipendentemente dai componenti del progetto.

I blocchi di controllo sono utilizzati per prendere decisioni.

Blocchi logici.

I blocchi matematici sono i blocchi che svolgono funzioni matematiche.

I blocchi di **testo** sono i blocchi che eseguono la gestione delle stringhe.

Con i blocchi **Liste** è possibile gestire le voci di una lista.

Con i blocchi delle **variabili** è possibile gestire le variabili globali e locali. Ci sono cinque tipi di blocchi di variabili (initialize global name to, get, set, initialize local name to in-do and initialize local name to in-return).

Blocchi specifici dei componenti

Molti dei componenti dell'applicazione possono avere blocchi di **chiamata di metodo** nel loro cassetto Component-Specific. Un blocco di **chiamata di metodo** è un insieme predefinito di istruzioni che permette di utilizzare le funzionalità del componente.

Blocchi di procedura

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Con i blocchi di **procedura** è possibile gestire le procedure. Una procedura è una sequenza di istruzioni che viene raggruppata sotto un nome ed esegue un compito specifico. Invece di avere programmi complessi è possibile creare procedure e invocarle ogni volta che si desidera utilizzarle.
La procedura e molti altri blocchi, tra cui 'if-else' e 'make list', forniscono un pulsante blu **mutatore**. Se si fa clic su di esso, è possibile trasformare il blocco, ad esempio, aggiungendo argomenti alle procedure o rami 'else' a un if.

Risorse online

Questa pagina spiega come funzionano i blocchi di base.
https://appinventor.mit.edu/explore/understanding-blocks.html

Una panoramica dell'architettura di un'App
http://www.appinventor.org/Architecture2

Questo tutorial mostra come creare procedure in App Inventor
http://appinventor.mit.edu/explore/ai2/support/concepts/procedures.html

Questo tutorial mostra come chiamare le procedure in App Inventor
http://ai2.appinventor.mit.edu/reference/blocks/procedures.html#do

Una rapida panoramica sull'utilizzo dei mutatori in App Inventor
https://appinventor.mit.edu/explore/ai2/support/concepts/mutators.html

2.4 - Collegamento a un telefono o a un tablet

Impostazione di App Inventor

È possibile eseguire le applicazioni sul dispositivo Android in due modi diversi: tramite connessione Wi-Fi (AI Companion) o via cavo USB. Nel primo caso è necessario che sia il computer che il dispositivo Android siano collegati alla stessa rete Wi-Fi. Se non si dispone di un telefono o tablet Android, è comunque possibile creare applicazioni con App Inventor. App Inventor fornisce un emulatore Android, che funziona proprio come un dispositivo Android, ma tutto appare sullo schermo del computer. E' possibile testare le App sull'emulatore e distribuirle ad altri anche attraverso il Play Store. In questo caso è necessario che l'App Inventor Companion App sia installata sul dispositivo.

Installazione e funzionamento dell'emulatore in AI2

I passi per l'installazione e l'esecuzione di App Inventor sono i seguenti:

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Passo 1. Installare il software di installazione di App Inventor

Fase 2. Avviare aiStarter (solo Windows & GNU/Linux)

Fase 3. Aprire un progetto App Inventor e collegarlo all'emulatore

Collegamento a un telefono o a un tablet con un cavo USB

I passi per iniziare a utilizzare App Inventor con il cavo USB sono i seguenti:

Passo 1: Installare il software di configurazione di App Inventor

Passo 2: Scaricare e installare l'applicazione MIT AI2 Companion App sul telefono

Fase 3. Avviare aiStarter (solo Windows & GNU/Linux)

Passo 4: Impostare il dispositivo per USB (Attivare il debug USB)

Fase 5: Collegare il computer e il dispositivo e, se necessario, autenticarsi

Passo 6: Testare il collegamento

Connettere il telefono o tablet via WiFi

I seguenti passi guidano attraverso il processo:

Passo 1: Scaricare e installare l'applicazione MIT AI2 Companion App sul telefono

Fase 2: Collegare sia il computer che il dispositivo alla stessa rete WiFi

Passo 3: Aprire un progetto App Inventor e collegarlo al dispositivo

Risorse online

Una panoramica sul settaggio di App Inventor

http://appinventor.mit.edu/explore/ai2/setup.html

Una panoramica dell'installazione e del funzionamento dell'emulatore

http://appinventor.mit.edu/explore/ai2/setup-emulator.html

Una panoramica del collegamento a un telefono o a un tablet con un cavo USB

http://appinventor.mit.edu/explore/ai2/setup-device-usb.html

Una panoramica della connessione a un telefono o a un tablet via WiFi

http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html

Co-funded by the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Capitolo 3: Creare giochi con App Inventor

3.1 - Come creare un gioco semplice

Di seguito sono elencati i passi indicativi che possono essere seguiti per creare un gioco semplice utilizzando App Inventor.

Disegnare l'interfaccia grafica del gioco.

Nell'App Inventor Designer:

- Selezionare i componenti (pulsante, etichetta, ecc.) dalla **tavolozza**.
- Trascinare i componenti selezionati nell'area del **Visualizzatore** e organizzali per disegnare l'aspetto del gioco.
- Dalla tavolozza **Layout**, è possibile utilizzare le disposizioni dello schermo per creare **HorizontalArrangement** (da sinistra a destra) , **VerticalArrangement** (dall'alto verso il basso allineato a sinistra), o **TableArrangement (**in modo tabellare) dei componenti.
- I componenti selezionati vengono posizionati nell'area **Componenti** e nominati da App Inventor. È possibile modificare il nome di questi componenti o cancellarli, se necessario.
- Aggiungere file multimediali (immagini, suoni) al gioco caricandoli dal computer. I file multimediali sono elencati nell'area Media sotto la colonna **Componente**.
- Facendo clic su un componente dall'area del Visualizzatore è possibile modificarne i dettagli (titolo, colore di sfondo, ecc.) che sono elencati nell'area Proprietà.
- Dalla Tavolozza Disegni e Animazione selezionare e configurare Canvas, se necessario. Sul pannello rettangolare bidimensionale sensibile al tocco di Canvas è possibile disegnare e spostare Balls o ImageSprites.
- Aggiungere un componente Clock, se necessario. L'orologio funziona come un timer che scatta a intervalli regolari. Si tratta di un componente non visibile che fornisce manipolazioni e calcoli che trattano il tempo utilizzando l'orologio interno del telefono o del tablet.

Definire il comportamento dei componenti - The Blocks Editor

- Creare e inizializzare le variabili globali e locali.
- Selezionare un cassetto incorporato (Controllo, Logica, Matematica, Testo, ecc.) o un cassetto specifico del componente (un cassetto per ogni componente creato nel Designer) per ottenere un gruppo di blocchi per quel componente. Ogni componente specifico ha blocchi per i gestori di eventi, le chiamate di metodo, l'impostazione di proprietà e l'ottenimento di proprietà.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Scegliere e trascinare i blocchi appropriati nell'area Viewer.
- Configurare questi blocchi e collegarli in modo appropriato per impostare il comportamento di gioco.
- Definire le procedure per creare nuovi blocchi di istruzioni che possono essere utilizzati ripetutamente.
- Utilizzare la procedura e le chiamate di metodo.

3.2 - Creare un gioco di pong con App Inventor

Pong è un gioco utilizzato per dimostrare le fasi di sviluppo dei giochi come descritto nella Sezione 3.1. I due documenti seguenti descrivono come si può sviluppare il gioco **Pong.**

Progettare l'interfaccia grafica del gioco del Pong
Definire la funzionalità dei componenti del gioco Pong -The Blocks Editor

Risorse aggiuntive

https://appinventor. mit. edu/explore/sites/.../Pong%205a_13.pdf
http://www. appinventor. org/content/ai2apps/intermediateApps/pong

3.3 - Creare un gioco MoleMash con App Inventor

MoleMash è un altro gioco utilizzato per dimostrare le fasi di sviluppo dei giochi descritte nella Sezione 3.1. I due documenti seguenti descrivono come può essere sviluppato il gioco **MoleMash.**

Progettare l'interfaccia grafica del gioco MoleMash
Definire la funzionalità dei componenti del gioco MoleMash - The Blocks Editor

Risorse aggiuntive

Mole Mash - App Inventor Tutorial Parte 1 https://youtu.be/Ya1ejdRwKvw
Mole Mash - App Inventor Tutorial Parte 1 https://youtu.be/IPSG7bYXN2M

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Capitolo 4 - La realtà aumentata di Metaverse

4.1 - Che cos'è Metaverse

Metaverse è una piattaforma web gratuita e di facile apprendimento che permette a insegnanti e studenti di creare le proprie esperienze interattive di Realtà Aumentata (AR) senza bisogno di codice. Le esperienze di AR vengono sviluppate sul web nello Studio Metaverse e poi visualizzate sulla Metaverse Mobile App (iOS e Android).

Cosa possono creare insegnanti e studenti con Metaverse? Le possibilità sono infinite, ma ecco alcuni esempi indicativi:

- Cacce al tesoro AR,
- Giochi AR,
- Storie interattive AR,
- Puzzle interattivi e quiz AR,
- Viaggi virtuali sul campo AR
- Pareti multimediali

Quali sono i vantaggi per gli studenti? La creazione di esperienze interattive di AR può migliorare l'alfabetizzazione e le competenze digitali degli studenti, la creatività e il pensiero logico. Inoltre, gli studenti si impegnano maggiormente nella procedura di apprendimento, applicano le loro conoscenze e iniziano a pensare e ad agire come ricercatori, a seconda della materia su cui si basa il loro progetto di AR. Gli insegnanti possono utilizzare Metaverse in qualsiasi materia con studenti di qualsiasi grado, promuovendo l'apprendimento basato su progetti, l'apprendimento attraverso il fare e l'apprendimento collaborativo.

Risorse online

Che cosa hanno da dire gli educatori su Metaverse: In questo video gli educatori che usano Metaverse con i loro studenti, condividono le loro esperienze e le risposte e i feedback dei loro studenti.
Anteprima Metaverse: Una rapida anteprima dello Studio Metaverse e della Metaverse Mobile App.
Portali AR in diretta: Questo video mostra i portali AR che "teletrasportano" gli utenti in vari luoghi del mondo. I portali sono stati creati con una telecamera a 360° e Metaverse.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Classe elementare di Joe Merrill's Lake Park: Gioco di vocabolario AR per studenti delle elementari, creato con Metaverse.

4.2 - Iniziare ad utilizzare lo Studio Metaverse

Per iniziare a utilizzare Metaverse Studio è necessario prima creare il proprio account nella piattaforma e poi familiarizzare con l'ambiente delle app. Il seguente documento fornisce i passi da seguire per iniziare ad utilizzare Metaverse e una breve presentazione dell'ambiente di Metaverse Studio.

Utilizzate questo documento per **familiarizzare con Metaverse Studio**

Risorse online

Panoramica dello Studio Metaverse: Dimostrazione delle modalità di base dell'utilizzo dello Studio Metaverse per creare Esperienze di Realtà Aumentata.
Panoramica del cruscotto Metaverse: Una rapida panoramica della pagina Metaverse Dashboard.

4.3 - Creare un'esperienza di AR

Metaverse Studio offre molti strumenti per creare facilmente un'esperienza AR, semplice o più complicata. Il seguente documento presenta i passi fondamentali da seguire per iniziare a creare esperienze di AR.

Utilizzare questo documento per **creare un'esperienza di AR in Metaverse**

Risorse online

Panoramica di Metaverse Storyboard: Una rapida panoramica dello Storyboard Metaverse Experience.
Scene di personaggi: Dimostrazione di come si possono utilizzare le scene di personaggi.
Scene di inserimento testo: Dimostrazione di come possono essere utilizzate le scene di inserimento testo.
Esperienze editoriali: Una rapida panoramica su come pubblicare le vostre esperienze di Metaverse.
Aggiungere una posizione GPS: Tutorial su come aggiungere una posizione GPS ad una Metaverse Experience.
Costruire un breakout digitale: Imparare a costruire un'attività di Digital Breakout per la vostra classe in meno di 15 minuti!

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Guardare l'esperienza qui: https://studio.gometa.io/discover/me/...

Costruisci un semplice gioco Trivia: Imparare a costruire un semplice gioco a 3 domande che utilizza i timer di scena per le domande e traccia i punti utilizzando i blocchi delle proprietà.

Co-funded by the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.