



**UNIVERSITÀ DI PISA**

Corso di Laurea in Informatica Umanistica

RELAZIONE

**Tecnologie di Realtà Aumentata per i beni  
culturali.**

**Un'applicazione al Parco di Pinocchio.**

**Candidato:** *James Magnavacca*

**Relatore:** *Massimo Magrini*

**Correlatore:** *Cristina Savettieri*

Anno Accademico 2020-2021

## Indice Generale

1. Introduzione .....	3
2. Storia della realtà aumentata nei beni culturali .....	6
2.1 La Realtà Aumentata.....	6
2.1.1 Cos'è la realtà aumentata? .....	6
2.1.2 Breve storia .....	7
2.1.3 Settori di applicazione.....	8
2.2 Tecnologie di Realtà Aumentata applicate ai beni culturali.....	11
3. Il parco di Pinocchio e la Piazzetta dei Mosaici.....	16
3.1 Il progetto e la realizzazione del Parco .....	16
3.2 La Piazzetta dei Mosaici .....	17
4. Il progetto VERO.....	19
5. Tecnologia AR, Unity 3D .....	20
5.1 Hardware per la Realtà Aumentata .....	20
5.2 Tracciamento.....	22
5.3 Interazione con l'utente.....	25
5.4 I contenuti virtuali.....	25
5.5 Ambiente di sviluppo, Unity 3D .....	26
6. Strategie di sviluppo, problemi e soluzioni .....	30
6.1 Perché Unity 3D.....	30
6.2 Creazione del progetto .....	31
7. Futuri Sviluppi, migliorie .....	41
8. Conclusioni.....	43
9. Bibliografia.....	45
10. Sitografia .....	46
11. Appendici .....	48
11.1 Ordine dei mosaici secondo la cronologia del romanzo. ....	48
11.2 Codici Completi .....	52

# 1. Introduzione

Pinocchio dalla sua prima apparizione sul “Giornale dei bambini” ha ispirato autori, scultori e artisti che hanno offerto la loro personale interpretazione del burattino e degli altri protagonisti del romanzo. Questa pluralità di forme trova completa giustificazione all’interno della storia di Collodi: Pinocchio stesso muta continuamente durante le sue avventure. Da pezzo di legno diventa burattino, grazie all’esperta mano di Geppetto; dopo aver lasciato casa da persona a modo, con la promessa di andare a scuola, viene scambiato per una marionetta nel teatro di Mangiafoco e per poco non torna ad essere un semplice tronco da ardere. Subisce prima un’apparente trasformazione animale, sostituendo il cane da guardia di un vecchio contadino, e poi una apuleiana metamorfosi asinina nel Paese dei Balocchi.

Non è solo Pinocchio a cambiare forma nell’opera: la figura della fata turchina, prima bambina, abitante solitaria di una casa nel bosco, torna sull’isola delle Api Operose come donna di campagna, si ripresenta come spettatrice nello spettacolo del circo e capretta sull’isola in mezzo al mare, fino a diventare la principale protagonista nella conversione definitiva di Pinocchio a bambino. La maggior parte dei personaggi, cambia e si trasforma, alcune volte rappresenta una crescita interiore, altre volte invece si tratta solo di un mutamento esteriore.

Come il suo protagonista, un burattino di legno diventato bambino, anche il romanzo è diventato molte altre cose, reinterpretato e riletto in continuazione già subito dopo la sua pubblicazione. Le sole edizioni illustrate, comparse già ai tempi di Carlo Lorenzini, testimoniano l’incredibile carica immaginifica offerta dal romanzo, che con la sua ambientazione non definita in un luogo preciso, ha stuzzicato la fervida fantasia dei disegnatori che nel corso degli anni hanno fatto calpestare a Pinocchio luoghi sempre diversi. Enrico Mazzanti<sup>1</sup> interpreterà il romanzo come una favola per bambini, interpretazione che si rispecchierà pienamente nelle sue illustrazioni, dal tratto non troppo preciso, legato al meraviglioso mondo della fantasia. Stile completamente

---

<sup>1</sup> Nato nel 1852.

diverso e antitetico è quello adottato dal suo successore, Carlo Chiostrì<sup>2</sup>, che percorrerà invece la strada del realismo, aiutato da un tratto preciso e ricco di dettagli. Non va dimenticata la - molto criticata - versione di Walt Disney, che preferì inserire Pinocchio in un ambiente dai tratti teutonici, invece della classica rappresentazione nella campagna toscana.

Troviamo anche tavole dalle tinte più fosche, gotiche e quasi orrorifiche. L'edizione curata da Giambattista Galizzi dimentica completamente la natura scanzonata e fanciullesca del burattino, rappresentando un mondo crudele che scatena tutte le sue forze maligne sul povero protagonista. Gris Grimly nel 2002 ne propose una versione ancora più oscura, basandosi sulle atmosfere scaturite dalla penna di Edgar Allan Poe, con abbondante uso del nero, che dona un aspetto sinistro e indefinito a ogni scena, facendo presagire una macabra fine.

Molto particolare fu il rapporto di Venturino Venturi con Pinocchio, figura a cui si avvicinò per la prima volta dopo il 1954, quando ottenne la commissione per la Piazzetta dei Mosaici all'interno del Parco di Pinocchio a Collodi. L'opera, circa 900 metri quadri di mosaici, segnerà per sempre la sua vita, portandolo anche ad un esaurimento nervoso della durata di due anni. Venturi non abbandonerà mai più la figura del burattino, su cui ritornerà più volte, non più in forma di mosaici ma di illustrazioni, caratterizzate da figure poco definite, quasi ideali o platoniche. Come scrisse più tardi Venturino Venturi, "Pinocchio è stato per me il pretesto di fare un uomo", e analizzando le sue opere, possiamo dire che la sua rappresentazione di Pinocchio e delle sue avventure rappresenti, in parte, le angosce e le fatiche della vita tipicamente umane.

Le avventure di Pinocchio non terminano nel 1883 con il burattino abbandonato esanime su una sedia, ma continuano nella folta schiera di riscritture, dette anche "pinocchiate"<sup>3</sup>. A soli dieci anni dalla conclusione del romanzo, nel 1893, compare la prima riscrittura, *Il figlio di Pinocchio* di Oreste Boni, che dà il via alle pinocchiate che marcheranno stretto tutto il Novecento, fino a sfociare nei primi anni duemila.

---

<sup>2</sup> Nato nel 1863.

<sup>3</sup> Termine già in uso nel '500 in riferimento a una pasta dolce. Fu Bruno Migliorini nel 900 ad associarlo per la prima volta a Pinocchio, con il significato di "azione degna di Pinocchio". Fu Rodolfo Biaggione ad usare ginocchiate in riferimento a quel filone letterario legato al romanzo di Collodi.

Pinocchio per tutto il XX secolo si lancia nel moderno, si attualizza, a volte anticipa le evoluzioni tecnologiche future. Lo vediamo abbandonare la campagna toscana per trasferirsi nella capitale, abbiamo un Pinocchio avventuriero e colonialista fino al Pinocchio reso protagonista della propaganda fascista, in camicia nera con manganello e olio di ricino.

Prima di scendere nella pura e semplice ideologia politica, Pinocchio veste ancora il mantello della fantasia, fa esperienza delle nuove tecnologie, prende il volo in *Pinocchio in dirigibile* di Epaminonda Provaglio e corre in auto in *Pinocchio in Automobile* di Giulio Erpianis. Numerose sono anche le riscritture che vedono protagonista il nostro burattino in avventure dal forte sapore salgariano, capaci di unire favola e avventura in ambienti esotici inesplorati.

La presenza tecnologico-futurista è una costante in tutte le pinocchiate del Novecento, troviamo un esercito di cloni in *Settemila fratelli di Pinocchio* di Luigi Barberi, fino a riconoscerlo in un moderno mostro di Frankenstein<sup>4</sup>, un costruito elettromeccanico ne *Il nuovo Pinocchio* dell'autore rumeno Dan Voivod, che ha dato alla sua storia il sottotitolo di *burattino atomico*.

Pinocchio non ha abbandonata la forma di burattino solo per diventare un bambino, ma ha continuato a cambiare, saltando anche tra una forma di medium e l'altra. Non vanno dimenticate le più moderne interpretazioni di Pinocchio: il film *AI* di Steven Spielberg e la traduzione in Emoji del capolavoro uscito dalla penna di Carlo Lorenzini, che ne offrono una visione contemporanea e attuale. In questo contesto, la possibilità di fargli fare il salto definitivo nel mondo digitale all'interno di un'applicazione in realtà aumentata rappresenta una nuova opportunità per continuare la sua storia infinita.

---

<sup>4</sup> Il tema dell'automa, del costruito umano animato artificialmente, non è completamente estraneo al romanzo di Carlo Lorenzini, che ne dà con il suo burattino una personale interpretazione, come precedentemente avevano fatto Hoffman nel suo *Der Sandmann* - l'uomo della sabbia - e Mary Shelley proprio con *Frankenstein*.

## 2. Storia della realtà aumentata nei beni culturali

### 2.1 La Realtà Aumentata

#### 2.1.1 Cos'è la realtà aumentata?

Le tecnologie di realtà aumentata hanno il potenziale di collegare in modo automatico e interattivo l'enorme quantità di informazioni oggi disponibili e la realtà in cui viviamo.

Dove la realtà virtuale (VR) trasporta l'utente in un mondo audio-visivo immaginario completamente isolato da ciò che lo circonda, la realtà aumentata (AR) cerca di colmare il divario sovrapponendo i contenuti digitali al mondo reale. Le informazioni diventano quindi parte dell'ambiente circostante, apparentemente contestualizzate nello spazio fisico.

Azuma<sup>5</sup> fu il primo a dare una definizione precisa di Realtà Aumentata elencandone le caratteristiche fondamentali di un'applicazione in realtà aumentata:

- Combinazione di reale e virtuale. La visione di poter vedere contenuti virtuali in qualsiasi punto dello spazio fisico offre numerose possibilità.
- Interazioni in tempo reale. L'interattività all'interno di una applicazione in realtà aumentata è fondamentale: anche quando l'utente ha a che fare con contenuti statici interagisce comunque con lo spazio fisico nel quale prendono vita le animazioni.
- Rendering 3D. La registrazione nello spazio fisico della scena in realtà virtuale consente all'utente di scegliere il proprio punto di vista personalizzato, adattando così l'esperienza alle scelte di ogni singola persona.

La combinazione di reale e virtuale traccia una linea netta con la realtà virtuale che immerge l'utente in un mondo completamente digitale; questa unione può essere ottenuta in vari modi: la soluzione oggi preferita è la combinazione di una o più videocamere e un display per mostrare insieme realtà e informazioni digitali.

L'interazione in tempo reale invece distingue la realtà aumentata dagli effetti speciali dei film; gli input dell'utente con il sistema di realtà aumentata può avvenire tramite azioni naturali come guardare e camminare, o azioni specifiche grazie a bottoni, gesti,

---

<sup>5</sup> Ronald Azuma è un informatico americano, unanimemente riconosciuto per il suo contributo allo sviluppo delle tecnologie di realtà aumentata.

tocchi. La rappresentazione in tre dimensioni è fondamentale per immergere ulteriormente l'utente all'interno dell'esperienza.

Questa definizione non limita la realtà aumentata ad un hardware specifico e non ne restringe i contenuti ai soli media visivi; sono infatti possibili esperienze tattili, uditive e olfattive.

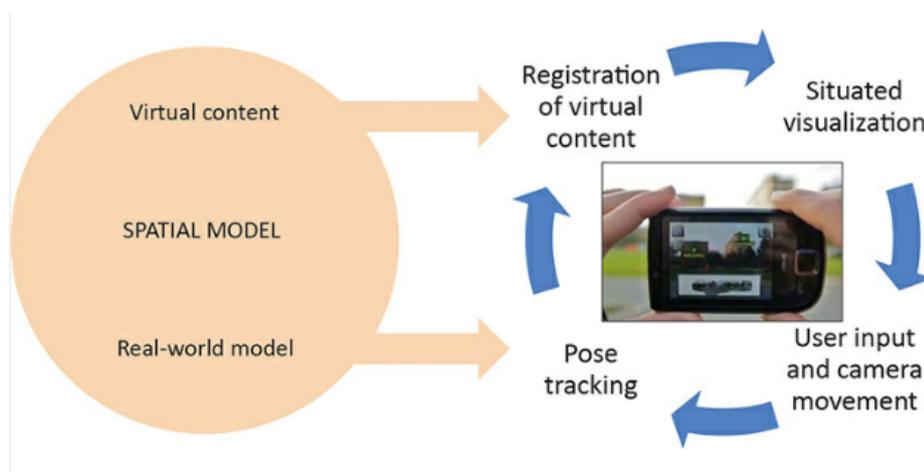


Figura 1 - Grafico che rappresenta il ciclo completo di un'applicazione in realtà aumentata

All'interno di un sistema di realtà aumentata basata su contenuti visuali abbiamo quattro passaggi che si ripetono in un ciclo continuo:

- Input dell'utente e visuale della fotocamera in tempo reale.
- Tracciamento della posizione tramite vari tipi di sensori (bussola, giroscopio e gps).
- Preparazione del contenuto in tre dimensioni.
- Visualizzazione dei contenuti sovrapposti in modo coerente alla realtà.

Tutto questo avviene senza soluzione di continuità, il sistema analizza e gestisce tutti i dati provenienti dai sensori, visuale della videocamera inclusa, mostrando le informazioni virtuale inserite, secondo le volontà degli sviluppatori, all'interno della scena reale.

### 2.1.2 Breve storia

Fu Ivan Sutherland a dare inizio agli studi che porteranno poi alla nascita di realtà aumentata e realtà virtuale. Nel 1968 costruì il primo prototipo di visore dotato di display semi-trasparente, ottiche e un sistema per il tracciamento della testa; a causa del

suo peso e ingombro, il dispositivo fu montato al soffitto guadagnandosi così il nome di Spada di Damocle.

Furono i progressi nella capacità di calcolo dei computer negli anni '80 e '90 a dare nuova linfa agli studi di realtà aumentata. Nel 1992 due ricercatori<sup>6</sup> alla Boeing coniarono il termine Augmented Reality (AR) in seguito alla creazione di un visore capace di aiutare gli operai durante la costruzione degli aerei.

Nel corso degli anni '90 si susseguirono numerose applicazioni di sistemi di realtà aumentata soprattutto nei campi medico e industriale.

In questo periodo vennero anche introdotte alcune nuove tecnologie, oggi alla base di numerose applicazioni di realtà aumentata. Iniziarono i primi esperimenti con l'uso del GPS, per la registrazione e il tracciamento della posizione dell'utente oppure per la creazione di una mappa virtuale dell'ambiente circostante. Emersero versioni primitive di dispositivi indossabili e portatili, muniti di schermi e numerosi sensori. Infine, vennero sviluppati i primi prototipi di ambienti in realtà aumentata condivisi dove più persone potevano interagire e usufruire degli stessi contenuti, partecipando alla stessa attività.

Tutte queste tecnologie, prevalentemente esclusive dei laboratori di ricerca, diventano nel 1999 disponibili per tutti in seguito al lancio di ARToolKit<sup>7</sup>, il primo software di sviluppo open-source per la programmazione di applicazioni in realtà aumentata.

La possibilità di poter disporre di un pacchetto di sviluppo gratuito e open source, i fondi messi a disposizione dai governi nazionali e, soprattutto, i notevoli passi avanti nelle tecnologie mobili, hanno reso possibile la rapida diffusione delle tecnologie di realtà aumentata in numerosi settori.

### **2.1.3 Settori di applicazione**

L'uso della realtà aumentata ha trovato sempre più spazio come ausilio al lavoro negli stabilimenti industriali e nei cantieri, che si basa sempre di più su modelli digitali preparati al computer (*Computer Aided Design*). L'aggiornamento dei modelli, con le

---

<sup>6</sup> Claudell e Mitchell.

<sup>7</sup> Kato, H., and Billinghurst, M. (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. Proceedings of the International Workshop on Augmented Reality

modifiche apportate durante la realizzazione vera e propria, è difficile e dispendioso in termini di tempo e quindi denaro; qui vengono in aiuto le applicazioni di realtà aumentata che permettono di avere il riferimento sempre a disposizione, sovrapposto al sito di costruzione con la possibilità di effettuare calcoli e modifiche in tempo reale.

Applicazioni in realtà aumentata si sono rivelate di grande aiuto durante l'uso e la manutenzione di numerosi macchinari, complessi e non. L'utente o l'esperto ha in sovrapposizione tutte le istruzioni necessarie sul funzionamento o la struttura interna dell'apparecchio, con tutte le informazioni sui componenti eventualmente da sostituire. Un impiego di questo tipo rende anche possibile varie forme di assistenza a distanza sotto la guida di un tecnico.

L'uso negli ospedali di tecnologie in realtà aumentata fu particolarmente precoce. Il settore medico si rivelò sin da subito particolarmente interessato alla possibilità di visualizzare contemporaneamente il paziente con i risultati di esami in sovrapposizione, impiego che è velocemente passato dalle visite alla sala operatoria.

Oggi è particolarmente inflazionato il mercato legato ad applicazioni che sfruttano alcune delle funzionalità legate alla realtà aumentata per fornire un valido ausilio e accompagnamento alle attività di tutti i giorni; alcune offrono informazioni su punti di interesse tramite l'uso del sistema di coordinate GPS, altre cercano di fornire una traduzione corretta di un testo inquadrato dalla fotocamera. La presenza di numerosi sensori forniti in tutti i moderni telefoni cellulare, uniti a fotocamera di alta risoluzione, ha permesso un'enorme diffusione di questa tipologia di software, ormai di uso quotidiano.

Sempre grazie a sistemi di geolocalizzazione, oggi discretamente precisi, esistono numerose applicazioni in realtà aumentata capaci di mostrare il tragitto, insieme ad altre utili informazioni, direttamente nel campo visivo del guidatore - o più semplicemente in un computer di bordo - con l'impiego di una o più fotocamere.

Per moltissime persone la prima esperienza con la realtà aumentata è probabilmente avvenuta davanti al televisore guardando una partita di calcio: la linea del fuorigioco, tracciata durante il replay di un'azione, è un'ottimo esempio infatti di informazioni digitali sovrapposte alla realtà in tempo reale.

Non possiamo assolutamente non citare il settore pubblicitario, che è riuscito ad applicare nei modo più svariati e originali il potenziale offerto da queste tecnologie. Gli acquirenti possono scegliere il nuovo arredamento, provando mobili e poltrone con l'applicazione "IKEA Place" o la funzione "View in your room" di Amazon, sono in grado di truccarsi virtualmente con le funzionalità offerte da "Sephora Virtual Artist" e hanno la capacità di trovare la giusta taglia per le loro prossime sneaker con "Nike Fit". Non dimentichiamo poi tutti i filtri, effetti e animazioni offerte dalle ultime versioni delle applicazioni fotografiche offerte dai social network più utilizzati, che sfruttano tecnologie proprie della realtà aumentata per migliorare e aggiungere contenuti a foto e video.

Finiamo cercando di inquadrare le possibilità offerte dalla realtà aumentata al mondo dell'intrattenimento virtuale. Cosa possono queste tecnologie aggiungere ad un settore ormai tecnologicamente molto avanzato come quello videoludico? Numerosi prodotti hanno dato una risposta proponendo di colmare il divario tra gioco analogico e digitale, unendo l'interazione con oggetti fisici e reali nel mondo virtuale. Una delle prime opere commerciali ad abbracciare questa nuova idea fu *The Eye of Judgement* per il sistema Playstation3: si trattava di un gioco di carte collezionabili, che inquadrare dalla camera proprietaria della console - *PSEye* - prendevano vita e combattevano sullo schermo del televisore. Altro progetto molto interessante è *Microsoft Illumiroom* che, grazie ad un proiettore particolarmente sofisticato, promette di estendere il mondo di gioco al di fuori dei limiti del pannello a cristalli liquidi.

## 2.2 Tecnologie di Realtà Aumentata applicate ai beni culturali

Sono sempre più numerose le applicazioni di realtà aumentata - o mista - legate al mondo dei beni culturali. Musei, siti archeologici e città d'arte sono molto interessati allo sviluppo di software in grado di valorizzare e riaccendere l'interesse per monumenti e opere. La possibilità di unire la visione della realtà ad una serie di contenuti digitali apre nuove opportunità di interazione con il sito e la grande diffusione di dispositivi mobili sempre più potenti ha reso incredibilmente accessibili queste tecnologie. Esistono applicazioni che mostrano come un edificio era al tempo della sua costruzione, grazie alla sovrapposizione di un modello digitale sopra le rovine oggi rimaste. Numerosi musei sfruttano la realtà aumentata per la creazione di installazioni, permanenti e non, in grado di unire l'analogico con il digitale, attivando l'interazione partecipe dell'utente con le opere e l'ambiente grazie ad applicazioni per smartphone, tablet e visori più sofisticati come *Hololens*<sup>8</sup> di Microsoft: percorsi ludici e narrativi prendono vita così direttamente nelle mani dei visitatori.

Inoltre, grazie all'uso della fotogrammetria e della scansione 3D, è oggi possibile vivere opere e monumenti anche a distanza, in realtà aumentata e virtuale. Moltissimi programmi permettono, inquadrando semplicemente una foto stampata su un libro o un opuscolo, di visualizzare il modello 3D di un'opera, un monumento o una scultura.

La stanza principale della biblioteca del Trinity College di Dublino, chiamata "Long Room", è protagonista di un'interessante installazione in realtà aumentata: questa camera, lunga quasi 65 metri e contenente circa 200.000 rari volumi antichi, ospita anche 14 busti di marmo raffiguranti i più grandi filosofi e scrittori del mondo occidentale. I ricercatori dell'università hanno sviluppato un prototipo in realtà aumentata dove un ridente Jonathan Swift, ricreato grazie all'uso di video volumetrici e interpretato da un attore professionista, guida il visitatore attraverso la splendida ambientazione offerta dalla biblioteca. I busti presenti nella stanza fungono da attivatori

---

<sup>8</sup> Si tratta di un visore di realtà mista, ergonomico e senza fili, che offre avanzate tecnologie come la mappatura in tempo reale di una stanza, tracciamento mani e occhi e controllo vocale. Al momento, dato il costo elevato, si tratta di un prodotto pensato per la produttività nei settori professionali.

per il contenuto digitale, ma, per invogliare l'esplorazione della collezioni, sono stati inseriti punti di interesse secondari, anch'essi attivati grazie all'uso della fotocamera e di un sistema di riconoscimento immagini. I ricercatori hanno svolto numerosi studi e sondaggi sull'uso dell'applicazione all'interno della biblioteca, che hanno dimostrato come queste tecnologie siano in grado di invogliare interazione e apprendimento. Si tratta anche di uno dei primi utilizzi di video volumetrici all'interno di esperienze di realtà aumentata e mista: questo ha permesso di creare un tour virtuale all'interno della biblioteca, presentando le informazioni in modo informale e umoristico grazie all'impiego di attori reali. I video, registrati con le più moderne tecnologie, sono stati poi applicati ad un modello 3D per offrire una visione a 360 gradi del personaggio.



Figura 2 - L'avatar di Jonathan Swift accoglie i visitatori all'interno della biblioteca

Un progetto ancora più imponente riguarda le grotte di Longmen, uno tra i più importanti beni culturali cinesi, situato nella vecchia capitale Luoyang. Le oltre 2300 grotte custodiscono innumerevoli sculture in pietra legate alla tradizione buddista. Nei primi anni del novecento il sito fu danneggiato e derubato e molte statue furono portate oltreoceano; proprio per questo oggi alcune nicchie sono inaccessibili al pubblico e numerose statue non rappresentano più l'antico splendore. La digitalizzazione delle opere, iniziata nel lontano 2005, ormai giunta quasi al termine, ha reso disponibili i modelli in tre dimensioni delle statue e delle scritte presenti. L'applicazione,

sviluppata dall'Università di Yale in collaborazione con l'associazione per i beni culturali cinese, abbraccia realtà aumentata e virtuale ed è stata resa possibile grazie all'enorme lavoro di digitalizzazione delle grotte. Le sculture tornano al loro antico splendore in realtà aumentata, con ricostruzioni accurate basate su studi storico-artistici molto approfonditi; in realtà virtuale invece, tramite visori e smartphone, è possibile visitare i siti momentaneamente chiusi al pubblico. Entrambe le esperienze reagiscono alle interazioni dei visitatori, grazie a un sistema avanzato di riconoscimento dei gesti e l'utilizzo di avanzate tecniche di *motion-capture* che assicurano un punto di vista reale e credibile dell'opera.

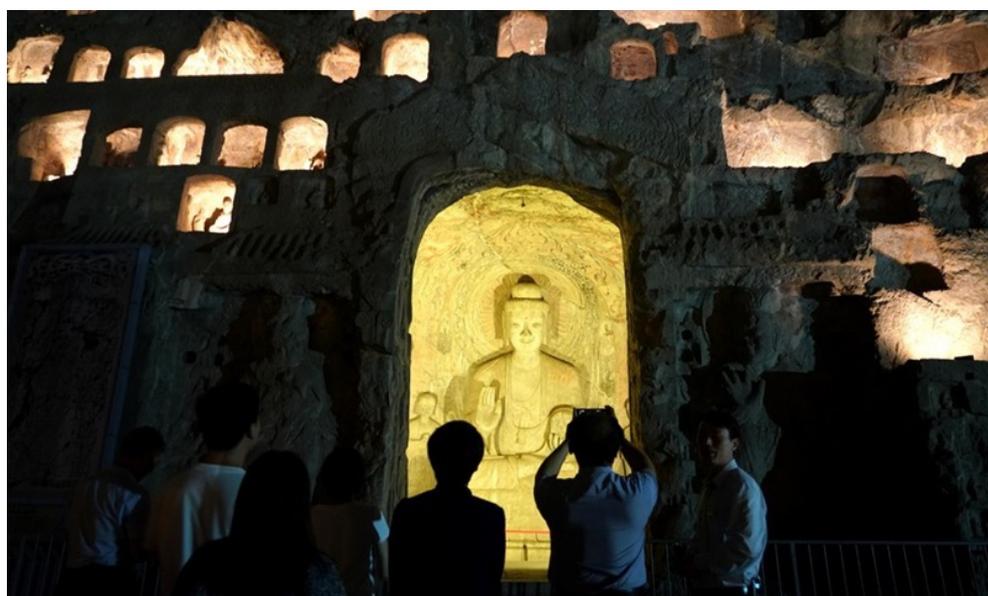


Figura 3 - Grotte di Longmen

Arte, narrativa digitale e storia si fondono nell'installazione in realtà aumentata all'interno del Kennin-ji, il tempio Zen più antico di Kyoto. Sviluppata in collaborazione con il Museo Nazionale di Kyoto, l'applicazione "MR Museum in Kyoto" sfrutta HoloLens di Microsoft per un'esperienza mista, legata ad un importantissimo dipinto raffigurante le divinità dell'aria e del tuono<sup>9</sup>. La rappresentazione olografica di un monaco del tempio guida i visitatori nelle varie parti del racconto mentre i vari motivi del dipinto prendono vita: vento, tuoni e fulmini, insieme alle loro divinità, si animano

---

<sup>9</sup> Shihon Kinji Chakushoku Fūjin Raijin-zu

in completo 3D volumetrico di fronte alle persone, che hanno la possibilità di apprezzare un tesoro nazionale in un modo nuovo e interattivo. L'uso della realtà aumentata ha permesso anche di rendere visibili altre opere, legate al tempo del vento e del tuono, presenti nelle altre stanze del tempio. Il progetto ha dimostrato come queste nuove tecnologie aprano le porte a nuove modalità di apprendimento all'interno dei musei, non solo grazie alla sovrapposizione di semplici informazioni di fronte ai visitatori, ma anche con la creazione di esperienze nuove legate alle opere da uno studio approfondito e alla sua traduzione in arte digitale. Inoltre, l'uso della realtà aumentata, richiede di visitare fisicamente il tempio e ha permesso di vivere l'opera in un modo alternativo capace di includere anche le nuove generazioni, ormai meno interessate al semplice aspetto culturale



Figura 4 - Le divinità dell'aria e del tuono prendono vita in realtà aumentata

Anche all'interno dei musei la realtà aumentata ha ottenuto enormi successi, sviluppando nuove opportunità di interazione con le opere, non legate al semplice sviluppo di guide virtuali.

All'interno del Museo Nazionale di Singapore, un gruppo del collettivo artistico nazionale giapponese, *teamLab*, ha trasformato le 69 opere, legate alla storia naturale delle isole, appartenenti alla collezione William Farquhar, in un'esperienza interattiva digitale presentata in tre dimensioni, con grandi schermi e proiezioni. I quadri e i

disegni della galleria, riguardanti principalmente flora e fauna della storia di Singapore e della penisola malese, sono stati re immaginati in una composizione artistica virtuale, caratterizzata da accesi colori dalle tonalità acquerello. Con l’ausilio di numerosi sensori posti all’interno della Gran Rotunda, la struttura che ospita l’installazione, le opere interagiscono con i visitatori lungo un percorso a spirale terminante in una cascata di petali che, all’approcciarsi delle persone, si trasformano in alberi rigogliosi. La galleria viene anche arricchita da un’applicazione in realtà aumentata per telefoni, che permette di trovare e catturare piante e animali presenti, registrandoli in una collezione virtuale e offrendo così numerose informazioni aggiuntive.

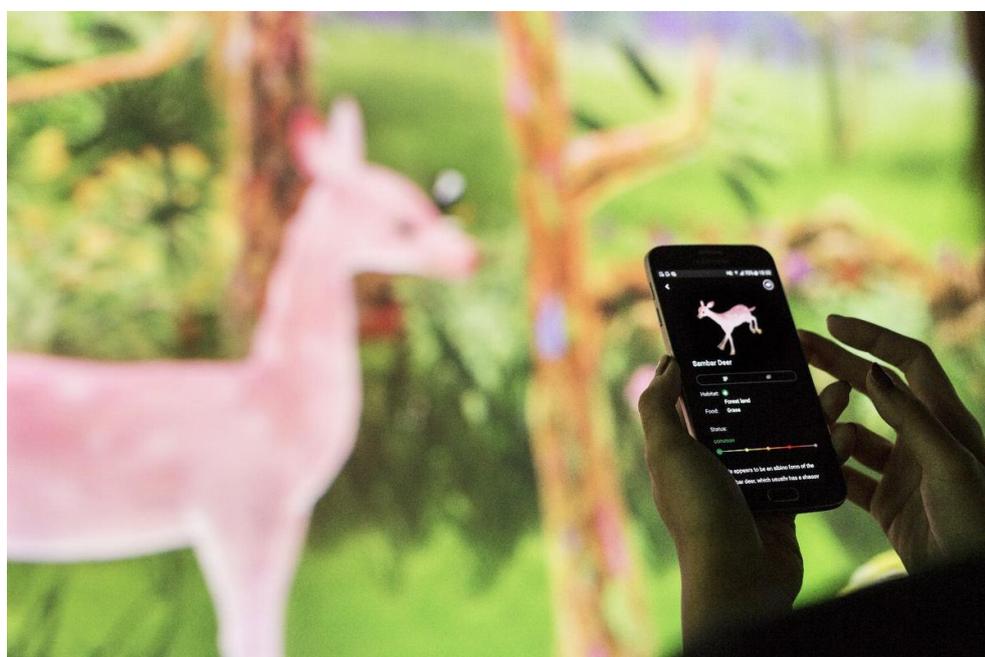


Figura 5 - Un visitatore cattura una abitante della Gran Rotunda

Un esempio tutto italiano è il progetto “Storie di Fiume e di Mare”, un’esperienza di narrativa virtuale interattiva in realtà aumentata, all’interno delle sale del Museo del Territorio “Luigi Ghirotti” a Riccione. L’applicazione, messa a punto dall’agenzia la Jetée di Firenze e dal laboratorio ISTI del CNR di Pisa, permette ai visitatori del museo di ascoltare e vivere quattordici storie riguardanti i fossili che abitano le teche: si tratta di tredici scene in realtà aumentata, completamente doppiate dagli attori della compagnia “Il paracadute di Icaro”, e una in realtà virtuale. Per attivare i contenuti in

AR è sufficiente inquadrare una delle tredici indicazioni poste a terra, attivando così una roccia, che una volta schiusa con un semplice tocco a schermo, rivelerà l'animale legato al fossile insieme alla sua storia.



Figura 6 - I visitatori ascoltano la storia del fossile

I progetti che abbiamo trattato in questa sezione dimostrano quanto l'applicazione di tecnologie di realtà aumentata legate ai beni culturali - musei, siti archeologici, collezioni - non punti alla completa scissione tra l'analogico e il digitale, bensì a un'unione di questi due mondi apparentemente distanti, in grado di arricchire l'esperienza del visitatore e aprire a nuove forme di interazione con la storia e la cultura passata.

### **3. Il parco di Pinocchio e la Piazzetta dei Mosaici**

#### **3.1 Il progetto e la realizzazione del Parco**

Nel settembre del 1951, a distanza di settant'anni dalla prima apparizione di Pinocchio sul Giornale per i Bambini, Rolando Anzilotti, al tempo sindaco di Pescia, ha l'idea di realizzare un monumento artistico in onore del burattino più famoso del mondo.

La volontà è quella di creare un centro culturale in grado di coinvolgere grandi e piccini. Proprio per questo il bando di concorso lascia ampia libertà agli artisti così da dare

carattere e valore al parco coinvolgendo così il visitatore in un'attiva partecipazione all'interpretazione delle opere.

Il parco - il primo parco a tema in Italia e uno dei primi in Europa - fu inaugurato il 14 maggio del 1956. Realizzato da artisti e architetti che hanno segnato in seguito il panorama italiano e internazionale, come Venturino Venturi ed Emilio Greco, questo monumento all'opera di Carlo Lorenzini non si limitava solo a riproporre e celebrare le avventure di Pinocchio, ma cercava di coinvolgere adulti e bambini in un viaggio di riscoperta tra arte e gioco.

Il nucleo centrale formato dal monumento di "Pinocchio e la Fata" di Emilio Greco e dalla "Piazzetta dei Mosaici" di Venturino Venturi venne in seguito ampliato con l'aggiunta dell'Osteria del Gambero Rosso e del Paese dei Balocchi, un percorso fantastico immerso nel verde dove trovano posto le sculture di Pietro Consagra e gli elementi architettonici di Marco Zanussi.

Il Parco di Pinocchio si presenta oggi come un grande percorso a sorpresa dove a ogni tappa adulti e bambini interagiscono insieme per rievocare e rivivere i momenti delle Avventure di Pinocchio, attraverso le attrazioni, le sculture e gli edifici dei grandi artisti che hanno lavorato negli anni all'opera.

### **3.2 La Piazzetta dei Mosaici**

La Piazzetta dei Mosaici fu realizzata dall'artista Venturino Venturi tra il primo novembre del 1954 e l'aprile del 1956. Il monumento è un quadrato di circa trenta metri per lato, chiuso da quattro muri perimetrali sagomati, su cui sono posti i mosaici raffiguranti personaggi e momenti delle *Avventure di Pinocchio*. Tutto è racchiuso in una cornice di lecci che inserisce la piazza all'interno dell'ambiente naturale del luogo; inoltre l'esterno, anch'esso decorato, e gli anfratti che intervallano l'opera contribuiscono alla coerenza compositiva del monumento.

Il progetto per la Piazzetta dei mosaici parte da un'idea innovativa: invece di seguire gli schemi classici del monumento pubblico o della statua commemorativa, Venturino, insieme agli architetti Renato Baldi e Lionello di Luigi, concepisce un contenitore di opere, una piazza aperta all'esplorazione e all'interazione con il visitatore.

Gli artisti abbracciarono completamente la filosofia del Bando di Concorso per il Monumento di Pinocchio a Collodi che così recitava: “È lasciata agli artisti la più ampia libertà di soluzioni architettoniche, plastiche e pittoriche; si fa presente, tuttavia, che l’opera convenientemente disposta in una sistemazione a verde dovrebbe ispirarsi al contenuto ed allo spirito del libro *Le Avventure di Pinocchio* di Carlo Collodi”<sup>10</sup>.

Venturi condensa nelle forme essenziali dei suoi mosaici tutta la forza espressiva dei personaggi del romanzo, proponendoli in una galleria all’aperto senza un ordine cronologico coerente con gli episodi della storia; in questo modo il visitatore è incoraggiato a percorrere più volte il percorso per scoprire ogni volta nuovi e improbabili accostamenti.

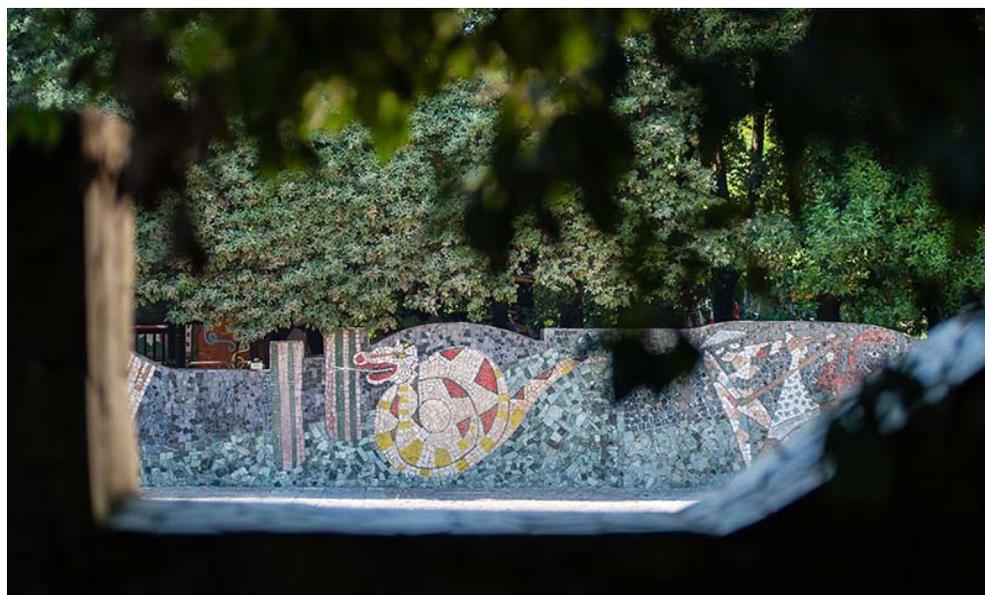


Figura 7 - Uno scorcio della Piazzetta dei Mosaici

Come punto di riferimento il progetto originale poneva al centro della piazza una statua gigante di Pinocchio, alta cinque metri. In sintonia con gli aspetti ludici e interattivi dell’opera, la statua, esattamente come lo *gnomone* di una meridiana, avrebbe dovuto proiettare la propria ombra sui mosaici scandendone il passaggio da una scena all’altra così come fa la meridiana con lo scorrere del tempo.

---

<sup>10</sup> Documento presente nell’Archivio della Fondazione Nazionale Carlo Collodi, (Collodi, Pescia).

Il Pinocchio gigante non fu mai realizzato, lasciando la piazza orfana del suo protagonista e l'esperienza del visitatore priva della bussola con cui orientarsi.

## **4. Il progetto VERO**

VERO - Virtualità intErattiva nel paRco di pinocchiO - è un progetto realizzato da dall'Istituto di Scienza e Tecnologia dell'informazione "A.Faedo" del Consiglio Nazionale delle Ricerche, in collaborazione con la Fondazione Carlo Collodi.

L'obiettivo è quello di realizzare un'applicazione in realtà aumentata per dispositivi mobili di ultima generazione, sfruttando le possibilità offerte dalle nuove tecnologie per valorizzare uno dei monumenti più importanti all'interno del parco: la Piazzetta dei Mosaici realizzata dall'artista Venturino Venturi.

Seguendo la filosofia alla base di tutta l'esperienza del parco, che ha da sempre incluso la visita partecipe e interattiva del pubblico con tutte le sue opere e attrazioni, l'applicazione permetterà di rivivere le scene immortalate nei tasselli dei mosaici, grazie a una serie di contenuti audio-visivi interattivi fedeli alla poetica dell'artista e al romanzo di Carlo Lorenzini.

Dopo aver scaricato l'app sul proprio dispositivo mobile - smartphone o tablet - i visitatori, inquadrando i mosaici con la fotocamera, attiveranno l'esperienza interattiva associata alla scena che prenderà vita in continuità con l'ambiente circostante della piazza.

Particolare attenzione sarà dedicata ai contenuti digitali, che consisteranno di animazioni in due e tre dimensioni, a cui lavoreranno artisti professionisti, e audio spazializzato virtualmente all'interno della piazza. Tutta l'esperienza si pone come ponte di collegamento tra la visione artistica di Venturino Venturi, basata sulla sua personale interpretazione della storia del burattino e che richiama fortemente luoghi e dintorni del Parco grazie alla scelta dei materiali, raccolti e lavorati personalmente dall'artista, e il romanzo di Collodi, che affascina da generazioni l'immaginario di adulti e bambini. L'applicazione conterrà anche una sezione più interattiva che consentirà ai visitatori della Piazza di riordinare i mosaici secondo la cronologia del romanzo,

invogliando a mettere in relazione la grande originalità delle opere di Venturi con le scene lette nel libro.

## **5. Tecnologia AR, Unity 3D**

### **5.1 Hardware per la Realtà Aumentata**

Una configurazione hardware adatta alla realtà aumentata deve essere in grado di gestire la fusione di contenuti reali e virtuali e mostrarli all'utente uniti in modo credibile e concreto. In pratica, devono essere presenti un'unità di calcolo per le operazioni, uno o più sensori per il tracciamento (tra cui la videocamera) e una forma di display per usufruire dei contenuti digitali. Oggi sono gli smartphone, molto diffusi e relativamente economici, a rappresentare la soluzione migliore per applicazioni commerciali legate alla realtà aumentata: il numero consistente di sensori di cui sono equipaggiati, le fotocamere sempre di maggior qualità e la facile portatilità li hanno resi perfetti per una diffusione a macchia d'olio delle tecnologie di realtà aumentata.

I primi prototipi hardware per la Realtà Aumentata presentavano uno schermo montato molto vicino agli occhi e una serie di sensori per il tracciamento della realtà circostante. Questa prima soluzione ha dato vita in seguito a due tipi distinti di display.

La prima tipologia è quella che prende il nome di *Optical See Through Display* (OST). Questa configurazione riflette i contenuti digitali su uno schermo trasparente, posizionati davanti agli occhi, sovrapponendoli a ciò che si trova nel campo visivo dell'utente, utilizzando stereoscopia od ologrammi; è così mantenuta inalterata la visione naturale che viene solamente potenziata dal sistema informatico. Il problema principale di questo "occhiale aumentato" risiede nella visualizzazione delle informazioni digitali che risultano spesso poco visibili a causa dell'intensità della luce solare.

Il secondo tipo di configurazione, denominata *Video See Through Display* (VST) è antitetica a quella precedentemente descritta. Questo tipo di visore sembra molto più simile a quelli dedicati ad esperienze di Realtà Virtuale completa. Davanti agli occhi è posizionato uno schermo a cristalli liquidi non trasparente, su cui vengono visualizzati

insieme realtà e contenuti digitali, grazie all'uso di uno o più videocamere montate sopra la testa. In questo caso la difficoltà principale si trova nella calibrazione delle fotocamera e nella scelta dell'angolo di visione più adatto. Meglio riprodurre fedelmente la vista naturale o offrirne una più adatta alla sovrapposizione delle informazioni digitali sul mondo reale? La risposta non è immediata e universale, ma spesso dipende dal tipo di esperienza che si vuole offrire.

Entrambe le soluzioni, oltre ai problemi propri delle tecnologie adottate, devono far fronte a tutti quelli relativi all'ergonomia dell'apparecchio. L'occhiale deve essere comodo, leggero e soprattutto poco invasivo; inoltre deve risultare socialmente accettabile al di fuori di contesti abituati all'uso di caschi e visori ingombranti oltre a essere abbastanza economico per diventare elettronica di consumo.

Esperienze in realtà aumentata non sono comunque un'esclusiva per apparecchi portatili, esistono infatti soluzioni più classiche o ancora più sperimentali e interessanti. Ne illustreremo velocemente alcune, partendo da quella più semplice ed economica, fino ad arrivare a quelle più originale particolarmente adatta a esibizioni e installazioni particolari, a tempo, o comunque limitate.

La configurazione più immediata è sicuramente quella offerta da un computer, fisso o portatile, munito di webcam, usb o integrata: la videocamera, esattamente come nelle configurazioni portatili, funge contemporaneamente da sorgente video e di tracciamento in tempo reale, con l'ausilio di tecniche di *image-recognition*<sup>11</sup>. La realtà unita agli elementi virtuali, con cui l'utente può interagire tramite metodi tradizionali, ad esempio mouse e tastiera, o inconsueti, come gesti o riconoscimento vocale, è così mostrata sul monitor del computer.

Esiste poi una particolare soluzione chiamata Specchio Virtuale; gli elementi, fotocamera frontale e display verticale, sono disposti in un sistema particolare in modo da simulare uno specchio. Questo formato si è dimostrato particolarmente adatto a tutte le applicazioni che richiedono la modifica dell'aspetto dell'utente, come la prova di nuovi vestiti, l'applicazione di filtri fotografici e giochi interattivi di gruppo. All'interno

---

<sup>11</sup> Tecniche che fanno parte degli studi informatici legati alla Computer Vision. Si tratta di tecnologie che mirano a rendere un sistema capace di raccogliere informazioni da immagini e video, basandosi sulla visione umana.

di questa configurazione è possibile sostituire la fotocamera, applicando uno specchio semitrasparente davanti al monitor a cristalli liquidi.

Grandi eventi o installazioni hanno anche implementato all'interno di esperienze in realtà aumentata l'uso dei proiettori, che sono stati capaci di ampliare il numero di persone in grado di interagire insieme all'interno della scena, interagendo in gruppo con i contenuti virtuali proiettati insieme alla realtà ripresa dalla videocamera. Non solo le immagini sono proiettate su grandi schermi di proiezione, vengono infatti impiegate numerose varianti, come pannelli trasparenti, oggetti reali in tre dimensioni e anche muri di nebbia artificiale, che permettono un'interazione fisica con l'immagine.

## **5.2 Tracciamento**

Un sistema di Realtà Aumentata deve registrare costantemente almeno la posizione e l'orientamento dell'utente, convertendo in tempo reale i dati ottenuti nel sistema di coordinate della corrispondente scena virtuale. In questo modo, con la sovrapposizione dei contenuti digitali alla visione del mondo circostante, si mescolano realtà e finzione. Anche se alcuni tipi di applicazione in AR possono funzionare con un sistema di tracciamento statico, cioè con una serie di sensori fissi in posizione arbitraria intorno all'utente o nel luogo interessato, la maggior parte delle applicazioni richiede una lettura dell'ambiente mobile e capace di adattarsi al punto di vista della persona.

La dotazione di sensori di cui è munito ormai ogni smartphone o tablet, sebbene a volte poco efficace o precisa, racchiude e riassume tutto ciò di cui una configurazione hardware per la realtà aumentata ha bisogno per una completa consapevolezza del mondo circostante. Giroscopio, ricevitore GPS, accelerometro, bussola e fotocamera sono una costante in tutti i telefoni cellulari - o tablet - in commercio; conseguenza di questo sono i numerosi software - app - che infestano i negozi virtuali presenti su Android e iOS, che in qualche modo sfruttano o credono di sfruttare le possibilità offerte dalle tecnologie di realtà aumentata. Analizzeremo di seguito, a grandi linee, il funzionamento di questi sensori e il tipo di dato che restituiscono al sistema.

La registrazione e il tracciamento costante di utente e oggetti reali avviene sui dispositivi mobili grazie a due tipologie di dati ben distinte: *dati grezzi*, offerti dal

pacchetto di sensori presente su ogni apparecchio e *dati visuali*, provenienti dall'analisi della sorgente video della fotocamera frontale e/o posteriore.

Esclusa la fotocamera, il resto dei sensori presenti sui telefoni riporta, in vari modalità, informazioni legate a posizione e orientamento del dispositivo e di chi lo sta usando. Il ricevitore GPS - Global Position System - permette di stimare altitudine, latitudine e longitudine di un punto terrestre; anche se non particolarmente rigorosi, i dati resi disponibili permettono lo sviluppo di applicazioni in realtà aumentata legate a luoghi specifici, come percorsi cittadini. Anche le connessioni senza fili, Wi-Fi e Bluetooth, permettono la triangolazione della posizione del dispositivo, all'interno di reti locali; soluzioni di questo tipo sono state adottate, in particolare, all'interno di mostre e musei per guide o tour virtuali. I dati raccolti da accelerometro e bussola, invece, consentono una transizione più fluida dei movimenti dell'utente all'interno della scena in realtà aumentata, registrando velocità e orientamento in tempo reale.

Spesso, per la scarsa precisione del singolo dispositivo, viene adottata una tecnica particolare, detta *Sensor Fusion*<sup>12</sup>, che unisce i dati provenienti da uno o più sensori per un risultato migliore. Sensori come quelli descritti in precedenza sono fondamentali per avere un sistema di tracciamento continuo e portatile, ma, come abbiamo visto, sono spesso poco precisi e non sempre affidabili; un'accuratezza superiore richiederebbe maggiore costo e volume dell'apparecchio, sacrificando così la portabilità del dispositivo. Così, per ottenere un'analisi migliore, gli sviluppi nelle tecnologie di realtà aumentata hanno portato a preferire i dati video ottenuti grazie alle fotocamere, poco ingombranti e soprattutto economiche. Analizzati con sofisticati algoritmi di Image Recognition, i milioni di pixel registrati mettono a disposizione dello sviluppatore un pozzo senza fondo di dati visuali.

Il riconoscimento e il tracciamento per immagini è forse il metodo più diffuso e affidabile per tracciare, identificare e memorizzare la posizione di oggetti e caratteristiche nell'ambiente reale.

---

<sup>12</sup> Tecnica complessa, nata per sopperire alle mancanze dei vari sensori. I dati provenienti da multipli sistemi di tracciamento vengono uniti per ottenere una maggiore precisione di insieme.

ARToolKit, il primo pacchetto software per lo sviluppo di applicazioni in realtà aumentata, mise fin da subito a disposizione un sistema di riconoscimento in grado di distinguere uno speciale formato di marchio in bianco nero personalizzabile, facilmente stampabile; in questo modo era possibile sovrapporre e attivare contenuti in base al simbolo acquisito dalla webcam.

Oggi le evoluzioni nelle tecniche di Image Recognition e i miglioramenti delle fotocamere digitali permettono il riconoscimento di oggetti, edifici e scritte a partire da una semplice fotografia, senza la necessità di utilizzare etichette particolari: questo ha permesso lo sviluppo di applicazioni in realtà aumentata anche per quei luoghi dove l'invasione dell'ambiente con segnali artificiali non era possibile (palazzi, siti archeologici).

I dati visuali ottenuti dalla sorgente video proveniente dalla videocamera sono alla base anche di tutta una serie di tecnologie, che non solo accrescono le modalità di interazione con persone e ambiente, ma aiutano anche a rendere più immersivi e credibili i contenuti digitali fusi con la realtà. La stima dell'intensità e del colore di una fonte luminosa aiuta, ad esempio, a inserire contestualmente gli oggetti nella scena, e, unita a un calcolo corretto della posizione e dell'angolo della luce, è possibile creare ombre fedeli a quelle reali.

L'analisi dei fotogrammi provenienti dalla fotocamera permette anche una serie di avanzate tecniche di tracciamento e riconoscimento, non più basate su immagini statiche. I software più avanzati per la realtà aumentata - *Vuforia*<sup>13</sup> - permettono di individuare e tracciare oggetti reali, come giocattoli e soprammobili, basandosi sul loro modello in tre dimensioni, utilizzando tecniche assimilabili alla fotogrammetria. Non sono rare invece, anche su dispositivi mobili - grazie a ARKit di Apple - funzionalità che rendono possibile un tracciamento completo del corpo umano, fermo o in movimento. Ormai comunissima è invece il riconoscimento facciale, usato in particolare in tutti i filtri fotografici, presenti prevalentemente sui più diffusi social network.

---

<sup>13</sup> Vuforia è forse la libreria più avanzata e completa disponibile per lo sviluppo di applicazioni in realtà aumentata. Oltre a un efficiente tracciamento immagine, include numerose funzionalità avanzate, come il riconoscimento oggetti basato su modelli 3D o anche la navigazione in un'intera area - al chiuso o all'aperto- scanzonata digitalmente.

## 5.3 Interazione con l'utente

La seconda caratteristica fondamentale per un'esperienza di Realtà Aumentata è l'interazione in tempo reale tra utente e contenuti, digitali e reali. Le persone ovviamente interagiscono con la realtà, si muovono e inquadrano, in modo volontario o meno, elementi dell'ambiente circostante con la fotocamera. Molto più complessa è l'interazione con i contenuti virtuali che mano a mano prendono vita, compaiono e scompaiono, nelle mani o davanti agli occhi dell'utente. Tutto questo è reso anche più difficile dall'immensa varietà di dispositivi che supportano in qualche modo applicazioni in realtà aumentata. Non tutti presentano lo stesso tipo di input, alcuni hanno esclusivamente un *touch screen*, altri nemmeno quello; la presenza di più bottoni e joystick è legata a sistemi specifici, come visori o console.

A prescindere dalle opportunità offerte dal dispositivo - in questo contesto non considereremo le potenzialità di input esterni o ausiliari - le interazioni all'interno di un'esperienza in realtà aumentata devono essere il più naturali possibili, dopotutto l'utente non è inserito in un'ambiente completamente virtuale. Proprio sull'interattività cadono molte applicazioni commerciali oggi disponibili, che risultano spesso eccessivamente artificiali e ricalcano le meccaniche tipiche di software classici, richiedendo numerosi input.

In un'app per smartphone, ad esempio, è auspicabile un'esperienza di questo tipo: la persona si muove nell'ambiente aumentato in modo naturale, attivando contenuti digitali inquadrando gli elementi a essi legati. Una volta presenti sulla scena, gli oggetti virtuali, invece di richiedere un'azione umana, si possono animare in base alla posizione del telefono.

## 5.4 I contenuti virtuali

Un'applicazione in realtà aumentata, presenta oggetti e ambienti digitali all'interno di una rappresentazione della realtà fornita, a seconda della configurazione hardware adottata, da una visione diretta o mediata dall'obiettivo della fotocamera. Tutto questo rappresenta la differenza principale con la realtà virtuale, che rimuove completamente il contatto tra utente e realtà. Abbiamo visto come un sistema hardware e software

dedicato alla realtà aumentata prenda coscienza del luogo circostante e come poi questi dati vengano messi a disposizione per unire le informazioni virtuali.

Per una esperienza completa è necessario che tutte le aggiunte virtuali siano in tre dimensioni, in grado così di unirsi in modo naturale alla visione della realtà, esprimendo fisicità e presenza materiale simili agli oggetti realmente presenti con cui condivideranno lo spazio. La visualizzazione di modelli in tre dimensioni permette anche l'uso di un'illuminazione naturale sulle superfici e la proiezione delle ombre, fondamentali per offrire una coesione ancora maggiore. Non vanno dimenticate tutte le altre possibilità offerte dalla terza dimensione, come l'opportunità di sfruttare il modello 3D per il calcolo delle collisioni con altri elementi reali o virtuali oppure la possibilità di inserire un sistema di calcolo della fisica per ogni oggetto.

Tutto questo vale anche per i contenuti non visuali, come la musica, i suoni ambientali ed effetti acustici. Anch'essi devono essere pensati in tre dimensioni, spazializzati nell'ambiente, capaci così di offrire informazioni sulla loro provenienza.

Lo scopo è sempre quello di creare una coesione naturale tra la realtà e la finzione, l'analogico e il digitale.

## **5.5 Ambiente di sviluppo, Unity 3D**

Unity 3D è un motore di gioco nato con l'obiettivo di rendere democratica e accessibile la creazione di videogiochi. In seguito al lancio nel 2005 durante la WWDC - World Wide Developers Conference - di Apple come strumento di sviluppo esclusivo per i computer dell'azienda di Cupertino, Unity ha aggiunto numerose frecce al suo arco: nuove funzioni, ma soprattutto il supporto multiplatforma. Oggi è infatti possibile esportare il proprio progetto su quasi tutti i sistemi hardware e software disponibili tra cui Mac, Windows, Android, iOS, le principali console attualmente in commercio e dispositivi di realtà mista (XR, AR e VR) come i visori di Oculus<sup>14</sup>, Microsoft HoloLens e Magic Leap<sup>15</sup>.

---

<sup>14</sup> Oculus VR è un marchio di Facebook Technologies che produce due linee di visori Oculus Rift e Oculus Quest, oltre a supportarne le librerie di sviluppo e gestire la vendita delle applicazioni con un negozio digitale proprietario.

<sup>15</sup> Magic Leap è un visore per la realtà aumentata che proietta direttamente negli occhi dell'utente le immagini 3D generate dal computer, sovrapponendole alla visione reale.

Unity si è rivelato subito particolarmente adatto a sviluppatori in erba, grazie ad un'interfaccia pulita e ad un sistema drag & drop semplice e intuitivo (è possibile creare un clone del classico Pac Man senza scrivere una riga di codice).

Unity permette di sviluppare giochi ed esperienze in due e tre dimensioni. In ambiente 2D Unity supporta l'uso di Sprite<sup>16</sup>, avanzate tecnologie di rendering e strumenti per facilitare la creazione degli ambienti; in 3D invece supporta modelli di vario formato, texture, ombre dinamiche e numerosi effetti di post processing avanzati.

Offre inoltre un'API in C#, linguaggio utilizzato anche per i plugin e lo scripting all'interno dei progetti.

Oggi Unity può essere utilizzato per lo sviluppo di molteplici applicazioni, non tutte legate al mondo dei videogiochi come film di animazione, effetti speciali, esperienze in realtà aumentata e realtà virtuale, applicazione di intelligenze artificiali e simulazioni in tre dimensioni per l'ambito medico, scientifico, di ricerca, industriale e automobilistico.

### **Concetto di GameObject**

Unity 3D basa il suo sistema sul concetto fondante e fondamentale di GameObject. Ogni elemento che fa parte di una scena di "gioco" (in senso lato) luci, personaggi, scenari, immagini, interfaccia sono nella loro forma primitiva GameObjects, che con l'aggiunta di componenti specifici, acquisiscono comportamenti particolari.

Questo sistema rende particolarmente intuitivo il lavoro sul progetto: lo sviluppatore a sempre a ne fare con la stessa classe di elementi a cui può via via aggiungere proprietà particolari, anche personalizzate con l'uso di script, che ne specializzano e differenziano le funzioni. Inoltre tutti i pacchetti aggiuntivi, forniti da Unity per supportare le numerose tecnologie aggiuntive, come la realtà aumentata e la realtà virtuale, presentano tutte la stessa struttura e basano la propria interfaccia sfruttando il concetto di Game Object. I nuovi elementi incorporati da questi pacchetti - nella maggior parte dei casi anche quelli forniti da sviluppatori terzi seguono questa struttura - contengono tutte le funzionalità, legate ad un compito specifico, all'interno dello stesso sotto forma

---

<sup>16</sup> Si tratta di oggetti di gioco in due dimensioni, sostanzialmente costituiti da una semplice texture, ma che ricevono particolare attenzione e ottimizzazione per permetterne l'uso all'interno di un videogioco.

di numerose proprietà; è così possibile combinare tutte le novità con le caratteristiche base dello Unity Engine, legate normalmente allo sviluppo di videogiochi.

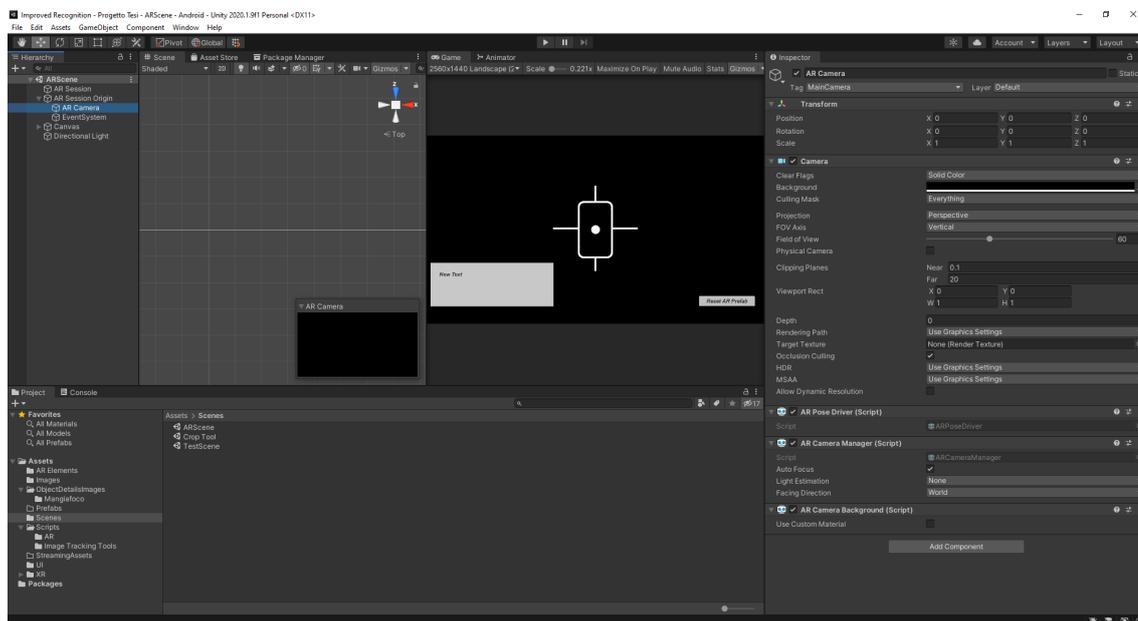


Figura 8 - Visione di insieme della finestra di Unity

## Unity Editor

La finestra principale di Unity presenta all'utente un'interfaccia semplice e intuitiva, composta da una classica barra degli strumenti e da una serie di pannelli modulari che possono essere scelti e disposti in base alle necessità del progetto. Nota distintiva dell'interfaccia, è la possibilità di estenderne e personalizzarne le caratteristiche con l'uso del codice.

## Project Window

Qui troviamo tutte le risorse disponibili a essere utilizzate all'interno del progetto, importate da file esterni come modelli 3D, immagini, tracce audio, o create direttamente all'interno del programma, file di codice e prefab.

## Scene View

Questa finestra mette a disposizione una panoramica grafica di tutti i contenuti all'interno della scena corrente; è così possibile avere un controllo diretto sulla posizione all'interno del livello di tutti gli elementi. Questa visuale è anche molto utile per analizzare in tempo reale tutti ciò che accade durante l'esecuzione del software;

unita alla console, posta in fondo alla finestra del programma, è uno strumento indispensabile per correggere eventuali errori o bug presenti.

### **Hierarchy**

Questo pannello mostra una lista di tutti gli oggetti all'interno della scena disposti secondo un ordine gerarchico, sottolineando in modo preciso i rapporti di dipendenza e parentela. Questi rapporti sono fondamentali per comprendere e gestire pienamente tutti gli elementi, in particolare quelli visivamente presenti all'interno di un livello. La posizione degli oggetti è infatti relativa al proprio genitore: nel caso generale è l'origine della scena a fungere da punto di riferimento per le coordinate, ma la posizione, la scala e la rotazione di tutti gli elementi figli di un altro sono relativi a quella di chi li precede nella gerarchia. Una corretta gestione di questa struttura permette maggiore facilità nell'identificare e raggruppare gli oggetti, in particolare all'interno del codice o nella scelta di cosa mostrare e non.

### **Inspector Window**

Tutte le proprietà di un oggetto selezionato vengono elencate all'interno di questa finestra: lo sviluppatore può prendere in esame ogni singolo componente associato, aggiungerne altri o modificarne i valori. In questa sezione sono anche mostrate tutte le variabili interne al codice che lo sviluppatore ha deciso di rendere visibili all'esterno, oltre a tutte quelle estensioni che ha deciso di applicare all'interfaccia.

### **Unity 3D Toolbar**

Nella barra degli strumenti trovano posto tutte le funzioni principali per lo spostamento e la modifica degli oggetti all'interno della scena, insieme ai pulsanti per l'avvio e il controllo dell'esecuzione del progetto in tempo reale.

### **Game View**

In questa finestra viene eseguita la scena in tempo reale. L'utente può controllare il corretto funzionamento del progetto, eventuali errori all'interno del codice e la generale presentazione grafica. Nella maggior parte dei casi è possibile eseguire in modo completo il software, ma alcune volte, come nel caso di alcune applicazioni che sfruttano tecnologie non presenti in un computer, è necessario provare il progetto direttamente sul dispositivo corretto.

## **Linguaggio di Programmazione C#**

Unity ha adottato C# per l'API ufficiale e la scrittura di codice all'interno dell'editor.

C# (pronuncia inglese "See Sharp") è un linguaggio di programmazione orientato a oggetti e componenti, sviluppato da Microsoft per la creazione di applicazioni all'interno dell'ecosistema .NET<sup>17</sup>; ha le proprie radici all'interno della famiglia C e condivide molte caratteristiche con i linguaggi C++, Java e JavaScript.

Il linguaggio C# è stato concepito per essere una via di mezzo tra la complessità dei suoi affini C e C++ e la semplicità tipica dei linguaggi dedicati prevalentemente al solo scripting, come Javascript; è possibile quindi creare codice per software molto complessi, senza sacrificare troppo l'ottimizzazione e la gestione delle risorse. Al momento dell'esportazione finale del progetto, il codice scritto in C# viene compilato nel linguaggio di riferimento per la piattaforma scelta.

## **6. Strategie di sviluppo, problemi e soluzioni**

### **6.1 Perché Unity 3D**

L'esperienza pensata per il parco, legata esclusivamente alla Piazzetta dei Mosaici di Venturino Venturi, richiede che il software sia in grado di riconoscere i mosaici, e, avvenuto il tracciamento, presentare un'animazione in tre dimensioni legata al contenuto artistico e letterario dell'opera. Per sviluppare prima il prototipo era necessario poter disporre di una facile gestione di modelli e animazioni 3D e del pieno supporto delle tecnologie di realtà aumentata per i principali sistemi operativi mobili, Android e iOS. Inoltre, non essendo esclusa la successiva estensione dell'esperienza a dispositivi differenti, come gli occhiali Epson Moverio o altri tipi di visori, era necessario un ambiente di sviluppo in grado di permettere il supporto ad altre configurazioni con relativa semplicità.

La scelta del software per lo sviluppo dell'applicazione al Parco di Pinocchio è quindi ricaduta su Unity 3D. Unity 3D, come abbiamo visto, nato come motore di gioco nel

---

<sup>17</sup> Piattaforma di sviluppo a carattere generale, che mette a disposizione vari funzionalità a numerosi linguaggi di programmazione, come interoperabilità ed esecuzione asincrona e simultanea.

lontano 2005 è ormai diventato uno strumento di sviluppo adatto a tutta una serie di applicazioni molto diverse dai videogiochi. Nell'ultimo periodo, con il rinnovato interesse per la realtà aumentata, numerose librerie software a pagamento legate alla realtà aumentata - Vuforia, Wikitude e EasyAR - hanno iniziato a supportare Unity 3D, scegliendolo come motore grafico multiplatforma d'elezione. Inoltre, gli sviluppatori di Unity, hanno recentemente rilasciato un pacchetto completamente gratuito che unisce al suo interno le funzionalità di ARKit e ARCore, le librerie AR rispettivamente di Apple e Google, insieme a gli elementi di sviluppo per HoloLens di Microsoft e MagicLeap, chiamato ARFoundation. In questo modo è possibile sviluppare quasi contemporaneamente su tutte queste piattaforme, riducendo al minimo le modifiche al codice legate al singolo hardware. ARFoundation non presenta possibilità avanzate come quelle offerte ad esempio da Vuforia, ma consente comunque un riconoscimento immagine efficace e continuo, incorpora funzionalità avanzate come le ancore e il tracciamento del corpo umano sui dispositivi supportati insieme a tutta una serie di altri effetti, come la stima della luce, i riflessi in tempo reale, il tracciamento delle superfici piane e la nuvola di punti. Tutto questo in modo completamente gratuito: anche per questo la combinazione Unity 3D con ARFoundation si è rivelata la migliore, non senza qualche criticità dovuta alla relativa giovinezza del pacchetto.

## **6.2 Creazione del progetto**

### **Riconoscimento dei mosaici**

Le prove in laboratorio avevano dati riscontri molto positivi riguardo la capacità del software nel riconoscimento dei mosaici: dopo aver inquadrato con la fotocamera dello smartphone una riproduzione in scala, stampata su un pannello, dell'opera, il programma riusciva a creare facilmente una corrispondenza con le foto inserite.

Un ambiente interno illuminato artificialmente, privo di tutte le variabili ambientali del mondo esterno, non si è rivelato però un buon banco di prova per l'applicazione che è crollata completamente durante la prova al parco, senza riuscire a individuare i mosaici con la libreria di immagini caricata. Nuvole, eventi atmosferici, il variare della posizione del sole durante la giornata e le stagioni, le ombre, la vegetazione circostante,

la presenza di visitatori: tutte queste cose influenzano e lasciano il proprio marchio sulla superficie del mosaico, rendendo inefficace la scelta di una foto per intero dell'opera.

La creazione di una collezione di foto, scattate in vari momenti della giornata - mattino, pomeriggio, sera - e in condizioni meteorologiche varie - cielo nuvoloso e sereno in particolare - ha portato risultati migliori ma non costanti, a causa anche della conformazione naturale del luogo che abbraccia la Piazzetta dei Mosaici, immersa completamente nel verde. Solo le numerose variazioni portate dalle ombre alla superficie mosaicata, sempre nuove e imprevedibili delle piante mosse dal vento, avrebbe richiesto un enorme lavoro nella creazione della libreria di immagini da inserire nel progetto, nascondendo comunque l'incognita di non aver considerato una particolare variazione del mosaico.

Torniamo alla libreria formata da una singola immagine per oggetto e analizziamo il problema dal punto di vista del software di sviluppo, Unity e ARFoundation, per tentare scoprire il motivo del mancato riconoscimento. Una volta inserita l'immagine all'interno della Reference Image Library, questa viene analizzata e memorizzata sotto forma di dati grafici che verranno poi comparati al video proveniente dalla fotocamera: per avere il riconoscimento di un oggetto devono esserci numerose corrispondenze tra le informazioni provenienti dal dispositivo di registrazione e una delle immagini inserite dallo sviluppatore.

Più grande e dettagliata è la rappresentazione digitale dell'oggetto, più preciso e completo dovrà essere il fotogramma proveniente dalla fotocamera; è possibile ingannare il sistema con una variazione di colore anche consistente, ma è impossibile ottenere il riconoscimento inquadrando una porzione parziale dell'oggetto raffigurato per intero nella libreria.

Prendiamo in esame un caso particolare, il mosaico raffigurante "Geppetto che scolpisce Pinocchio". Con l'inserimento della sola immagine completa all'interno del software, la sola porzione raffigurante Geppetto - o il tronco di legno ormai diventato burattino - non sarà mai sufficiente a riportare un riconoscimento di successo.

Le prove successive effettuate scegliendo più immagini per singola opera hanno riportato esiti completamente diversi. I mosaici venivano riconosciuti dal software con

grande rapidità, efficienza che sarebbe poi potuta aumentare con una maggiore attenzione alla cattura delle immagini da inserire nella libreria di riconoscimento. A questo punto l'unico ostacolo risiedeva nel posizionamento del modello o contenuto digitale da presentare: il software, dopo aver identificato una delle immagini, riporta il centro dell'oggetto nel sistema di coordinate della scena virtuale. Questo si rivela particolarmente utile, perché rende basilare la collocazione del modello davanti al corrispondente elemento presente nella realtà. Servendosi di più immagini per ogni mosaico questo non era chiaramente più possibile: ogni frammento del mosaico riconosciuto - almeno 4 per opera - ha una posizione ovviamente diversa da quella restituita invece identificando l'immagine corrispondente all'opera completa.

Per rendere più immediata la divisione delle molteplici fotografie, che saranno usate per assicurare un funzionamento ottimale dell'applicazione, e per facilitare il posizionamento dei contenuti in realtà aumentata in posizione corretta, è stato creato uno strumento che aggiunge all'interfaccia di Unity tutte le funzioni necessarie per ottemperare alle richieste del progetto. Questo strumento, chiamato *CropTool*, consiste essenzialmente in uno script che, una volta aggiunto ad un oggetto vuoto nella scena, estende le funzionalità dell'editor, presenta anche un'interfaccia capace di facilitare il lavoro collaborativo all'interno dello stesso progetto.

Lo sviluppo di un'applicazione mobile, con il supporto per la realtà aumentata, segue buona parte delle convenzioni comuni a tutte le piattaforme supportate dallo Unity Engine. Si parte scegliendo la presentazione grafica per il progetto, due o tre dimensioni - per il progetto VERO è stata scelta quella 3D - insieme alla Pipeline di rendering più adatta agli scopi e, infine, la piattaforma di sviluppo di riferimento. Terminata la lunga fase iniziale, nella quale vengono salvati tutte le cartelle e i file base per il software, lo sviluppatore si trova davanti l'Editor di Unity aperto su una scena elementare, che rispetta la configurazione visuale scelta in precedenza. Ora è necessario scaricare e installare i pacchetti e le librerie software, necessarie per accedere alle funzionalità in realtà aumentata proprietarie di Android e iOS. ARFoundation confeziona all'interno dei suoi elementi e script tutte le possibilità offerte da ARCore e ARKit, in modo molto efficace e perfettamente integrato nell'interfaccia del programma; chiaramente, è

possibile che alcune funzioni siano disponibili esclusivamente su una o l'altra piattaforma, ma è sufficiente controllare la documentazione online per fugare ogni dubbio. Oltre all'installazione di questi componenti aggiuntivi, sono fondamentali alcune piccole modifiche alle impostazioni di esportazione: su Android è necessario specificare un livello di API minimo per il funzionamento della realtà aumentata - al momento dello sviluppo di VERO 24 - e la scelta delle librerie grafiche automatica, mentre su iOS sono necessarie alcune informazioni riguardo l'impiego della fotocamera da parte dell'applicazione. Le librerie per lo sviluppo in realtà aumentata proposte da Unity coprono tutte le necessità basilari per un'applicazione, ma non mancano di proporre alcuni elementi avanzati.

- *Device Tracking*, supporta il tracciamento di orientamento e posizione del dispositivo all'interno dello spazio reale.
- *Plane Detection*, permette di riconoscere piani orizzontale e verticale nell'ambiente circostante e di aggiornarne la posizione in tempo reale.
- *Point Clouds*, si tratta di un gruppo di punti caratteristici registrati dal dispositivo nell'ambiente circostante, che viene salvato e tracciato costantemente dal sistema.
- *anchors* - o ancore - consentono allo sviluppatore di creare posizioni arbitrarie che, una volta attivate, continuano a essere aggiornate dal dispositivo, presentando così sempre la stessa locazione all'interno della scena.
- *Light Estimation*, script che stima colore, temperatura e luminosità della luce all'interno del paesaggio circostante.
- *Environment Probes*, tecnica particolarmente complessa che rende possibile convertire in una texture il video proveniente dalla fotocamera. Impiegando la texture risultante sugli oggetti in tre dimensioni, si riescono a creare riflessi coerenti e credibili con l'ambiente, rendendo così particolarmente credibile la simulazione di materiali riflettenti, come i metalli.
- *Face Tracking*, abilitando la fotocamera frontale, è possibile tracciare e sfruttare per numerose applicazioni
- *2D Image Tracking*, gli oggetti e il codice legati a questa funzione permettono di sviluppare applicazioni basate sul riconoscimento immagini; è possibile anche

tracciarne in tempo reale la posizione nel sistema di coordinate interno alla scena virtuale. In questo modo si possono mostrare modelli 3D vicini all'obiettivo, animare copertine di libri o fumetti oppure, semplicemente, attivare informazioni una volta inquadrato un oggetto corrispondente a una foto interna al software.

- *Raycast*, sistema avanzato che controlla collisioni con le proprietà dell'ambiente reale circostante, cioè superfici piane o punti di interesse registrati dal software.

Al termine dell'installazione di ARFoundation è possibile creare una scena in realtà aumentata, con tutte le funzioni principali in pochi semplici passi.

La configurazione base prevede l'uso di due componenti, AR Session Origin e AR Session.

*AR Session Origin* aiuta a tenere traccia della posizione del dispositivo e della camera a esso associata, provvedendo a unire in tempo reale due sistemi di coordinate diverse, quello costituito dalla posizione e rotazione reali e quello legato alla scena virtuale propria del motore grafico. In questo modo tutte le informazioni e contenuti digitali che il programma mostrerà durante l'uso saranno visualizzati correttamente in base alla posizione dell'utente. Tutto questo apre la possibilità di tracciare elementi presenti nell'ambiente, come piani orizzontali e verticali, punti di interesse, oggetti, immagini e corpi.

*AR Session* invece controlla tutti gli elementi legati alla scena in realtà aumentata, impedisce la sovrapposizione e l'inizio di istanze multiple del programma, aiuta a sincronizzare la frequenza di aggiornamento dei contenuti digitale con il conteggio dei fotogrammi forniti dalla camera del dispositivo, permette di avere informazioni sullo stato del tracciamento della posizione del dispositivo e segnala eventuali errori di esecuzione delle funzionalità base.

Grazie all'inserimento di questi due elementi è possibile ottenere una banale applicazione per il dispositivo e sistema preferito, in grado di attivare la visione della fotocamera e tracciare posizione, rotazione e altezza dell'utente.

A questo punto si possono iniziare a inserire contenuti statici nell'ambiente, con il rozzo posizionamento arbitrario in base all'origine della sessione, oppure si possono allargare

le funzionalità dell'applicazione con l'uso di componenti avanzati in grado di aprire nuove possibilità di interazione con l'ambiente.

Come abbiamo visto, il funzionamento base del prototipo per il progetto VERO si basa sulla possibilità di riconoscere il mosaico inquadrato dall'utente presentando in seguito un'animazione specifica legata all'opera.

Per abilitare il riconoscimento immagine, legato alle foto del mosaico, è necessario inserire nello stesso elemento contenente il componente AR Session Origin, lo script, fornito da Unity, AR Tracked Image Manager, che richiede l'inserimento di una libreria immagini su cui basare il tracciamento, chiamata Serialized Library. Per facilitare l'associazione del contenuto al mosaico corrispondente è stato essenziale creare una versione personalizzata del codice, chiamata AR Tracked Image Dynamic Prefab, capace di fornire un'interfaccia pulita, rapida e aggiornabile in base al numero di immagini contenute nella libreria.

In entrambi i casi, il funzionamento alla base è il medesimo: una volta riconosciuto un'oggetto corrispondente ad una delle immagini inserite nella collezione, il programma crea un elemento vuoto nella stessa posizione con associato lo script ARTrackedImage che mette a disposizione tutta una serie di informazioni legate all'immagine di partenze e al posizionamento nella scena virtuale. Questo oggetto vuoto può essere utilizzato in vari modi per il posizionamento di un qualsiasi contenuti sopra o nelle vicinanze dell'obiettivo reale: è possibile associare permanente il contenuto digitale all'immagine tracciata in tempo reale così da seguirne sempre la posizione, soluzione particolarmente adatta per soggetti in movimento.

Viste le caratteristiche del luogo e l'impossibilità di assicurare un tracciamento costante dell'opera, per la presentazione dei contenuti dei mosaici è stata presa la decisione di utilizzare una soluzione ibrida, unendo alcune funzionalità offerte dal software per il riconoscimento immagine con un metodo di posizionamento esclusivo per la realtà aumentata, le ancore.

L'uso corretto delle ancore richiede il tracciamento di un oggetto reale, come un piano orizzontale e verticale, capace di stabilire una connessione costante del sistema al mondo fisico, su cui basare l'aggiornamento della posizione dell'ancora che altrimenti si

troverebbe a fluttuare nella scena virtuale, rendendo instabile e soprattutto artificiale qualsiasi contenuto ad essa associato. Sono stati così aggiunti due ulteriori script resi disponibili da ARFoundation chiamati ARPlaneManager e ARAnchorManager: il primo permette la ricerca e il tracciamento di superfici piane orizzontali e verticali, come un tavolo o un muro, mentre il secondo attiva la possibilità di creare un'ancora all'interno della scena virtuale e aggiornarne costantemente la posizione.

Il ciclo base dell'app è quindi il seguente: all'avvio dell'applicazione sarà necessario cercare un piano di appoggio su cui basare l'aggiornamento delle ancore, da quel momento in poi l'utente potrà inquadrare una delle opere e, dopo il riconoscimento di uno dei numerosi segmenti inseriti sotto forma di foto, verrà mostrato il contenuto digitale appropriato posizionato al centro del mosaico. L'animazione rimarrà attiva finché nel campo visivo dell'utente e sarà possibile visualizzarne solo una per volta.

Per la gestione di tutta l'esperienza in realtà aumentata è stato creato uno script, denominato ARExperience, contenente una serie di variabili e condizioni accessibili anche dall'esterno. Prima di tutto viene attivata l'animazione che invita l'utente a ricercare un piano piana da tracciare: al primo ritrovamento, l'animazione è disattivata insieme alla funzionalità di ricerca delle superficie ormai non più necessaria. L'utente può ora iniziare a esplorare la Piazzetta inquadrando i mosaici.

Al riconoscimento di un'oggetto corrispondente ad un'immagine della Reference Library, viene eseguita la funzione, interna al codice AR Tracked Image Dynamic Prefab, *OnTrackedImageChanged*. Le istruzioni analizzano un particolare tipo di dati fornito come input - ARTrackedImagesChangedEventArgs - contenente tutta una serie di informazioni legate alle immagini riconosciute. Su ogni oggetto tracciato viene create un elemento particolare, vuoto se non specificato in altro modo dallo sviluppatore, che ne segue costantemente posizione, rotazione ed eventualmente scala. Visto il luogo d'uso dell'applicazione e le problematiche legate al riconoscimento di cui è già stato detto in precedenza, non è stato possibile sfruttare questa funzionalità totalmente; invece di usufruire dell'elemento ARTrackedImage primario, sarà realizzata un'ancora nella stessa posizione del primo segmento di fotografia riconosciuto e, sfruttando i dati ottenuti durante la divisione con il CropTool, il contenuto digitale verrà posizionato al

centro del mosaico intero. In questo modo è stato ottenuta una grande affidabilità durante il riconoscimento delle opere, grazie alle numerose immagini per ognuna, senza sacrificare un'unione credibile e realistica del contenuto digitale con la realtà.

Per ottenere una registrazione migliore della posizione dell'immagine inquadrata, l'ancora non è creato al primo tracciamento, ma, utilizzando una costante inserita nello script ARExperience, si attenderà un numero di fotogrammi arbitrario - nel nostro caso sessanta - in cui l'immagine l'immagine risulta tracciata prima di attivare i contenuti. Tutto questo per evitare una creazione affrettata dell'ancora che andrebbe a inficiare il risultato finale con una sovrapposizione imprecisa.

```
private void OnTrackedImageChanged(ARTrackedImagesChangedEventArgs args){}18
```

### **Crop Tool, strumento per la divisione dell'immagine in dettagli.**

La creazione di una libreria immagini così vasta e dettagliata ha richiesto lo sviluppo di uno strumento personalizzato, in grado di rendere la divisione delle singole foto più rapida e precisa possibile. L'obiettivo del software, in breve, è quello di offrire un utile appoggio al riconoscimento immagine: una volta associato lo script *CropTool* ad un oggetto vuoto all'interno dell'editor, l'utente ha la possibilità di suddividere una foto in più parti, creando contemporaneamente gli oggetti vuoti, in grado di presentare il contenuto digitale a loro associato al centro dell'oggetto, esattamente come se l'immagine fosse stata riconosciuta per intero.

Prima di utilizzare il software è obbligatorio inserire l'immagine da dividere, il fattore di divisione per l'asse x e l'asse y, larghezza e altezza dell'oggetto in metri; i campi restanti riguardano la creazione dell'oggetto vuoto, il primo serve a invertire l'asse y e z e il secondo viene usato per distanziare leggermente il contenuto dalla posizione dell'immagine tracciata.

Una volta premuto il bottone *Crop Image*, vengono create le cartelle necessarie per una visualizzazione gerarchica e ordinata all'interno del progetto, l'immagine è suddivisa secondo i parametri inseriti e le informazioni legate al posizionamento al centro sono

---

<sup>18</sup> Codice completo della funzione in appendice a pagina 52.

stampate nel titolo di ogni segmento. La versione attuale del software richiede l'inserimento di fattori di divisione capaci di non lasciare resto.

```
void DivideWithRealMesures(){}19
```

La funzione *DivideWithRealMeasures*, non solo segmenta l'immagine secondo i parametri inseriti ma calcola anche la distanza del centro di un tassello da quello dell'immagine completa usando le dimensioni in metri riferite all'oggetto reale; queste informazioni saranno in seguito utilizzate per posizionare correttamente al centro del mosaico il contenuto in tre dimensioni. Viene memorizzato per prima cosa il centro in metri dell'oggetto reale all'interno di un *Vector2*, nelle sue componenti x e y. In seguito sono salvate larghezza e altezza del segmento, in pixel per la divisione della texture e in metri per la distanza dal centro. Un doppio ciclo percorre tutta la matrice di righe e colonne: all'interno viene calcolata la distanza dal centro per ogni segmento, viene salvata la nuova immagine con tutte le informazioni nel nome con la chiamata della funzione *DivideImage* ed è salvato un oggetto vuoto nel formato prefab di Unity, per facilitare il posizionamento delle animazioni in seguito.

La funzione *DivideImage* memorizza il colore dei pixel del segmento desiderato, crea una nuova texture in due dimensioni della dimensione voluta e, una volta applicati i punti memorizzati, salva la nuova immagine nella cartella corretta in formato *png*.

```
/// <summary>
/// Creates a blueprint prefab for parenting AR custom content, enabling correct centre
positioning on a 2D object.
/// </summary>
/// <param name="name">String containing Cropped Texture Name</param>
void CreatePrefab(string name){}20
```

La funziona *CreatePrefab* è impiegata nella realizzazione di un elemento vuoto con la posizione già modificata per posizionare qualcosa al centro dell'oggetto reale, nel nostro caso il mosaico. Viene preso in esame il nome della texture e sono estratte dalla stringa le distanze dal centro. Dopo essere stati convertiti nel formato corretto, i due numeri

---

<sup>19</sup> Codice completo della funzione in appendice a pagina 53.

<sup>20</sup> In appendice a pagina 53.

sono usati per riposizionare il prefab in modo che venga visualizzato all'incirca al centro dell'oggetto. Al termine di tutta l'operazione viene aggiornato l'Editor di Unity così da rendere visibili le nuove risorse create.

A questo punto si possono sfruttare le funzionalità basilari del riconoscimento immagini proposto da ARFoundation: i numerosi segmenti di ogni mosaico sono inseriti all'interno della libreria XRReferenceImageLibrary, che viene poi associata al programma per la presentazione dei contenuti.

### **Gioco Interattivo all'interno della Piazzetta dei Mosaici.**

Il prototipo dell'applicazione include anche una sezione interattiva, sempre in realtà aumentata, in cui il visitatore deve ricostruire le avventure di Pinocchio inquadrando i mosaici in ordine cronologico. Missione apparentemente banale ma che richiede particolare attenzione per la peculiarità della rappresentazione di alcune scene, apparentemente slegate dall'immaginario tradizionale del burattino. Dopo un'attento confronto tra le figure, i passi del romanzo più adatti a descriverle e le - poche - informazioni sull'immaginario di Venturino Venturi, le opere sono state ordinate secondo un ordine cronologico coerente al romanzo.

La logica del gioco consiste in un ciclo continuo fino alla conclusione della cronologia del libro - il primo mosaico rappresenta la creazione di Pinocchio e l'ultimo raffigura il gatto e la volpe al ciglio della strada puniti per le loro cattiverie - che l'utente dovrà ricomporre. Una volta avviato il gioco, il sistema attende che l'utente inizi la ricostruzione inquadrando un mosaico e, una volta riconosciuto, ne confronta la posizione all'interno della successione con quella corrente. Se le due posizioni corrispondono, quella corrente viene aumentata di uno, altrimenti viene registrato un errore: in entrambi i casi il programma attende un nuovo riconoscimento.

Il codice per il gioco si basa nuovamente sul tracciamento e riconoscimento immagine che abbiamo già visto in precedenza per la presentazione dei contenuti digitali.

Al momento del tracciamento di uno dei dettagli inseriti all'interno della libreria, viene registrato in una variabile globale - *m\_RecognizedMosaic* - la stringa corrispondente al titolo identificativo scelto.

La logica del gioco invece è stata inserita all'interno della funzione primaria di Unity *Update()*. Questa funzione, che troviamo in ogni script base creato dentro il motore, viene invocata a ogni fotogramma di gioco, quindi, normalmente trenta o sessanta volte al secondo.

Prima di tutto l'ordine corretto dei mosaici è stato registrato in un dizionario associativo - nome del mosaico associato al suo numero nell'ordine - in modo da rendere immediato il recupero del valore di ogni elemento; è stata creata anche una versione inversa di questo dizionario - numero associato al nome - in modo da evitare operazioni per ricondurre il valore al titolo dell'opera. Trattandosi di un'esecuzione costante e in tempo reale, non controllata da input strettamente volontari, sono state create variabili di controllo per evitare il controllo multiplo di uno stesso riconoscimento e, quindi, il conteggio sbagliato degli errori.

Due variabili, *m\_RecognizedMosaic* e *m\_PrevRecognizedMosaic*, registrano rispettivamente il mosaico riconosciuto attualmente e quello precedentemente tracciato. Sono utilizzate insieme al booleano *m\_CurrentMosaicChecked* per assicurare un solo controllo finché l'utente non inquadra un mosaico diverso da quello precedente: una volta controllati i valori, non è possibile effettuare una prova ulteriore finché le due variabili relative ai mosaici non presentano un valore differente. Se l'opera scelta è quella corretta il valore di *m\_TargetMosaic* - inizialmente zero - viene incrementato di uno, in caso contrario viene registrato un errore all'interno di *m\_Fails*. Il gioco quindi procede finché l'utente non completa la sequenza dei mosaici

## **7. Futuri Sviluppi, migliorie**

Il prototipo sviluppato è riuscito ad assicurare un'affidabile e costante riconoscimento dei mosaici all'interno della Piazza, riuscendo a sopperire in parte alle mancanze del pacchetto gratuito per la realtà aumentata offerto da Unity e riducendo al minimo le variazioni, provocate dai mutamenti meteorologico ambientali sulla superficie dei mosaici.

Eventuali sviluppi futuri riguardano, dal punto di vista prettamente tecnologico, il *CropTool* e di conseguenza l'uso di numerose immagini per il riconoscimento di oggetti e opere, senza identificatori arbitrari applicati nelle vicinanze. Non è sempre possibile, come nel caso del Parco di Pinocchio, aggiungere targhe o etichette per un'attivazione più semplice e immediata del contenuto digitale nei pressi di sito reale: proprio per questo nel campo dei beni culturali, in particolare edifici e monumenti all'aperto, è quasi sempre necessaria una soluzione senza marchio, detta *Marker Less*.

Nell'attuale versione, gli strumenti sviluppati sono strettamente legati al riconoscimento e alla successiva visualizzazione del contenuto virtuale su un oggetto piano - il mosaico - esteso in due dimensioni, sull'asse x e l'asse y.

Non dovrebbe essere particolarmente difficile estendere queste funzionalità al riconoscimento di opere e oggetti in tre dimensioni, come sculture e statue: dovrà essere aggiunta la terza dimensione nello strumento per la segmentazione delle immagini e sarà fondamentale un nuovo metodo di calcolo del centro dell'oggetto. Per la collocazione appropriata delle informazioni digitali sarà forse necessario individuare un punto centrale arbitrario, esterno all'oggetto, ad esempio di fronte a una scultura, per evitare la compenetrazione del soggetto fisico con quello digitale.

Dovrebbe essere anche possibile supportare il riconoscimento di palazzi e edifici storici. Un palazzo potrebbe essere considerato scomposto in tutte le sue facciate: considerando ogni facciata come un elemento esclusivamente in due dimensioni, sarebbe sufficiente applicare la stessa soluzione adottata per i mosaici. Con il desiderio di offrire una miglior connubio tra il reale e il digitale, potrebbe essere utile sfruttare le informazioni offerte da giroscopio, accelerometro e fotocamera sulla posizione dell'utente per allineare i contenuti con l'edificio.

Entrambi questi suggerimenti propongono alternative a tecnologie di riconoscimento oggetti, legate a scansioni di modelli o aree esistenti, complesse e soprattutto molto costose come quelle offerte da librerie software come *Vuforia* o *Wikitude*.

Non è chiaramente possibile offrire strumenti in grado di coprire ogni possibile applicazione in realtà aumentata, le necessità dipendono strettamente dal tipo di esperienza che lo sviluppatore vuole creare; animazioni e modelli digitali

completamente legati alla posizione dell'oggetto reale richiederanno sicuramente correzioni ad-hoc in loco.

Per quanto concerne i contenuti e l'applicazione al Parco in generale, l'esperienza può essere ampliata e ingrandita in vari modi. Il percorso monumentale, che accoglie 21 sculture dei personaggi più rappresentativi del romanzo, create da grandi artisti del Novecento, si presta in modo particolare ad un'installazione virtuale in realtà aumentata, capace di arricchire la visita con elementi attivati in prossimità delle opere. Si può immaginare un cammino virtuale, complementare alle statue che escono dal verde del parco, modulato sulla posizione di ogni singolo visitatore, che aggiunga all'esperienza fisica delle statue esperienze visive e sonore aggiuntive.

## **8. Conclusioni**

Le tecnologie di realtà aumentata sono in grado di colmare il divario tra realtà e contenuti virtuali, tra l'analogico e il digitale. Nell'era dell'informazione, dove ormai tutti portano con sé numerosi dispositivi elettronici connessi alla rete, la realtà aumentata si presenta come un'alternativa alla classica frazione artificiale dei contenuti digitali. Il primo ostacolo a questa transizione, da tecnologia di nicchia a tecnologia di consumo, non solo legata, come è oggi, ad applicazioni di intrattenimento o pubblicitarie, è il legame con gli smartphone, che accolgono quasi tutte le forme odierne di software in AR. Trattandosi di dispositivi polivalenti, non legati a una funzione specifica, ma a molteplici, non possono chiaramente essere ottimizzati per la realtà aumentata a tutto tondo: i sensori devono consumare meno energia possibile e gli sviluppatori devono contenere al massimo l'uso della batteria. Negli ultimi tempi abbiamo già assistito a progressi in questa direzione: il GPS è diventato più efficiente e affidabile, è aumentato il numero di fotocamere presenti ed è ora possibile usare contemporaneamente quella posteriore e quella frontale. Un'altra soluzione potrebbe essere quella di offrire un occhiale o visore, dotato di sensori e videocamera, in grado di connettersi con il telefono; in questo modo, gli utenti potranno interagire con il software in modo più naturale, relegando il telefono solo a una funzione di calcolo e gestione

dell'esperienza. In ogni caso, i dispositivi di riferimento per la realtà aumentata rimarranno ancora a lungo telefoni e tablet, soprattutto per applicazioni dedicate al grande pubblico, più restio, rispetto a nicchie di appassionati e professionisti, ad adottare hardware dedicato. Anche per la realtà aumentata sarà probabilmente il settore dell'intrattenimento - includiamo in questo settore anche le applicazioni a sfondo culturale - a dettare e a definire i passi successivi. Oggi cinema e videogiochi ci hanno abituato a livelli altissimi di qualità e fedeltà visiva, non è quindi possibile presentare un'esperienza in realtà aumentata poco rifinita, con un tracciamento ambientale approssimativo e un funzionamento generale poco fluido. Molte delle riflessioni precedenti riguardo l'avanzamento tecnologico dei dispositivi, valgono anche come soluzione a questo punto.

Crediamo, però, sia anche necessario che le esperienze in realtà aumentata smettano di scimmiettare modalità tipiche di altre forme di intrattenimento, per crearne una propria, modulata sulle peculiarità di questa tecnologia tecnologia. Queste peculiarità sono riassunte nelle tre caratteristiche fondamentali stilate da Azuma: l'unione di contenuti digitali, interattivi in tempo reale e registrati nello spazio, capaci di reagire al punto di vista e alla posizione dell'utente, con contenuti reali la troviamo, al momento, solo nei dispositivi e nelle applicazioni di realtà aumentata. Basandoci su questo, possiamo auspicare applicazioni basate sempre di più sul punto di vista dell'utente, che rimane sempre protagonista e centrale a tutta l'esperienza, perché ha in mano l'occhio sulla scena reale e virtuale. Inoltre, immaginiamo interazioni dinamiche centrate sulla posizione e la direzione dello sguardo di chi sta usufruendo dell'applicazione, con contenuti che si attivano senza interazioni artificiali con il dispositivo, riducendo al minimo l'uso di sistemi di controllo esterni.

Nuove esperienze, caratterizzate da una narrativa centrata sull'utente, che decide come e cosa inquadrare all'interno dell'ambiente, che deve invogliarlo a esplorare e scandagliare la scena in cerca di interazioni.

## 9. Bibliografia

- Curreri, Luciano, 2017, *Play it again, Pinocchio: Saggi per una storia delle pinocchiate*, Bergamo, Moretti & Vitali.
- Marcheschi, Daniela, 2016, *Il naso corto: una rilettura della avventure di Pinocchio*, Bologna, Centro Editoriale Dehoniano.
- Baldacci, Valentino, Andrea Raucha, 2006, *Pinocchio e la sua immagine*, Firenze, Giunti.
- Ruffilli, Massimo, 2011, *Pinocchio a Firenze*, Reggello, Firenze Libri.
- Collodi, Carlo, 2019, *Le avventure di Pinocchio*, Padova, Edizioni Crescere.
- Dedola, Rossana, 2019, *Pinocchio e Collodi: sul palcoscenico del mondo*, Bertoni Editore, Perugia.
- Catelli, Nicola, Simona Scattina, 2017, *Il corpo plurale di Pinocchio: metamorfosi di un burattino*, Arabeschi, Catania.
- Schnalstieg, Dieter, Tobias Höllerer, 2016, *Augmented reality: principles and practice*, Crawfordsville USA, Pearson.

## 10. Sitografia

Silk Road, *Museums*

<http://www.musesilkroad.com/en/index.php?c=news&a=view&id=97>

Global Times, *China Longmen Caves*

<https://www.globaltimes.cn/content/1203699.shtml>

Trinity College Dublin, *Long Room*

<https://www.tcd.ie/library/old-library/long-room/>

Trinity College Dublin, *V-SENSE*

<https://v-sense.scss.tcd.ie/creative-experiments/jonathan-swift-in-vr-ar-long-room-project/>

Buddhistdoor, *Augmented Reality in Zen Temple*

<https://www.buddhistdoor.net/news/augmented-reality-brings-kyotos-quos-oldest-zen-temple-into-the-future>

Microsoft, *Kyoto Museum with Hololens*

<https://news.microsoft.com/apac/features/mixed-reality-museum-kyoto-unique-insight-centuries-old-japanese-artwork/>

Hakuhodo Global, *New Unimagined Ways for AR*

<https://www.hakuhodo-global.com/news/an-unimagined-way-from-a-new-use-mr-museum-in-kyoto-1.html>

Cladglobal, *Glass Rotunda*

<https://www.cladglobal.com/CLADnews/architecture-design/National-Museum-of-Singapore-Glass-Rotunda-Story-of-the-Forest-teamLab-gallery-augmented-reality-technology/329219>

Nhb, *National Museum of Singapore*

<https://www.nhb.gov.sg/nationalmuseum/our-exhibitions/exhibition-list/story-of-the-forest>

Patrimonio Culturale, *Museo del territorio di Riccione*

<https://patrimonioculturale.regione.emilia-romagna.it/notizie/2021/il-museo-del-territorio-di-riccione-diventa-piu-interattivo-con-due-app>

Silab Isti, *Museo di Riccione*

<http://silab.isti.cnr.it/index.php/news/229-news-museo-riccione>

Unity Engine, *Unity Documentation*

<https://docs.unity3d.com/Manual/index.html>

Unity Engine, *AR Foundation*

<https://unity.com/unity/features/arfoundation>

Pinocchio, *Parco di Collodi*

<https://www.pinocchio.it/>

# 11. Appendici

## 11.1 Ordine dei mosaici secondo la cronologia del romanzo.

Mosaico 1 - La creazione di Pinocchio.

Capitolo III

*“(...)Appena entrato in casa, Geppetto prese subito gli arnesi e si pose a intagliare e a fabbricare il suo burattino(...)”*

Mosaico 2 - Il Carabiniere.

Capitolo III

*“(...)Alla fine, e per buona fortuna, capitò un carabiniere il quale, sentendo tutto quello schiamazzo, e credendo si trattasse di un puledro che avesse levata la mano al padrone, si piantò coraggiosamente a gambe larghe in mezzo alla strada, coll’animo risoluto di fermarlo e d’impedire il caso di maggiori disgrazie.(...)”*

Mosaico 3 - Il Grillo Parlante.

Capitolo IV

*“(...)Ma quella contentezza durò poco, perché sentì nella stanza qualcuno che fece: « Cri-cri-cri »(...)”*

Mosaico 4 - Pinocchio, Arlecchino e Pulcinella.

*“(...)A questo affettuoso invito, Pinocchio spicca un salto, e di fondo alla platea va nei posti distinti; poi con un altro salto, dai posti distinti monta sulla testa del direttore d’orchestra, e di lì schizza sul palcoscenico(...) “*

Mosaico 5 - Mangiafoco.

*“(...)Dopo avere starnutito, il burattinaio, seguitando a fare il burbero, gridò a Pinocchio: « Finiscila di piangere! I tuoi lamenti mi hanno messo un’uggiolina qui in fondo allo stomaco... sento uno spasimo, che quasi quasi... Etcì! Etcì!» e fece altri due starnuti. « Felicità! » disse Pinocchio.(...)”*

Mosaico 6 - L’osteria del Gambero Rosso.

*“(...)Il povero Gatto, sentendosi gravemente indisposto di stomaco, non poté mangiare altro che trentacinque triglie con salsa di pomodoro e quattro porzioni di*

*trippa alla parmigiana: e perché la trippa non gli pareva condita abbastanza, si rifece tre volte a chiedere il burro e il formaggio grattato! La Volpe avrebbe spelluzzicato volentieri qualche cosa anche lei: ma siccome il medico le aveva ordinato una grandissima dieta, così dovè contentarsi di una semplice lepre dolce e forte con un leggerissimo contorno di pollastre ingrassate e di galletti di primo canto. Dopo la lepre, si fece portare per tornagusto un cibreino di pernici, di starne, di conigli, di ranocchi, di lucertole e d'uva paradisa; e poi non volle altro. Aveva tanta nausea per il cibo, diceva lei, che non poteva accostarsi nulla alla bocca. (...)*”

Mosaico 7 - Il Bosco degli Assassini.

*“(...)Ma Pinocchio non poté finire il suo ragionamento, perché in quel punto gli parve di sentire dietro di sé un leggerissimo fruscio di foglie.(...)”*

Mosaico 8 - La Fata.

*“(...)Allora si affacciò alla finestra una bella Bambina, coi capelli turchini e il viso bianco come un'immagine di cera, gli occhi chiusi e le mani incrociate sul petto, la quale, senza muover punto le labbra, disse con una vocina che pareva venisse dall'altro mondo.(...)”*

Mosaico 9 - I tre Medici.

*“(...)E i medici arrivarono subito uno dopo l'altro: arrivò, cioè, un Corvo, una Civetta e un Grillo-parlante.(...)”*

Mosaico 10 - L'albero degli zecchini.

*“(...)«E se invece di mille monete, ne trovassi su i rami dell'albero duemila?... E se invece di duemila, ne trovassi cinquemila? e se invece di cinquemila, ne trovassi centomila? Oh che bel signore, allora, che diventerei!... Vorrei avere un bel palazzo, mille cavallini di legno e mille scuderie, per potermi baloccare, una cantina di rosoli e di alchermes, e una libreria tutta piena di canditi, di torte, di panattoni, di mandorlati e di cialdoni colla panna.»(...)”*

Mosaico 11 - Il Serpente.

*“(...)Aveva veduto un grosso Serpente, disteso attraverso alla strada, che aveva la pelle verde, gli occhi di fuoco e la coda appuntata, che gli fumava come una cappa di camino.(...)”*

Mosaico 12 - I carabinieri e il cane Alidoro.

*“(...)I carabinieri, giudicando che fosse difficile raggiungerlo, gli aizzarono dietro un grosso cane mastino, che aveva guadagnato il primo premio a tutte le corse dei cani. Pinocchio correva, e il cane correva più di lui: per cui tutta la gente si affacciava alle finestre e si affollava in mezzo alla strada, ansiosa di veder la fine di un palio così inferocito. Ma non poté levarsi questa voglia, perché fra il can mastino e Pinocchio sollevarono lungo la strada un tal polverone, che dopo pochi minuti non era possibile di veder più nulla.(...)”*

Mosaico 13 - Il pescatore verde.

*“(...)E nel tempo stesso vide uscire dalla grotta un pescatore così brutto, ma tanto brutto, che pareva un mostro marino. Invece di capelli aveva sulla testa un cespuglio foltissimo di erba verde; verde era la pelle del suo corpo, verdi gli occhi, verde la barba lunghissima, che gli scendeva fin quaggiù. Pareva un grosso ramarro ritto sui piedi di dietro.(...)”*

Mosaico 14 - La lumaca.

*“(...)Aspetta, aspetta, finalmente dopo mezz'ora si aprì una finestra dell'ultimo piano (la casa era di quattro piani) e Pinocchio vide affacciarsi una grossa lumaca, che aveva un lumicino acceso sul capo(...)”*

Mosaico 15 - L'omino di burro.

*“(...)Figuratevi un omino più largo che lungo, tenero e untuoso come una palla di burro, con un visino di melarosa, una bocchina che rideva sempre e una voce sottile e carezzevole, come quella d'un gatto, che si raccomanda al buon cuore della padrona di casa.(...)”*

Mosaico 16 - Il Paese dei Balocchi.

*“(...)Intanto era già da cinque mesi che durava questa bella cuccagna di baloccarsi e di divertirsi le giornate intere, senza mai vedere in faccia né un libro, né una scuola; quando una mattina Pinocchio, svegliandosi, ebbe, come si suol dire, una gran brutta sorpresa, che lo messe proprio di malumore.(...)”*

Mosaico 17 - Pinocchio trasformato in asino.

*“(...)Fatto sta che il compratore, appena pagati i venti soldi, condusse il ciuchino sulla riva del mare; e messogli un sasso al collo e legatolo per una zampa con una fune che teneva in mano, gli diè improvvisamente uno spintone e lo gettò nell’acqua. (...)”*

Mosaico 18 - La capretta.

*“(...)La cosa più singolare era questa: che la lana della caprettina, invece di esser bianca, o nera, o pallata di più colori, come quella delle altre capre, era invece tutta turchina, ma d’un turchino così sfolgorante, che rammentava moltissimo i capelli della bella Bambina.(...)”*

Mosaico 19 - Geppetto nella pancia del Pescecaone.

*“(...)E più andava avanti, e più il chiarore si faceva rilucente e distinto: finché, cammina cammina, alla fine arrivò: e quando fu arrivato... che cosa trovò? Ve lo do a indovinare in mille: trovò una piccola tavola apparecchiata, con sopra una candela accesa infilata in una bottiglia di cristallo verde, e seduto a tavola un vecchietto tutto bianco, come se fosse di neve o di panna montata, il quale se ne stava lì biascicando alcuni pesciolini vivi, ma tanto vivi, che alle volte mentre li mangiava, gli scappavano perfino di bocca.(...)”*

Mosaico 20 - Il Gatto e la Volpe Punite.

*“(...)Non avevano ancora fatti cento passi, che videro seduti sul ciglione della strada due brutti ceffi, i quali stavano lì in atto di chiedere l’elemosina.(...)”*

## 11.2 Codici Completi

### Posizionamento in realtà Aumentata

```
private void OnTrackedImageChanged(ARTrackedImagesChangedEventArgs args)
{
    if (m_ARExperience.Initialized && !m_ARExperience.PauseTracking)
    {
        #region Tracking
        // loop events added when a new image is recognized for the first time
        foreach (var trackedImage in args.added){...}
        // loop events added when a recognized image is updated
        foreach (var trackedImage in args.updated)
        {
            if (trackedImage.trackingState == TrackingState.Tracking)
            {
                if (m_ARExperience.Counter >= m_ARExperience.TargetCounter &&
                    !m_ARExperience.ObjectInstantiated)
                {
                    // create anchor on top of the current tracked image position
                    anchor = m_ARAnchorManager.AddAnchor(new
                    Pose(trackedImage.transform.position, trackedImage.transform.rotation));
                    // position prefab with the anchor transform as a parent
                    GameObject prefab = ReturnPrefab(trackedImage);
                    if (prefab != null)
                    {
                        Instantiate(prefab, anchor.transform);
                        m_ARExperience.ObjectInstantiated = true;
                    }
                }
            }
            else
            {
                // update frame counter
                m_ARExperience.Counter = m_ARExperience.Counter + 1;
            }
        }
        else if (trackedImage.trackingState == TrackingState.None)
        {
            // enable prefab variables
            if (m_ARExperience.Counter > 0 && !m_ARExperience.ObjectInstantiated)
            {
                m_ARExperience.ResetObjectVariables();
            }
        }
    }
}
```

## Segmentazione immagine

```
void DivideWithRealMesures()
{
    // calculate object centre position using provided real measures (meters)
    Vector2 imagePivot = new Vector2(m_RealObjectWidth / 2, m_RealObjectHeight / 2);
    // calculate each segment pixel dimensions for the cropped texture output
    int cropWidth = m_Texture2D.width / m_DivisionFactorX;
    int cropHeight = m_Texture2D.height / m_DivisionFactorY;
    // calculate each segment meter dimensions for AR positioning data
    float realCropWidth = m_RealObjectWidth / m_DivisionFactorX;
    float realCropHeight = m_RealObjectHeight / m_DivisionFactorY;
    // image total number
    int imageNumber = 0;

    for (int r = 0; r < m_DivisionFactorY; r++)
    {
        for (int c = 0; c < m_DivisionFactorX; c++)
        {
            // distance from whole image centre (metri)
            float realDistanceX = imagePivot.x - ((realCropWidth / 2f)
                + (realCropWidth * c));
            float realDistanceY = imagePivot.y - ((realCropHeight / 2f)
                + (realCropHeight * r));
            string name = m_ImageName + "Crop" + imageNumber + "X" + realDistanceX
                + "Y" + realDistanceY;
            // create new texture with the cropped portion using pixels
            DivideImage(cropWidth * c, cropHeight * r, cropWidth, cropHeight, name);
            // create corresponding prefab for cropped detail
            CreatePrefab(name);
            // update image counter
            imageNumber++;
        }
    }
}
```

## Creazione GameObject per ogni tassello

```
/// <summary>
/// Creates a blueprint prefab for parenting AR custom content, enabling correct centre
/// positioning on a 2D object.
/// </summary>
/// <param name="name">String containing Cropped Texture Name</param>
void CreatePrefab(string name)
{
    #if UNITY_EDITOR
        GameObject newGameObject = new GameObject();
        // extract position data from string
        Vector2 data = ExtractDataFromName(name);
        if (m_InvertYZ)
        {
            newGameObject.transform.position = new Vector3(data.x, m_DistanceFromImage,
                data.y);
        }
        else
        {
            newGameObject.transform.position = new Vector3(data.x, data.y,
                m_DistanceFromImage);
        }
        // save the game object as a prefab inside object folder
        PrefabUtility.SaveAsPrefabAsset(newGameObject, m_ObjectFolderPath + "/" + name +
            ".prefab");
        // remove game object from editor scene
        DestroyImmediate(newGameObject);
    #endif
}
```

## Modalità Interattiva

```
public class PinocchioGameScript : MonoBehaviour, ISerializationCallbackReceiver
{
    ARTrackedImageManager m_ARTrackedImageManager;
    [SerializeField]
    XRReferenceImageLibrary m_ImageLibrary;
    [Serializable]
    struct NamedAnimation
    {
        public string imageGuid;
        public GameObject animationPrefab;
        public NamedAnimation(Guid guid, GameObject animation)
        {
            imageGuid = guid.ToString();
            animationPrefab = animation;
        }
    }

    [SerializeField]
    [HideInInspector]
    List<NamedAnimation> m_AnimationPrefabList = new List<NamedAnimation>();
    // if negative number game hasn't started yet
    int m_TargetMosaic = 0;
    bool m_GameStarted = false;
    int m_RecognizedMosaic = -1;
    int m_PrevRecognizedMosaic = -1;
    bool m_CurrentMosaicChecked = false;
    int m_Fails = 0;
    float m_TimePassed = 0;
    ARTrackedImage m_CurrentTrackedImage;
    ARTrackedImage m_PreviousTrackedImage;
    Dictionary<Guid, GameObject> m_AnimationPrefabDictionary = new Dictionary<Guid,
        GameObject>();

    Dictionary<string, int> m_GameOrder = new Dictionary<string, int>
    {
        { "Geppetto", 0 },
        { "Carabiniere", 1},
        { "Grillo", 2},
        { "Burattini", 3},
        { "Mangiafoco", 4},
        { "Gambero", 5},
        { "Assassini", 6},
        { "Fata", 7},
        { "Medici", 8},
        { "Zecchini", 9},
        { "Nagini", 10},
        { "Alidoro", 11},
        { "Dagon", 12},
        { "Lumaca", 13},
        { "Burro", 14},
        { "Balocchi", 15},
        { "Lucio", 16},
        { "Capretta", 17},
        { "Pesceccane", 18},
        { "Gatto", 19}
    };

    Dictionary<int, string> m_InverseGameOrder = new Dictionary<int, string>
    {
        {0, "Geppetto"},
        {1, "Carabiniere"},
        {2, "Grillo"},
        {3, "Burattini"},
        {4, "Mangiafoco"},
    }
}
```

```

        {5, "Gambero"},
        {6, "Assassini"},
        {7, "Fata"},
        {8, "Medici"},
        {9, "Zecchini"},
        {10, "Nagini"},
        {11, "Alidoro"},
        {12, "Dagon"},
        {13, "Lumaca"},
        {14, "Burro"},
        {15, "Balocchi"},
        {16, "Lucio"},
        {17, "Capretta"},
        {18, "Pescecane"},
        {19, "Gatto"}
};

public XRReferenceImageLibrary imageLibrary
{
    get => m_ImageLibrary;
    set => m_ImageLibrary = value;
}

[SerializeField]
Text m_DebugText;
[SerializeField]
Text m_MistakesText;
[SerializeField]
Text m_PrevMosaicText;

public void OnBeforeSerialize()
{
    m_AnimationPrefabList.Clear();
    foreach (var kvp in m_AnimationPrefabDictionary)
    {
        m_AnimationPrefabList.Add(new NamedAnimation(kvp.Key, kvp.Value));
    }
}

public void OnAfterDeserialize()
{
    m_AnimationPrefabDictionary = new Dictionary<Guid, GameObject>();
    foreach (var entry in m_AnimationPrefabList)
    {
        m_AnimationPrefabDictionary.Add(Guid.Parse(entry.imageGuid),
            entry.animationPrefab);
    }
}

private void Awake()
{
    m_ARTrackedImageManager = GetComponent<ARTrackedImageManager> ();
    ResetGame();
}

// Update is called once per frame
private void Update()
{
    m_DebugText.text = m_TargetMosaic.ToString();
    if (m_RecognizedMosaic < 0)
    {
        m_GameStarted = false;
    }
    else
    {
        m_GameStarted = true;
    }
}

```

```

if (m_GameStarted)
{
    if (!m_CurrentMosaicChecked)
    {
        // game logic
        // if recognized mosaic matches the target one, update target and move
        on
        if (m_RecognizedMosaic == m_TargetMosaic)
        {
            m_TargetMosaic++;

            m_PrevMosaicText.text = m_InverseGameOrder[m_RecognizedMosaic];
        }
        // if player chose the wrong mosaic, increase fail number and don't
        update target
        else
        {
            m_Fails++;
            // update mistakes ui
            m_MistakesText.text = "Mistakes: " + m_Fails.ToString();
        }
        // flag current recognized mosaic as checked
        m_CurrentMosaicChecked = true;
        m_PrevRecognizedMosaic = m_RecognizedMosaic;
    }
    else
    {
        // check if we can restart game logic
        // restart when current recognized mosaic is not the previous checked
        one
        if (m_RecognizedMosaic != m_PrevRecognizedMosaic)
        {
            m_CurrentMosaicChecked = false;
        }
    }
}

private void OnEnable()
{
    m_ARTrackedImageManager.trackedImagesChanged += OnTrackedImageChanged;
}

private void OnDisable()
{
    m_ARTrackedImageManager.trackedImagesChanged -= OnTrackedImageChanged;
}

private void OnTrackedImageChanged(ARTrackedImagesChangedEventArgs args)
{
    foreach (var trackedImage in args.added)
    {
    }

    #if PLATFORM_ANDROID
    foreach (var trackedImage in args.updated)
    {
        if (trackedImage.trackingState == TrackingState.Tracking)
        {
            m_CurrentTrackedImage = trackedImage;
            m_RecognizedMosaic =
                m_GameOrder[ParseTitle(trackedImage.referenceImage.name)];
        }
    }
    #endif
}

```

```

#if UNITY_IOS
foreach (var trackedImage in args.updated)
{
    if (trackedImage.trackingState == TrackingState.Tracking ||
        trackedImage.trackingState == TrackingState.Limited)
    {
        m_CurrentTrackedImage = trackedImage;
        m_RecognizedMosaic =
            m_GameOrder[ParseTitle(trackedImage.referenceImage.name)];
    }
}
#endif
}

void ResetGame()
{
    m_RecognizedMosaic = -1;
    m_PrevRecognizedMosaic = -1;
    m_GameStarted = false;
    m_TargetMosaic = 0;
    m_TimePassed = 0f;
    m_Fails = 0;
}

public GameObject GetPrefabForReferenceImage(XRReferenceImage referenceImage)
=> m_AnimationPrefabDictionary.TryGetValue(referenceImage.guid, out var
prefab) ? prefab : null;

string ParseTitle(string name)
{
    int cropIndex = name.IndexOf("Crop");
    string mosaic = name.Substring(0, cropIndex);
    return mosaic;
}
}

```

## Codice ARExperience

```

/* Script used to initialize and manage the Augmented Reality Experience.
 * Before the user can actually start framing real objects with the phone-camera, the
program needs to do some preliminary work.
 */
public class ARExperienceManager : MonoBehaviour
{
    [SerializeField]
    private Image TrackPlaneUI;
    public Text debugText;
    [SerializeField]
    ARPlaneManager m_ARPlaneManager;
    // variable used to start app operations
    [SerializeField]
    bool m_Initialized = false;
    public bool Initialized
    {
        get { return m_Initialized; }
        set { m_Initialized = value; }
    }

    // variable used to signal the presence of an AR Animation inside the scene
    [SerializeField]
    bool m_ObjectInstantiated = false;
}

```

```

public bool ObjectInstantiated
{
    get { return m_ObjectInstantiated; }
    set { m_ObjectInstantiated = value; }
}

[SerializeField]
int m_Counter = 0;
public int Counter
{
    get { return m_Counter; }
    set { m_Counter = value; }
}

[SerializeField]
int m_TargetCounter = 0;

public int TargetCounter
{
    get { return m_TargetCounter; }
}

bool m_PauseTracking = false;

public bool PauseTracking
{
    get { return m_PauseTracking; }
    set { m_PauseTracking = value; }
}

private void Awake()
{
    if (m_ARPlaneManager == null)
    {
        m_ARPlaneManager = GetComponent<ARPlaneManager> ();
    }
    // add function to run when a new plane is recognized and stored in the queue
    m_ARPlaneManager.planesChanged += PlanesChanged;
}

private void PlanesChanged(ARPlanesChangedEventArgs args)
{
    if (!Initialized)
    {
        Activate();
    }
}

private void Activate()
{
    // remove UI when the plane is found
    TrackPlaneUI.enabled = false;
    // disable ARPlaneManager to free resources
    m_ARPlaneManager.enabled = false;
    // set app initialization variable to true
    Initialized = true;

    debugText.text = "INITIALIZED";
}

public void ResetObjectVariables()
{
    ObjectInstantiated = false;
    Counter = 0;
}
}

```

