POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Leap Pharaohs Experience: sistemi interattivi touchless per la valorizzazione del patrimonio culturale

Relatori
prof. Riccardo Antonio Silvio Antonino

Candidato
Lara Fantone

Anno Accademico 2021-2022

Abstract

Il presente lavoro si propone di descrivere e contestualizzare il progetto di realizzazione di un'installazione touchless per la fruizione interattiva dei contenuti della Lista dei Re del Museo Egizio di Torino. Essa, realizzata in collaborazione con Robin Studio, è stata inaugurata presso le sale del Museo Egizio il 27 settembre 2022 in occasione del bicentenario della decifrazione dei geroglifici ad opera dell'archeologo Jean-Francois Champollion. Sarà in seguito spostata e resa permanente all'interno della Sala della Scrittura, la cui apertura è prevista nel mese di dicembre del 2022.
Lo studio colloca le scelte compiute nel quadro del processo di digitalizzazione dei musei e della situazione post pandemica, ponendo in evidenza lo stretto legame tra musei, società e tecnologia. L'obiettivo è stato quello di valorizzare il patrimonio culturale attraverso le potenzialità delle nuove tecnologie e coinvolgere visitatori di ogni età, rendendoli attivamente partecipi nell'esplorazione del prezioso papiro, esposto al pubblico dallo scorso settembre nella versione restaurata.
Si illustrerà come la necessità di offrire un'interazione efficace e sicura sotto il profilo igienico abbia portato, in seguito all'analisi delle possibilità tecnologiche dei dispositivi di input in modalità touchless, all'utilizzo del Leap Motion Controller all'interno di un'applicazione sviluppata per mezzo del software Unity. Ci si è proposti di garantire un'esperienza più fluida e intuitiva possibile per il visitatore, a partire dall'invito all'azione tramite esplicite indicazioni di comportamento. Per lo stesso motivo come modalità di interazione si è scelta la simulazione del movimento del mouse, strumento familiare a tutti i potenziali fruitori.
L'importanza del papiro è dovuta alla presenza sul suo retro dell'unico elenco completo dei Sovrani d'Egitto, a partire dal periodo mitologico primordiale fino alla fine del Secondo Periodo Intermedio (1650 a.C.). Seppur il papiro si presenti in condizioni frammentarie, grazie ai lavori di studio e restauro effettuati nel corso degli anni è stato rivelato un grande numero di nomi di Re. Il contenuto dell'applicazione si sviluppa proprio a partire da questi nomi che, localizzati nei vari frammenti, permettono di accedere ad un ulteriore livello di dettaglio su di essi. È inoltre resa possibile la visualizzazione e consultazione del lato frontale del papiro, su cui è presente un registro tributario. L'utente ha dunque la possibilità di esplorare a proprio piacimento il documento e le acquisizioni degli studi su di esso, accedendo ad informazioni che altrimenti sarebbe difficile mostrare.
Infine, si ripercorrerà la fase di sviluppo dell'applicazione, soffermandosi sulle decisioni riguardo l'organizzazione dei contenuti e sulla strutturazione modulare dell'applicazione stessa, indispensabile per permettere l'integrazione di materiali di cui eventualmente emerga la necessità nel tempo.

Ringraziamenti

Prima di tutto ci tengo a ringraziare il prof.Antonino, che mi ha dato l'opportunità di partecipare ad un progetto così stimolante e nelle corde di quello che più mi piace fare; il Museo Egizio e il team di Robin Studio, che mi ha accolta e grazie al quale tutto ciò è stato possibile. Un grande grazie anche a Chiara, che ha passato con me giorni e notti a programmare e sbattere la testa su dannati bug.
Grazie ai miei genitori, che mi hanno sempre sostenuta in quello che ho voluto fare, senza mai smettere di credere in me ed incoraggiarmi.
A Ele, l'altra parte di me.
A Luca, Sara, Letizia, Matteo, Marco, Lorenzo, Filippo, Eugenio e Vittorio. Ho avuto la fortuna di incontrarvi al secondo anno anno e di avervi al mio fianco in tutti i successivi, nei momenti più belli, ma anche nei più difficili.
A Marti, che mi ha sempre spronata senza smettere di credere in me neanche un istante.
A Din e a Maddi, che ci siete sempre state nonostante le distanze e che mi sopportate ancora dopo tutto questo tempo.
A Giulia, del bene che mi vuoi e delle fughe che abbiamo fatto insieme, ad alta quota e non.
Ad Andre, che sei presto passato dall'essere coinquilino all'essere amico.

Impossibile nominare tutti quelli a cui devo l'essere arrivata fin qui oggi. Vi voglio bene, dovreste saperlo già.

Indice

Introduzione

Un vasto cambiamento sta attraversando il settore museale. Nel volgere di una ventina d'anni si é passati da una visione incentrata prevalentemente su conservazione ed esposizione protetta e in loco ad una visione più dinamica, che non rinuncia ai principi fondamentali su cui i musei si fondano, ma li sviluppa e potenzia grazie al contributo di nuove e sofisticate tecnologie, offrendo ai visitatori esperienze interattive e coinvolgenti.
Il presente lavoro si propone di tracciare l'ideazione e la realizzazione di un'installazione con sensore Leap Motion per l'esplorazione del Papiro dei Re, uno dei più importanti reperti conservati, studiati e restaurati dal Museo Egizio di Torino. Il progetto é stato guidato dall'intento di rendere accessibile a tutti un testo delicato e prezioso, che non é possibile manipolare e che contiene innumerevoli informazioni difficili da decodificare senza una mediazione. Si è dunque voluto progettare uno strumento agile, alla portata di ogni fascia di visitatore, che al tempo stesso desse conto della ricchezza dei contenuti e degli studi su di esso effettuati.
La trattazione si articola in quattro sezioni.
Nel primo capitolo si studia il quadro di contesto: la rivoluzione digitale all'interno dei musei e le nuove forme di fruizione che si stanno affermando, con particolare riguardo alle installazioni. Ne emerge un panorama ampio e molto variegato, caratterizzato da alcune linee costanti, quali la digitalizzazione del patrimonio conservato, il ricorso a tecnologie touch e l'offerta di esperienze immersive.
Il secondo capitolo è dedicato al reperto museale da valorizzare: la lista reale, scritta sul Verso del Papiro dei Re. L'opera risale al nucleo originario della collezione del Museo Egizio di Torino e la sua importanza fu riconosciuta dallo stesso Champollion: contiene infatti l'unico elenco completo, non soggetto ad omissioni ideologiche, dei faraoni dalle origini mitiche al regno di Ramses II. Si tratta di un papiro che ci è pervenuto in condizioni frammentarie ed ha richiesto una lunga attività di studi e restauri, culminati negli ultimi interventi, affidati all'egittologo danese Kim Ryholt, e alla restauratrice tedesca Myriam Krutsch,. Il Museo egizio lo ha posto al centro degli eventi della celebrazione del bicentenario della decifrazione dei geroglifici da parte di Champollion il 27 settembre 2022. In tale occasione sono stati presentati al pubblico gli esiti dell'attività di riclassificazione e restauro, una sala dedicata al papiro e l'installazione descritta nel presente lavoro.
Nel terzo capitolo si spiega la scelta del touchless e se ne prendono in esame alcune possibili soluzioni tecnologiche. La decisione di ricorrere al touchless deriva dalle necessità post-pandemiche e ad una precisa richiesta del committente e si inserisce in una tendenza che si sta affermando proprio in questo momento: la rivalutazione di dispositivi

e tecnologie touchless già esistenti, le cui potenzialità non erano state sviluppate fino in fondo, allo scopo di rendere più sicura la fruizione, prescindendo dal contatto. L'analisi tecnica e funzionale di due famiglie di sistemi di motion tracking, whole body e mid-air, accompagnata dalla descrizione di alcuni esempi di utilizzo in ambito museale, è condotta nell'intento di intraprendere una direzione che renda l'esperienza del fruitore fluida ed immediata, nonostante la novità dell'approccio. A tale scopo si valuta in particolare Leap Motion Controller, sensore a infrarossi in commercio dal 2012 che permette di tracciare in modo preciso il movimento delle mani e la cui integrazione è semplificata da Touch Free, software creato dalla stessa azienda nel 2020 per offrire un pacchetto di sviluppo completo verso il touchless.
Nel quarto ed ultimo capitolo si percorre il processo che ha portato alla creazione dell'applicazione Leap Pharaoh Experience: la fase iniziale di studio delle fonti, di scelta della tecnologia e di organizzazione dei contenuti, seguita dalla descrizione accurata della sua struttura e del funzionamento dei principali meccanismi che la sorreggono. Segue un accenno alla grafica e ai metodi di animazione, incentrati sull'utilizzo della libreria DOTween.
Si riportano infine alcune osservazioni sul comportamento dei visitatori durante l'inaugurazione e commenti che derivano dal monitoraggio condotto dal personale museale nelle settimane successive all'apertura della sala.

Capitolo 1

La digitalizzazione nei musei

1.1 Cambia la definizione di museo

Il settore museale è al centro di un processo di rapida trasformazione, segnato dall'evoluzione digitale e dalle possibilità che essa offre, al punto che nel 2019 l'ICOM (International Council of Museums) propone di rivedere la definizione di museo al fine di comprendere anche le nuove acquisizioni introdotte dalla rivoluzione digitale. Dopo tre anni di percorso partecipato che ha visto coinvolti 126 comitati in tutto il mondo, il 24/8/22 a Praga l'Assemblea Generale Straordinaria di ICOM approva la nuova definizione di museo, che va a modificare l'art.3[20] dello Statuto ICOM:

> *Il museo è un'istituzione permanente senza scopo di lucro e al servizio della società, che effettua ricerche, colleziona, conserva, interpreta ed espone il patrimonio materiale e immateriale.*
> *Aperti al pubblico, accessibili e inclusivi, i musei promuovono la diversità e la sostenibilità.*
> *Operano e comunicano eticamente e professionalmente e con la partecipazione delle comunità, offrendo esperienze diversificate per l'educazione, il piacere, la riflessione e la condivisione di conoscenze.*

Risulta interessante un confronto con la definizione precedente (Vienna 2007), che recitava:

> *Il museo è un'istituzione permanente, senza scopo di lucro, al servizio della società e del suo sviluppo, aperta al pubblico, che effettua ricerche sul patrimonio tangibile e intangibile dell'uomo e del suo ambiente, lo acquisisce, lo conserva, lo comunica e in particolare lo espone per scopi di studio, educazione e diletto.*

Vengono infatti introdotti, per quanto attiene al presente studio, i concetti di accessibilità, inclusività, condivisione ed esperienza: si sottolinea cioè la necessità di un'offerta ampia e diversificata affinché la visita si trasformi in un'esperienza che educhi e dia piacere. In particolare la parola chiave è *esperienza*, l'esperienza del visitatore sempre più coinvolto in ambienti immersivi che lo rendono parte attiva.

Ciò è dovuto a ragioni concomitanti: l'emergere di nuove istanze socio-culturali, le possibilità offerte dallo sviluppo tecnologico nel campo del digitale, l'avvento dell'Industria 4.0 e, a partire dal marzo del 2020, la pandemia di COVID19.[30]
Una rassegna del 2020 [27] della bibliografia relativa alla rivoluzione digitale nei musei e nel patrimonio culturale evidenzia come il settore museale stia vedendo nelle trasformazioni digitali una delle principali aree di politica e ricerca e rileva come l'attenzione si stia spostando sempre di più dalla centralità della tecnologia a quella del fruitore.

1.2 Il coinvolgimento dei musei nella rivoluzione digitale

La nuova definizione di museo recepisce dunque e al tempo stesso riflette un vasto e variegato panorama di innovazioni che a vario livello e con differente intensità stanno attraversando le istituzioni museali ed hanno ricevuto un inaspettato quanto decisivo impulso dalla situazione pandemica verificatasi a partire dal 2020 [30].
I musei, nel panorama dell'industria culturale, rappresentano un settore in rapida diffusione internazionale: una recente relazione dell'UNESCO [34] documenta l'esistenza di 95000 organizzazioni museali nel mondo tracciandone la distribuzione: il 65% in Nord America ed Europa occidentale, il 33% in Europa orientale, America Latina e Asia e lo 0,9% in Africa e lo 0,5% nei paesi arabi; dimostra inoltre che si è verificato nell'ultima decade un incremento del 60% dei musei del mondo.
Tuttavia occorre osservare che la situazione pandemica ha costituito un momento di discontinuità nella storia dei musei, in quanto ha comportato la chiusura del 90% delle strutture, imponendo la ricerca di forme alternative di fruizione e producendo come effetto nell'80% dei casi la digitalizzazione del rapporto con l'utente, ma nel restante 10% la definitiva chiusura di quei musei che non sono stati in grado di adeguarsi alle nuove esigenze.
Emerge un nuovo profilo di museo, una vera e propria industria, che segna il superamento della concezione tradizionale e si avvale della tecnologia sia a livello di promozione della cultura sia per la fruizione in loco sia per la gestione. L'uso di website, la presenza sui principali social networks, le biglietterie online, l'introduzione data base dei visitatori e chat sono stati i primi passi di un processo le cui linee di sviluppo stanno dando esiti quali la digitalizzazione del patrimonio museale, la sua fruizione digitale e il potenziamento del coinvolgimento esperienziale del pubblico.
Resasi urgente per l'impossibilità di fruire di persona dei musei nel periodo pandemico, la digitalizzazione del patrimonio è peraltro condizione necessaria per poter offrire contenuti digitali a supporto di nuove esperienze di fruizione.
Una recente indagine[14] condotta dall'Osservatorio Innovazione Digitale Nei Beni e Attività Culturali del Politecnico di Milano ha documentato come la pandemia abbia inciso sulla digitalizzazione del patrimonio: le istituzioni che hanno reso disponibile la collezione online sono passate dal 40% nel 2020 al 70% nel 2021 e il 24% dei musei considera la digitalizzazione l'attività prioritaria nei prossimi due anni.
In Italia proprio nel 2022 è stato redatto dall'Istituto Centrale per la Digitalizzazione della Cultura del Ministero dei Beni Culturali il PND, Piano Nazionale di Digitalizzazione del

patrimonio culturale, con lo scopo di favorire e coordinare il processo di trasformazione nel quinquennio 2022-2026, rivolgendosi in prima istanza ai musei[15] e agli altri luoghi della cultura che tutelano, gestiscono, valorizzano i beni culturali.
Intorno alla digitalizzazione del patrimonio in corso, di cui si riconosce l'utilità sia per la fruizione sia per la tutela, è ancora aperto il dibattito, che verte soprattutto sulla possibilità di riutilizzo dei contenuti digitalizzati e sugli elevati costi che essa comporta.
Il primo e più immediato esito della digitalizzazione è il virtual tour, che permette al visitatore di fruire del patrimonio museale da remoto. Nel giro di pochi anni si è diffusa così che è diventata prassi consolidata la possibilità di accedere a tour virtuali, fruire di laboratori, attività didattiche e podcast da remoto. E decine sono ad oggi i musei che offrono ai loro utenti visite guidate online. In Italia, dopo le esperienze pionieristiche del Museo Egizio di Torino e del Museo degli Uffizi di Firenze, il fenomeno si è diffuso a macchia d'olio e lo stesso MIC propone un Gran Virtual Tour[19] che si configura come un viaggio virtuale in tutta la penisola e permette di visitare da casa musei, teatri, parchi archeologici, biblioteche. Ancor più ambizioso é Europeana, il progetto di digitalizzazione

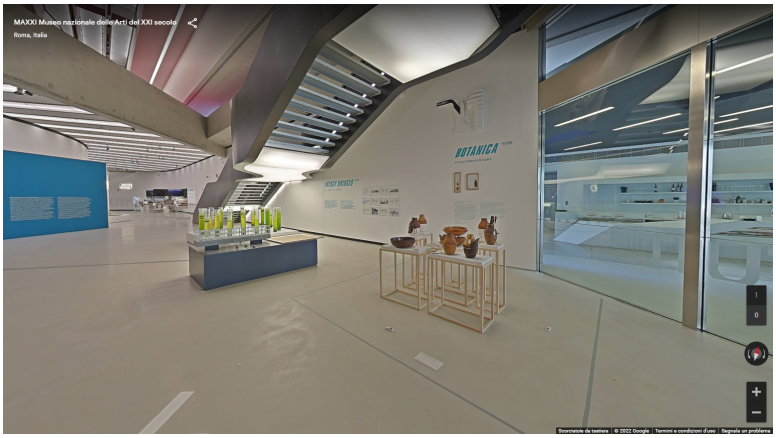


Figura 1.1: Esempio di virtual tour: MAXXI, Roma

del patrimonio culturale europeo, promosso dall'Unione Europea.[5]
Ma il digitale non è solo un sostituto della visita in loco, bensì uno strumento per attrarre nuove fasce di visitatori, come bambini e giovani, ed offrire al pubblico già fidelizzato forme diverse di esperienza ed attività che integrino la visita tradizionale.

1.3 Installazioni museali, un vastissimo e variegato panorama

In questo quadro sono proliferate, con un incremento esponenziale nel periodo di riapertura post pandemica, le forme di coinvolgimento dei visitatori, rese possibili da svariate nuove tecnologie e dalla loro combinazione: realtà aumentata (AR), realtà virtuale (VR) e ambienti immersivi sono tra le più utilizzate per catturare lo spettatore e renderlo protagonista di esperienze psicofisiche che costituiscono per lui un canale di apprendimento

e divertimento. In tale processo si è peraltro rivelata decisiva la competenza tecnologica degli utenti, abituati dall'uso di tablet e smartphone e smart tv a forme di comunicazione rapide e intuitive.
Allo scopo di contestualizzare il presente lavoro nel vasto panorama contemporaneo, si individuano alcune delle vie percorse: l'introduzione di installazioni all'interno di sale museali, le mostre immersive ed infine l'affermarsi di veri e propri musei dell'arte digitale.

1.3.1 L'introduzione di installazioni nelle sale museali

L'introduzione delle nuove tecnologie nei musei, rallentata inizialmente anche dai costi elevati, ha sollevato un dibattito tra gli intellettuali e gli addetti ai lavori: la questione riguarda il modo in cui le tecnologie possono produrre valore per i musei che le inseriscono come auhsilio [21][24]. Il confronto ha portato alla definizione di alcune coordinate all'interno delle quali debba muoversi la tecnica: prima fra tutte non costituire ostacolo o distrazione alla fruizione, ma arricchirla ed aumentarne il valore.
Esempi di efficace realizzazione di questo intento sono i serious games: il primo in ambito museale in Italia é stato "Father and son" [6], commissionato dal Museo Archeologico Nazionale di Napoli. Si tratta di un gioco di una sessantina di minuti che offre al visitatore la possibilità di viaggiare nel tempo e visitare le collezioni del museo , identificandosi con la storia di un figlio alla ricerca del padre. Il percorso gli fornirà la consapevolezza dei principali nuclei tematici che costituiscono il museo e della complessità delle epoche e delle situazioni attraversate. Diversamente dai giochi, che solitamente preparano o svi-

Figura 1.2

luppano dall'esterno la conoscenza dei musei, le installazioni ne arricchiscono la fruizione dall'interno. Le prime ad essere introdotte e le più diffuse nello scorso decennio sono stati gli schermi multimediali touch. Un museo pioniere nella sensibilizzazione ai cambiamenti climatici, il MoCC di Hong Kong (Jockey Club Museum of Climate), dal 2013 un polo di ricerca, insegnamento e formazione alla responsabilità verso il pianeta, le ha impiegate ampiamente sia nell'allestimento stabile sia nelle mostre. In una delle sale, ad esempio, viene simulata la Stazione di ricezione per il telerilevamento satellitare CUHK. Questo offre al visitatore la possibilità di sperimentare come ci si senta nel monitorare da vicino i cambiamenti climatici in differenti luoghi e tempi: infatti mediante un grande schermo

touch può autonomamente navigare su Hong Kong e vedere come aree specifiche diventerebbero se si innalzasse il livello del mare o si portassero alle estreme conseguenze le tendenze climatiche in corso [12].

Figura 1.3

Altrettanto coerente con l'esigenza di valorizzare i contenuti materiali del museo, ma anche con le nuove esigenze pandemiche, è l'installazione predisposta per la visita della Cattedrale di S. Agata a Catania, finanziata dall'Assessorato regionale per i beni culturali e l'identità siciliana. In questo caso si ricorre in esterno ad una postazione Leap Motion dotata di schermo olografico su cui vengono proiettati gli interni del monumento con possibilità di approfondimento tematico senza alcun contatto fisico per il visitatore.

1.3.2 Le mostre immersive

Nel 2021 si è verificato un boom delle mostre immersive, grazie allo sviluppo della tecnologia e dell'utilizzo del digitale nel periodo della prolungata chiusura pandemica. L'offerta si è arricchita con storytelling attrattivi tramite la correlazione multimediale di immagini, musiche, voci e testi, utilizzati innanzitutto a livello promozionale, tramite nuovi siti web sempre più accessibili, e quindi in loco con la modernizzazione delle collezioni stesse [29].
Si sono sviluppate installazioni sia permanenti sia temporanee ed itineranti, in luoghi adattati allo scopo, con impiego di *Augmented Reality* come, ad esempio, nelle *Infinity Mirror Rooms* presso la Tate Gallery (Figura 1.5)[16], nelle mostre *Future World-where art meets science* all'Art Science Museum di Singapore, dove installazioni immersive reagiscono alla presenza o al tocco dei visitatori (Figura 1.6)[7].

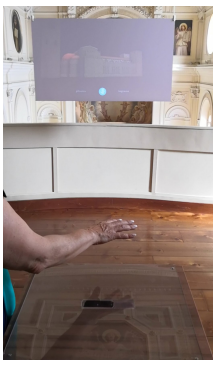


Figura 1.4

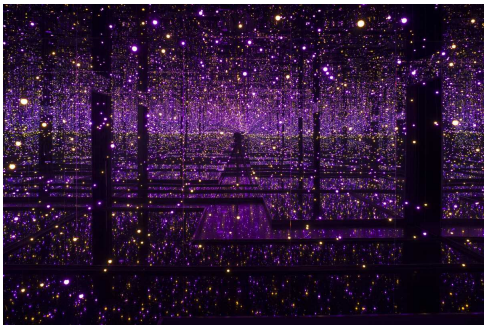


Figura 1.5: Mirror Rooms

Figura 1.6: Future World: Sketch Aquarium

1.3.3 I musei dedicati all'arte digitale

La più recente tappa del processo di trasformazione in atto ha portato alla nascita di musei appositamente dedicati all'arte digitale. La Comunità Europea li promuove con la creazione della rete internazionale V-MusT.net con l'obiettivo di introdurre linee guida e pratiche comuni. Li accomunano le dimensioni degli spazi ed il carattere spettacolare dell'offerta, ma anche il superamento della fruizione tradizionale dell'opera d'arte: il visitatore non viene a contatto con l'opera stessa, ma con il suo simulacro virtuale, e sperimenta nuovi percorsi conoscitivi grazie alle sofisticate proiezioni e animazioni tridimensionali.
Tra i pionieri in questo campo si ricorda l'Atelier des Lumierès di Parigi, il primo centro di arte digitale della capitale francese, aperto al pubblico nell'agosto 2018. Ricavato in una fonderia dell'Ottocento, ospita mostre digitali immersive, in cui le opere di grandi artisti vengono proiettate sul pavimento e sulle altissime pareti dell'edificio. Segue l'esempio parigino la Lichthalle-Maagdi Zurigo, che propone al visitatore l'immersione nei capolavori della storia dell'arte mediante animazioni e musica, favorendo un approccio coinvolgente che attrae anche bambini e ragazzi, avvicinandoli all'arte in modo ludico.
Il TeamLab Borderless di Tokyo, inaugurato nel 2019, si presenta come il più spettacolare dei musei digitali. Il suo concept sviluppa il tema del mondo senza confini: il visitatore entra in uno spazio senza confini, dove non viene guidato e non è previsto un percorso prestabilito, ma la ricerca di esperienze inedite.
Non mancano infine istituzioni di ricerca ed avanguardia artistica: nell'agosto del 2020 apre ad Amsterdam, in un quartiere post industriale, il Nxt Museumper l'esposizione di installazioni immersive di artisti, designer e scienziati con l'obiettivo di acquisire e conservare le installazioni in forma permanente e di offrire opere all'avanguardia nel campo della New Media Art.

Capitolo 2

Caso di studio: Papiro dei Re

Il caso di studio, The List of Kings, rappresenta uno dei reperti più preziosi, ma anche uno dei più importanti lavori di ricerca e di promozione culturale del Museo Egizio di Torino.

2.1 Il Museo Egizio di Torino

Il Museo Egizio di Torino, il più antico al mondo dedicato esclusivamente a reperti nilotici, vanta una storia bicentenaria. Fu infatti istituito nel 1824 da Carlo Felice di Savoia, che acquisì la preziosa collezione del console Drovetti e la unì alla collezione privata dei Savoia e alla collezione Donati, dando vita al Museo delle Antichità Egizie di Torino. Tra gli apporti successivi i più consistenti risalgono al periodo tra il 1903 e il 1937, in cui il museo si arricchì dei reperti provenienti dalla Missione Archeologica Italiana in Egitto. A partire dal nuovo millennio il museo ha avviato un processo di trasformazione che nel giro di una ventina d'anni ne ha fatto un importante polo di ricerca e promozione culturale a livello internazionale, oltre che una delle principali attrazioni turistiche italiane.

Le fasi della recente e rapida trasformazione, con cui si è realizzato il superamento della concezione tradizionale di museo, costituiscono un paradigma, che verrà seguito da altri musei, quali, ad esempio, nel 2009 il MAXXI, Museo Nazionale delle Arti del XXI secolo.[23] Se ne ripercorre l'iter al fine di contestualizzare il presente lavoro.

Nel 2004 si costituisce la fondazione del Museo delle Antichità Egizie di Torino, il primo esempio italiano di partecipazione del privato alla gestione di un patrimonio culturale pubblico[23]: ne fanno parte Il Ministero dei Beni e delle Attività Culturali, la Regione Piemonte, la Compagnia di San Paolo e la Fondazione CRT. Lo scopo è "la valorizzazione, promozione, gestione e adeguamento strutturale, funzionale ed espositivo del Museo, dei beni culturali ricevuti o acquisiti a qualsiasi titolo e la promozione e valorizzazione delle attività museali"[13]. Dal 2010 al 2015 si svolgono i lavori di rifunzionalizzazione, restauro e messa in sicurezza della sede. Nel 2011 il museo viene riconosciuto come ente di ricerca. A partire dal 2016 promuove e organizza mostre in Germania, Russia e Cina.

Oggi presenta tutti i tratti di digitalizzazione che caratterizzano un'istituzione museale all'avanguardia: pagina web, presenza sui social, collezione digitalizzata ed accessibile agli

studiosi, offerta di visite virtuali, laboratori ed attività didattiche, organizzazioni di eventi culturali, comunicazione web con il pubblico, raccolta dati del pubblico.

Nell'ottica della pianificazione delle attività di ricerca e della creazione di eventi rivolti sia a nuove fasce di pubblico attratte dall'occasione sia al pubblico già fidelizzato, nel 2022 il Museo Egizio sceglie di celebrare il bicentenario della decifrazione dei geroglifici ad opera dell'archeologo Jean-Francois Champollion con la presentazione del lavoro di studio e restauro promosso dalla Fondazione sulla Lista dei Re.

2.2 La Lista dei Re

La lista dei Re si trova in un papiro databile al XIII sec. a.C, appartenente alla collezione originaria del console Drovetti, la cui importanza venne riconosciuta dallo stesso Champollion.

Figura 2.1: Lista dei Re (verso del papiro)

Esso contiene sul retro una lista dei faraoni dalla creazione fino al regno di Ramses II, l'unico elenco completo che possediamo.

> *"Una lista cronologica scritta in ieratico, in 11 colonne, dei sovrani d'Egitto a partire dal periodo mitologico primordiale, con i regni divini di Geb, Osiride, Horo, Seth, Maat, fino alla fine del Secondo Periodo Intermedio (1650 a.C). Di ciascun sovrano sono ricordati titolatura, nomi e durata del regno in anni, mesi e talvolta persino giorni. La lista dei re menziona sovrani di grande importanza come Menes-Namer (circa 2900 a.C) primo sovrano non divino di cui si fa menzione, o come Djoser (2592-2566 a.C) che eresse la prima grande piramide della storia."*[10]

L'unicità del documento consiste nel fatto che non si configura come un canone selezionato in base a criteri politici, come altre liste parziali che possediamo[1], nelle quali non compaiono tutti i nomi senza che peraltro venga esplicitata la ragione dei casi di *dominatio memoriae*. Si tratta invece di un elenco completo. É probabile che sia stato uno dei documenti ufficiali consultati in epoca tolemaica dallo studioso egiziano Manethon per redigere l'opera in lingua greca *Ta Aigyptiakà*, attribuitagli dalla tradizione, su cui si fonda la periodizzazione dei regni dell'Antico Egitto attualmente adottata dagli studiosi.

[1] ad esempio, la Lista di Abido e la Lista di Saqqara

La lista si trova sul verso (Figura 2.1) di un papiro il cui recto (Figura 4.58) presenta un registro tributario del tempio di Amon a Karnak datato al tempo di Ramses II: vi sono elencati i nomi di coloro che in tutto l'Egitto sono tenuti a versare tributi al tempio, dai semplici pescatori agli alti funzionari, ai sovraintendenti delle fortezze della regione meridionale. Quando l'elenco non fu più utile, forse per una nuova organizzazione del sistema tributario, venne riutilizzato e destinato ad altro, cioè alla lista reale.

Figura 2.2: Registro tributario (recto del papiro)

2.3 Gli interventi di restauro

Il papiro contenente la Lista dei Re, pur essendo stato acquistato integro dal Drovetti intorno al 1820, arriva al Museo Egizio in condizioni frammentarie. Inizia così un'attività di studio, classificazione e restauro durata duecento anni.

Il primo ad occuparsene , nel 1824, è proprio Champollion, che identifica 47 dei 300 frammenti e realizza i primi fac-simili, seguito da Richard Lepius che crea il primo fac-simile completo, mentre John Wilkinson redige la prima trascrizione fronte-retro comprendente la lista tributaria e finalmente nel 1938 Giulio Farina pubblica la prima edizione corredata di foto e trascrizione in caratteri geroglifici. Negli anni Cinquanta dello scorso secolo Alan Gardiner apporta alcune modifiche alla collocazione dei frammenti del Farina e a partire dal 1997 riprendono gli studi. Diversi ricercatori scrivono note e rivedono parzialmente la disposizione fino all'opera di ripubblicazione di Kim Rhyolt dell'Università di Copenaghen che crea una nuova ricostruzione e disposizione delle colonne, avvalendosi di strumenti digitali, eliminando errori ed aggiungendo una ventina di pezzi precedentemente considerati estranei alla lista. Parallelamente alle attività di ricerca sul contenuto del papiro, si svolgono alcuni interventi di restauro: nell'Ottocento Gustav Seyffarth identifica duecento frammenti e li incolla su carta vegetale, all'inizio del Novecento Hugo Ibscher li stacca dalla carta vegetale, li pulisce e ricolloca unendoli con sottili strisce di seta, affiancato nel suo lavoro da Erminia Caudana, che recepisce le variazioni introdotte dal Farina con cui lavora a stretto contatto; nel 2021 il Museo Egizio affida il lavoro di restauro a Myriam Krutsch, restauratrice dell'Agyptisches Museum und Papyrussammlung di Berlino, che, seguendo la ricostruzione di Kim Ryholt, distacca le strisce di seta, pulisce e restaura oltre 300 frammenti, montandoli su un foglio di carta giapponese con una tecnica usata per la prima volta al Museo Egizio di Torino.
La nuova ricostruzione di Kim Ryholt ed il restauro di Myriam Krutsch costituiscono un significativo apporto agli studi di egittologia e mostrano quanto gli sviluppi della tecnologia

abbiano accelerato i processi di comprensione e tutela del patrimonio papiraceo conservato: un felice connubio tra intelligenze artificiali, sofisticati algoritmi, saperi accademici e tecniche manuali.

2.4 La celebrazione del bicentenario della decifrazione dei geroglifici

Intorno agli esiti delle attività di riclassificazione e restauro del Papiro dei Re la fondazione del Museo Egizio pianifica un evento rivolto ai suoi visitatori e alla comunità accademica in occasione della celebrazione del bicentenario della decifrazione dei geroglifici da parte di Jean-Francois Champollion.
L'evento, a cui il sito del museo, i social e la stampa conferiscono grande rilievo, si sviluppa nell'arco della giornata del 27 settembre, con apertura serale ad ingresso gratuito, ed ha tre momenti di forza: l'inaugurazione di una sala in cui è possibile ammirare il Papiro dei Re ed approfondirne la conoscenza tramite un'installazione interattiva e un pannello grafico illustrativo; la conferenza dal titolo *History, Content and Restoration of the so-called "Turin King List"* con gli interventi del direttore Christian Greco, dell'egittologo Kim Ryholt, della restauratrice Myriam Krutzsh e della responsabile della Collezione Papiri del Museo Egizio Susanne Töpfer; il percorso di visita sulla scrittura egizia "Sulle tracce di Champollion", guidato da esperti egittologi, in esclusiva per la giornata delle celebrazioni.
L'installazione oggetto del presente lavoro è stata ideata per l'esposizione temporanea del papiro nella sala dedicata e sarà definitivamente collocata, a partire dal mese di dicembre, nel settore del Museo destinato alla lingua ed alla scrittura degli Egizi. Resa possibile dal finanziamento degli "Scarabei", l'Associazione Sostenitori Museo Egizio, fondata nel 2007 e composta prevalentemente da sostenitori radicati nel territorio piemontese, e dalla Consulta per la valorizzazione dei Beni artistici e Culturali di Torino, l'opera è stata realizzata in collaborazione con Robin Studio di Torino.

Capitolo 3

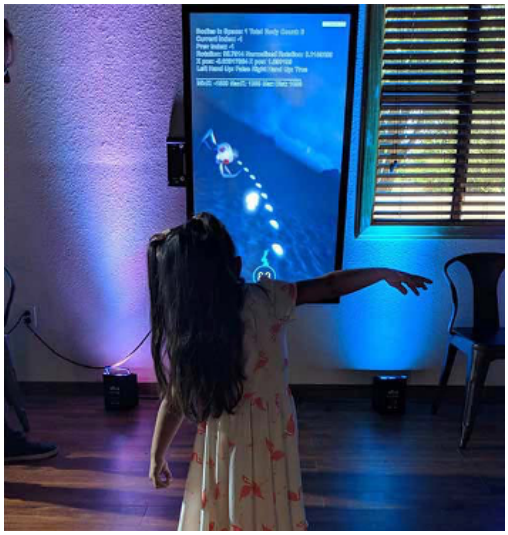
Opzioni Touchless

Per le esigenze descritte nel caso di studio, la scelta cade su tecnologie touchless.
Le tecnologie di interazione touchless esistono da più di un decennio, ma non hanno ancora preso il sopravvento sul touch. E' evidente che il contatto fisico diretto con un'interfaccia sia il modo più intuitivo di interagire, e fino ad ora non c'era stato motivo di sostituirlo con uno più instabile e meno intuitivo.
Negli ultimi anni i musei avevano ampiamente installato touchscreen e pulsanti che consentissero l'interazione. Ad esempio la New York Historical Society, nella ristrutturazione della sua sede, un investimento da 65 milioni di dollari, aveva installato stazioni touchscreen: totem touch, tavoli interattivi, pannelli e ogni genere di allestimenti multimediali interattivi.
Ma la pandemia del COVID19 ha invertito questa tendenza e, come affermato da Janice Majewski[8], consulente museale dell'Institute for Human-Centered Design, i musei in questo momento stanno esplorando la possibilità di introdurre funzionalità multisensoriali e interazioni basate sui gesti: la situazione post-pandemica ha reso il touchless un'alternativa non solo utile, ma necessaria. Si sono quindi rivalutati quei dispositivi già esistenti da tempo, ma le cui potenzialità non erano state sviluppate fino in fondo perché non vi era una vera necessità di utilizzo.

Gli studi in merito all'efficacia dell'applicazione di tali tecnologie in ambito museale non sono ancora molti[25]. Tuttavia rivelano che i due vantaggi più evidenti delle interazioni basate sui gesti sono proprio il fatto che offrono un'alternativa al touch più sicura sul piano igienico e che possono risultare coinvolgenti per utenti di età diversificate. Gli studiosi Koutsabasis e Vosinakis[25] in particolare osservano che l'approccio cinestetico[1] è considerato di alto valore pedagogico per l'educazione museale specialmente per coinvolgere i visitatori negli aspetti immateriali del patrimonio culturale, quali attività quotidiane, riti e abitudini, in quanto consentono un approccio più immediato. Inoltre le interazioni naturali ed intuitive rendono più divertente l'esperienza del visitatore e così mantengono vivo più a lungo il suo interesse e il suo coinvolgimento.

[1] che riguarda il movimento

(a) bare-hand interaction



(b) full-body interaction

Figura 3.1: [9]

Un punto di debolezza tuttavia, secondo Jim Spadaccini, fondatore di Ideum, sta nella necessitá di una estrema precisione dei sensori e di fornire istruzioni semplici e chiare per i gesti in modo che la fruizione sia naturale ed immediata.[28]
Le applicazioni cinestetiche adatte al patrimonio museale sono al momento poche, ma sembra esservi un crescente interesse verso questa direzione di ricerca e di sviluppo[25].
Qui si prendono in esame in particolare la fotocamera di profondità Microsoft Kinect e la fotocamera ad infrarossi Leap Motion Controller[11].
La differenza principale tra questi due sistemi di *motion traking* è che il primo riconosce il movimento dell'intero corpo, mentre il secondo il movimento delle mani. Si può infatti in generale distinguere il tipo di interazione tra *whole-body*, dove tutto il corpo è coinvolto ed influisce sul contenuto digitale e *a mani libere*, dove il protagonista è il movimento delle mani, rendendo possibile un'interazione più precisa in cui anche le singole dita possono influire sul sistema.

3.1 Microsoft Kinect Sensor

Kinect è un sensore di movimento che registra le variazioni di posizione del corpo nello spazio 3D, ed è in grado di elaborarle in tempo reale combinando a queste caratteristiche un sistema di riconoscimento vocale, grazie al quale rileva l'identità degli utenti quando si avvicinano.
Rilasciato da Microsoft nel 2010 nel settore dei videogiochi, nasce come accessorio per XBOX360 e permette per la prima volta al giocatore di interagire con la console senza impugnare nessun controller. L'impatto di Kinect si è poi esteso oltre il settore dei giochi, grazie ai costi contenuti e alla grande disponibilità e sembrava avere tutti i presupposti

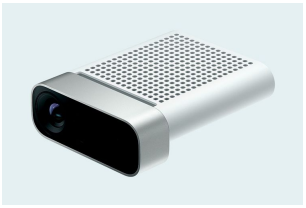
per rivoluzionare il mondo dell'esperienza videoludica, ma non solo. La prima versione (Kinect) viene presto migliorata e nasce Kinect 2. Entrambe le versioni sono oggetto di ricerca nella comunità scientifica: realtà virtuale, realtà aumentata, gesture recognition[35], interazione uomo-macchina, mappinge SLAM, rilevamento di oggetti e riconoscimento di oggetti tridimensionali, medicina e riabilitazione sono solo degli esempi[32].

Nonostante l'attenzione che questa nuova tecnologia ha attirato su di sé e i numerosi pezzi venduti, Microsoft decide di interrompere definitivamente la produzione delle due versioni del Kinect. Probabilmente la scelta viene compiuta a fronte dell'insoddisfazione dei consumatori dovuta alla ridotta presenza di videogiochi: pochissimi sviluppatori hanno deciso di investire nella creazione di prodotti videoludici con alla base questa tecnologia appena nata e quindi non senza imperfezioni. Infatti, nell'esperienza di gioco non mancavano problemi legati al tempo di risposta o errori grossolani di riconoscimento che attivavano comandi non voluti o che non li facevano partire affatto.

Nel 2019 Microsoft rilascia una nuova versione di Kinect, la Azure Kinect[2] che questa volta è indirizzato soprattutto agli sviluppatori piuttosto che ai videogiochi: si tratta di un sensore per la *mixed reality* equipaggiato da avanzati sensori AI e dalle prestazioni ricche di potenzialità. Si analizzano le caratteristiche tecniche nelle sezioni successive.

(a) Kinect

(b) Kinect 2



(c) Azure Kinect

3.1.1 Kinect e Kinect 2

La prima versione di Kinect si compone di una camera RGB (8-bit VGA resolution of 640x480), un proiettore infrarossi (IR), una camera IR e un array di quattro microfoni. Nella Figura 3.3 si localizzano sul sensore i componenti citati.

La camera IR consiste in un sensore CMOS[3] a infrarossi che percepisce l'ambiente attraverso lo spettro del bianco e nero, dove il nero puro rappresenta l'infinito lontano e il bianco puro l'infinito vicino. Il proiettore IR è un laser a infrarossi che, passando attraverso una grata di diffrazione, si trasforma in una serie di punti infrarossi. L'unione di proiettore e camera IR compone il sensore di riconoscimento della profondità, tecnologia ottenuta in licenza dalla azienda sviluppatrice israeliana PrimeSense.

Tecnologia PrimeSense I dettagli sul funzionamento della tecnologia non sono noti, ma gli esperti ritengono che la mappa di profondità sia stata ottenuta combinando il

[2]https://azure.microsoft.com/en-us/products/kinect-dk/

[3]complementary metal oxide semiconductor

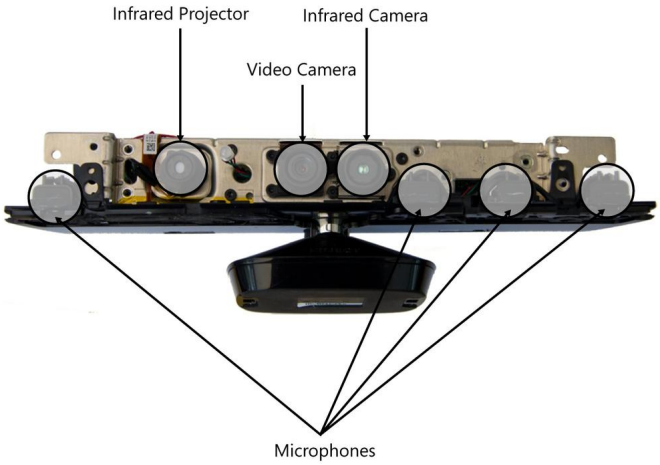


Figura 3.3: Kinect v1

metodo della *structured light* con il *depth from focus* e la *visione stereo*. La structured light (Figura 3.4) stima le distanze a partire dalla distorsione di un pattern noto proiettato nella scena.

Nel caso del Kinect il pattern proiettato è un insieme di punti. (Figura 3.5) Grazie al

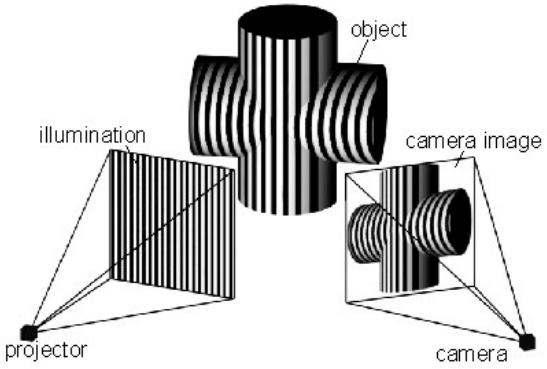


Figura 3.4: Structured Light method

principio per cui il grado di sfocatura dipende dalla distanza *(depth from focus)* la presenza di lenti con lunghezze focali diverse in X e Y fanno sì che i cerchi proiettati diventino ellissi la cui orientazione dipende dalla distanza.

La *visione stereo* entra in gioco poiché la camera IR e il proiettore IR si trovano in due posizioni diverse (Figura 3.6): osservando la scena da due posizioni diverse (visione stereoscopica) è possibile stimare la profondità. La stima in questo caso avviene quindi

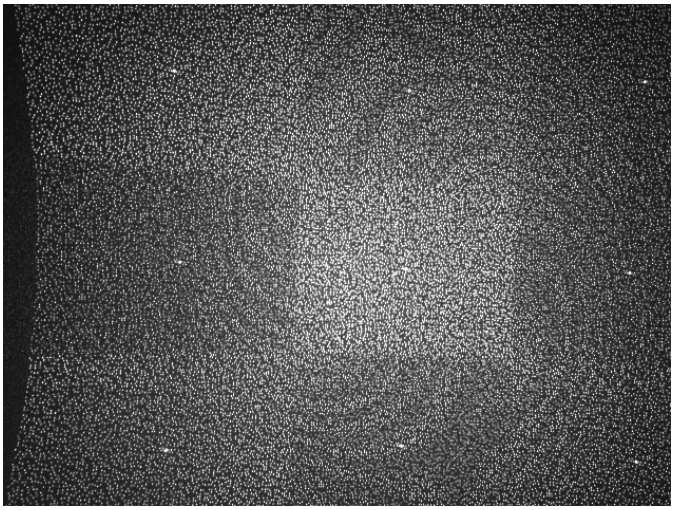


Figura 3.5: MS Kinect infrared pattern. [26]

a partire dalla differenza di posizione tra i punti proiettati e quelli catturati con calcoli trigonometrici.

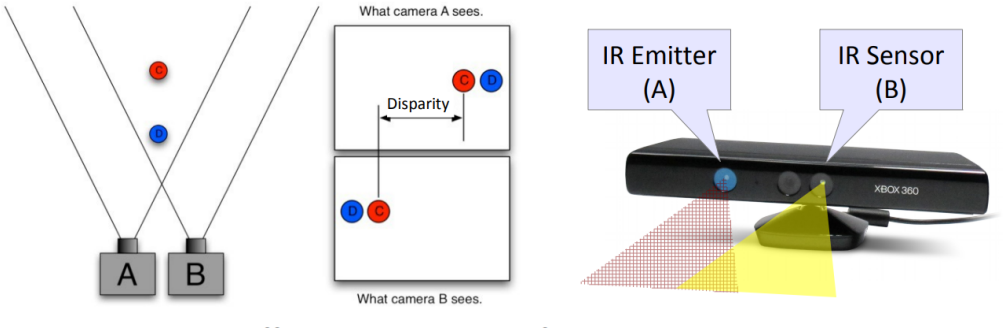


Figura 3.6: Kinect - visione stereo

I limiti tecnologici della tecnologia PrimeSense consistono nella presenza di elementi in grado di compromettere il pattern proiettato dall'emettitore a infrarossi, e quindi generare conseguenti errori. Esempio di fonte di disturbo non di poco conto è la luce solare diretta così come anche l'uso contemporaneo di più sensori con conseguente sovrapposizione di aree di proiezione.
Il tracking dello scheletro avviene tramite software, che nel caso di Microsoft con la prima versione di Kinect, è in grado di individuare 20 *joints*[4] del corpo umano. Lo scheletro

[4] nodi

viene individuato a partire dalla mappa di profondità dell'utente che stima delle parti del corpo tramite tecniche di computer vision e mashine learning.
Con Kinect 2, pensata per Xbox One, Microsoft abbandona PrimeSense a favore di un sistema proprietario non più basato sulla structured light, apportando miglioramenti da più punti di vista.
La risoluzione video passa dal livello VGA a 1080p @30fps, garantendo così un'esperienza più fluida. La connessione USB 3.0 riduce il ritardo a 2 fame al secondo. Cambia l'ap-

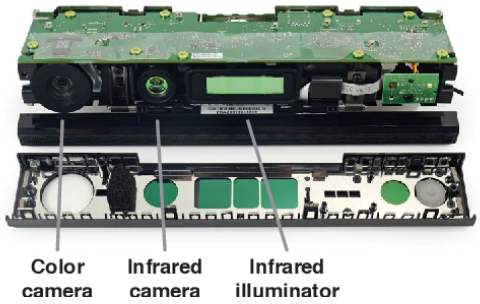


Figura 3.7: Kinect 2

proccio, che ora si basa sulla modulazione dell'intensità ad onda continua, comunemente utilizzata nelle telecamere a tempo di volo. Nella telecamera a tempo di volo ad onde continue, la distanza viene calcolata come differenza di fase tra il segnale emesso e quello riflesso: la luce proveniente da una sorgente luminosa modulata in ampiezza viene retro diffusa dagli oggetti del campo visivo della telecamera e ne viene misurato il ritardo di fase, tradotto in un valore di distanza per ciascun pixel. (Figura 3.8)

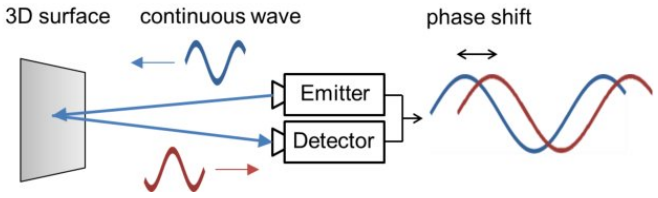


Figura 3.8: Principio della telecamera a tempo di volo

Ad ogni pixel della mappa di profondità corrisponde dunque una misura reale. Grazie a ciò, Kinect 2 è più veloce ed accurato della prima versione, può supportare il riconoscimento di sei individui con scheletro completo ovvero informazioni su 25 joints. Migliora inoltre il tracking del volto e si introduce il riconoscimento delle gesture tramite il tool *Gesture Builder*[5].

[5]https://www.nuget.org/packages/VisualGestureBuilder/2.0.1410.19000

Rimangono tuttavia, gli stessi i limiti tecnologici legati all'uso delle tecnologie IR che presentava il predecessore, quali la luce solare diretta, l'uso di più emettitori IR e la presenza di elementi in grado di attenuare il segnale IR.

3.1.2 Kinect Azure DK

L'ultima versione di Kinect è basata anch'essa sulla telecamera a tempo di volo a corrente continua, ma con profondi miglioramenti a partire dalla risoluzione della camera RGB che ora raggiunge i 3840 x 2160 pixels. É composta (Figura 3.9) da un sensore di profondità

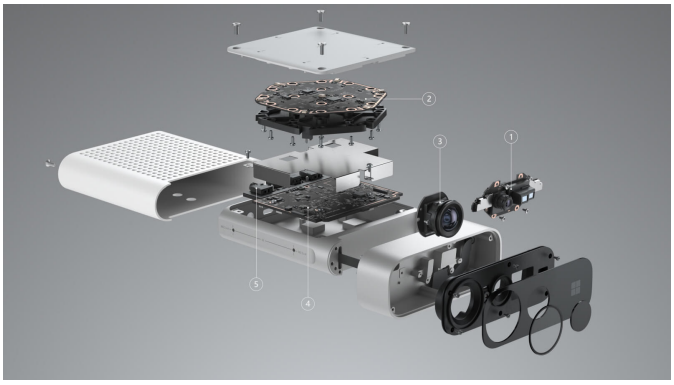


Figura 3.9: Microsoft Azure Kinect. Source: https://azure.microsoft.com/

(1), un array di 7 microfoni (2), una videocamera RGB 12-MP (3), un accelerometro e giroscopio (4) e pin di sincronizzazione esterni per facilitare la sincronizzazione tra più Kinect (5).
Funziona in quattro modalità diverse: *NFOV* [6] *unbinned*, *WFOV*[7] *unbinned*, *NFOV binned* e *WFOV binned*.
Azure Kinect DK[8] è venduto a 399$ su Microsoft Store, ma è al momento disponibile solo presso Stati Uniti, Cina, Germania, Giappone e Regno Unito[4]

Di seguito la tabella riassuntiva di confronto delle tre versioni di Kinect.

[6] narrow field-of-view depth mode

[7] wide field-of-view depth mode

[8] Developer kit

Funionalità	Kinect	Kinect 2	Azure Kinect
Color camera resolution	1280 × 720 px @12 fps 640 × 480 px @30 fps	1920 x 1080 px @30 fps	3840 x 2160 px @30 fps
Depth camera resolution	320 × 240 px @30 fps	512 × 424 px @30 fps	NFOV unbinned—640 × 576 px @30 fps NFOV binned—320 × 288px @30 fps WFOV unbinned—1024 × 1024px @15 fps WFOV binned—512 × 512px @30 fps
Depth sensing technology	Structured light–pattern projection	ToF (Time-of-Flight)	ToF (Time-of-Flight)
Field of view (depth image)	57 ° H, 43° V alt. 58.5° H, 46.6°	70° H, 60° V alt. 70.6° H, 60°	NFOV unbinned—75° × 65° NFOV binned—75° × 65° WFOV unbinned—120° × 120° WFOV binned—120° × 120°
Specified measuring distance	0.4–4 m	0.5–4.5 m	NFOV unbinned—0.5–3.86 m NFOV binned—0.5–5.46 m WFOV unbinned—0.25–2.21 m WFOV binned—0.25–2.88 m
Weight	430 g (without cables and power supply); 750g (with cables and power supply)	610 g (without cables and power supply); 1390g (with cables and power supply)	440 g (without cables); 520 g (with cables, power supply is not necessary)

3.1.3 Esempio di interazione *whole body* : *Be a Bug* by Ideum

Come affermato precedentemente, Kinect è stato ed è tutt'ora utilizzato come strumento base per la realizzazione di sistemi in cui l'interazione uomo-macchina avviene attraverso il tracking del movimento dell'intera figura umana. Se l'esperienza può essere pensata e sviluppata per singolo utente oppure per più utenti, andando a creare veri e propri ambienti immersivi; si prende di seguito in esame un esempio di applicazione della tecnologia Kinect dove l'esperienza è concepita per singolo utente. Ideum[9] è una compagnia del New Mexico specializzata nell'ideazione nella creazione di esperienze digitali e installazioni museali touchless. I lavori di Ideum basati sulla tecnologia Kinect di Microsoft iniziano nel 2015 e la prima installazione touchless che creano è *Be a Bug* presso il ABQ BioPark[3] (USA).

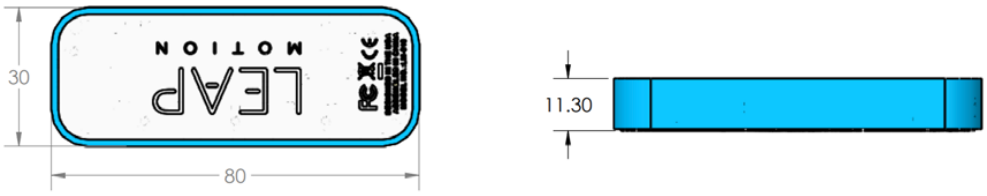
Figura 3.10: *Be a Bug* presso ABQ BioPark

L'installazione (Figura 3.1b), progettata per la fruizione singola, permette al visitatore di scegliere tra una serie di insetti *(bugs)* di cui a questo punto potrà andare a controllare alcuni movimenti, come se fosse l'insetto stesso: aprendo e muovendo le braccia come per volare, l'insetto virtuale si muoverà di conseguenza.
Perché il sistema funzioni, è necessario che durante la fruizione l'utente sia posizionato in uno specifico punto di fronte allo schermo. I movimenti del corpo, infatti, sono riconosciuti e tracciati solo se il visitatore si trova in un'area specifica di fronte allo schermo. A fronte di ciò, è stata posizionata una grafica circolare sul pavimento per chiarificare ai visitatori quale fosse la posizione per vivere in prima persona l'esperienza.
Si è poi optato per inserire ulteriori grafiche sul pavimento (visibili in Figura 3.10) per attirare il pubblico allo schermo, incoraggiarlo e chiamarlo all'azione: ciò si rende necessario

per via della scelta di Ideum di utilizzare il touch come prima interazione in particolare per l'individuazione dell'insetto da "interpretare".
Come fa notare Spadaccini, fondatore di Ideum, in questo tipo di installazioni è importante sì rendere il visitatore consapevole di come interagire, del fatto che può farlo, e del risultato che il suo specifico movimento determina, ma la fila di visitatori sempre presente in attesa del proprio turno impara il funzionamento osservando chi viene prima. In *Be a Bug* un'ulteriore feedback real time è dato all'utente grazie alla presenza sullo schermo di indicatori animati che suggeriscono l'azione. Sempre secondo Spadaccini, la chiave per il successo dell'esperienza si trova proprio nel rendere consapevole l'utente dei propri movimenti, di come questi vengano ricevuti dal sistema e l'uso di indizi visivi.[31]
Be a Bug è stato creato utilizzando l'ambiente di sviluppo Unity 3D; lo schermo utilizzato per l'installazione è Presenter 65[9] di Ideum.

3.2 Leap Motion Controller

Il Leap Motion Controller viene immesso sul mercato da Leap Motion Inc. nel 2012. Si tratta di un piccolo dispositivo che - grazie alla presenza di due fotocamere infrarossi e tre LED infrarossi - è in grado di tracciare il movimento delle mani e delle dita in modo piuttosto preciso. Inizialmente pensato per poter controllare applicazioni sul PC senza dover toccare nulla, il suo utilizzo si diffonde fino al mondo della realtà virtuale (VR) proprio come dispositivo di hand tracking. Nel 2019 Leap Motion Inc. viene acquistata da Ultrahaptics, il cui nome viene quindi cambiato in Ultraleap. Oggi, la tecnologia Leap Motion, già utilizzata principalmente in ambito Videogame, entra a pieno titolo nell'ammodernamento dell'offerta museale con particolare riguardo all'uso integrato della Gamification con la tecnica Touchless, a seguito della ricerca di modalità di fruizione dei device di controllo e comando interattivo senza contatto fisico.



(a) Caption1 (b) Caption 2

Figura 3.11: Dimensioni in mm; Peso: 32g

Esaminando ora il lato tecnico, il Leap Motion Controller presenta un'area di interazione che si estende da 10 cm a 60 cm o più, estendendosi dal dispositivo in un campo visivo tipico di 140x120° (Figura 3.12).

Il tracciamento funziona in diverse condizioni ambientali. Il funzionamento del sensore è determinato da ciò che troviamo al suo interno (Figura 3.13): due telecamere infrarossi (IR) e tre emettitori infrarossi (IR). Il produttore dichiara che la precisione di rilevamento delle dita è di circa 0.01 mm. Le telecamere IR si basano su CCD (Charge-Coupled Device). I dipositivi ad accoppiamento di carica (CCD) sono composti da una matrice di sensori basati su strutture MOS (metallo opto semiconduttore). I semiconduttori possono accumulare una carica elettrica proporzionale all'intensità della radiazione elettromagnetica che li colpisce e vengono accoppiati proprio per rendere possibile il trasferimento di energia tra elementi adiacenti.

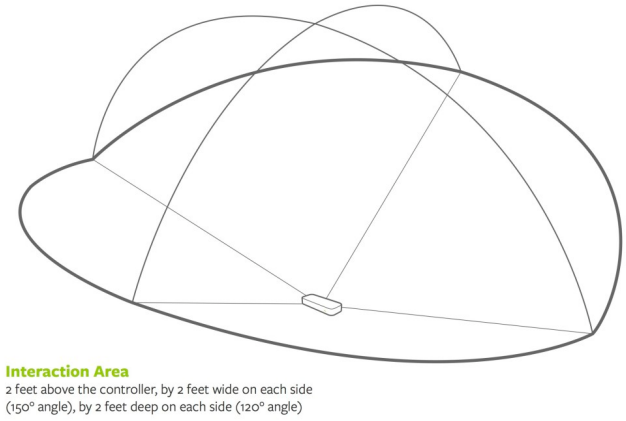
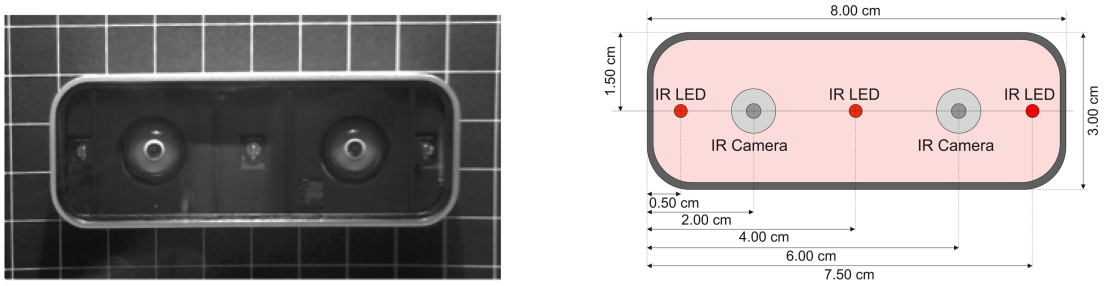


Figura 3.12: Interaction area Leap Motion Controller



(a) **(b)**

Figura 3.13: Leap Motion Controller[36]

3.2.1 Leap Motion SDK

I dati di input provenienti dal sensore sono processati sul computer grazie al Software Development Kit (SDK) fornito da Leap Motion Inc. per permettere agli sviluppatori di integrarli facilmente nelle loro applicazioni. Tramite SDK infatti si accede ad API di alto livello e dati astratti come il numero di dita e mani riconosciute, la loro posizione, la posizione del palmo, l'esatta posizione che un dito o uno strumento sta puntando sullo schermo, una serie di movimenti gestuali come, ad esempio, il tocco o lo scorrimento, oltre alle variazioni di posizione e orientamento tra frames. I dati vengono forniti sotto forma di frames: ogni frame contiene le informazioni di ciò che è stato tracciato e queste informazioni sono distribuite all'interno di una gerarchia che inizia con l'oggetto *mano* (Figura 3.14) che include: posizione e velocità del palmo, vettori di direzione e normali, base ortonormale, dita - che a sua volta includono posizione e velocità

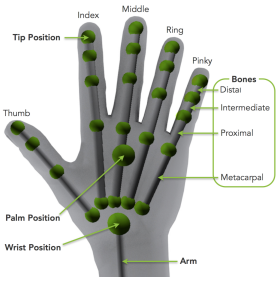


Figura 3.14

dell'estremità *(tip position)*, vettore di direzione, base ortonormale, dimensioni e posizione dei giunti delle ossa - e ancora informazioni di posizione del polso.
I movimenti della mano sono stimati a partire dalle variazioni di posizione tracciate nel tempo: il calcolo del movimento nel tempo viene fatto mettendo a confronto due *frames* che contengono la stessa mano.
Le *gestures* - modelli di movimento riconosciuti come eventi che possono essere utilizzati per innescare determinate azioni - incluse nel Leap Motion API sono *swipe, circle e tap.*
Ad oggi, con lo sviluppo da parte di Ultraleap di nuovi software di tracciamento sempre accurati, l'uso delle *gesture* è sempre meno presente.
Il software di tracking più recente è *Gemini*, che sarà trattato più nel dettaglio nella successiva sezione dedicata.

3.2.2 Gemini

Disponibile nella sua ultima versione dal 2021, Gemini è una piattaforma di quinta generazione per il tracciamento della mano, che si presenta come la più robusta e flessibile finora realizzata. Viene migliorato in particolare il riconoscimento delle due mani in contemporanea, rendendolo stabile anche in situazioni limite come la sovrapposizione.
I requisiti minimi hardware sono:

- Windows® 10, 64-bit
- Intel® Core™ i3 processor 5th Gen (must support AVX instructions)
- 2 GB RAM
- USB 2.0 port

Il software per Windows include un pannello di controllo dal quale è possibile visualizzare le riprese della telecamera e modificarne le impostazioni; il software per servizio di tracking che fa partire i moduli della telecamera e invia i dati di tracciamento alle applicazioni che li richiedono; il Software Development Kit, necessario per lo sviluppo di applicazioni che usano la libreria LeapC API (nel caso in cui si utilizzi come ambiente di sviluppo Unity o Unreal, esistono invece i rispettivi plugins); la libreria OpenX API Layer, adatta per le applicazioni che utilizzano OpenXR.
Il *setup* per il corretto utilizzo di Gemini prevede semplicemente il collegamento di una *Ultraleap Hand Tracking Camera* e il download del software stesso. A questo punto, accedendo al pannello di controllo è possibile visualizzare le immagini infrarosse provenienti direttamente dal modulo della telecamera e il tracciamento delle proprie mani (Figura 3.15).

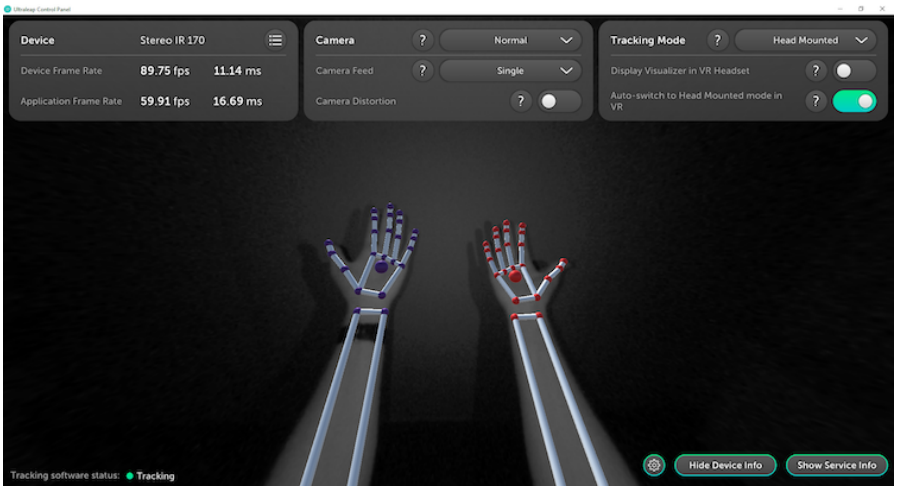


Figura 3.15: Interfaccia del pannello di controllo del sistema di tracciamento by Ultraleap

3.2.3 TouchFree

É ottobre 2020 quando Ultraleap lancia la sua nuova applicazione TouchFree[2] che, in risposta alla pandemia Covid-19, offre una soluzione di integrazione dei dispositivi di tracciamento delle mani, trasformando facilmente applicazioni basate sull'interazione *touch* in *touchless*. La particolarità di TouchFree si trova proprio nella sua semplicità di integrazione anche in interfacce utente già esistenti. Il pacchetto include l'applicazione, il servizio

Figura 3.16: Installazione con TouchFree e Ultraleap 3Di.[18]

Windows e TouchFree Tooling per la conversione dei dati di tracciamento.
TouchFree Tooling consente di trasformare i dati del dispositivo di tracciamento in modo che siano compatibili e utilizzabili come sistema di input: in concreto sullo schermo si genera un cursore controllabile in modalità touchless. É attualmente disponibile per due ambienti di integrazione, quali Unity e Web (JavaScript) e fornisce una connessione Client al servizio di TouchFree, tramite il quale riceve i dati di posizione e interazione necessari per l'interazione touchless.

Nel capitolo 4 si svilupperà nel dettaglio l'integrazione di TouchFree in Unity, in quanto strumento utilizzato nello sviluppo dell'applicazione del caso di studio.

3.2.4 Ultraleap 3Di

Ad inizio 2022 la compagnia Ultraleap lancia la nuova telecamera per l'hand tracking che completa la proprietaria applicazione TouchFree. Citando Steve Cliffe, presidente e

Figura 3.17: Ultraleap 3Di.[18]

CEO di Ultraleap, le interfacce touchless basate sui gesti saranno sicuramente preferite come opzione futura rispetto a quelle touchscreen.[33] Si tratta di un'affermazione che deriva dalle ricerche effettuate dall'azienda: esse hanno evidenziato come i consumatori degli Stati Uniti e del Regno Unito abbiano modificato il loro comportamento a seguito della pandemia Covid-19 e siano ora maggiormente consapevoli della necessità di limitare la contaminazione batterica e per una maggiore sicurezza gradiscano l'utilizzo della tecnologia touchless, soprattutto in ambienti pubblici dove sarebbe impossibile garantire un'igiene adeguata.
Dal punto di vista tecnico la telecamera Ultraleap 3Di è pensata per essere posizionata su schermi orizzontali fino a 42" e schermi verticali fino a 32". É inoltre caratterizzata da un elevato frame rate (90 fps).

3.2.5 Esempio di interazione *Mid-Air*: *The MAX Pottery Studio*

La tecnologia Leap Motion, come illustrato precedentemente, permette un tipo di interazione mid-air. Le finalità di utilizzo di questa tecnologia sono numerose, così come le modalità di fruizione, basate ad esempio sull'utilizzo di HMD (Head Mounted Display) o desktop (su schermo) o ancora in funzione dell'ambiente di interazione (tridimensionale o bidimensionale). L'applicazione che si prende a titolo d'esempio è stata progettata dal Mississipi Arts + Entertainment Experience (The MAX) a Meridian (Mississipi) nel

2019[22]. La tecnologia utilizzata è il Leap Motion Controller: qui l'interazione a mezz'aria con protagoniste le mani consente ai visitatori di realizzare virtualmente ceramiche seguendo il processo utilizzato da famosi ceramisti locali. Il sensore è disposto come in Figura 3.18 tra schermo e seduta. Interessante l'organizzazione della postazione che rende intuitivamente evidente al visitatore la modalità di fruizione: l'utente deve essere seduto, quindi non dovrà utilizzare il proprio corpo intero; è distante dallo schermo, quindi non sarà necessario interagire direttamente con esso toccandolo; le mani possono invece muoversi liberamente sopra la piattaforma circolare che separa l'utente dallo schermo. L'interazione in questo caso è tridimensionale ovvero influisce tridimensionalmente all'in-

Figura 3.18

terno dello schermo, dove è presente l'oggetto 3D della ceramica che l'utente modella con le sue mani.

Capitolo 4

Sviluppo

A partire dagli elementi emersi nel caso di studio, trattato in dettaglio nel capitolo 2, si è passati all'ideazione e allo sviluppo del progetto. In questo capitolo se ne traccerà il percorso fino alla creazione della versione finale dell'applicazione *Leap Pharaohs Experience*. Si tratterà dunque la fase iniziale di studio delle fonti che, parallelamente alla scelta delle tecnologie per l'interazione, ha portato alla realizzazione della prima bozza di storyboard; l'organizzazione dei contenuti; l'importanza della modularità legata alla creazione di un database modificabile a cui seguirà la descrizione dell'applicazione stessa in ogni sua parte strutturale tramite un'analisi dettagliata di ciascuna pagina; una parte finale che corrisponde all'inserimento dei contenuti definitivi, alla grafica e alla gestione delle animazioni.
É bene sottolineare che nelle prime fasi dello sviluppo non si era in possesso di contenuti grafici e storici definitivi: per questo motivo, si è inizialmente lavorato con elementi fittizi che sono stati successivamente sostituiti ed adattati alle necessità definitive di progetto. Il percorso è stato scandito da una serie di incontri di dialogo con la responsabile curatrice della collezione papiri del Museo Egizio Susanne Töpfer e con Piera Luisolo, referente grafica del Museo. Questi momenti hanno reso possibile la comunicazione e il confronto tra reparti diversi quali informatico, grafico e storico.

4.1 Fase iniziale

4.1.1 Scelta dell'interazione

La prima grande difficoltà che si incontra nello sviluppo di un'applicazione touchless non è - come si potrebbe pensare - il funzionamento della tecnologia da utilizzare, quanto piuttosto lo studio e la scelta del tipo di interazione allo scopo di rendere l'esperienza dell'utente più fluida possibile. Infatti si tratta di una tecnologia che la maggior parte delle persone non ha mai avuto modo di utilizzare e quindi la sfida consiste nel trasmettere al visitatore indicazioni di comportamento, senza che il mezzo diventi una barriera nella fruizione.
Per questo motivo si è scelto di simulare il funzionamento del cursore del mouse sullo

schermo, controllato in questo caso dal movimento della mano a mezz'aria: l'utente ha così un riscontro immediato della posizione della sua mano nello spazio, rispetto alla posizione del cursore nello spazio finito dello schermo con cui si trova ad interagire.
Come illustrato nel capitolo precedente, la realizzazione di un sistema siffatto può avvenire tramite utilizzo di tecnologia touchless, quale Leap Motion Controller o la più recente Ultraleap 3Di. Nello specifico caso di studio, la scelta è ricaduta sulla prima: inizialmente il Museo proponeva come idea per l'installazione una proiezione su parete con cui interagire a distanza, e a questo scopo Ultraleap 3Di non sarebbe stata adatta poiché destinata ad essere applicata su schermi, motivo per cui l'opzione è stata scartata. Il sensore Leap Motion, invece, ha reso possibile la creazione di un sistema di interazione scalabile ed adattabile a seconda della variabile schermo/proiezione: è stato infine stabilito che per una prima apertura al pubblico - avvenuta il 27 settembre 2022 in corrispondenza del bicentenario della decifrazione dei geroglifici - l'installazione sarebbe stata proposta nella versione con schermo, posta in una sala temporanea fino al 21 novembre 2022[13] per poi essere spostata definitivamente nella nuova sala della scrittura - la cui apertura è prevista il 6 dicembre 2022 - sotto forma di proiezione interattiva.
Si è dedicata particolare attenzione alla tipologia di interazione, la cui scelta è stata subordinata anche all'analisi del possibile comportamento del pubblico: infatti giova ricordare che si sta trattando un tipo di istallazione il cui funzionamento touchless non può ad oggi essere dato per scontato, anche se, come documentato precedentemente, inizia ad essere presente in contesti museali internazionali. Gli schermi in particolare presentano il rischio di non riuscire a comunicare all'utente la possibilità di interazione, e a ciò si aggiunge la tendenza - condizionata dall'esperienza - a considerarli touchscreen. Da qui l'idea di distanziare l'utente dallo schermo, così come dalla proiezione, inserendo in loco una barriera fisica che lo aiuti a non cadere nell'errore di trovarsi di fronte ad un'esperienza touch. Si aggiunge quindi un ulteriore motivo di esclusione dell'Ultraleap 3Di, in quanto non consente una distanza dell'utente superiore al metro.

4.1.2 Studio delle fonti

Per prima cosa è stato necessario un esame approfondito del papiro. Sì è dunque studiata in particolare la struttura del documento contenente la lista dei Re, a partire dalla sua organizzazione originale dei contenuti. In particolare sono presenti undici colonne che, lette da destra verso sinistra, contengono i nomi dei faraoni che si sono susseguiti cronologicamente nei diversi periodi di regno.
Inizialmente non erano disponibili materiali informativi, quindi il grande lavoro è stato di ricerca e approfondimento per la comprensione dell'oggetto di studio e quindi su ciò che si sarebbe potuto rendere evidente ed interessante per la fruizione museale. Si nota inoltre che non si era in un primo tempo a conoscenza del numero di nomi di sovrani che, rispetto al presunto totale, sarebbe stato necessario mostrare e visualizzare.
Si è pertanto consultata la letteratura in merito a partire dagli studi effettuati sul papiro con il lavoro effettuato da Jean-François Champollion al più recente e attuale lavoro dell'egittologo dell'Università di Copenaghen, Kim Ryholt.

4.1.3 Primo storyboard

In questo paragrafo si mostra lo storyboard nella sua prima versione risalente ai primi giorni di luglio. Attraverso una suggestione grafica si mostra l'idea iniziale riguardo all'organizzazione dei contenuti dell'applicazione. Vediamo ora nel dettaglio le considerazioni fatte inizialmente, di cui verrà svelato il risultato finale nelle sezioni successive.
La prima immagine che l'utente vede è una *call to action*, la cui importanza è emersa fin da subito ed è dovuta alla necessità di mostrare al visitatore in modo chiaro l'azione da svolgere, come spiegato ad inizio capitolo (Figura 4.1).
A seguito della prima interazione l'utente si trova nella *homepage* contenente il *core concept* dell'installazione: la lista dei nomi dei faraoni (Figura 4.2).A questo punto, andando a selezionare un nome, è possibile entrare nella pagina dedicata ai dettagli riguardanti il faraone scelto (Figura 4.3).

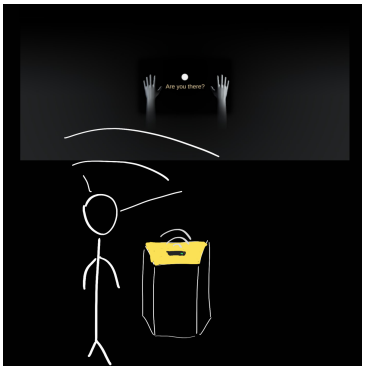


Figura 4.1

All'interno della pagina dedicata al faraone, le informazioni da mostrare sono: nome,

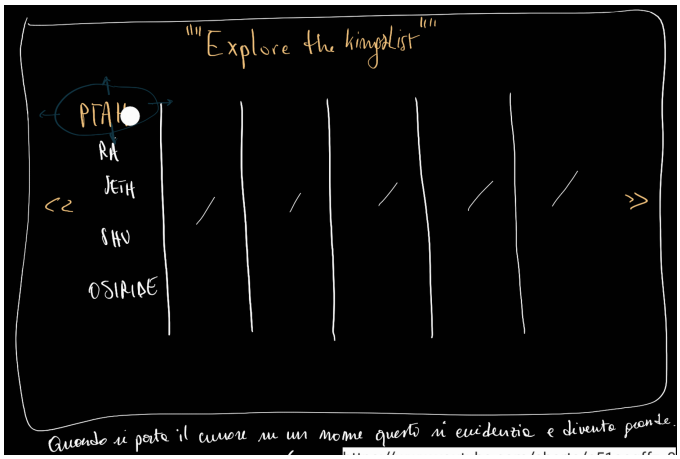


Figura 4.2

periodo, dinastia, cartiglio, presenza su altre liste, biografia e un'immagine.
A tale proposito si evidenzia il fatto che al fine di rispondere alle direttive del Museo, i testi devono essere sempre trascritti sia in lingua italiana che in lingua inglese e le due versioni devono essere sempre visibili, poste una sotto l'alta.
L'utente può a questo punto tornare indietro all'elenco completo. Inizialmente si prevedeva una *exit area*, nella parte bassa dello schermo, tramite la quale intuitivamente sarebbe stato possibile chiudere il pannello del Re per tornare al livello precedente. Successivamente è risultato superfluo dedicare un'area così ampia per tornare al livello precedente: data la precisione del sensore nel tracciare la posizione della mano e quindi del cursore sullo schermo, si è reputato sufficiente l'uso di normali bottoni per la navigazione tra le pagine.

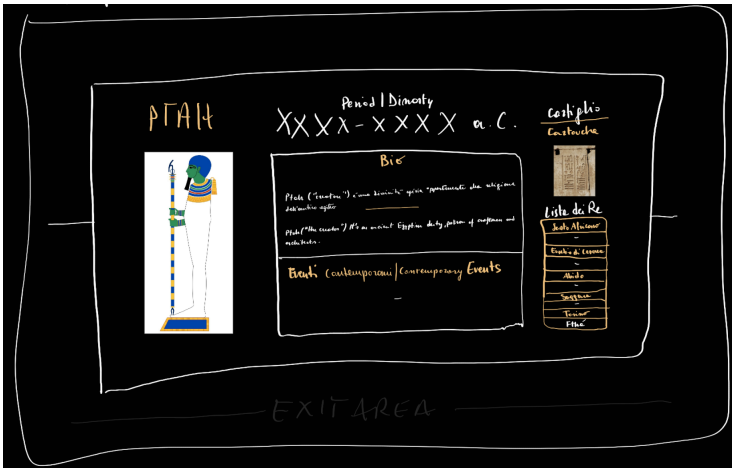


Figura 4.3

Quello finora descritto è il meccanismo cuore dell'applicazione e per questo motivo è stato il punto di partenza sia nell'ideazione che nella creazione dell'applicazione.

4.1.4 Software: Unity e TouchFree

Ambiente di sviluppo

Come ambiente di sviluppo si è utilizzato Unity, uno tra i motori grafici più usati per la realizzazione di videogiochi e di contenuti interattivi. Può essere installato su un computer con sistema operativo Windows o macOS ed è considerato multipiattaforma, in quanto permette di realizzare un progetto che potrà poi essere fruito su dispositivi diversi per ambiente e/o circuito. All'interno di Unity è possibile gestire applicazioni che spaziano da interfacce interattive a videogiochi 3D complessi. Le interazioni e tutto ciò che deve essere gestito via script è sviluppato utilizzando due linguaggi di programmazione: Javascript e C#.

TouchFree Tooling per Unity

Come si è descritto nel terzo capitolo (3.2.3), Ultraleap ha recentemente rilasciato il software TouchFree per semplificare la trasformazione di schermi interattivi *touch* in *touchless*. Per sfruttare le sue potenzialità direttamente su Unity, mette a disposizione un *client* che, una volta inserito all'interno dell'ambiente di sviluppo dell'applicazione, permette di ricevere facilmente i dati di *tracking* della mano generando un cursore a schermo controllabile senza tocco.

Impostazione del progetto Unity con TouchFree

Come prima operazione, è necessario installare *TouchFree Tooling for Unity*[17] sul PC: si tratta di un file *.unitypackage* pronto per essere importato su Unity. Il suo funzionamento

è subordinato alla presenza di un sensore di tracciamento, collegato allo stesso computer, nonché del software TouchFree stesso. É bene assicurarsi dunque che il servizio *TouchFree* sia in esecuzione e, accedendo alle impostazioni, se ne effettui quindi la configurazione. A questo punto, con sensore Leap Motion alla mano, il lavoro inizia con la creazione di un progetto Unity3D.
Si è deciso di lavorare su un'unica scena, in cui l'elemento chiave, gerarchicamente padre di tutti i contenuti, è il *gameobject Canvas* (Figura 4.4). Esso viene settato per una riso-

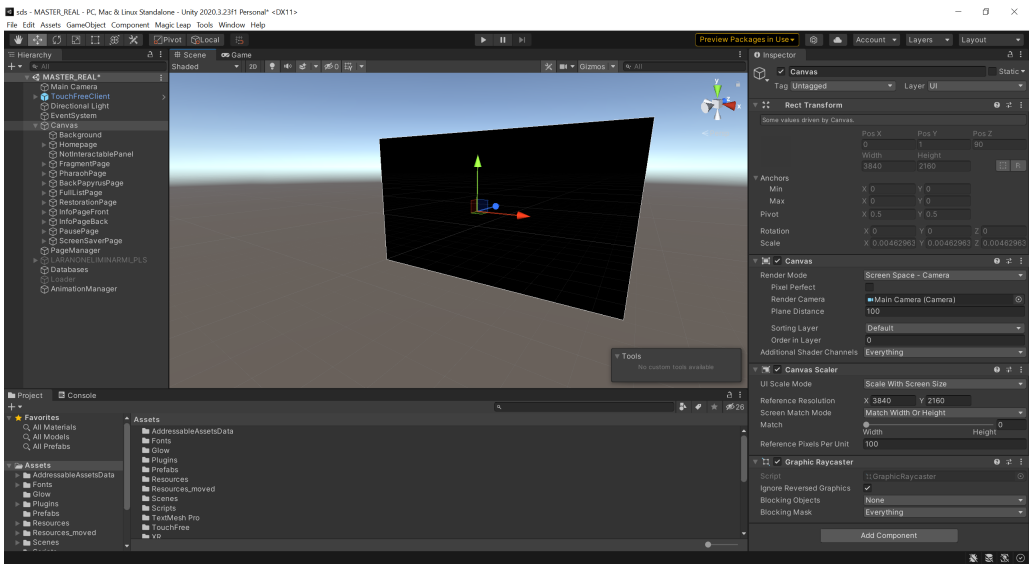


Figura 4.4: Caption

luzione schermo di 3840 x 2160 pixel ed è lo spazio finito entro il quale saranno mostrati tutti i contenuti.
Per importare il file *.unitypackage* di TouchFree nel progetto Unity si può trascinare il file nel corrispondente editor. All'interno del pacchetto è presente una scena di esempio, grazie alla quale è possibile testare la corretta comunicazione tra *TouchFree Tooling* e *Service* e comprendere il procedimento di integrazione. Una volta importato il pacchetto, la scena in questione si trova nel percorso *TouchFree/Tooling/Examples* degli Assets.
L'elemento principale per il funzionamento del sistema è il *prefab TouchFree Client* che si trova all'interno della suddetta scena. Per importarlo all'interno della nostra scena creata ex novo, lo si recupera nel percorso *TouchFree/Tooling/Prefabs* e semplicemente lo si aggiunge all'interno della gerarchia parallelamente al *Canvas* precedentemente creato. Come visibile nell'*inspector* dell'oggetto (Figura 4.5), sono presenti gli script fondamentali alla connessione che rendono possibile lo scambio dei dati: *Connection Manager* gestisce la connessione con il Server; *Message Receiver* si occupa della ricezione dei messaggi dal *Service* in modo ordinato e ne distribuisce i risultati ai rispettivi gestori; *InputActionManager*, gestore di tutte le *<InputActions>*, recupera i dati ricevuti dall'*<InputActionPlugins>* e li trasmette tramite l'evento *<TransmitInputAction>* che dovrebbe essere osservato da qualsiasi classe che voglia utilizzare le *<InputActions>* in arrivo; *UnityUIInputController*

fornisce l'input dell'interfaccia utente di Unity in base ai dati in arrivo dal servizio di *TouchFree* attraverso una *<ServiceConnection>*; *CameraConnector* garantisce che tutti i canvas utilizzati da *TouchFree Tooling* facciano riferimento all'oggetto camera corretto per il rendering. A questo punto la scena è correttamente impostata ed aggiungendo ele-

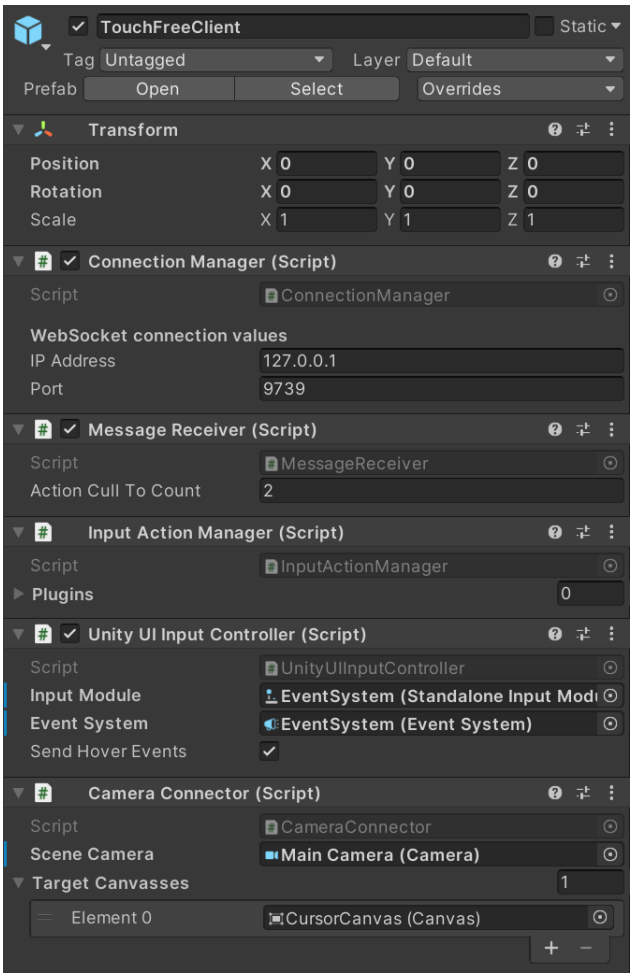


Figura 4.5: TouchFree Client Prefab - componenti

menti UI all'interno del *Canvas* sarà possibile interagire, oltre che con il mouse, anche in modalità touchless con il sensore Leap Motion. Il cursore che comparirà sullo schermo è gestibile all'interno di Unity: gerarchicamente figlio del *client*, il prefab *RingCursor* è accessibile e personalizzabile nelle sue parti (Figura 4.6).

Configurazione

Dalle impostazioni di *TouchFree* si accede alla sezione di configurazione del sensore (Figura 4.7) tramite la quale si determina una corrispondenza tra posizione del sensore, spazio di interazione e spazio dello schermo. Si esegue quindi, con il sensore collegato al computer,

Figura 4.6: TouchFree Client

il *Quick Setup*, che permette la configurazione tramite brevi e semplici passaggi.

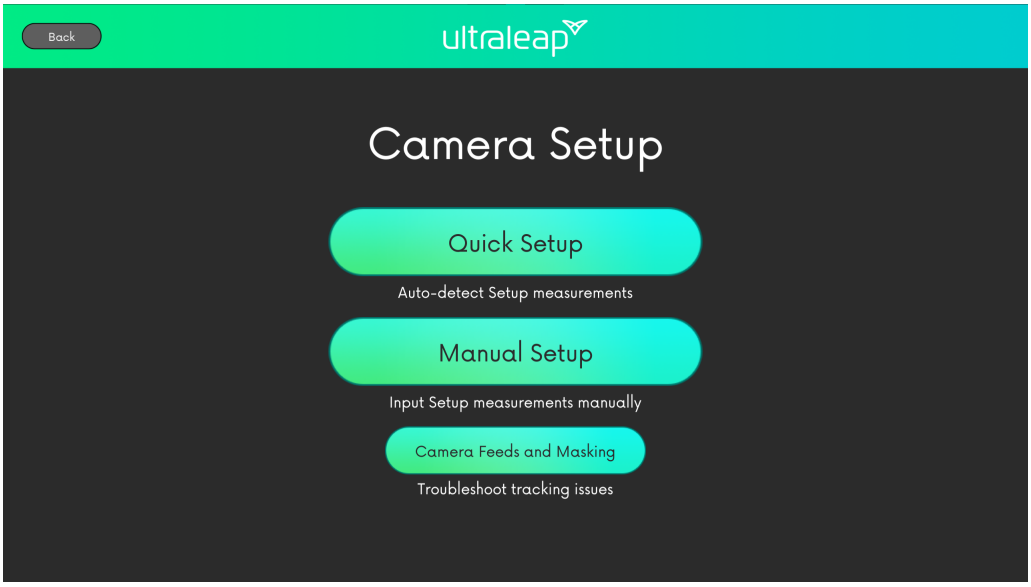


Figura 4.7

Come primo passaggio si indica la posizione (Figura 4.8) di montaggio del sensore ovvero se si trova sopra o sotto il display fisico, e quindi - tramite procedura guidata - si indicano due punti di riferimento a mezz'aria che segnalano al software dove si trovano i limiti dell'area di interazione "virtuale" (Figura 4.9). Nel nostro caso si definisce uno spazio di interazione in prospettiva rispetto all'effettiva dimensione dello schermo (si veda Figura 4.10), come se fosse una versione ridotta del display fisico. Una configurazione di questo tipo permette di interagire con schermi e proiezioni di grandi dimensioni ad una certa distanza rispetto ad essi. Il setup può essere effettuato tutte le volte in cui si renda necessario.La configurazione finale sarà comunque effettuata in sede presso il museo con installazione e postazione definitiva.
Attraverso le impostazioni è altresì possibile accedere alla gestione semplificata dell' *"Appearance"* relativa al cursore (Figura 4.11), ovvero le sue caratteristiche estetiche, quali dimensione e colore, e alle impostazioni di interazione, dove si definisce la modalità di interazione e

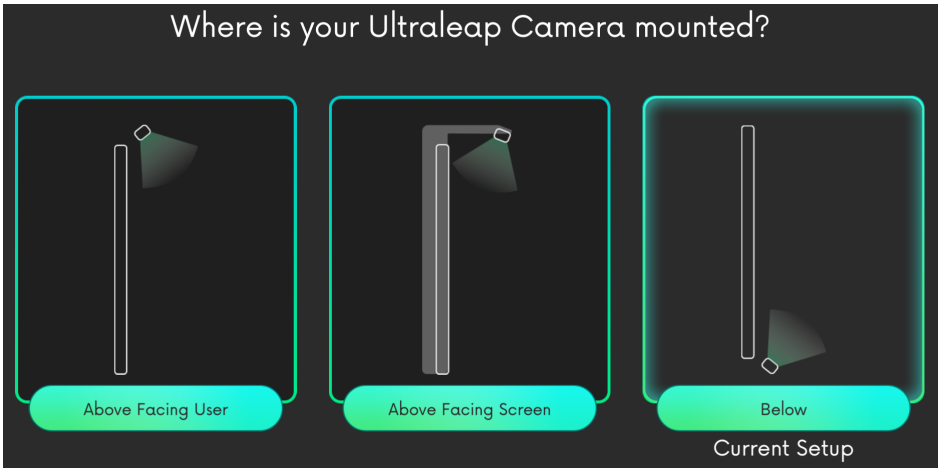


Figura 4.8

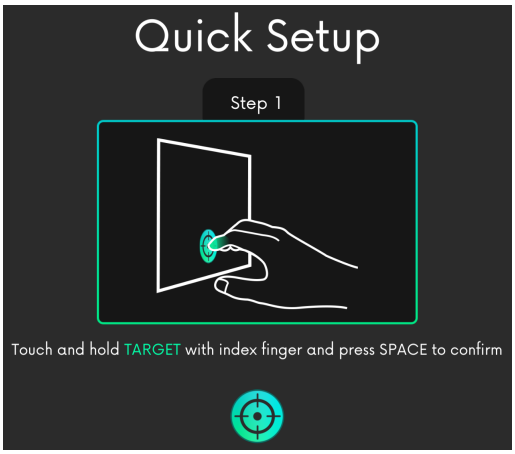


Figura 4.9

se ne regolano le specifiche. (Figura 4.12). I tipi di interazione possibili sono tre:

- Air Push: per attivare l'evento click, l'utente muove la mano verso lo schermo, come se dovesse toccarlo; l'interazione può avvenire a qualsiasi distanza dallo schermo purchè all'interno del campo visivo della camera.

- Hover & Hold: per attivare l'evento di click, l'utente deve tenere il cursore fermo per un breve periodo di tempo.

- Touch Plane: l'utente interagisce toccando un piano virtuale invisibile davanti allo schermo; l'evento di click è attivato quando la mano dell'utente si muove attraverso tale piano; l'interazione deve avvenire ad una distanza fissa dallo schermo (regolabile dalle impostazioni).

Figura 4.10: Large screen setup

Figura 4.11

Si è scelta la modalità di interazione *Hover & Hold* poichè, dopo alcuni test con utenti "nuovi" alla tecnologia, è risultata la modalità più intuitiva, mentre le altre si sono rivelate di complessa interpretazione generando frustrazione e disinteresse nell'utente. Si tratta infatti di un'esperienza più lenta e per questo capace di evitare situazioni di selezione errata, mancata o non voluta. Ad anticipare l'evento di click, il cursore mostra una breve animazione temporizzata. Il tempo prima dell'attivazione di tale evento può essere regolato in questa schermata delle impostazioni *(Complete Time)*, così come il tempo necessario perchè l'interazione abbia inizio *(Start Time)* dal momento in cui la mano si ferma in una posizione e, ancora, la precisione di rilevamento del movimento da parte del sensore *(Cursor Movement)*

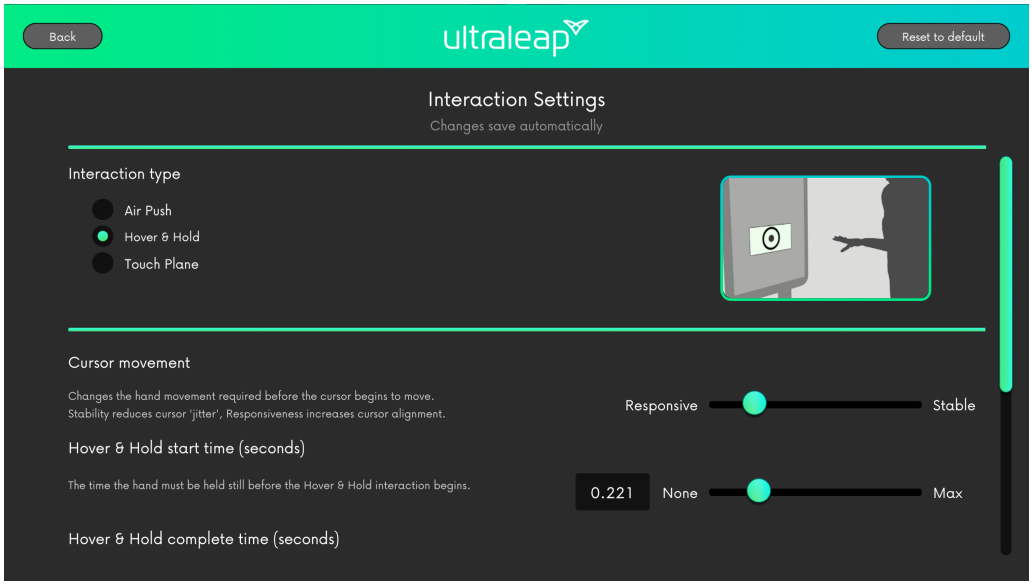


Figura 4.12

4.2 Organizzazione contenuti e user experience

Nella presente sezione si illustrerà la realizzazione dell'applicazione attraverso le fasi del suo sviluppo pratico. Di seguito in figura il diagramma della struttura dell'applicazione.

4.2.1 Struttura modulare dell'applicazione

Per struttura modulare si intende lo sviluppo di un sistema che non sia costruito ad hoc per un certo tipo di contenuti immutabili, ma possa essere implementato e modificato a piacimento senza dover cambiare radicalmente il codice. Oltre ad essere un paradigma di programmazione sempre preferibile nello sviluppo di progetti di grandi dimensioni, nel caso specifico questo approccio ha permesso di iniziare a lavorare per poi testare e modificare in corso d'opera non solo i contenuti, ma la struttura stessa dell'applicazione.

4.2.2 Creazione di un database e importazione su Unity

La creazione di un database è stata necessaria per la gestione di nomi e informazioni su ogni faraone. In concreto i dati sono stati caricati su un file excel organizzato per righe corrispondenti a ciascun faraone e per colonne contenenti informazioni riguardo nome del periodo, nome della dinastia, periodo di regno, nome del faraone, altre liste antiche in cui ricorre, biografia, frammento del papiro in cui si trova, periodo del frammento del papiro in cui si trova ed infine eventuale altro nome del faraone. Inizialmente riempito con dati fittizi, il foglio di calcolo esportato in .csv (valori separati da virgole) può essere importato su Unity.
La lettura e il caricamento dei dati contenuti nel file *.csv* sono gestite tramite due script:

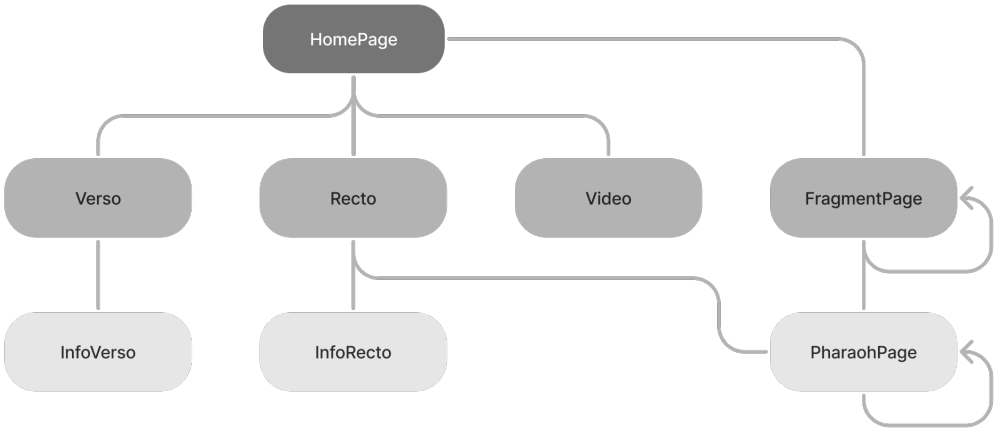


Figura 4.13: diagramma dell'applicazione

CSVReader e Load Excel. Nella classe CSV, si definisce la funzione *Read*, metodo statico e pubblico che riceve come parametro il percorso del file *csv* da leggere e restituisce una lista di tipo `List<Disctionary<string, object>`.
Per la gestione degli elementi contenuti all'interno del database, si crea una nuova classe chiamata *Pharaoh* all'interno della quale si definiscono le variabili che andranno a caratterizzare ogni elemento, corrispondenti a quelle associate ai valori del foglio di calcolo creato.

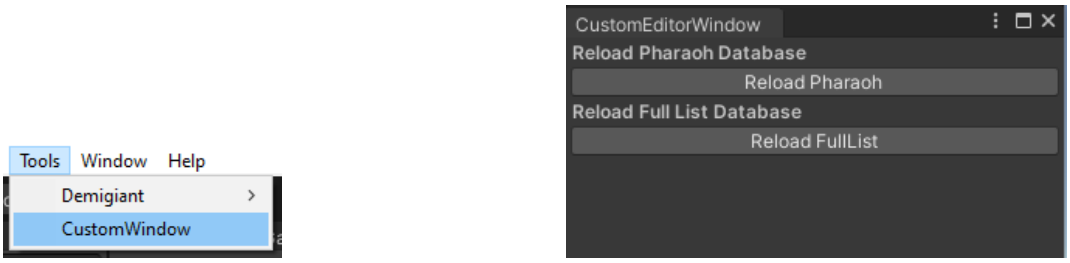
Pharaoh Class

La classe dell'elemento faraone è un concetto fondamentale per l'intera applicazione. Grazie alla classe è possibile avere un oggetto *Pharaoh* contraddistinto da una serie di variabili che vengono definite proprio all'interno di essa e assegnate in fase di caricamento del *database*. Ogni faraone possiede:

- *PharaohID*: codice alfanumerico che lo identifica univocamente
- *Period Name*: nome del periodo
- *Dinasty Name*: nome della dinastia
- *Reign Period*: periodo di regno espresso in date numeriche
- *Lists*: nomi di altre liste dei Re in cui compare il faraone in questione

- *Bio*: biografia
- *Other Name*: eventuale altro nome attribuito al faraone

Per quanto riguarda *Period Name*, *Dinasty Name* e *Bio* si sono dedicate due variabili distinte per la rispettiva trascrizione del testo in italiano e in inglese. Oltre alle variabili sopra citate, a completare la suddetta classe concorrono ancora tre variabili di tipo *Sprite*, i cui valori si assegnano in fase di *Start* dell'applicazione all'interno dello script di gestione generale, *PageManager*, di cui si tratterà nel dettaglio più avanti. Le tre immagini richieste dalle tre variabili *Sprite* corrispondono a cartiglio, faraone e *highlight* del nome del faraone sul frammento del papiro.
Tornando a `LoadExcel`, qui si definiscono i metodi che, quando chiamati, azionano l'importazione del *database*. In particolare `LoadPharaohData` è la funzione principale: al suo interno, come prima operazione dopo il `Clear` della variabile a cui è assegnato il database importato `(pharaohsDatabase)`, `CSVRead("filename")` legge e salva in una lista i dati di `filename`. A questo punto, si itera la lista e, associando ad ogni elemento i-esimo la classe faraoni e i corrispondenti sotto valori, si va a riempire `pharaohDatabase`. Alla fine di questo processo, l'intero database è caricato all'interno del nostro progetto Unity.
Lo script `LoadExcel` è componente di un *empty Game Object* presente in scena, tramite il quale si accede al metodo `LoadPharaohData` e al database stesso. La chiamata della funzione è gestita tramite il `CustomEditorWindow`, script grazie al quale è possibile personalizzare elementi di menu di Unity: in questo caso si è aggiunto l'elemento *CustomWindow* nel menu *Tools* (Figura 4.14a) tramite il quale si accede alla finestra (Figura 4.14b) contenente l'elemento `GUILayout.Button`. Quest'ultimo è programmato per recuperare il componente LoadExcel in scena e chiamare quindi la funzione `LoadPharaohData` attivando il processo sopra descritto.



(a) Tools menu (b) Custom window

Figura 4.14

Risulta necessario sottolineare a questo punto che riguardo al papiro vi è una distinzione tra i nomi dei faraoni che sono stati "semplicemente" riconosciuti e quelli di cui invece si sa abbastanza da dedicarvi una sezione. Questa distinzione ha portato alla creazione di due database distinti, il secondo dei quali - che chiameremo *"FullListDatabase"* - contiene semplicemente una lista completa di nomi. Il procedimento di caricamento su Unity è il medesimo, ma viene salvato in una lista di oggetti di tipo *String*, a differenza di *pharaohsDatabase*, dove gli elementi della lista sono oggetti di classe *Pharaoh*.

Si è scelto di utilizzare due database a fronte della necessità di frequenti modifiche durante la fase di sviluppo della struttura dei database stessi rispetto ai contenuti.

4.2.3 Page Manager

L'applicazione è strutturata a partire da *Canvas*, *Game Object* caratterizzato dalla presenza del componente *Canvas*, che definisce l'area all'interno del quale si inseriscono tutti gli elementi UI. Le pagine principali dell'applicazione, organizzate in *Homepage, Fragment Page, Pharaoh Page, Full List Page, BackPapyrus Page* e *Restorartion Page* sono *Panels*, oggetti a loro volta gerarchicamente padri di tutti gli elementi UI che li compongono. La gestione della navigazione tra le diverse pagine e il "riempimento" con i contenuti corretti, avviene a partire da *PageManager*, *Empty Game Object* presente in scena a cui è legato l'omonimo script.
All'interno dello script avviene l'inizializzazione dell'applicazione e dei contenuti principali.
Di seguito si dà spazio alla spiegazione della struttura delle diverse pagine.

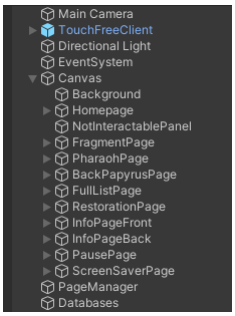


Figura 4.15

4.2.4 Homepage

Nella *Homepage* vengono visualizzati i frammenti del papiro, la cui disposizione definitiva è stata resa nota a fine del mese di agosto a seguito dell'ultimo lavoro di restauro effettuato sul documento (si veda la sezione 2.3). Il lavoro è stato quindi inizialmente effettuato a partire da un *facsimile* di frammenti (Figura 4.16). In questa pagina l'utente finale ha

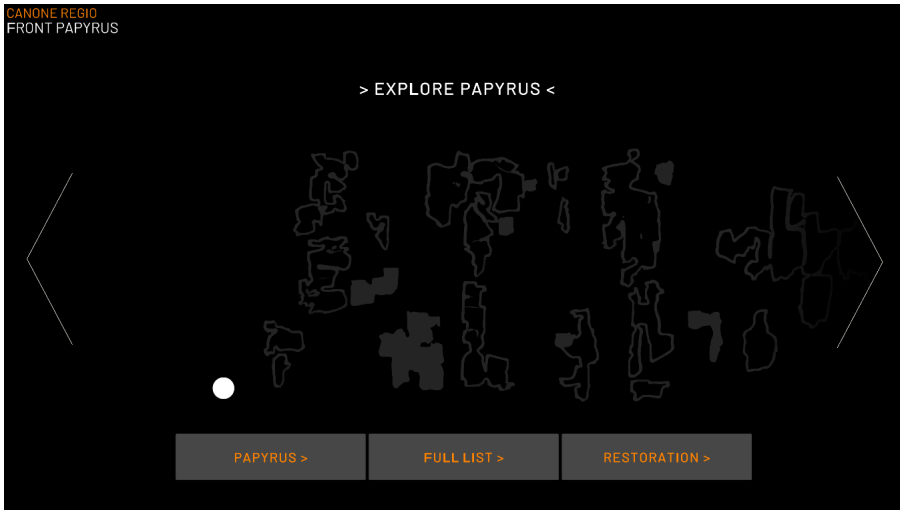


Figura 4.16

più possibilità di comportamento, tra cui scorrere la Lista dei Re nella lunghezza del papiro o scegliere un frammento in particolare. Altrimenti, utilizzando i rispettivi *UI Buttons*, può esplorare recto e verso del papiro oppure la pagina dedicata al restauro. I frammenti si distinguono tra cliccabili e non, a seconda che contengano o meno nomi di faraoni appartenenti a *PharaohsDatabase*. A livello di componenti ciò che li distingue è la presenza nel primo del componente *Button* - che, oltre a renderlo cliccabile, trasmette al *gameobject* tutte caratteristiche di comportamento tipiche di un bottone e del *Fragment Script* di classe Fragment.

Fragment Class

Questa è la classe che caratterizza ogni frammento cliccabile ovvero che - come descritto precedentemente - contiene almeno un faraone di classe Pharaoh. Ciò è dovuto al fatto che la lista completa dei nomi leggibili sul papiro comprende anche tutti quei faraoni di cui non si sa (quasi) nulla e di cui quindi non ci può essere, al momento, un ulteriore livello di approfondimento.
Quindi se un frammento contiene uno o più nomi appartenenti al *PharaohsDatabase*, è anche caratterizzato dalla classe *Fragment* definita all'interno del `FragmentScript`. Grazie alla suddetta classe, ogni frammento che risponde alle caratteristiche citate poco innanzi porta con sé un numero identificativo *(fragmentID)*, una lista di elementi di classe *Pharaoh* dei faraoni che contiene e la variabile *Sprite* all'interno del quale viene salvata l'immagine

del frammento stesso.
Partendo dal presupposto che nel database ogni faraone possiede un campo in cui si specifica il *fragmentID* del frammento in cui è contenuto, il meccanismo è il seguente.
Nel database

- ogni riga e dunque ogni faraone possiede anche il campo *fragmentID*
- i *fragmentID* sono gli stessi per faraoni che si trovano fisicamente sullo stesso frammento.

In scena

- si creano e dispongono tanti elementi *UI Buttons* quanti sono i frammenti *cliccabili* del papiro
- si aggiunge a ciascuno di essi il *FragmentScript*
- li si ordina e li si rende gerarchicamente figli di uno stesso *empty* (`fragmentButtons`).

Nella fase di inizializzazione (in Start() di PageManager)

- si scorrono gli elementi figli di *FragmentButtons*
- si assegna ad ognuno un *fragmentID* (parametro della classe *Fragment*) in modo incrementale, avvalendosi della disposizione ordinata dei figli
- si itera il database dei faraoni
- se verifica se il campo *fragmentID* del faraone corrisponde all'*id* assegnato al frammento in scena e in caso positivo si aggiorna la lista di faraoni del frammento, aggiungendovi il parametro di tipo *Pharaoh*.

All'interno di questo processo iterativo di lettura di `pharaohsDatabase` vengono effettuate altre operazioni quali l'assegnazione delle immagini relative ai singoli faraoni:

- immagine identificativa del faraone, specifica di ciascuna divinità (Re del periodo Mitologico) e condivisa per periodo nel caso degli altri Re
- cartiglio del nome
- immagine del frammento del papiro con il nome del faraone evidenziato.

Si descrive quindi il procedimento generale di assegnazione dinamica delle immagini.

Meccanismo di assegnazione del riferimento delle immagini

Si prende a titolo d'esempio il meccanismo di assegnazione dell'immagine relativa al cartiglio del nome di ogni faraone. Le immagini di tutti i cartigli vengono inserite manualmente in un array definito all'interno di *LoadImage script*, componente del *gameobject* di tipo *UI* destinato a contenere la figura in scena (Figura 4.18). Trattandosi di un'immagine specifica di ogni elemento di tipo *Pharaoh*, si aggiunge un parametro a `PharaohsDatabase`

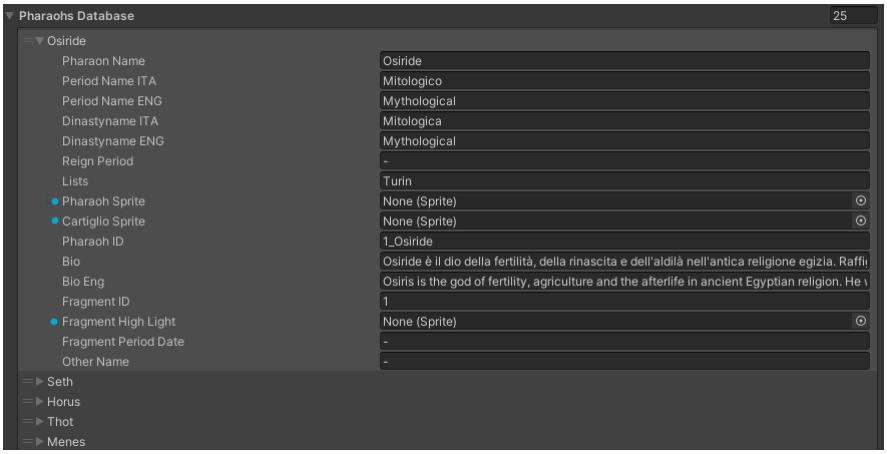


Figura 4.17

destinato a contenere il riferimento (Figura 4.17). In questo modo, l'assegnazione avverrà solamente una volta, che in questo caso si fa coincidere con l'avvio dell'applicazione, inserendola, come sopra descritto, tra le operazioni di *Start* di *PageManager*.
L'assegnazione avviene tramite confronto tra i nomi dei file inseriti nell'array e l'*id* del faraone. Se questi coincidono, il database viene aggiornato aggiungendo - per quel faraone - il riferimento al cartiglio corretto. In questo modo, per tutta la fase di utilizzo dell'applicazione, fino a un successivo riavvio, immagine e faraone saranno univocamente collegati semplificando il passaggio tra immagini diverse in funzione dei diversi faraoni. All'interno

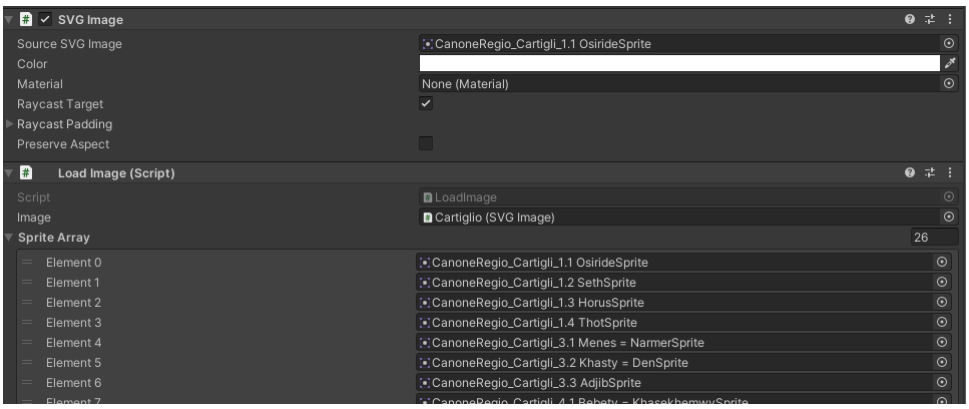


Figura 4.18

di `LoadImage` è anche presente una funzione adibita a recuperare direttamente dall'array - tramite indice - una specifica immagine. L'implementazione di ciò ha reso possibile recuperare in modo semplice e dinamico immagini non direttamente dipendenti dal faraone, come ad esempio nel caso dei diversi frammenti. Il recupero dell'indice corretto è facile, se l'array è ordinato rispetto ai valori del *fragmentID*: conoscendo il frammento corrente (il suo *id*), non solo sarà possibile accedere a tutti i relativi valori precedentemente citati,

ma se ne può utilizzare l'id come indice per recuperare l'immagine corretta.
Ne consegue l'importanza di avere una variabile globale sempre aggiornata rispetto a frammento e faraone da visualizzare. I valori di queste variabili vengono dinamicamente assegnati in base alle scelte dell'utente in fase di utilizzo dell'applicazione e contengono il valore relativo all'*id* identificativo di frammento e faraone selezionato.
La scelta di un frammento dalla *homepage* determina il passaggio a *Fragment Page*.

4.2.5 Fragment Page

In questa pagina si vogliono visualizzare il frammento selezionato ed i nomi presenti su di esso. Per fare ciò - nell'intento di creare una struttura modulare - si è dovuto tener conto di una serie di aspetti. Prima di tutto, il numero di nomi presente all'interno di un frammento non è fisso e dipende dal frammento stesso. Si vuole inoltre creare una corrispondenza visiva tra scritta su papiro e nome e, infine, abilitare il passaggio al livello di approfondimento successivo nonché alla scheda di uno specifico faraone. Si intende dunque ottenere una pagina che contenga l'immagine di un frammento, i singoli nomi evidenziati su di essa e tutti i nomi contenuti che - se cliccati - portano alla visualizzazione di una nuova pagina contenente le relative informazioni.

Figura 4.19: materiali provvisori

A partire dall'array di nomi (si veda la sezione Fragment Class) legati ad uno specifico frammento se ne conosce la lunghezza e quindi il numero di elementi di cui è composto. Per ogni elemento dell'array del frammento corrente, viene quindi creato un bottone a cui si assegna un determinato comportamento. I bottoni vengono creati dinamicamente nel momento in cui il "frammento corrente" si aggiorna: il prefab di tipo *Button* precedentemente creato (con annesso il componente *Pharaoh Script*) viene istanziato per ogni faraone presente nella lista del frammento. In ogni bottone si memorizzano nome e relativi parametri della classe *Pharaoh* e vi si aggiungono dei `listener` tramite `Event Trigger` di tipo `PointerClick`, per il passaggio alla scheda dettagliata, e di tipo `PointerEnter e Exit`, responsabile invece del comportamento nella situazione di *Hover* del cursore, per svelare la posizione del corrispondente nome sul papiro (Figura 4.19).
Ricapitolando, dalla *Homepage* la scelta di un frammento attiva la funzione `GoToSelectedFragment(FragmentScript fragment)`, a cui trasmette i dati relativi a se stesso. Il metodo permette il passaggio a *Fragment Page* eseguendo le seguenti operazioni:

- aggiorna la variabile globale che contiene l'*id* del frammento corrente
- aggiorna la foto del frammento
- chiama la funzione `LoadPhaoraohButtons(thisFragment)`, che si occupa di istanziare un bottone per ogni faraone contenuto in thisFragment:
  - distrugge, se presenti, bottoni già creati

- itera l'array contenente i faraoni del frammento.

  e per ogni faraone:

  * istanzia il prefab `pharaohButtonPrefab` (gerarchicamente figlio di `pharaohButtonsObj`, empty già presente in scena)
  * assegna i valori del faraone, copiandoli nei campi della classe *Pharaoh*
  * assegna il *pharaohName* al testo del bottone
  * aggiunge un listener onClick che aggiorna il faraone corrente con quello legato a se stesso e porta a *Pharaoh Page*, chiamando `GoToSelectedPharaoh(Pharaoh pharaoh)`
  * aggiunge un listener onEnter che, al passaggio del cursore sopra il nome, attiva l'immagine che evidenzia la corrispondente scritta sul papiro
  * aggiunge un listener onExit che fa tornare l'immagine base del frammento senza nome in evidenza.

Infine è presente un bottone *Home* grazie al quale è possibile tornare al livello precedente, ovvero la *Homepage*.

4.2.6 Full List Page (Verso)

Il *verso* del papiro corrisponde al lato visibile nella Homepage, ovvero il retro del papiro che contiene la Lista dei Re. In questa sezione, si vuole mostrare l'intera lista dei nomi rinvenuti grazie agli studi e ai restauri effettuati negli anni. Qui troveremo dunque anche quei faraoni di cui non si conosce altro che il nome. Si è dunque pensato di realizzare una lista percorribile orizzontalmente e composta da `PharaohButtons` (le cui caratteristiche si sono descritte nel paragrafo precedente) e ulteriori bottoni, non cliccabili, di coloro che fanno parte della lista, ma di cui non si ha un approfondimento. In questo modo si rende consultabile l'elenco nella sua totalità, permettendo però l'accesso al livello di approfondimento ulteriore per coloro che lo possiedono.

Figura 4.20

Anche in questo caso i bottoni vengono generati dinamicamente, così che non ci siano complicazioni nel momento in cui i database di riferimento siano modificati: questa operazione viene effettuata in fase di avvio dell'applicazione. Il database di riferimento è `FullListDatabase` che, come si è descritto precedentemente, contiene l'elenco completo di nomi sotto forma di stringhe. Il procedimento si basa sul confronto di esso con `pharaohDatabase`: quando si incorre in una corrispondenza tra i due si istanzia un `pharaohButton` nello stesso modo descritto in *Fragment Page*, mentre per tutti i valori con mancata uguaglianza si istanzia un `notInteractableButtonPrefab`, gameobject con le stesse caratteristiche degli altri bottoni, ma disabilitato e quindi non interagibile. Passando con il cursore sopra di essi non succede nulla, mentre per quelli cliccabili il nome si ingrandisce e cambia colore. In figura 4.20 i tre casi: Sneferu e Netjerikara sono disabilitati e dunque non cliccabili, mentre Menkahor, Djedu e Unas sì; Djedu si trova nella situazione in cui il cursore è posizionato

su di esso.
L'insieme di bottoni è organizzato in scena grazie al *panel* padre, che con il componente *Grid Layout Group* li distribuisce all'interno di una griglia con numero prestabilito di righe. Essendo la lista potenzialmente molto lunga, si inseriscono delle frecce che, una volta definita la porzione da mostrare, permettono di spostare orizzontalmente il *panel* svelando i nomi "nascosti". Il funzionamento delle frecce è gestito dallo script `FullListArrowButtonManager`.
Sono infine presenti due bottoni *Home* e *Info* che permettono rispettivamente di tornare alla *Homepage* ed accedere a una sezione di approfondimento riguardo il *verso* del papiro: *InfoVerso Page*, di cui si tratterà nel dettaglio nella sezione dedicata.

4.2.7 Pharaoh Page

In questa pagina si approfondisce il singolo faraone riportandone biografia, immagine, informazioni riguardo al periodo di regno, dinastia di appartenenza, altre liste in cui compare e cartiglio del nome. Si è inoltre deciso di aggiungere una corrispondenza visiva del frammento di appartenenza rispetto all'intero papiro (Figura 4.21). I dati sono prelevati dal database e visualizzati in modo organizzato nella schermata. Come da linee guida dettate dal Museo, ogni testo italiano deve essere accompagnato dalla traduzione inglese, entrambi sempre visibili. I testi, come si è descritto nella sezione *Creazione di un data-*

Figura 4.21

base, vengono inseriti all'interno del database e quindi associati ad ogni specifico faraone utilizzando la classe *Pharaoh*. Essi sono dunque modificabili esternamente a Unity e possono essere aggiornati ogni qualvolta necessario, anche in futuro grazie a nuovi eventuali approfondimenti e informazioni derivanti da scoperte o studi archeologici. L'interfaccia utente di questa pagina deve essere quindi organizzata in modo da poter contenere le diverse informazioni variabili di faraone in faraone. La biografia, che rappresenta la variabile testo più incerta come lunghezza e come varianza, si inserisce in due elementi UI di testo ordinati verticalmente, uno per il testo italiano e uno per quello inglese. Il comportamento che si vuole ottenere è che la casella si adatti alla quantità di testo ed influenzi quindi la posizione della seconda. BioArea (Figura 4.22) è un *empty* con *"Vertical Layout Group"*, componente che permette di organizzarne i figli secondo un layout verticale. Per ottenere quanto detto riguardo le dimensioni del "contenitore" del testo, si aggiunge ad entrambi il componente *Content Size Fitter* con Vertical Fit di tipo *Preferred Sized*: in questo modo lo spazio occupato dal testo, che si manterrà della stessa dimensione per quanto riguarda il font, determinerà la grandezza della cella che lo contiene, effettuando un *override* della dimensione stabilita nel Vertical Layout Group del padre.
Si illustra a questo punto come si comunica al sistema lo specifico faraone e i relativi

contenuti che devono essere mostrati.
Dopo aver predisposto all'interno della scena gli elementi che andranno "riempiti" con i dati, l'operazione principale è rendere noto al sistema da dove recuperarli: la variabile globale che definisce il faraone corrente è `currentSelectedPharaoh`, che contiene appunto il riferimento dell'ultimo faraone selezionato, quello di cui si devono mostrare i dettagli. La variabile in questione viene tenuta aggiornata grazie a una funzione che ne ridefinisce il valore ogni qualvolta `GoToSelectedPharaoh(pharaoh)` venga chiamata.
Come visibile dal diagramma (Figura 4.13) Pharaoh Page si raggiunge da:

- Fragment Page
- FullList Page
- Pharaoh Page stessa

É importante tenere in memoria la pagina di provenienza per un funzionamento coerente del *back*, bottone che permette di tornare alla pagina precedente. Per questo motivo, ma - come vedremo - non l'unico, è bene distinguere l'arrivo da Fragment Page piuttosto che da FullList Page. Si definisce quindi un metodo che aggiorna il valore di una variabile booleana globale utilizzata a tale scopo. Il metodo, `ItsFromFullList(bool boolean)` viene chiamato `onClick` grazie ad un listener aggiunto ai `pharaohButtons` nelle fasi precedentemente descritte di generazione degli stessi. Il parametro `boolean` inserito è pari a `false` per quanto riguarda i bottoni generati in Fragment Page, mentre a `true` per quelli di FullList Page. In questo modo, alla selezione di un faraone corrisponde anche l'aggiornamento di tale variabile e da Pharaoh Page basterà controllarne il valore per conoscere la pagina temporalmente precedente. Il controllo avviene nella funzione chiamata al click del tasto *back*.

4.2.8 BackPapyrus Page (Recto)

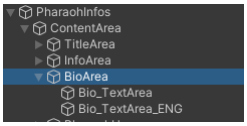


Figura 4.22

Il recto del papiro, che contiene il registro tributario, non è visibile in esposizione al museo per l'importanza che riveste l'altro lato (il verso), e rimane così nascosto. Grazie a questa sezione si dà la possibilità al visitatore di osservarlo nella sua versione digitalizzata. Si tratta semplicemente di un pannello contenente l'immagine che - tramite frecce laterali - l'utente può scorrere orizzontalmente. Da qui si può inoltre accedere alla sezione info che, come si vedrà nel dettaglio più avanti, contiene informazioni generali su questo lato del papiro. Il movimento del pannello contenente l'immagine del papiro è gestito tramite script come si vedrà nel dettaglio nella sezione 4.2.12.

4.2.9 Restoration Page (Video)

Si inserisce all'interno dell'applicazione un video (senza audio) riguardo l'ultimo lavoro di restauro effettuato sul papiro (Figura 4.24). In una pagina accessibile a partire dalla *homepage* si aggiunge il `Video Player` in scena (Figura 4.23). Il video viene azionato e

fermato via script in corrispondenza dell'entrata e uscita dalla pagina tramite i comandi `videoPlayer.Play()` `videoPlayer.Stop()`. In questo modo il video partirà dall'inizio ogni volta che si accede alla pagina in questione. Grazie alla spunta *Loop* nell'*Inspector*, una volta terminato, inizia nuovamente.
Per mostrare il progresso del video si era inizialmente inserita una *slidebar* il cui riempimento era controllato via script, ma per motivi grafici è stata rimossa e inserita direttamente all'interno del video.

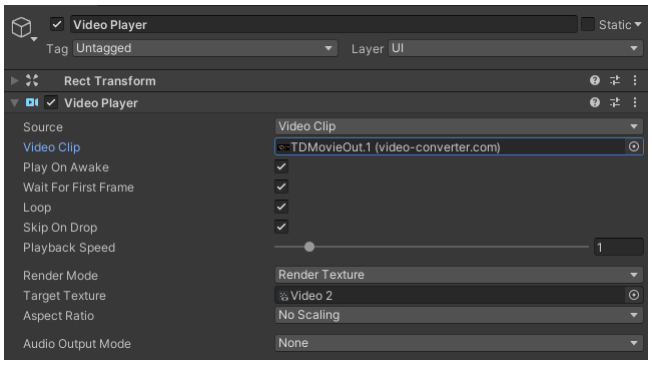


Figura 4.23

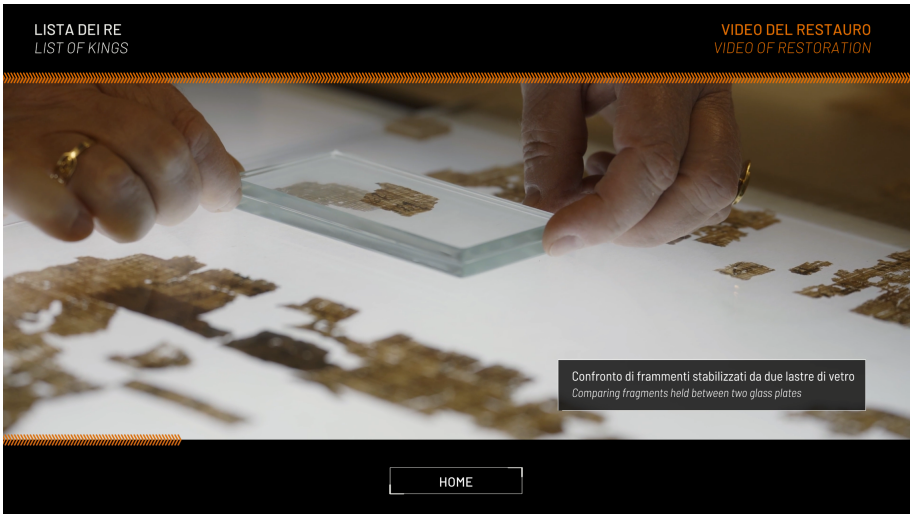


Figura 4.24

4.2.10 Info Pages

Le pagine di approfondimento sono due e riguardano il recto e il verso del papiro, offrendo una spiegazione generale su di essi. Sono accessibili rispettivamente dalla pagina dedicata al documento tributario e da Full List (verso).

Recto Info

Qui si inserisce semplicemente un pannello destinato a contenere un'immagine con testo esplicativo e un bottone di *back*, che riporta al registro tributario (Figura 4.25).

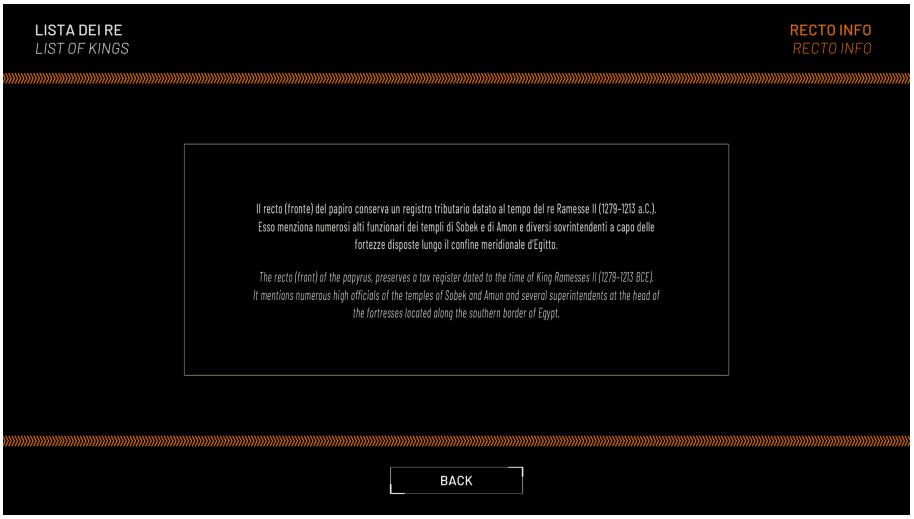


Figura 4.25

Verso Info

Il contenuto delle informazioni circa il verso è nettamente maggiore. L'idea è quella di offrire un approfondimento riguardo la lista dei Re e in particolare i Grandi Nomi, cinque titoli assegnati a ciascun sovrano nel momento dell'incoronazione, in funzione di ciò che compare sul papiro. Questi contenuti vengono organizzati in schede descrittive (card): una prima di introduzione generale al *verso*, a cui ne seguono cinque, ciascuna con i dettagli di un titolo in particolare. Le diverse card sono posizionate in scena orizzontalmente all'interno di un pannello *(InfoVersoPanel)* (Figura 4.26 e 4.27).

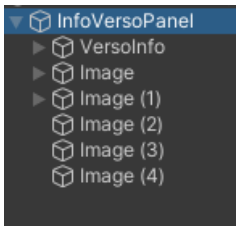


Figura 4.26

Il pannello viene fatto scorrere in modo da scoprire man mano le diverse card tramite le due frecce laterali che ne aggiornano la posizione, agendo anche sulle singole schede animandole e ponendole o meno in primo piano (Figura 4.29).
Lo script `InfoAnimationManager` - legato a *InfoVersoPanel* - (Figura 4.28) è responsabile della gestione del comportamento appena descritto. Lo script legato alle frecce (`InfoArrowButtonManager`) si occupa di spostare orizzontalmente il pannello nelle diverse posizioni, secondo uno schema di funzionamento identico a quello trattato nella *homepage*.

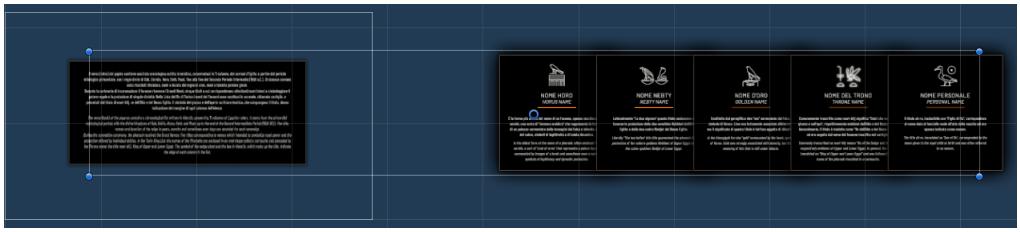


Figura 4.27

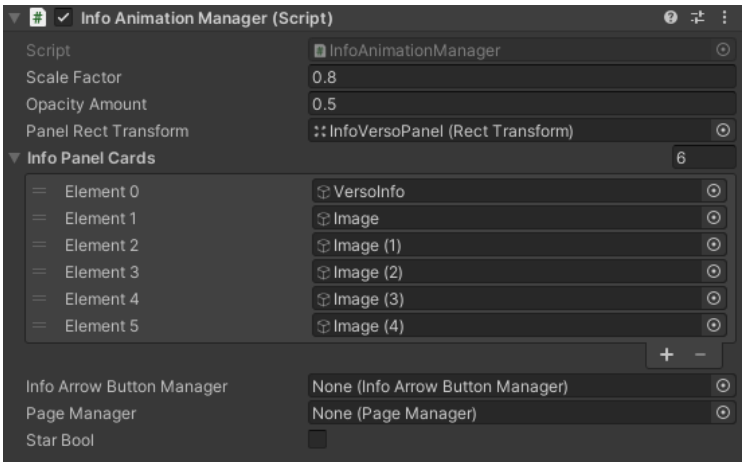


Figura 4.28

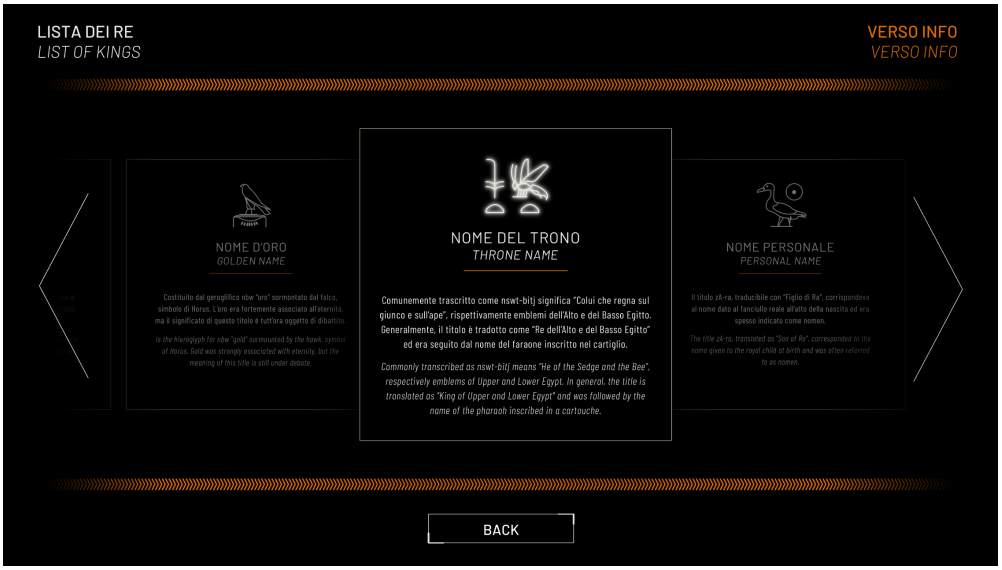


Figura 4.29

Ogni card possiede come componente `CardScript`, che definisce la classe `Card` caratterizzata dai seguenti parametri:

- `GameObject card`
- `Vector3 cardDefaultScale`
- `bool selected`
- `Vector2 panelRectTransformValue`

Nello start si salvano le condizioni iniziali quali cardDefaultScale.

Il funzionamento, attivato dal click sulle frecce, può essere riassunto nel seguente modo:

- si confronta la nuova posizione del pannello con `panelRectTransformValue` di ogni card
- se corrisponde, si assegna `card.selected=true;`
- per la card selezionata si eseguono le seguenti operazioni:
  ```
  child.GetComponent<Image>().DOFade(1f, 2f);
  child.GetComponent<RectTransform>()
  .DOScale(child.GetComponent<CardScript>().GetCard().GetDefaultScale(),
  1f);
  ```
- per le altre invece il *fade* si imposta al minimo così come lo *scale*.

4.2.11 Passaggio tra le pagine

Il passaggio da una pagina all'altra si basa sul componente `Canvas Group` che ogni pagina - intesa come gameobject padre di tutti i contenuti appartenenti ad una specifica pagina - possiede (Figura 4.30). Il processo di prima inizializzazione consiste nell'impostare `alpha=0` e `blocksRaycasts=false` per tutte le pagine eccetto *homepage*. Il valore di *alpha*, che determina l'opacità degli elementi UI all'interno del gruppo, varia da 0 a 1, dove 0 è completamente trasparente e 1 completamente opaco. Il passaggio tra le pagine si basa su questi componenti che permettono di mostrare e nascondere le pagine, consentendo altresì un'animazione dinamica di comparsa e scomparsa, andando ad agire sul canale *alpha* gradualmente, effetto che non sarebbe possibile ricorrendo semplicemente all'attivazione e disattivazione dei gameobject in quanto funzione booleana. Siccome la trasparenza delle pagine non determina consequenzialmente anche la disattivazione fisica degli elementi, è necessario agire via script, impostando *Blocks Raycasts* a `false` per le pagine non visibili e a `true` per la sola visibile. In questo modo si annullano gli "ostacoli" fisici determinati dagli elementi contemporaneamente attivi.
Dunque, nel momento in cui si ha un passaggio da una pagina ad un'altra, le operazioni da eseguire sono le seguenti:

- `pageToDeactive.blockRaycasts = false`
- `pageToDeactive.DOFade(0f, 0.7)`
- `pageToActive.blockRaycasts = true`
- `pageToActive.DOFade(1f, 0.7)`

DOFade è una funzione della libreria DOTween che trasforma l'*alpha* del target al valore e nel tempo indicati: `DOFade(float to, float duration)`.

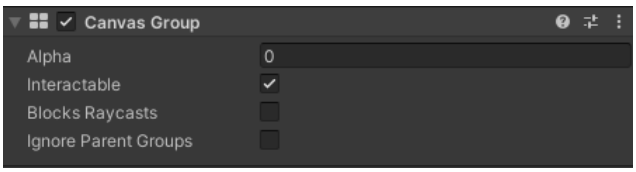


Figura 4.30

4.2.12 Arrows

Le due frecce laterali presenti nella maggior parte delle pagine hanno principalmente due modalità di utilizzo: in *Fragment Page* e *Pharaoh Page* aggiornano il contenuto (rimanendo nella stessa pagina), mentre in *Homepage, Verso, Recto* e *VersoInfo* la posizione in x di un pannello. Si descrive quindi il meccanismo che le caratterizza.

Aggiornamento dei contenuti all'interno di una pagina

Nell'organizzazione della struttura dell'applicazione si è ritenuto utile dare la possibilità all'utente di scorrere tra i frammenti e tra le schede dei sovrani senza dover necessariamente risalire al livello precedente. La soluzione adottata è stata l'inserimento di frecce laterali che consentono di passare al frammento/faraone adiacente. Nel caso particolare dei faraoni vi è un'ulteriore aspetto da considerare, ovvero se la pagina di provenienza sia Full List o Fragment Page: nel primo caso le frecce permettono di passare da un faraone all'altro tra tutti quelli del papiro, mentre nel secondo caso solo tra quelli presenti nel medesimo frammento.
I GameObject delle frecce sono figli di *ArrowButton* (Figura 4.31), empty a cui è legato lo script che gestisce lo specifico comportamento da azionare, che chiameremo genericamente *ArrowManager*.
LeftButton e *RightButton* sono elementi *UIButton* che contengono l'immagine delle frecce e si identificano grazie al *TypeOfButton*, impostato rispettivamente come Left e Right (Figura 4.32). Al click di una delle due frecce *ArrowManager* è in grado di capire su quale di esse si è verificato l'evento, proprio grazie al parametro che ne distingue il tipo. Fino

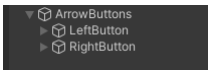


Figura 4.31

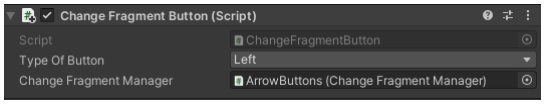


Figura 4.32

a questo punto il funzionamento è il medesimo per ogni freccia presente nell'applicazione, vediamo ora il caso più specifico delle pagine del frammento e del faraone.
ArrowManager, che in Figura 4.33 corrisponde a *ChangeFragmentManager* poiché si riporta il caso specifico del passaggio tra frammenti, si occupa di:

- se in PharaohPage, recuperare e controllare il valore di `itsFromFullList` per conoscere la provenienza
- salvare in un array temporaneo la lista dei frammenti/faraoni
- recuperare il valore del frammento/faraone corrente
- recuperare l'indice corrente, confrontando il valore del frammento/faraone corrente con la sua posizione nella lista precedentemente salvata
- controllare il tipo della freccia che è stata cliccata (Left/Right) e incrementare o decrementare l'indice di conseguenza
- aggiornare quindi il frammento/faraone corrente in funzione del nuovo indice
- cambiare di conseguenza il contenuto della pagina chiamando le funzioni `GoToSelectedPharaoh()` / `GoToSelectedFragment()`

All'interno di Arrow Manager è presente anche una funzione che viene chiamata non solo ad ogni cambiamento di pagina, ma anche al primo ingresso in essa, la quale si occupa

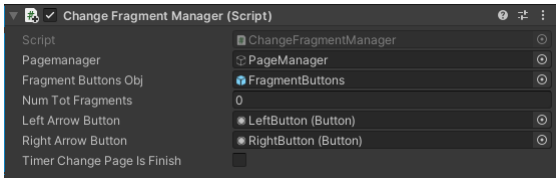


Figura 4.33

di verificare se ci si trova agli estremi della lista di elementi scorribili. Questo controllo aggiuntivo risulta necessario per evitare errori di compilazione e per disabilitare o attivare le frecce rendendole visibili o meno. Questo tipo di verifica si effettua controllando se la posizione dell'elemento corrente nella lista è zero oppure pari alla lunghezza della lista stessa.

Aggiornamento della posizione in x di un pannello

Nel caso invece delle frecce utilizzate per spostare un pannello, come nel caso dell'immagine del papiro o della lista di nomi, le operazioni eseguite in *ArrowManager* sono leggermente diverse, ma gerarchicamente l'organizzazione è la stessa vista sopra, così come il tipo di componenti legati agli oggetti *Button*.
Vediamo quindi in cosa si distingue.
Oltre al riferimento al pannello da spostare, si definisce una lista di valori in cui si inseriscono le posizioni in x che si vogliono far assumere (in Figura 4.34 il componente relativo al caso del pannello con l'immagine del papiro nella *Homepage*). Nello script, la funzione

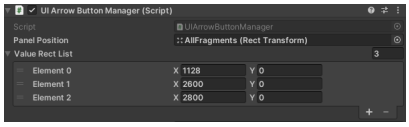


Figura 4.34

`UpdatePanelPosition()` aggiorna la posizione del pannello, animando lo spostamento verso la posizione finale data dalla posizione recuperata nella lista secondo l'indice corrente:
`panelPosition.DOAnchorPos(valueRectList[currentPageIndex], 0.5f, true);`
DOAnchorPos è una funzione della libreria DOTween che sposta la *anchoredPosition* del target al valore dato: `DOAnchorPos(Vector2 to, float duration, bool snapping)`.
L'indice ha un valore minimo pari a zero e un valore massimo pari alla lunghezza di `ValueRectList`. Il suo valore di partenza è inizializzato in Start() e viene incrementato o decrementato in funzione del `TypeOfButton` coinvolto (freccia destra o sinistra).
Anche qui si controlla la posizione entro gli estremi per gestire l'attivazione delle frecce.

Scelte per l'interazione

Per ottenere un'interazione con Leap Motion più fluida possibile si è deciso di considerare come *trigger* per l'attivazione delle frecce il semplice *hover* del cursore che corrisponde fisicamente al gesto intuitivo della mano che si sposta (anche velocemente) verso l'estremo destro o sinistro dello schermo in corrispondenza delle due frecce. Questo ha d'altra parte reso necessario un'operazione aggiuntiva che limitasse lo scenario che vede una perdita di controllo da parte dell'utente che esplora l'applicazione: tenere la mano nella stessa zona, dopo aver attivato una freccia, determina una successiva e ravvicinata riattivazione della stessa. Per ovviare a questo inconveniente, si è dunque inserito un evento intermedio di controllo. Di seguito il codice inserito negli script *ArrowManager* precedentemente descritti.

```
public bool timerChangePageIsFinish=true;    //in start

public void ArrowHoverEvent(TypeOfButton typeOfButton)
{
        CancelInvoke();
    Invoke("WaitToChangePage", 1f);

    if (timerChangePageIsFinish)
    {
        timerChangePageIsFinish = false;
        ChangePharaoh(typeOfButton);
    }}

public void WaitToChangePage()
    timerChangePageIsFinish = true;
```

La funzione `ArrowHoverEvent(typeOfButton)` è chiamata all'Hover del cursore al posto della funzione che innesca il processo di aggiornamento delle pagine descritto precedentemente, anticipandolo. Ogni volta che è chiamata, tramite `Invoke` si fa partire un timer (in questo caso pari a 1 secondo) al termine del quale setta `timerChangePageIsFinish=true`. Nella stessa funzione si innesca il cambio di pagina, in questo caso `ChangePharaoh(typeOfButton)`, con condizione `timerChangePageIsFinish=true` e parallelamente se ne imposta il valore a `false`, così che, se ArrowHoverEvent è nuovamente chiamato prima dello scadere del timer, il cambio pagina non possa essere richiamato.

4.2.13 Call to action: Pause e ScreenSaver

Essendo l'applicazione un'installazione all'interno di un contesto museale, si deve considerare il possibile comportamento dei visitatori e tenere conto di diversi scenari di prima interazione. In questa sezione si vogliono delineare e descrivere le scelte che riguardano il "richiamo all'azione" del visitatore, nonché la gestione di ulteriori due pagine per le situazioni di *Pause* e *ScreenSaver*. La prima, che precede temporalmente la seconda, si utilizza per richiamare l'attenzione dell'utente dopo un tempo relativamente breve di inattività. Il

pannello animato riportato in Figura 4.35 è ciò che compare sullo schermo dopo una certa quantità di secondi. Il tempo è stato deciso tenendo conto della durata del video presente in *RestorationPage*, ovvero 45": in questo modo si permette di vedere senza interruzioni il video ed è inoltre un tempo ragionevole anche nel caso in cui l'utente si trovi, ad esempio, a leggere la biografia di un faraone o un'altra sezione dell'applicazione che non richiede interazione. Quella che si è appena descritta è la situazione *Pause* (Figura 4.61): il pannello compare centrale sullo schermo, ma non lo copre interamente, lasciando attorno a sé leggermente visibile la pagina precedente, diventata layer inferiore a un pannello nero con alpha 0.5. Se in questa fase l'utente effettua un movimento riconosciuto dal sensore, il pannello di pausa si disabilita e si può riprendere l'esperienza dal punto in cui ci si trovava. In questo modo, la richiesta di attenzione da parte del sistema è lecita e poco invasiva. Se invece dopo l'attivazione della pausa il sistema non riceve alcun input nei successivi

Figura 4.35

N secondi, si attiva lo *ScreenSaver*. Si tratta di una pagina che ricopre interamente lo schermo, sostituendo quella di provenienza. Vi compare un video suggestivo e riassuntivo del contenuto dell'applicazione con grafiche animate del papiro, accostate ad un indizio di interazione per l'utente posizionato nella parte inferiore del video stesso sotto forma di animazione (Figura 4.36). In questo caso, che presuppone (con N=60) un tempo di

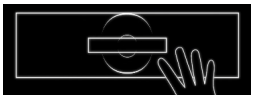


Figura 4.36

inattività di 1'45" secondi (45+N), si suppone che l'utente si sia spostato definitivamente. É dunque questa una condizione probabile in cui il visitatore potrebbe incorrere nell'applicazione, a meno che si verifichi un flusso continuo di visitatori. A questo punto, se l'utente interagisce, si effettua un *reload* completo della scena, resettando così le precedenti azioni e portandolo direttamente alla situazione iniziale in *Homepage*.
Si vede quindi come si siano sviluppati i meccanismi appena descritti.
Le funzioni di cui si tratta di seguito sono tutte contenute nello script `TestInActive` legato in scena a *PageManager*.
Per controllare la presenza di input da parte dell'utente nell' `Update()`, che è chiamato per ogni frame, si salva localmente il valore `Vector3` della posizione del cursore:

`prevCursorPosition = cursor.transform.position;`
dove cursor è il riferimento al GameObject del cursore presente in scena.
A questo punto è possibile confrontare l'attuale posizione con quella appena salvata (nel frame precedente) e quindi verificare l'attività dell'utente. All'interno della condizione `cursor.transform.position != prevCursorPosition`, quindi di attività, si struttura uno `switch` con tre casi: 0, 1 e 2, che corrispondono a tre stati diversi. Lo stato è inizializzato a 0. Le funzioni che gestiscono i timer sono Invoke() in cascata che oltre ad attivare le specifiche pagine di pausa, aggiornano lo stato.
Di seguito il codice interno alla condizione di attività nell'Update di `TestInActive`.

```
switch (stato)
{
case 0:
    StartTimerInvoke();
    break;
case 1:
    Debug.Log("ero in pausa, la disattivo");
    pageManagerScript.pausePage.DOFade(0f, 0.5f);
    stato = 0;
    break;
case 2:
    Debug.Log("ero in screen saver, torno in home");
    pageManagerScript.infoPageHome
        .GetComponentInChildren<InfoAnimationManager>()
            .ExitInfoPanelReset();
    pageManagerScript.ActivateAllPages(false);
    pageManagerScript.screenSaverPage.DOFade(0, 1f)
        .OnComplete(()=>
            SceneManager
            .LoadScene(SceneManager.GetActiveScene().buildIndex)
        );
    stato = 0;
    break;
}
```

I diversi stati distinguono tre possibili contesti e quindi - ricordando che ci si trova nella condizione di attività dell'utente - che cosa si deve verificare:

- stato=0 : stato di attività; non è scaduto il timer, quindi l'utente sta interagendo normalmente; deve resettarsi il timer di inattività

- stato=1 : se ci si trova in questo stato, significa che è scattata la pausa, quindi ora va disattivata

- stato=2 : si è attivato lo *screensaver*, quindi, quando l'utente interagisce, deve avvenire il *reload* dell'intera scena.

I timer sono gestiti dalle funzioni `Invoke()`, così che al termine del tempo sia chiamata la funzione. La funzione `CancelInvoke()` si occupa di resettare e rifar partire il timer. `StartTimerInvoke()`, che è chiamata ad ogni frame in cui l'utente ha effettuato un movimento rispetto al frame precedente, esegue le seguenti operazioni:

- stato=0
- CancelInvoke()
- Invoke("Pause", pauseTimer).

Dopo aver impostato lo stato a zero, resetta il timer e invoca la funzione `Pause()` con un ritardo di tempo pari a pauseTimer (45"). In questo modo, se l'utente in `case 0` muove la mano il timer si resetta e riparte. Se il tempo di inattività raggiunge il valore di `pauseTimer`, la funzione `Pause()` è invocata.
All'interno di questa funzione si attiva quindi il pannello di pausa precedentemente descritto (Figura 4.61) e si eseguono le seguenti operazioni:

- stato=1
- Invoke("ResetApp", screenSaverTimer).

In questo modo si determina il passaggio ad uno stato diverso e si introduce il timer successivo - relativo all'attivazione dello screen saver - pari a screenSaverTimer (60"), al termine del quale è invocata la funzione `ResetApp()`. Si tratta dell'ultima funzione in cascata di questo meccanismo: ci si trova in tale stato dopo un tempo di inattività di pauseTimer+screenSaverTimer ed è dunque adibita ad attivare la pagina ScreenSaver, disattivare il pannello di pausa, assegnare il valore 2 allo stato e iniziare il video presente all'interno della pagina stessa.

Call to action interne alle pagine

In aggiunta ai primi inviti all'interazione descritti, in ogni pagina è stato inserito un testo contente informazioni (alternativamente in lingua italiana e inglese) con indizi di comportamento specifici alla pagina stessa, come d'esempio in Figura 4.37 e 4.38.

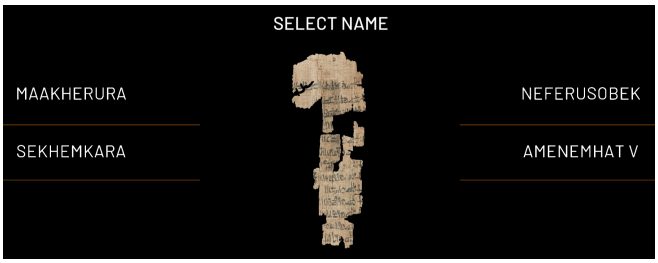


Figura 4.37

Figura 4.38

4.3 Materiali definitivi

In questa sezione si mostra il risultato visivo dell'applicazione attraverso una carrellata di immagini; si descrivono quindi i processi di importazione delle grafiche e la gestione delle animazioni.

4.3.1 Grafica e animazioni

La grafica dell'applicazione è stata guidata da due soggetti: da una parte il reparto grafico del Museo, che ha delle specifiche linee guida per quanto riguarda la palette (Figura 4.39) e la gestione dei testi italiano/inglese, i quali devono essere entrambi presenti e visibili; dall'altro lato Robin Studio, sotto controllo periodico e supervisione del Museo, ha creato ed interamente realizzato i contenuti grafici. Parallelamente allo sviluppo su Unity, la

Figura 4.39: Palette Sala della Scrittura

responsabile grafica di Robin Studio ha realizzato un mockup con *Invision* per mostrare gli effetti grafici da implementare e inserire all'interno dell'applicazione. In questo modo, man mano che le grafiche definitive erano pronte, esse venivano importate all'interno di Unity in un sistema pronto a riceverle perchè sviluppato come richiesto. Di seguito si

mostrano alcuni esempi di istruzioni di implementazione presenti su *Invision*, quali gli effetti desiderati al passaggio del cursore su determinati elementi: la freccia che da bianca diventa arancione (Figura 4.40), il contorno e il *glow* arancione del frammento (Figura 4.41), il cambio di colore per i bottoni di navigazione, lo zoom e l'ispessimento dei nomi per i *pharaohButtons* (Figura 4.42).

Figura 4.40

Figura 4.41

Figura 4.42

Figura 4.43

Libreria DOTween

Le animazioni, ideate dal reparto grafico di Robin Studio, sono state implementate all'interno di Unity, sfruttando principalmente la libreria DOTween.
Nell'insieme di animazioni si possono distinguere quelle di comparsa e scomparsa delle pagine e dei loro contenuti dalle animazioni di feedback grafico, come l'ingrandimento del testo e il cambio di colore al passaggio del cursore.
DOTween [1] è una libreria di *tweening*, ovvero una libreria di codice che fornisce delle specifiche funzioni per eseguire semplici animazioni - chiamate *"tweens"* - basate su codice. Si tratta infatti di un motore di animazione veloce ed efficiente, utilissimo nella realizzazione di animazioni all'interno dell'ambiente Unity. Si può scaricare in forma gratuita direttamente dallo Unity Asset Store, dove si trova anche l'estensione *DOTween Pro* a pagamento. La versione *Pro* si distingue principalmente per l'integrazione di un *visual editor* che permette di controllare le animazioni tramite interfaccia da *component*.

Storyboard delle animazioni

Homepage:

- Papiro completamente visibile (opacità 100%) in posizione centrale
- Frammenti non cliccabili diventano di opacità 70% e il papiro si sposta automaticamente verso sinistra fino a posizione di start
- Testo con indicazioni per l'utente con animazione *fade-in/fade-out* (in alternanza tra frase in italiano e poi in inglese in loop)
  - "MUOVI LA MANO" / "ESPLORA LA LISTA DEI RE" / "SELEZIONA UN FRAMMENTO"
  - "MOVE YOUR HAND" / "EXPLORE THE LIST OF KINGS" / "SELECT FRAGMENT"
- Animazione di selezione di un frammento con *glow* arancione (Figura 4.41)

Lista completa dei re:

- La lista completa si comporta come il papiro
- Testo delle indicazioni, come in homepage
  - "ESPLORA LA LISTA DEI RE"
  - "EXPLORE THE LIST OF KINGS"
- Animazione hover e selezione "nome re" con leggero zoom e glow (Figura 4.42)

InfoVerso:

- Animazione slide delle card

Recto:

- Il registro si comporta come la lista dei re in homepage
- Testo con indicazioni come in homepage
  - "GUARDA IL REGISTRO TRIBUTARIO"
  - "LOOK AT THE TAX REGISTER"

Fragment Page:

- Apparizione del frammento in zoom + fade in
- Contemporaneamente fade in delle linee arancioni
- Apparizione nome del re con animazione fade in (in successione come da numeri in Figura 4.44)
- Quando il cursore va in hover al nome del re
  - geroglifici sul papiro che si illuminano
  - zoom e cambio colore (in arancione) del nome
- Testo delle indicazioni (animazione come in homepage)
  - "SELEZIONA IL NOME"
  - "SELECT NAME"

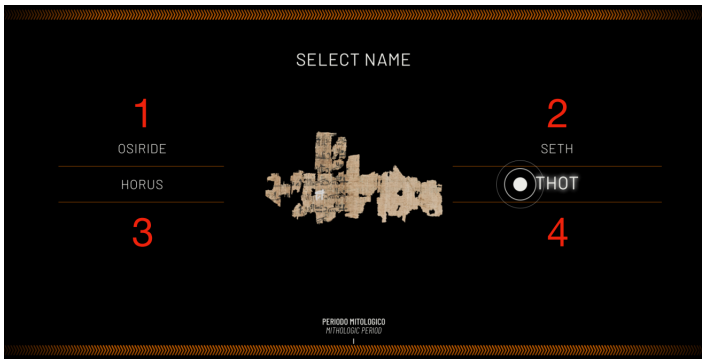


Figura 4.44

Pharaoh Page:

- Fade in del/i nome/i e della linea
- Fade in della biografia in italiano
- Fade in della biografia in inglese
- Fade in dal basso dell'illustrazione del re (Figura 4.45)
- Fade in e arrivo da destra delle card contenenti le informazioni (sulla destra) una alla volta (Figura 4.46)

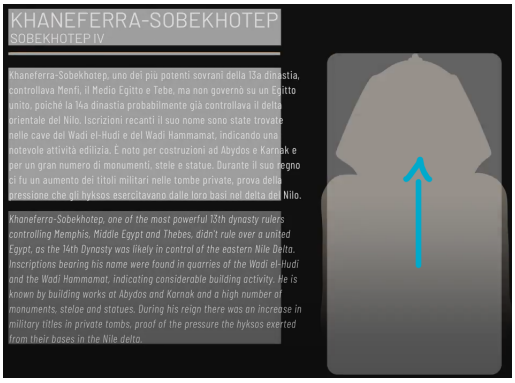


Figura 4.45

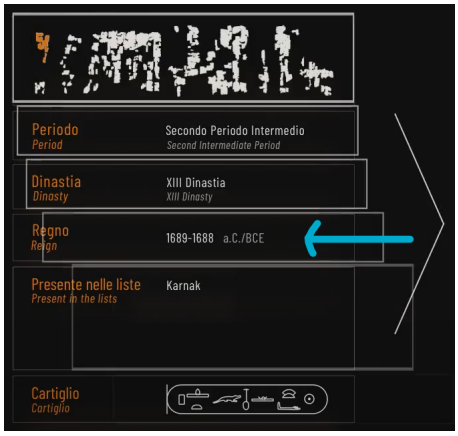


Figura 4.46

Animazione in hover

Si descrive ora il meccanismo generale di funzionamento delle animazioni attivate al passaggio del cursore su determinati oggetti, quali i frammenti cliccabili, i bottoni di navigazione, i *pharaohButtons* e le frecce laterali. I bottoni di questo tipo sono organizzati gerarchicamente in modo tale da avere come *children* la loro versione "normale" dell'immagine e parallelamente la propria versione *hover*. Quest'ultima, che è solitamente non visibile, è portata ad opacità 100% all'occorrenza tramite script. I testi sono trattati diversamente in quanto è necessario accedere al loro valore di *Color* per poterne cambiare il valore via script.
Ad ogni oggetto a cui bisogna legare questo tipo di comportamento, si aggiunge un componente *Event Trigger*, che permette di inserire una specifica funzione da eseguire in base al tipo di evento. In questo caso si sfruttano *PointerEnter* e *PointerExit* per chiamare la funzione necessaria, aggiungendo l'informazione booleana per distinguere le due situazioni. In Figura 4.47 il componente legato ad una delle frecce laterali; *SelectionButton* è il nome dello script in cui è stata definita la funzione qui chiamata *HoverArrow(bool boolean)*. All'interno del metodo richiamato si eseguono le operazioni necessarie che possono

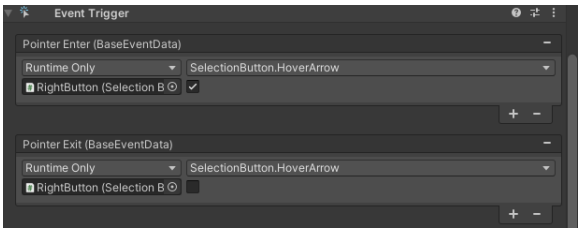


Figura 4.47

essere, in funzione dell'elemento: rendere visibile un'immagine, cambiare il colore di un testo e/o scalare la dimensione di un bottone.
Per rendere visibile o meno un'immagine, si agisce normalmente sul suo canale alpha, che

però non è supportato dai formati vettoriali. Poiché le immagini che si utilizzano sono di tipo *.svg*, risulta necessario aggiungere al corrispondente GameObject il componente *Canvas Group* che, come si è già illustrato in merito al passaggio tra le diverse pagine, possiede come proprietà anche il canale alpha. Controllando quest'ultimo si influenza quindi l'opacità degli elementi UI legati ad esso, cioè l'immagine: si utilizza anche qui la funzione `canvaGroup.DOFade(float to, float duration)`
Il colore del testo è accessibile e modificabile sfruttando nuovamente la libreria DOTween attraverso la funzione `DOColor(Color to, float duration)`, utilizzabile con TextMeshPro - asset impiegato per i testi - solo se in possesso dell'estensione *Pro* di DOtween.
Il testo di cui è necessario scalare la dimensione al passaggio del cursore è quello relativo ai *pharaohButton* che, come si è trattato in precedenza, sono istanziati dinamicamente in scena a partire dal relativo *prefab*. Per gestire lo *scale-up* e *scale-down* si ricorre a *DOTween Animation Component* così come riportato rispettivamente nelle Figure 4.48 e 4.49. Questo componente, disponibile con l'estensione DOTween *Pro*, offre una gamma di tipi di animazione - tra cui *SCALE* - e trova, se presente, il componente dello stesso GameObject adatto ad essere animato.
Si inseriscono i parametri di animazione desiderati, quali il tipo *(Scale)*, la durata, l'*ease* e il valore di arrivo. Per distinguere i due componenti inseriamo due diversi *id*.
A questo punto, per quanto attiene al codice, si crea una funzione che si occupi di azionare le due animazioni:

```
public void AnimatePharaohButton(bool eventTriggerBoolean)
    {
     if (eventTriggerBoolean)
          textToScale.GetComponent<DOTweenAnimation>()
                  .DORestartById("scaleUp");
     else
          textToScale.GetComponent<DOTweenAnimation>()
                  .DORestartById("scaleDown");
     }
```

DORestartById(string id) riavvia tutti i *tween* con l'*id* dato. La funzione sopra mostrata

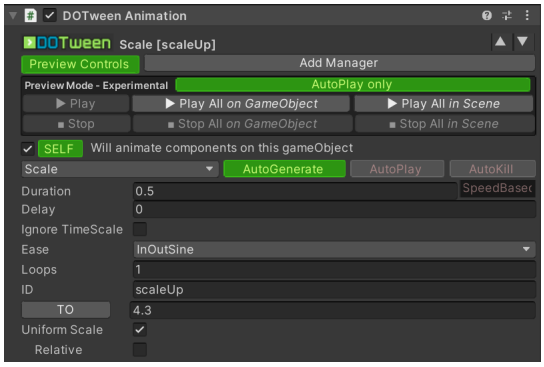


Figura 4.48

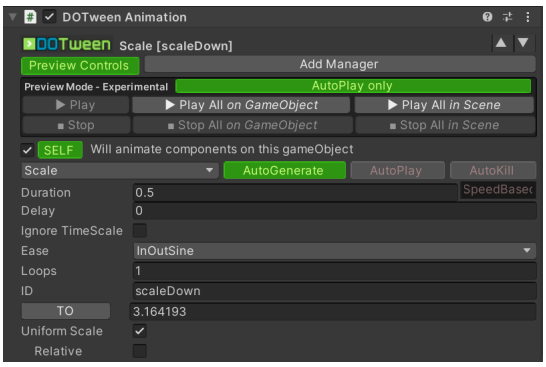


Figura 4.49

è aggiunta nell'*Event Trigger* del *pharaohButton*, come mostrato in Figura 4.50. In questo modo l'animazione di *scaleUp* sarà azionata *on Pointer Enter*, mentre quella di *scaleDown* nella condizione di *on Pointer Exit*, portando il testo alla dimensione di default.

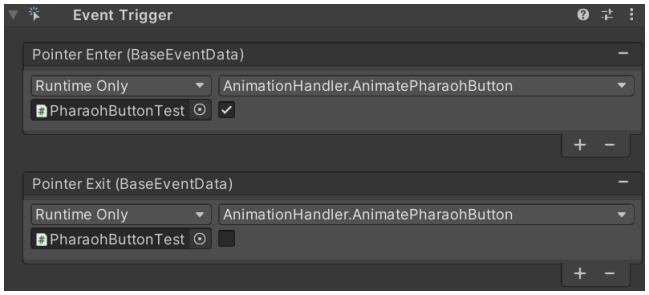


Figura 4.50

Animazioni di apertura delle pagine e comparsa dei contenuti

Nel passaggio tra le pagine, per evitare il cambiamento repentino tra di esse, si effettuano animazioni di *fade out* e *fade in*, rispettivamente per la pagina in uscita e in entrata, così come si era anticipatamente descritto nella sezione 5.2.11.
La comparsa dei contenuti all'apertura delle pagine è riportata nello *Storyboard delle animazioni*; in particolare di seguito ci si concentra sull'ingresso dei contenuti all'interno di *Fragment Page* e *Pharaoh Page* (Figure 4.44, 4.45 e 4.46). Per ottenere una sequenza di animazioni in cascata degli elementi, si è inserito per ciascuno di essi il componente *DOTween Animation* e se ne è sfruttato in particolare il parametro di *delay*, utilizzato come tempo di ritardo alla partenza rispetto al primo degli elementi che compare in scena: esso, d'altro canto, deve azionare gli altri tutti insieme. Per fare ciò, se ne sfrutta l'evento *onStart()*, accessibile dal componente stesso, nel quale si inseriscono i riferimenti agli *Animation Component* degli altri elementi, come nell'esempio in Figura 4.51: su di essi si richiama la funzione DORestart(), che riavvia l'animazione, a cui sarà associato il ritardo stabilito. Per le *card* di contenuti (a destra in *Pharaoh Page*), oltre che l'anima-

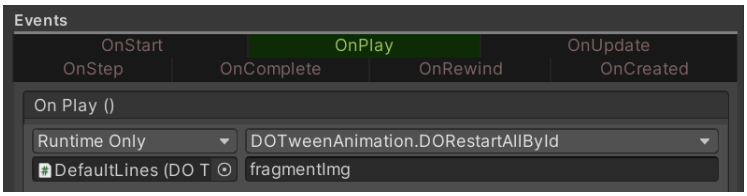


Figura 4.51

zione di *fade* è inserito un *DOTween Animation Component* aggiuntivo, questa volta di tipo *MOVE* per permettere l'ingresso delle caselle da destra: si inserisce la destinazione finale (valore *xyz*). Importante notare che, affinché il meccanismo funzioni, è necessario posizionare ogni volta il punto di partenza delle *card* al di fuori dello spazio visibile del

Canva. Allo stesso modo, l'opacità degli elementi da animare deve essere inizialmente allo 0% così che si possano animare fino ad essere visibili.
Leggermente più complesso è il caso dei vari *pharaohButton* a cui, essendo creati dinamicamente ogni volta, non è possibile pre-assegnare un ritardo, poiché dipende da quanti sono i bottoni totali presenti. Si è quindi optato per un'assegnazione dinamica dei *delay* all'interno del processo che porta alla creazione delle nuove istanze del *prefab*. Si definiscono due nuove variabili: *pharaohButtonsDelay* e pharaohButtonsDefaultDelay. La prima è aggiornata e assegnata per ogni faraone di cui si istanzia il bottone, mentre la seconda contiene il valore scelto a priori di *delay* che si vuole ottenere tra la comparsa di un elemento e il successivo. Nella funzione già descritta in precedenza *LoadPharaohButtons(Fragemnt fragment)* si inizializza *pharaohButtonsDelay* a zero, e ad ogni faraone appartenente al frammento ne viene incrementato il valore, eseguendo l'operazione seguente:

```
pharaohButtonsDelay += pharaohButtonsDefaultDelay;
```

A questo punto, se ne può assegnare il valore al parametro del rispettivo componente (già presente nel *prefab*):

```
tempButton.GetComponent<DOTweenAnimation>()
                                .delay = pharaohButtonsDelay;
```

In tutti gli altri casi, dove questo passaggio non è necessario, in quanto gli elementi sono già presenti in scena e se ne conosce pertanto la quantità finale, da codice è sufficiente azionare l'animazione del primo elemento tramite il comando DORestart(). Tale comando riavviare l'animazione così da risolvere possibili problematiche dovute alla mancata conclusione della stessa nel passaggio da una pagina all'altra.

4.3.2 Apertura al pubblico

In questa sezione si riportano i commenti ricevuti ed i dati rilevati mediante l'osservazione diretta durante l'inaugurazione dell'installazione e quelli che derivano dal monitoraggio condotto dal personale museale nelle due settimane successive all'apertura della sala.

La prima reazione del pubblico all'ingresso dell'ambiente dedicato al Papiro dei Re é stata di interesse nei confronti del sensore di movimento. Nei momenti di maggiore affluenza si è creata una coda di visitatori incuriositi e attenti a carpire le informazioni d'uso.
Per molti l'utilizzo é stato naturale ed immediato: in questi casi la fruizione è risultata prolungata ed accompagnata da entusiasmo e sorpresa, soprattutto per la possibilità di approfondimento tematico sui singoli faraoni e di agevole passaggio tra i frammenti e per il contatto ravvicinato con un materiale delicato

e prezioso. Diversamente da quanto ci si potrebbe aspettare, non si è trattato prevalentemente di persone appartenenti alla fascia d'età che va dai 15 ai 35 anni, i cosiddetti "nativi digitali" o comunque "alfabetizzati digitali": è stata invece la fascia d'età dai 50 anni in su a trascorrere più tempo di fronte al sensore.
Per alcuni visitatori tuttavia la fruizione non si é rivelata immediatamente efficace. La difficoltà principale è stata quella di intuire, nonostante un apposito cartello fornisse le informazioni per l'utilizzo del sensore, che tra di esso e la mano del visitatore andasse mantenuta una distanza di circa 20-30 cm. L'immagine accanto alla postazione simulava la corretta postura da assumere, quindi si potrebbe azzardare, contro ogni aspettativa, la considerazione che il testo scritto attiri maggiormente l'attenzione, per lo meno quando si tratta di trasmettere istruzioni, e che l'immagine si percepisca come subordinata ad esso. Chi non riusciva a far rispondere il sensore ai propri comandi reagiva o abbandonando la postazione interattiva o toccando il sensore, simulando cioè l'approccio ai dispositivi touchscreen, che ormai sono entrati nell'esperienza comune. In questi casi risultava molto gradito l'intervento del personale di sala, quale supporto per l'avvio dell'esplorazione.
Al di là delle differenze di approccio, la risposta del visitatore é stata nel complesso entusiasta, non solo per l'innovativo approccio tecnologico, ma anche per la grafica e i contenuti dell'interfaccia digitale. Soprattutto si é manifestato un grande interesse per la possibilità di usufruire della traduzione del testo del Papiro dei Re e di esplorarlo autonomamente.

4.3.3 Risultati

Di seguito si riportano le immagini dell'applicazione definitiva presentata al pubblico per la prima volta il 27 settembre 2022 in occasione del bicentenario della decifrazione dei geroglifici da parte di Jean-Francois Champollion (si veda sezione 2.4).

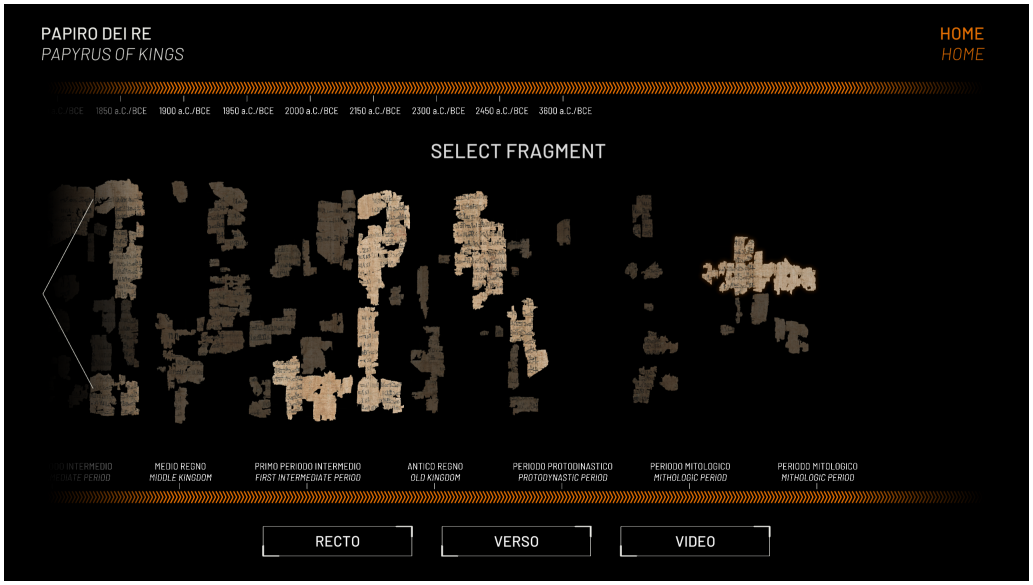


Figura 4.52: Homepage

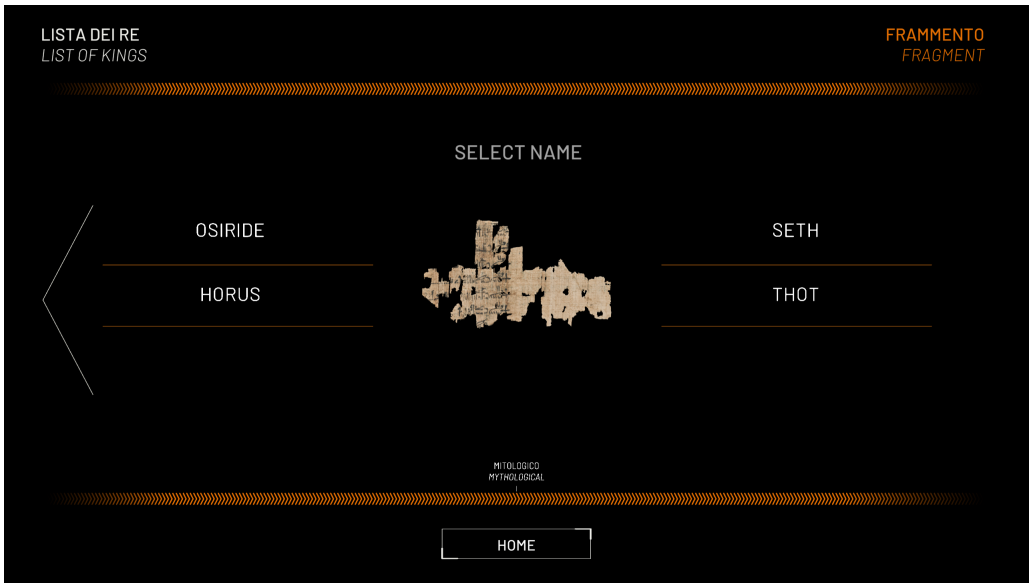


Figura 4.53: Fragment Page

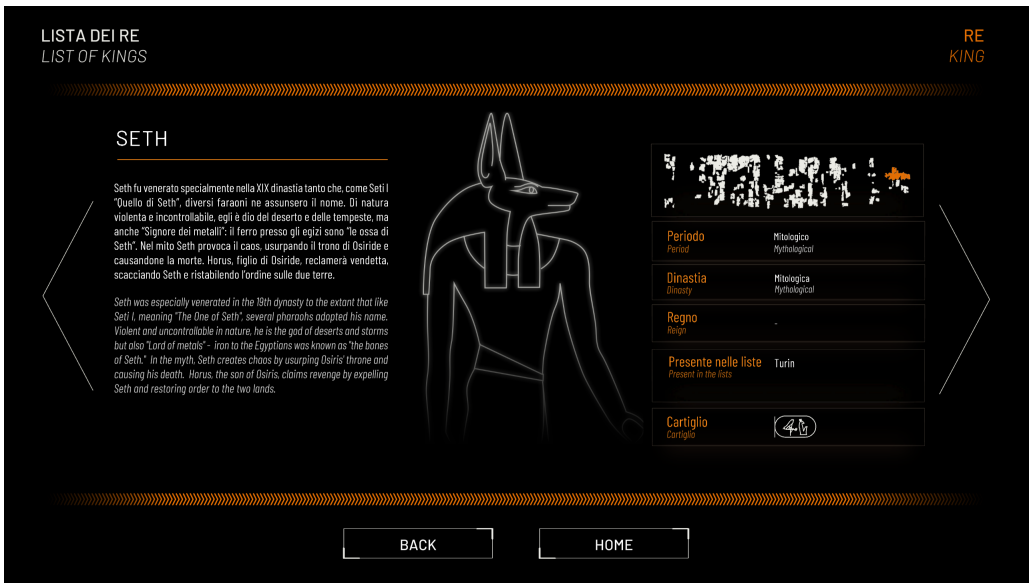


Figura 4.54: Pharaoh Page

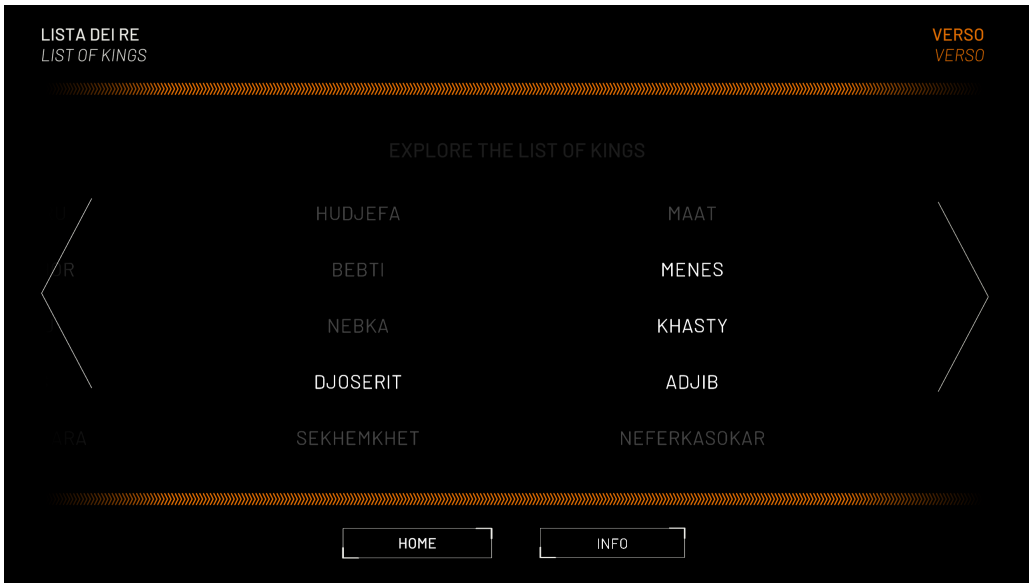


Figura 4.55: Full List Page

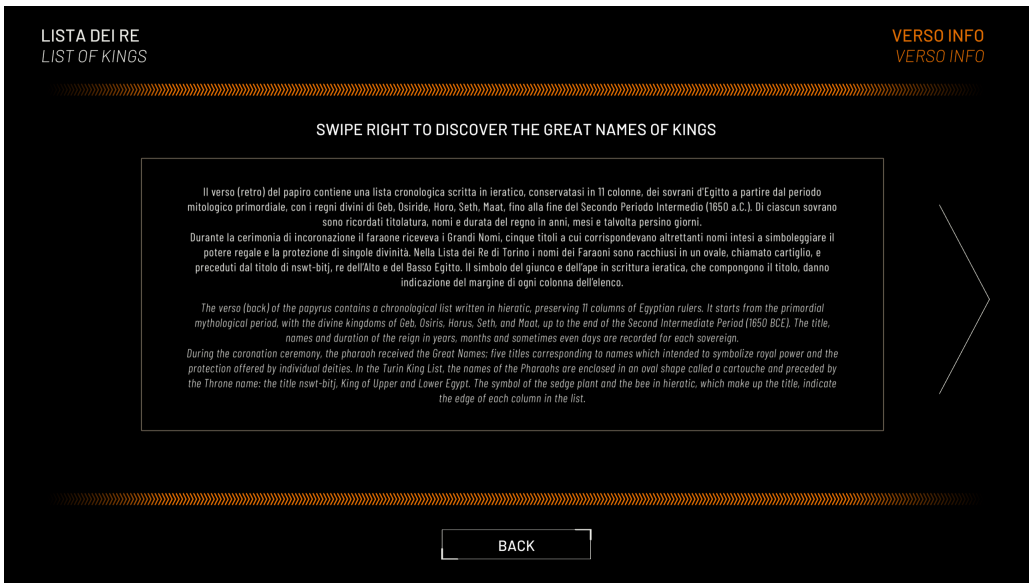


Figura 4.56: Verso Info Page

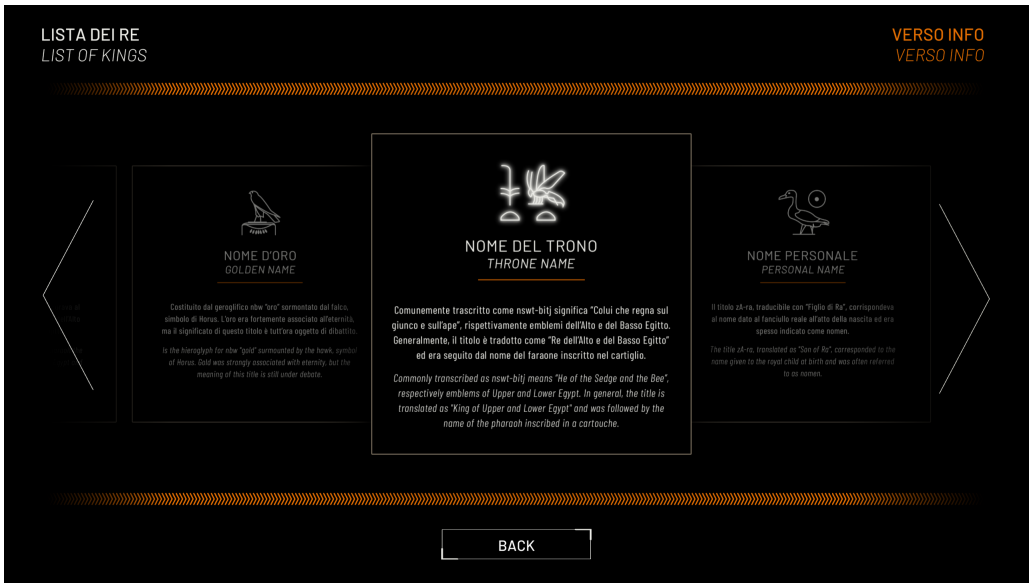


Figura 4.57: Verso Info Page 2

Figura 4.58: Recto

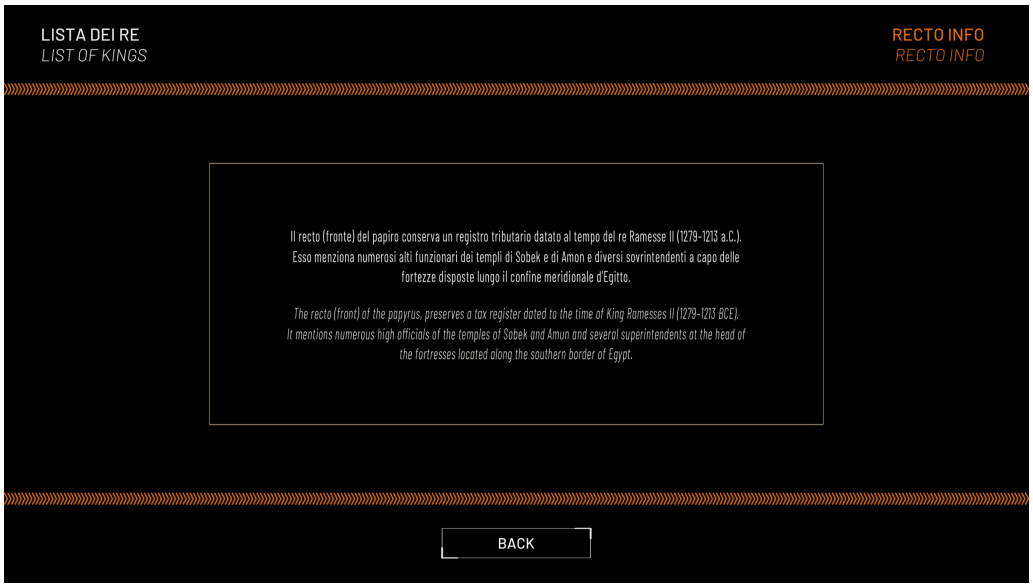


Figura 4.59: Recto Info Page

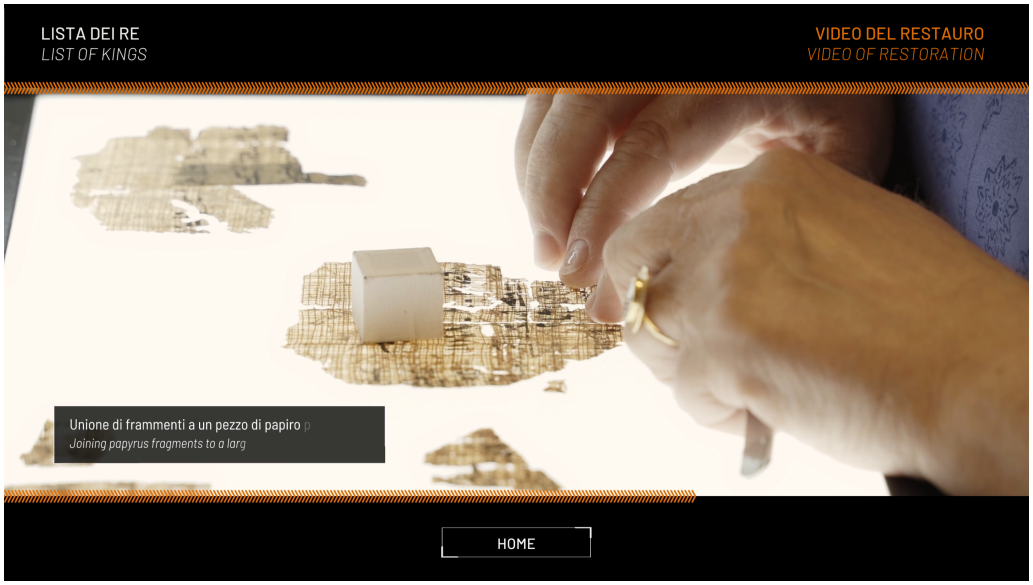


Figura 4.60: Video of Restoration Page

Figura 4.61: Pause

Figura 4.62: Screen Saver

Bibliografia

[1] Dotween documentation. http://dotween.demigiant.com/documentation.php.

[2] Touchfree user manual. https://docs.ultraleap.com/touchfree-user-manual.

[3] https://www.cabq.gov/artsculture/biopark".

[4] Azure kinect dk. https://azure.microsoft.com/en-us/products/kinect-dk".

[5] https://www.europeana.eu/it.

[6] mann-napoli.it/father-and-son-the-game/.

[7] www.marinabaysands.com/museum/exhibitions/future-world.html.

[8] www.humancentereddesign.org.

[9] https://ideum.com/.

[10] https://collezionepapiri.museoegizio.it/it-IT/section/Collezione-Papiri/eb10456f3f324a7493730546ed080456/Highlights-Lista-dei-Re/.

[11] https://www.ultraleap.com/product/leap-motion-controller/.

[12] https://www.mocc.cuhk.edu.hk/en-gb/.

[13] https://www.museoegizio.it/chi-siamo/.

[14] https://www.osservatori.net/it/ricerche/osservatori-attivi/innovazione-digitale-nei-beni-e-attivita-culturali.

[15] Piano nazionale di digitalizzazione. https://digitallibrary.cultura.gov.it/il-piano/.

[16] https://www.tate.org.uk/whats-on/tate-modern/yayoi-kusama-infinity-mirror-rooms.

[17] https://developer.leapmotion.com/touchfree-tooling-unity.

[18] https://www.ultraleap.com.

[19] https://www.beniculturali.it/virtualtour.

[20] 2022. https://www.icom-italia.org/definizione-di-museo-di-icom.

[21] N. Bonacasa. *Il Museo on line. Nuove prospettive per la museologia.* OADI-DIGITALIA. Osservatorio per le Arti Decorative in Italia "Maria Accascina", 2011.

[22] J. Cohen and J. Gallagher. Covid-19 and the rise of hands-off museum experiences. https://blooloop.com/museum/opinion/covid19-hands-off-museum-experiences.

[23] P. Forte. I musei statali in italia: prove di autonomia. Fascicolo 01, 2011.

[24] Frangione. Digitale a chi? *Il Giornale delle Fondazioni*, Studi e Ricerche, 2018.

[25] P. Koutsabasis and S. Vosinakis. Adult and children user experience with leap motion in digital heritage: The cycladic sculpture application. 10 2016.

[26] P. M. Kurt Konolige. Technical description of kinect calibration. http://www.ros.org/wiki/kinect_calibration/technical.

[27] H.-T. Liao, M. Zhao, and S.-P. Sun. A literature review of museum and heritage on digitization, digitalization, and digital transformation. In *Proceedings of the 6th International Conference on Humanities and Social Science Research (ICHSSR 2020)*, pages 473–476. Atlantis Press, 2020. https://doi.org/10.2991/assehr.k.200428.101.

[28] X. Ma. Touchless technologies for museum engagement, 2021. https://amt-lab.org/blog/2021/6/touchless-technologies-museum-engagement.

[29] A. Magrini, 2022. https://frameblog.unibo.it/index.php/2022/05/05/mostre-immersive.

[30] N. Raimo, I. D. Turi, A. Ricciardelli, and F. Vitolla. Digitalization in the cultural industry: evidence from italian museums. *International Journal of Entrepreneurial Behavior & Research*, 2021. https://doi.org/10.1108/IJEBR-01-2021-0082.

[31] J. Spadaccini. Touchless gesture-based exhibits, part two: Full-body interaction, 2020. https://ideum.com/news/touchless-interaction-public-spaces-part2.

[32] M. Tölgyessy, M. Dekan, L. Chovanec, and P. Hubinský. Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, 21(2), 2021. https://www.mdpi.com/1424-8220/21/2/413.

[33] Ultraleap. The end of the touchscreen era, 2020. https://www.ultraleap.com/company/news/press-release/end-of-public-touchscreens/.

[34] UNESCO. Museums around the world in the face of covid-19. 2020. unesdoc.unesco.org/in/rest/annotationSVC/DownloadWatermarkedAttachment/attach_import_94a8eedf-4246-4000-aba4-32f33f12ac61?_=373530eng.pdf&to=31&from=1.

[35] C. Wang, Z. Liu, and S.-C. Chan. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE Transactions on Multimedia*, 17(1):29–39, 2015.

[36] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013. `https://www.mdpi.com/1424-8220/13/5/6380`.