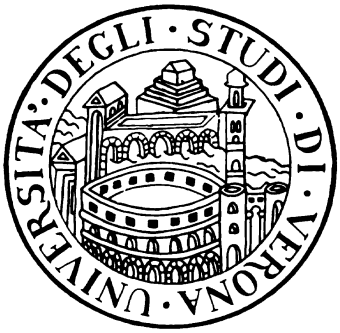
DOCUMENTAZIONE PROGETTO VITAS

A CURA DEI STUDENTI

GIOVANNI MENEGOZZO **ENRICO MAGNABOSCO**

CON LA SUPERVISIONE

PROF. PAOLO FIORINI DOTT. DIEGO DALL'ALBA



Introduzione:

Il progetto Vitas nasce da un tirocinio universitario presso l'università di Verona. Vitas è composto da un insieme di applicazioni software basate su reali test fisioterapici e da alcuni Serious Game.
Questa relazione ha l'intento di spiegare in dettaglio il codice sorgente che troverete su …(git-link)... .
Se volete conoscere le fasi di progettazione, l'obbiettivo e il target clinico di Vitas troverete nella stessa cartella github un file chiamato presentazione.pdf contenente ulteriori informazioni.

Hardware e software utilizzato:

Il Progetto è stato interamente sivuppato in unity 3D nella versione 4.8. L'utilizzo di versioni successive ne può compromettere il funzionamento.
Il sistema operativo utilizzato è Windows 8.1;
Si necessitano inoltre del dispositivo hardware Leap Motion e opzionalmente di Oculus Rift SDK 2.
L'interazione tra unity 3D, Leap Motion e Oculus è configurato tramite le rispettive sdk e i plugin Unity che troverete nel link sottostante.

Unity Leap Motion
Unity Oculus Rift

Organizzazione Guida:

Sebbene ogni script sia adeguatamente commentato abbiamo deciso di fornire un'ulteriore documentazione che dia l'idea di come è composta la scena e le interazioni tra Script.
Lavorando in ambiente grafico, infatti, si utilizzano molto utility e oggetti già prefrabbricati dall'ambiente che a volte abbiamo modificato a seconda dell'esigenza.
Divideremo la spiegazione del progetto di scena in scena, descrivendo tutte le classi e gli oggetti importanti (ossia quelli contenenti script) per la completa comprensione del funzionamento.

Scena Main Menù:

Funzinamento: L'utente inserisce i dati nei campi indicati, successivamente premendo prima il tasto 'save' e poi 'start' i dati inseriti vengono salvati in un file 'settings.txt' e una copia di backup viene salvata nella cartella 'BackupDatiPazienti'.

Script:
menuIniziale.cs: Crea la schermata da visualizzare dove l`utente inserisce i dati, li

salva e si preoccupa di farne una copia di backup. Dopo aver premuto 'start' chiama la scena successiva.

Oggetti rilevanti: Camera

Scena Ambiente Tuscany:

Funzinamento: È la scena principale in cui in ogni posto della casa abbiamo i cartelli che ci permetteranno di avviare i vari test.

Script:
start<test>.cs: Abbiamo uno script per ogni cartello che visualizza la scritta 'press E' quando ci si avvicina e avvia la scena desiderata.

Oggetti rilevanti: Cartello<test>

Scena Test Box and Blocks:

Funzinamento: L'utente deve afferrare gli oggetti e portarli dalla parte opposta della scatola in un determinato tempo.

Script:
GrabbableObject.cs: script assegnato a cubo che insieme allo script Grabbing hand assegnato alla mano facilita la presa degli oggetti. Questo script funziona tramite un ray-cast dalla mano al primo oggetto incontrato. Il Grabbing hand riconoscosce la posizione della mano controllando l'avvicinarsi di pollice ed indice. La funzione radius settata all'inizio del gioco setta la distanza del raggio minima affinche avvenga la presa dell'oggetto tra mano e oggetto più vicino individuato tramite ray-cast.

Punteggio.cs: è lo script che conta i punti controllando la posizione dell'oggetto al rilascio, inoltre si occupa della stampa del punteggio su file e su schermo.

TornaHome.cs: è lo script su camera per tornare alla schermata d'inizio dopo aver premuto il tasto 'Q'

SpawnCubiRandom.cs: Crea i cubi in modo casuale a seconda del numero scelto dall`utente nella schermata iniziate.

TrackingManoDestra.cs: si occupa del tracking della mano destra, salva 5 volte al secondo le posizioni x,y e z della mano.

TrackingManoSinistra.cs: si occupa del tracking della mano sinistra, salva 5 volte al secondo le posizioni x,y e z della mano.

Oggetti rilevanti: CuboPiccoloDaPrendere, HandController, Camera, TavoloDaGioco

Scena Test Star:

Funzinamento: L'utente deve toccare le stelle per circa un secondo e successivamente le stelle selezionate verranno distrutte, ha un tempo massimo di esecuzione in cui deve riuscire a cancellare il maggio numero di stelle.

Script:
GeneratoreFig.cs: si occupa di istanziare le stelle di diverse dimensioni (stellaGrande, stellaPiccola) nello spazio in maniera casuale, evitando la sovrapposizione.

Collider.cs: cancella l'oggetto a collisione avvenuta, aggiorna e stampa il punteggio, stampa su file e genera suono e luci. Stampa la posizione delle stelle rimaste sulla scena in un file di log

TrackingManoDestra: si occupa del tracking della mano destra, salva 5 volte al secondo le posizioni x,y e z della mano.

TrackingManoSinistra: si occupa del tracking della mano sinistra, salva 5 volte al secondo le posizioni x,y e z della mano.

Oggetti rilevanti: Tutte le parole sono oggetti “Text” a se stanti generati e posizionati manualmente, camera, estrellica

Scena Test Bart (Bilateral Arm Reaching Test):

Funzinamento: L'utente deve raggiungere con la mano il pulsante verde e tornare nella posizione iniziale in un determinato tempo che viene inserito nella schermata iniziale.

Script:
Bart.cs: Questo script inizializza 2 vettori che poi combina per creare i punti sulla scena. I punti creati corrispondono ad un arco tra i 10° e 170° composti da 4 file distanziati di 5 cm. Controlla che i bottoni non appaiano due volte nella stessa posizione
I bottoni vengono chiamati tramite la funzione invoke a seconda del valore impostato dall'utente.

ProvaBartCollider.cs: Si occupa dei punteggi e della stampa su file

TrackingManoDestra: si occupa del tracking della mano destra, salva 5 volte al secondo le posizioni x,y e z della mano.

TrackingManoSinistra: si occupa del tracking della mano sinistra, salva 5 volte al secondo le posizioni x,y e z della mano.

Oggetti rilevanti: redButton, greenButton, Camera

Scena Serious Game Lego:

Funzinamento: le istruzioni sono presenti sul gioco, l'utente deve ricostruire la figura data rispettando colori e forma.

Script:
spawnOggetti.cs: crea gli oggetti a seconda della difficoltà selezionata da utente

provaCollaider.cs: crea dei lego del colore del bottone premuto

ColliderLego.cs: Stampa il tempo. Permette ad un lego una volta afferrato di atterrare in maniera naturale bloccando la rotazione. Si occupa di posizionare il blocco nella posizione desiderata dall'utente a seconda del tasto premuto.

GrabbableObject.cs: è lo script assegnato a cubo che insieme allo script Grabbing hand assegnato alla mano facilita la presa degli oggetti. Questo script funziona tramite un ray-cast dalla mano al primo oggetto incontrato. Il Grabbing hand riconoscosce la posizione della mano controllando l'avvicinarsi di pollice ed indice. La funzione radius settata all'inizio del gioco setta la distanza del raggio minima affinche avvenga la presa dell'oggetto tra mano e oggetto più vicino individuato tramite ray-cast.

Oggetti rilevanti: legoBrick, HandControllerSandBox, Sphere, TorreSingola