UNIVERSITAT DE BARCELONA

Treball de fi de grau

App mòbil per a la realització de tests psicològics a infants

Autor:

Nicolás Forteza Ocaña

Professor:

Eloi Puertas Prats

 $22~{\rm de}$ Juny2017

Abstract

Resume

The study of children's interaction with technology can be very helpful to improve the process of learning through new media. This dissertation wants to create a tool so that pedagogues and psychologists can study how the youngest children react to the stimuli of the tablets. Through a face recognition test, records will be kept of all the gestures made on the screen.

This application is designed to run on both IOS and Android, thanks to Web technologies and the Ionic and PhaserJS frameworks.

Resum

L'estudi de la interacció dels nens amb la tecnologia pot ser molt profitós per millorar l'aprenentatge amb nous mitjans. Aquest treball vol donar una eina perquè pedagogs i psicòlegs puguin estudiar com els nens més petits reaccionen als estímuls de les tauletes tàctils. Mitjançant un test de reconeixement de cares, s'ira fent un registre de tots els gestos que faci sobre la pantalla.

Aquesta aplicació està pensada per executar-se tant en IOS com en Android, gràcies a les tecnologies web i els frameworks Ionic i PhaserJS.

Resumen

El estudio de la interacción de los niños con la tecnologia puede ser muy provechoso para mejorar el aprendizaje con nuevos medios. Este trabajo quiere dar una herramienta para que pedagogos i psicólogos puedan estudiar cómo los niños más pequeños raccionan a los estímulos de las tabletas táctiles. Mediante un test de reconocimiento de caras, se irá haciendo un registro de todos los gestos que haga sobre la pantalla.

Esta aplicación está pensada para ejecutarse tanto en IOS como en Android, gracias a las tecnologías Web i frameworks los frameworks Ionic i PhaserJS. Moltes gràcies a totes les persones que han fet possible aquest treball. A l'Eloi Puertas, per ser el meu mentor. A la Mariona Grané i la Lucrezia Crescenzi, per oferir-me els seus recursos i el seu temps. I als meus amics i familiars pel seu suport.

Índex

Introducció	1
Introducció i motivació	1
Objectius	1
Planificació	2
Estructura Memòria	2
Anàlisi del Problema	3
Requisits de l'aplicació	3
Descripció del test	4
Treballs relacionats	5
Disseny de la solució	6
Tecnologies	6
Disseny aplicació	7
Implementació	10
Gestió d'usuaris	12
El test	14
Proves i resultats	21
Observacions de les proves i errors detectats	21
Conclusions i treball futur	23
Conclusions	23
Treballs Futurs	24
Annexe	27

Introducció

Introducció i motivació

L'evolució que s'ha sofert en l'última dècada ha portat a l'expansió i la democratització de la tecnologia. Actualment quasi cada casa té una o més tauletes tàctils, tant per a treball com per ús recreatiu. Aquest avenç ha fet que, fins i tot, els infants puguin accedir a aquests recursos com si fos una joguina més. Però és possible utilitzar aquests aparells, no tan sols com una diversió, sinó que també com una eina per aprendre de forma interactiva? La Mariona Grané i la Lucrezia Crescenzi, membres del Laboratori de Mitjans Interactius, porten ja un bon temps treballant amb aquestes eines per poder estudiar i millorar l'aprenentatge dels més joves.

L'aplicació que es presenta aquí, vol servir com una nova eina per a ajudar a la investigació en la interacció entre els infants i les noves tecnologies interactives.

Objectius

El nostre objectiu és crear una aplicació per a tauleta tàctil, on s'enregistrin les interaccions que l'infant fa amb ella. Necessitem generar un estímul digital i ho farem mitjançant un test de memòria a curt termini.

El test consisteix en mostrar durant uns segons la fotografia d'una cara a la pantalla. Posteriorment, es mostra una imatge on surti un grup de persones i l'infant ha de trobar la cara anteriorment mostrada entre el grup de persones. En paral·lel, s'enregistrarà tots els tocs que faci en pantalla de forma detallada.

L'aplicació haurà de tenir la capacitat de mantenir diversos usuaris i que es puguin fer tests amb diferents infants, tot guardant els corresponents resultats. A més, es demana que sigui compatible amb diversos dispositius i sistemes operatius.

Planificació

Inicialment es va planificar començar fent unes demostracions dels tests al navegador. Seguidament, s'havia de crear tota l'aplicació i implementar un petit servidor per guardar els usuaris. Es pretenia acabar el més aviat possible per començar les proves amb els usuaris.

La planificació real ha sigut diferent. Les demostracions del test al navegador es van dur a terme en els terminis establerts. Tanmateix, el servidor fou cancel·lat per l'augment de la complexitat que suposava. L'aplicació havia de funcionar sobre les tabletes i els ordinadors Mac, dels quals no es va disposar fins a final de Maig. Es va haver de fer una migració completa del codi de Ionic 1 a Ionic 2. Les imatges per fer els tests no estigueren disponibles fins a principis de Juny, cosa que va endarrerir la finalització dels tests. Aquests contratemps han fet que la finalització de l'aplicació fos molt més tard, impedint així les proves amb usuaris.

Estructura Memòria

Aquesta memòria està dividida en sis parts, essent el primer la introducció.

En el segon capítol es presenta l'anàlisi del problema. Així doncs, analitzarem els requeriments que se'ns ha demanat que compleixi l'aplicació. Entrarem en detalls de com es demana que sigui la gestió d'usuaris i dels tests. Revisarem també alguns articles sobre el tema que aquí presentem: la interacció dels nens amb la tecnologia.

En el tercer capítol es parla del disseny de l'aplicació. Explicarem el funcionament del flux general de l'aplicació, tot detallant els tests i la gestió d'usuaris.

En el quart capítol es presenta la implementació. Aquí profunditzarem en les funcionalitats que compleix l'aplicació, així com comentar les decisions que s'han dut a terme durant el desenvolupament.

En el cinquè capítol es presenten les proves i els resultats. Podrem contrastar si realment l'aplicació és efectiva i assoleix els seus objectius.

En l'últim capítol, les conclusions, analitzarem el resultat final d'aquest treball, veient així si s'ha complert amb tots els requisists establerts, i es plantejarà el treball futur que es pot construir sobre aquesta aplicació.

Anàlisi del Problema

A diferència d'altres treballs de final de grau, aquest s'ha portat a terme conjuntament amb unes investigadores. Així, no només es tractava d'investigació, si no també de satisfer les necessitats com si d'un client es tractés. Les investigadores ens van presentar una sèrie de requisits i necessitats que l'aplicació havia de complir. Els presentem a continuació i els detallem.

Requisits de l'aplicació

Primerament, ens plantegen que l'aplicació ha de ser adaptable, és a dir, ha de funcionar en diversos dispotitius mòbils i també diferents sistemes operatius. En el seu centre disposen d'IPads i ordinadors amb sistema IOS, però tenen pensat adquirir mòbils amb Android en un futur. Així doncs, aquest requeriment és crític i és un punt clau en el disseny de l'aplicació.

Un cop aclarit la necessitat de fer una aplicació multiplataforma, ens expliquen els requisits que aquesta com a tal ha de complir.

Es vol implementar un test per infants menors de quatre anys; així doncs, l'aplicació no pot ser vistosa, ja que els nens tendeixen a tocar allò que els suscita curiositat.

El tests consisteix en mostrar unes imatges durant uns segons, i per tant, l'aplicació ha d'estar preparada per suportar el pès d'aquestes.

El test ha de controlar els encerts i els errors de l'infant, haurem de ser capaços d'enregistrar aquesta informació.

També és important tots els gestos que el nen fa durant el test, és per això que haurem de dissenyar algun mecanisme capaç de reconèixer i portar un compte dels comps que es prem i es dibuixa sobre la pantalla i de quina forma es fa.

Es vol guardar tota la informació associada a un usuari i per tant es demana que hi hagi una gestió dels mateixos. Aquesta informació s'ha de poder enviar a un ordinador i es desitja que això es fagi en un procés automàtic.

Finalment, es demana que l'aplicació sigui configurable oferint canviar el nom de l'examinador i canviar el temps del test.

Descripció del test

En aquest treball no s'ha dissenyat el funcionament del test, si no només implementat el que se'ns ha demanat. Les investigadores estan fent una feina de recerca i és per això que és molt important que seguim el que ens han demanat.

Mostrem a l'usuari la cara d'una persona, la imatge perdura uns segons.

Seguidament mostrem un grup de persones, d'entre les quals l'usuari ha d'identificar a la que ha sigut prèviament ensenyada, la imatge perdura uns segons.

Així, definim les pantalles:

- La pantalla d'estímul: on es mostra la cara de la persona que s'haurà de buscar.
- La pantalla de test: on es mostra el conjunt de persones.

Com ja hem esmentat abans, el temps que dura cada pantalla ha de ser prèviament fixat per l'examinador.



Figura 1: Exemple d'estímuls



Figura 2: Exemple de test

Sobre la quantitat dels tests és necessari saber:

L'aplicació constarà d'un total de 17 nivells, de manera que el primer sigui per a que l'usuari es pugui familiaritzar amb el seu funcionament.

Els 16 nivells restants estan agrupats en quatre grups.

El test finalitza si se superen tots els nivells o si es fallen els 4 nivells sota un mateix grup.

La intenció final del test, no és únicament veure si l'usuari és capaç de relacionar correctament les cares, si no que també es vol fer un seguiment de tots els gestos que faci. Per així poder fer una anàlisi més exhaustiu del test, i l'examinador pugui fer conclusions més acurades.

Treballs relacionats/anteriors

Trobarem aquest tipus de tests de memòria a curt termini amb cares en els tests d'intel·ligència, com per exemple en les bateries de test creades per Leehey i Cahn o el K-ABC. En el nostre cas, tanmateix, no tenim l'objectiu de mesurar les capacitats d'un infant, si no que tenen una finalitat de recerca. És a dir, nosaltres pretenem observar de quina manera les TICs poden ser útils en el procés del test. Així, per exemple, podem registrar més detalls que en un test en paper o, addicionalment, podem detectar nous patrons que amb els mitjans usuals no es pot.

D'altra banda, també es vol poder contrastar amb estudis fets sobre la interacció de nens amb les tauletes i els gestos que són capaços de fer segons la seva edat. Un exemple d'això és l'estudi que va fer Nor Azah Abdul Aziz, en Children's Interaction with Tablet Applications: Gestures and Interface Design.

Disseny de la solució

Tecnologies

Com ja s'ha introduït a l'anàlisi la intenció és fer que l'aplicació es pugui executar en diferents plataformes mòbil com Android o IOS, encara que prioritzant en IOS.

Això es podria fer de dos formes: creant una aplicació nativa en cada plataforma, copiant en cada una d'elles tots els dissenys i processos, o crear una aplicació hibrida, on solament s'hagi de crear un únic disseny i única implementació per a totes les plataformes. Ja que fer una aplicació nativa per a cada plataforma és una gran càrrega de treball, es decideix fer l'aplicació hibrida.

Després d'investigar respecte a totes les tecnologies híbrides disponibles, es va decidir utilitzar el framework Ionic, per a fer tota l'aplicació, combinant-ho amb PhaserJS per totes les pantalles de test.

Ionic és un SDK (software development kit) per al desenvolupament d'aplicacions mòbil hibrides. Es basa en AngularJS 2 i Apache Cordova(un altre conegut framework per crear aplicació hibrides). Ionic permet crear les aplicacions mòbils utilitzant tota mena de tecnologies Web, com HTML5, CSS o JavaScript. I amb el pas de la versió 1 a la versió 2 d'Ionic s'han aplicat diferents millores com: TypeScript, un llenguatge de programació evolucionat de JavaScript que inclou organització en classes i definició de variables amb tipus; millora en l'execució en els dispositius i tambe ha millorat en la organització de directoris, cada pantalla te el seu directori per separat.

PasherJS és un framework per crear contingut interactius (com jocs) amb HTML5. Es basa en un altre framework que es diu Pixi.js. Permet explotar al maxim motors de render per web com WebGL o Canvas i es soportat en tota mena de dispositius. Es pot executar tant al navegador com compliar-ho per a ser executat per a dispositius mobils o ordinadors.

S'ha elegit PhaserJS per que permet carregar imatges amb facilitat i per que ens permet

fer un seguiment detallat de totes les interaccions. Permet saber coordenades exactes dels tocs a la pantalla i relacionar-les amb el pixel que te a sota.

Disseny aplicació

Primer de tot hi haurà una pantalla inicial (home) que ens permetrà tant iniciar un test, amb un botó gran, com navegar pel menú d'opcions.

Aquest menú d'opcions ens ha de permetre, tant tornar a la pantalla d'inici, com anar a la pantalla de gestió d'usuaris (UserList) i la de configuració de l'aplicació (Settings).

A la pantalla de gestió d'usuaris hi haurà llistats de tots els usuaris creats en aquest dispositiu. Dins del perfil de cada usuari s'emmagatzemarà, a banda de la informació personal com el nom i l'edat, també es guardarà tots els gestos fets en cada test i una taula d'estadístiques generades de cada test.

A banda s'ha de permetre seleccionar els usuaris a fer el test, poder eliminar un usuari i enviar per correu totes les dades d'un usuari. També cal que es puguin crear nous usuaris, obrint així una nova pantalla on s'ha d'omplir un formulari amb totes les dades (CreateUser).

A la pantalla de configuració podrem canviar diverses opcions del test, com els temps de durada de les pantalles.

A banda, també tenim 2 pantalles més: la de test, que s'accedeix des de la pantalla inicial, i la de finalització de test (EndTest) que s'era la pantalla posterior al test i que portarà directament a la inicial.



Figura 3: Flux de les pantalles de l'aplicació

Pantalla de test

Les pantalles de test han de ser com hem descrit abans. Hi han d'haver 2 parts per cada nivell: una pantalla d'estímul, on solament es veurà en gran la cara que es vol memoritzar, i un altre de test, on s'ha de trobar la cara abans exposada. El test conte 17 nivells, sent el primer de prova. Els 16 nivells restants es divideixen en blocs de quatre, si es fallen tots els nivells d'un bloc, es finalitza el test (tal com es veu en el diagrama).



Figura 4: Flux del test

Es vol detectar el que fa l'usuari durant la prova, tant en les pantalles d'estímul com en les de test. Primer es defineix quines zones es volen detectar que l'usuari passa el dit pel damunt. En la pantalla d'estímul, com solament és la cara en gran i el fons de pantalla, definirem 3 zones diferents de touch: els ulls, la cara i altres(la resta de la pantalla). En el cas de la pantalla de test, en ser més petita la cara, no detectarem els ulls. En aquest cas detectarem cara, cos i altres, on en altres, també inclourem les altres persones que apareixen a la fotografia. Per donar com a superat un nivell l'usuari haurà de clicar a la cara correcta.

Respecte als gestos detectats en definirem cinc:

- Tap: Quan cliquem en un mateix punt durant any de 1.5 segons.
- Pres: Al contrari del tap, quan cliquem en un mateix punt durant més de 1.5 segons.

- Stroke: Quan tracem una línia recta amb el dit sobre la pantalla.
- Curve: Quan tracem una corba simple amb el dit sobre la pantalla.
- Altres: gestos més complexos que no es poden classificar com els anteriors.

En finalitzar el test, sortirà una pantalla on podràs guardar o no el test. Quan es guarda, internament, haurà de generar dues taules de resultats. La primera serà una taula amb tots els gestos fets durant el test, amb columnes com el nivell de test, el tipus de pantalla, tipus de gest, temps de durada, zona de la fotografia pressionada o si és resposta correcte al test.

La segona taula serà de resultats generals del test on s'expondran dades com: duració del test, duració mitjana de cada pantalla, puntuació, freqüència mitjana de tocs per pantalla, percentatge de tocs segons la zona de la pantalla o percentatge de gestos de cada tipus.

Implementació

Primer de tot, s'ha de posar en marxa generant el projecte, que en el cas d'Ionic amb una simple comanda ja generem tots els directoris i arxius necessaris. Inicialment el projecte estava fet amb Ionic versió 1. Quan ja es tenia bastant avançat el projecte, es va decidir fer una migració a Ionic versió 2, per certs problemes amb el funcionament de l'aplicació.

En Ionic versió 2, millora en moltes característiques, com Angular 2 en lloc d'Angular 1, millor organització dels projectes, utilització de TypeSricpt i millora del rendiment en general.

S'han creat sis pantalles diferents en l'aplicació: la pantalla inicial, la de gestió d'usuaris, la de creació d'usuaris, la de configuració, la de test i la de finalització del test.

Com s'ha mostrat en el diagrama d'abans es vol que tant la pantalla d'inici com les d'usuaris i de configuració estiguessin interrelacionades. Per això s'han posat en totes aquestes pantalles un menú lateral per a navegar entre elles.

En la pantalla inicial, a banda del desplegable del menú, només hi haurà un gran botó per iniciar el test. Aquest botó estarà bloquejat fins que no hi hagui un usuari seleccionat, des de la llista d'usuaris.



Figura 5: Pantalla Home amb el menú deplegat

Per guardar estats com l'usuari seleccionat, tenim un servei darrere de l'aplicació al que anomenem "DataProvider". Aquest servei proveeix tota la informació que necessita les pantalles i s'encarrega de la persistència de les dades. Quan s'executa l'aplicació, el "DataProvider" és l'encarregat de carregar tota la informació guardada en memòria. També té els mètodes que es criden per guardar tots els canvis nous a la memòria. El "DataProvider" guardar la llista de tots els usuaris amb la seva informació, a més també guardarà tot el registre de gestos entre d'altres.

Altra informació que es guardarà en "DataProvider" serà la de configuració de l'app. Aquestes dades seran editables des de la pantalla de configuració i estan formades per:

- Nom de l'examinador: Aquest serà el nom que es posarà en les taules de dades com qui ha supervisat la prova.
- Temps d'estímul: temps que s'expondra la pantalla d'estímul.
- Temps de test: temps màxim que s'expondra la pantalla de test.
- Temps després d'encert: temps que es mantindrà la pantalla una vegada encertat el test.

Sense SIM 😨	6:16 Settings	81% 🔳 🗲
	Settings	
Name of examinator Nico		
Time stimulus(ms) 10000		
Time test(ms) 10000		
Time post clicked face(1000		
	Save	

Figura 6: Pantalla Settings

Gestió d'usuaris

Uns dels objectius més importants que es van marcar per l'aplicació era poder gestionar tots els usuaris amb els seus tests. En la pantalla de gestió d'usuaris es pot fer tot això.

Com a primera funcionalitat hi ha l'opció d'afegir un nou usuari. Quan es clica en aquesta opció, s'obre una nova pantalla on s'ha d'omplir un petit formulari amb el nom, el cognom i la data de naixement. Una vegada omplert es pot clicar al botó de confirmació. En prémer aquest botó es generarà internament el perfil d'usuari amb la informació donada i generarà de nova, com un identificador d'usuari, generat amb la data de naixement i les inicials. Just després l'afegirà a la llista d'usuaris i guardarà els canvis a memòria.

Sense SIM 주	0:50	93% 📼
< Back	Create User	
Name Aina Lastname Ferrer		
Birthday		01/01/2014
	Add user	



A banda de crear nous usuaris, també podem fer accions sobre els ja creats. Podem seleccionar l'usuari per fer el test amb ell, amb la icona del tick.

També esborrar un usuari. Que per si un cas, ens demanarà per una confirmació per a executar l'acció. En la implementació del fet d'esborrar tenim en compte els diferents casos en esborrar. Per exemple, si eliminem l'element que tenim seleccionat, desselecciona l'element i s'ha de seleccionar un de nou.

I per últim, també podrem enviar tota la informació de l'usuari via e-mail. Una vegada premem el botó ens apareixerà una finestra emergent on podrem completar la informació de l'e-mail. L'e-mail vindrà amb el nom de l'usuari, i adjunta, tota la informació i taules dels tests realitzats. Això ho farem gràcies al plugin per ionic Email Composer, que permet construir tot un e-mail i obrir-lo amb l'aplicació per defecte de correu.



Figura 8: Pantall List users

El test

Com ja s'ha esmentat en la part de disseny, es va decidir utilitzar PhaserJS en la pantalla de test per les facilitats que té a l'hora de carregar imatges i detectar tota mena de tocs a sobre d'aquestes. La flexibilitat d'aquest Framework permet construir-lo damunt de qualsevol pantalla d'Ionic, sense cap problema en el moment de desplegar l'aplicació al dispositiu mòbil.

Durant la fase inicial del desenvolupament, es va crear un projecte web per testejar el funcionament de Phaser, gràcies al fet que està basat en tecnologies HTML5. Això va permetre des d'un bon principi centrar el desenvolupament en les algorítmies i la implementació del test, sense la necessitat de crear cap aplicació mòbil. Des del navegador web s'ha creat l'estructura inicial del test, posteriorment es va iniciar el projecte de l'aplicació mòbil i es va dur a terme la integració del projecte de Phaser amb Ionic.

Inicialment, per plantejar l'estructura de pantalles del test d'una forma intuïtiva, creant una nova pantalla cada vegada que es vol carregar una imatge. Però s'havia d'anar amb compte, ja que cada vegada que s'instància una nova pantalla aquesta s'afegeix a la pila de pantalles, i al llarg del test es poden generar fins a 34 pantalles diferents (17 nivells amb 2 pantalles cada un). Però segons la documentació d'Ionic, es pot netejar la pila cada vegada que crides a una pantalla nova (fent que aquesta pantalla sigui la primera de la pila). Es va implementar aquesta solució a l'aplicació, que no va causar cap problema durant el testeig de l'emulació ni amb el desplegament amb Android, però quan s'executava el test en dispositius IOS, el programa es parava inesperadament. Després d'una intensa depuració del codi es va detectar que la memòria de pantalles no es netejava correctament en IOS.

Aquest descobriment va obligar a canviar l'estructura de com es plantejava el test. Es va passar d'utilitzar 34 pantalles a solament una. A continuació explicarem l'algoritme que es va implementar en aquest últim cas.

Algorisme general

Primer de tot, i a causa de no tenir pantalles independents, és necessari controlar l'estat de la prova. Aquest estat estarà definit pel nivell actual i la pantalla (estímul o test). A partir d'aquí es podrà saber quines imatges toquen carregar i amb quina resolució (ja que la disposició de les imatges d'estímul i les de test és diferent).

A partir d'aquí, s'activarà un temporitzador que per finalitzar la pantalla en un temps determinat i es començarà a capturar tots els gestos dels usuaris. Cada vegada que un usuari toqui la imatge amb el dit, s'enregistrarà tots els punts per on passi el dit fins que el deixi anar. Una vegada succeeixi això, s'afegirà a una llista de gestos amb la informació amb el tipus de gest (tap, stroke, curve, etc.), la zona en la qual s'ha iniciat (cara, ulls, etc.), durada i, en el cas que sigui la pantalla test, si és resposta correcte o no.

Quan salti el temporitzador o, en la pantalla test, es faci un toc sobre la cara objectiu, finalitzarà aquesta pantalla. Aleshores, si aquesta pantalla era d'estímul, saltarà a la pantalla test corresponent. En cas que sigui la pantalla test, si és l'última imatge de totes o és l'última d'un bloc en què s'han fallat els 4 tests, anirà a la pantalla de finalització de la prova. En altre cas continuarà amb la imatge d'estímul següent.

En finalitzar el test, sortirà una pantalla on es permet guardar els resultats si es vol. Aquest resultat són dues taules en format csv. La primera és la llista de tots els gestos recollits durant la prova. La segona és un resum general de tots els resultats, traient estadístiques respecte als gestos i el resultat.

Definició de les zones de l'imatge

Com s'ha decidit durant el disseny, cada pantalla estarà dividida en tres zones diferents. En el cas de la pantalla d'estímul es dividirà en cara, ulls i altres. I en el cas de la pantalla de test, en cos, cara i altres.

Per definir aquestes seccions, una de les formes més senzilles trobades és crear un groundtruth. Un groundtruth és una imatge de resolució igual a la que vol fer referencia, on hi ha on s'omplen amb colors plans les àrees dels objectes que es volen representar, tenint així un color per cada element a classificar.

En aquest cas, les classificacions que s'han fet ha sigut: blanc, per a la cara, gris, per als ulls o el cos, i negre, per a la resta.



Figura 9: Pantalla estimul



Figura 10: Groundtruth de la pantalla estimul

Aleshores, el procediment a seguir per saber on ha clicat l'usuari s'era el següent.

Mirem les coordenades de la imatge on s'ha produït el gest. Agafem aquestes coordenades i busquem el píxel del groundtruth al que equival. El color d'aquest pixel, ens marcarà com classificar el gest.



Figura 11: Pantalla estimul



Figura 12: Groundtruth de la pantalla estimul

Detecció de gestos

Com s'ha dit en el disseny, hi ha cinc grups per classificar els gestos: tap, press, stroke, curve i others.

Gràcies a l'objecte Pointer de PhaserJS, s'han pogut detectar amb més facilitat els gestos que feia l'usuari. Pointer té un atribut intern que diu la distància recorreguda pel gest actual, això permet classificar en dos grups: si es concentra al voltant d'un punt (tap i press) o si es desplacen per la pantalla (stroke, curve i other).

Quan tenim que es concentren en un punt solament cal mirar un altre atribut de Pointer, la durada. Si dura més de 1.5 segons, aleshores serà un press, en cas contrari, ser un tap.

En el cas que el gest sigui més gran, utilitzarem un sistema bastant senzill per classificarlos. Mitjançant el producte escalar entre el vector del punt inicial al punt final i el vector del punt inicial al punt central, podem accedir al cosinus de l'angle entre els dos vectors. Això en donarà un valor entre 1 i -1. Suposarem si el valor és més petit de 0.75 que forma una corba (curve). En el cas contrari, es classificarà com a recta (Stroke).

Sortida dels resultats

A finalitzar el test es generen dues taules de valors en format CSV: un dels gestos fets i l'altre de resultats de la prova.

Sense SIM ᅙ	9:05	100% 🕮 🗲
	endTest	
Save Don't	save	

Figura 13: Pantalla EndTest

La taula de gestos vol ser un registre detallat de tots els gestos fets durant la prova, així els examinadors poden repassar la sessió amb exhaustivitat. Aquesta taula consta de les següents columnes:

- ID: identificador de l'usuari que fa la prova
- Nombre de mesos d'edat
- Dia de la prova
- ID prova: Pers i es vol posar algun identificatiu a la prova
- Qui li ha fet la prova
- Tipus de gest (Press, Tap, Stroke, etc.)
- Nivell al qual pertany
- Duració del touch (en mil·lisegons).
- Zona touch (Cara, ulls, etc.)

- Temps des de l'estímul
- Resposta: Si és resposta correcte per superar el test.

La taula de resultat presentarà certes estadístiques respecte a l'evolució de la prova. Aquesta taula solament crea una fila cada vegada que es fa una prova. Està pensada perquè es vagin acumulant totes les proves que fa un mateix usuari i així poder comparar resultats. Aquesta taula consta de les següents columnes:

- ID: identificador de l'usuari que fa la prova
- Nombre de mesos d'edat
- ID prova: Pers i es vol posar algun identificatiu a la prova
- Qui li ha fet la prova
- Dia i hora de l'inici de la prova
- Dia i hora del final de la prova
- Duració total de la prova
- Mitja durada per nivell
- Desviació estàndard de la durada.
- Puntuació: nombre de nivells superats satisfactòriament
- Puntuació cronològica: Nivells que hauria d'haver superat segons la seva edat.
- Freqüència mitjana d'errors a la pantalla de test
- Freqüència mitjana de tocs a la pantalla de test
- Percentatge de tocs en els ulls en la pantalla estimulo
- Percentatge de tocs a la cara en la pantalla test
- Percentatge de tocs en "altres" en totes les pantalles.
- Temps fins al primer touch.

- Temps mitja entre l'estímul i el primer touch de cada nivell.
- Percentatge de taps
- Percentatge de Press
- Percentatge de Strokes.

Proves i resultats

S'ha intentat, que durant tot el procés de desenvolupament, hi hagués una retroalimentació per refinar el producte. Mitjançant reunions puntuals s'ha fet control de l'evolució de l'aplicació per veure que estigues ben dirigida.

Les proves s'han fet de diverses formes durant el desenvolupament. Gràcies al fet que s'utilitzen tecnologies web a l'inici de la producció es va fer una demo en el navegador de com funcionaria el test. Això va permetre que ja es pogués discutir possibles correccions al funcionament que veiessin les investigadores interessades.

Quan ja vam iniciar el desenvolupament de l'aplicació en Ionic, vam poder començar ja a fer proves en els dispositius. Primer en un dispositiu android personal, i posteriorment en un IOS cedit pel LMI (Laboratori de Mitjans Interactius). Posteriorment, i fins a l'actualitat, s'ha utilitzat Testflight pel testeig.

Testflight és un servei online de Apple que permet descarregar i testejar aplicacions en IOS. El desenvolupador de l'aplicació pot invitar a gent a un grup de testers i aquests poden descarregar l'aplicació i fer comentaris respecte a forma molt senzilla.

A banda de les proves que s'han fet fins ara es vol mantenir un parell de mesos més la retroalimentació per millorar l'aplicació. I així les investigadores poden fer algunes proves amb infants.

Observacions de les proves i errors detectats

Durant les proves hem detectat diferents errors en l'algorisme. El més sonat podria ser que la classificació dels gestos en corba funciona en molt pocs casos. Sembla que l'objecte Pointer de Phaser no registra de forma uniforme tots els punts per on passa el gest, el que fa que l'angle que formen els vectors amb què calculem sigui pròxim a zero.

Altres problema que hem detectat és que si es fan tests d'un mateix usuari en el mateix

dipositiu tots els resultats estan en una mateixa taula, en canvi, si es fan els test en dispositius independents, els resultats estan en taules separades.

Una cosa que s'ens ha comentat al veure l'aplicació és la falta de retroalimentació en el test. L'usuari no te cap indici visual de que ha clicat on toca.

Conclusions i treball futur

Conclusions

Un cop explicat el gruix del treball, hem de dir que, en termes generals, s'han complert els objectius del projecte. El requisit d'adaptabilitat, un dels majors reptes que ha presentat aquesta aplicació, ha sigut plenament assolit donant la possibilitat d'executar l'aplicació en un navegador, en dispositius mòbil i tabletes Android o IOs i en ordinadors que disposin de Linux, Windows o Mac. A més, existeix la possibilitat d'exportar els tests per a que es pugui fer un executable de forma externa a la pròpia aplicació.

D'altra banda, tota la gestió dels usuaris s'ha dut a terme correctament, encara que el servidor fou cancel·lat ja que suposava una construcció des de zero, implementant la seva pròpia API Restful i control de versions. Addicionalment, requeria de manteniments i coneixements que elevava molt el nivell d'ús de l'aplicació, així que es va decidir apartar-lo de moment.

A canvi d'això, s'ha implementat l'opció de poder enviar tota la informació mitjançant un correu electrònic. Així, s'ha construit un sistema d'organitzar la informació dins de la pròpia aplicació, estructurar-la, transformar-la i generar de manera automàtica un correu per enviar-lo. Això ha sigut molt profitós per les investigadores, ja que l'aplicació s'encarrega d'obrir i adjuntar per sí mateixa els arxius necessaris, de manera que l'examinador no necessita tenir pràcticament cap coneixements d'informàtica.

Aquest treball de recerca pot resultar poc vistós degut a que l'aplicació com a producte final no és especialment maca o pot semblar que el flux de funcionament no és essencialment difícil. Tot i així, hem de resaltar que la dificultat ha residit especialment en les tecnologies emprades. Aquest tipus de coneixements no s'adquireixen en el transcurs de la carrera i aquí he pogut apendre com fer una aplicació que sigui adaptable per qualsevol sistema operatiu. No sols això, si no que hem de tenir en compte que els recursos que utilitza (per exemple, la capacitat d'emmagatzematge) han de servir de la mateixa forma si l'aplicació s'executa des d'un ordinador, des d'un mòbil o des d'una tableta.

Així doncs, cal comentar que s'ha hagut de posar especialment èmfasi en l'optimització de la memòria. Es va veure, un cop vaig disposar dels dispositius Mac, que Ionic no gestiona correctament la memòria amb IOS i això va suposar un problema per finalitzar l'aplicació en els terminis imposats. Tot i així, recomanaria a altres programadors a utilitzar Ionic, ja que en les últimes versions s'ha desenvolupat més aquest framework.

Per el cas de Phaser, he trobat una limitació important, ja que no registra bé tots els punts dels gestos. Això pot provocar que no es classifiquin correctament els patrons, que és un dels objectius crítics de la nostra aplicació. Concretament, el més problemàtic ha sigut la classificació de la corba, que no ens retornava punts intermitjos per poder fer el producte escalar i classificar correctament.

Finalment, fem una mica de reflexió de perquè no s'han assolit alguns dels objectius que s'havien plantejat en aquest projecte. La relació que hem tingut amb les investigadores ha sigut, gairebé, una relació amb un client. Així, nosaltres teníem una dependència amb la informació i material que ens proporcionaven. De totes maneres, he pogut desenvolupar un tipus de projecte que em prepara i em dóna oportunitats de cara al món laboral; he pogut tastar els tipus de relacions amb clients que segurament tindré i he tingut la oportunitat de desenvolupar un projecte de principi a fi, amb els seus contratemps inclosos.

Treballs Futurs

De cara a noves iteracions, estaria bé plantejar com es poden assolir els objectius que aquí s'han deixat.

Com ja he explicat en les conclusions, millorar l'enregistrament dels gestos és crític. Estaria bé buscar nous mètodes de càlcul, que no requereixin un temps crític de resposta i que optimitzin la classificació dels gestos. Així mateix, estaria bé detectar figures més complexes. Tot això, és clar, és necessari que sigui en temps real.

D'altra banda, estaria bé trobar una alternativa al servidor. És clar que es necessita una gestió d'usuaris, o sigui que estaria bé disposar d'una eina per unificar els usuaris, tenir un recompte dels tests que s'ha fet amb cada un i poder emmagatzemar aquesta informació. Així, també es permetria fer tests en el mateix usuari però en dispositius diferents.

Durant el desenvolupament va sorgir la idea de reproduir un test ja fet. És a dir,

guardar versions simplificades dels gestos dels usuaris i reproduir-los. Així, l'examinador podria estudiar detalls que potser no ha vist en el mateix moment i treure conclusions més adequades. Des del punt de vista tècnic, tanmateix, això s'hauria de fer tenint cura de no ocupar massa memòria.

Mirant l'aplicació des del punt de vista de la usabilitat, una bona idea seria donar algun tipus d'informació positiva mentre es fa el tests. Així com estan implementats ara, l'usuari no sap si la resposta és bona quan clica la cara corresponent i això pot enterbolir el resultat final. De totes maneres, s'ha de comentar que la decisió de no donar aquesta informació fou presa per les investigadores i no ha sigut fins al termini de l'aplicació que s'ha tornat a rependre la idea.

Finalment, i a banda de tots els canvis que s'hi puguin fer, durant els pròxims dos mesos (juliol i agost del 2017) es donarà suport mitjançant el Testflight per a tots els possibles errors que sorgeixin.

Bibliografia

- Nor Azah Abdul Aziz Children's Interaction with Tablet Applications: Gestures and Interface Design International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 02– Issue 03, May 2013
- [2] Angular linkWhat is Angular? [en línia] 2017 [consulta: 3 de juny de 2017]. Disponible
 a: https://angular.io/docs
- [3] Documentation- TypeScript [en línia] 2017 [consulta: 3 de juny de 2017]. Disponible a: https://www.typescriptlang.org/docs/home.html
- [4] Ionic Framework Documentation [en línia] 2017 [consulta: 18 de juny de 2017]. Disponible
 a: https://ionicframework.com/docs/
- [5] Phaser 2.6.2 API Docs Learn Phaser [en línia] 2017 [consulta: 15 de juny de 2017].
 Disponible a: https://phaser.io/docs/2.6.2/index

Annex

Manual de l'usuari.

Com gestionar un usuari

Obrim el menú lateral i seleccionem User.

Per crear un usuari, clica en el botó + en la barra superior. Hauràs d'omplir les dades de l'usuari:

- Nom
- Cognom
- Data de naixement.

Clica en *add user* per confirmar la creació. Sota del user creat tenim 3 opció:

- Seleccionar usuari per fer el test
- Esborrar usuari
- Enviar totes les dades d'usuari per mail.

Fer el test

Per poder fer el test primer hem de seleccionar un usuari. Anem al menú Users i seleccionem l'usuari amb la icona **V**. Ens tornarà a la pantalla d'inici i podrem clicar en el botó gran per iniciar el test. El test s'iniciarà i hi hauràs de seleccionar la cara que veus inicialment. Una vegada finalitzat el programa ens donarà l'opció de guardar els resultats o descartarlos.

Configurar

En cas de voler canvia algun dels paràmetres disponibles del test Vés al menú i clica en *Settings*. Canvia els paràmetres que vulguis i després clica a *save*.

Instal·lació en Android

Primer de tot comprova que tinguis Node.JS instal·lat al teu ordinador. Instal·la Ionic executant: **npm install -g ionic cordova** Instal·la el JavaJDK i l'Android SDK Connecta el dispositiu Android via USB i activa la depuració per USB al dispositiu. Executa: **ionic cordova run android –device**