PROJECTE FI DE CARRERA

Títol: Pissarra Multitàctil

Autor: Jorge Novoa Portocarrero

Titulació: Enginyeria Tècnica en Informàtica de Gestió

Director: Marta Fairen Gonzalez

Departament: Llenguatges i Sistemes Informàtics

Data: 18/01/11

Índex de Continguts

1.Introducció....6

2.Objectius....7

2.1Descripció del Projecte....7

2.2Context del Projecte....7

2.3Definició d'Objectius....8

2.4Abast del Projecte....9

3.Anàlisi d'Antecedents i Alternatives....10

3.1Antecedents....10
3.1.1ReacTable....11
3.1.1.1Funcionament....11
3.1.1.2Cost....12
3.1.1.3Especificacions....12
3.1.2Jefferson Y. Han....13
3.1.2.1Productes....13
3.1.2.1.1Cost....14

3.2Alternatives de Mercat....15
3.2.1Promethean....15
3.2.1.1Tecnologia....15
3.2.1.2Software....16
3.2.1.3Cost....16
3.2.2Smart Technologies....17
3.2.2.1Hardware....17
3.2.2.2Software....18
3.2.2.3Cost....18
3.2.3ebeam....19
3.2.3.1Hardware....19
3.2.3.2Software....20
3.2.3.3Cost....20
3.2.4Mimio....21
3.2.4.1Hardware....21
3.2.4.2Software....22
3.2.4.3Cost....22
3.2.5Reflexió....23

4.Eines Utilitzades....24

4.1Software per al Processament d'Imatges i Detecció de Blobs....25
4.1.1BBTouch....26
4.1.2CCV (tBeta) - Community Core Vision....26
4.1.3ReacTIVision....27
4.1.4TouchLib....27
4.1.5Touché....28

4.1.6Funcionament del CCV....29
4.1.6.1Pantalla Principal....29
4.1.6.1.1Configuració....31
Consideracions FTIR:....31
4.1.6.2Pantalla de Calibratge....31
4.1.6.2.1Configuració....32

4.2Protocol TUIO....33
4.2.1Conceptes bàsics....34
4.2.2OSC....35
4.2.3TuioSimulator....36

4.3Llenguatge de programació....37

4.4Entorn de desenvolupament....38

4.5Patrons de Disseny de Software....39
4.5.1Patró Singleton....39
4.5.2Patró Observador....39

4.6Software per la Interfície Gràfica: Qt....40
4.6.1Signals i Slots....41
4.6.2Sistema d'Esdeveniments de Qt....42
Com es reben....42
Com s'envien....42

4.7Software de Reconeixement Gestual....43
4.7.1Grafiti....44
4.7.2AME Patterns Library....44
4.7.3Sparsh_UI....45

4.8Altres....46
4.8.1Microsoft Visio 7.0....46
4.8.2Open Office....46

5.Metodologia de Treball....47

5.1Planificació....48
5.1.1Qué és una iteració?....49
5.1.2Cicle de vida del RUP....50
5.1.3Reflexió....51

5.2C++ Coding Standards....52

5.3Nom dels Fitxers:....53

5.4Confecció del Subversion....54

6.Tecnologia Utilitzada....55

6.1Tecnologies Tàctils....55
6.1.1Superfícies Tàctils Basades en Resistència....56
6.1.2Superfícies Tàctils Basades en Capacitància....57
6.1.2.1Panell Tàctil Basat en Capacitància de Superfície....57
6.1.2.2Panell Tàctil Basat en Capacitància de Projecció....58
6.1.3Superfícies Tàctils basades en Superfície d'Ona (SAW)....59
6.1.4Superfícies Tàctils Òptiques (Optical Based Touch Surfaces)....59
6.1.4.1Frustrated Total Internal Reflection (FTIR)....60
6.1.4.2Diffuse Illumination (DI)....61

6.1.4.3Diffuse Surface Illumination (DSI)....62
6.1.4.4Laser Light Plane (LLP)....63

7.Prototipus....64

7.1Disseny....65
7.1.1Tecnologia utilitzada....65
7.1.2La Reflexió Total Interna Òptica....66
7.1.2.1La Reflexió de la Llum....66
7.1.2.2La Refracció de la Llum....66
7.1.2.3La Reflexió Total Interna [1]....66
7.1.3La Superfície de la Pantalla en Detall....67
7.1.3.1Difusor (pantalla de projecció posterior)....67
7.1.3.2Capa compatible....67
7.1.3.3Acrílic....67
7.1.3.4Webcam....67
7.1.3.5Projector....67
7.1.4El Circuit Elèctric....68
7.1.4.1LEDs Infrarojos....68
7.1.4.2Electrònica i electricitat....68
7.1.5Projector i Pantalla de projecció posterior (retroprojecció)....69
7.1.6Capa Compatible (Compliant Layer)....69

7.2Muntatge....70
3.1. Consideracions de Disseny....70
3.2. Materials utilitzats (el que necessitem)....70
7.2.1Tractament de l'Acrílic....71
7.2.2Pantalla de Projecció i la Capa Compatible....72
7.2.3Cablejat del circuit electrònic....74

8.Arquitectura del Sotfware....75

8.1Introducció:....75

8.2Casos d'Ús Significatius....75

8.3Vista Lògica....77
8.3.1Arquitectura del Software....77
8.3.2Vista....78
8.3.2.1La Recepció de les Dades....78
8.3.2.2L'Emissió de les Dades....79
8.3.2.2.1GestureEventHandler....80
8.3.2.2.2ScribbleArea....80
8.3.2.2.3Scribbling Action....81
8.3.2.2.4Esdeveniments Implementats....82
8.3.3Controlador....83
8.3.3.1Reconeixement Gestual....83
8.3.3.1.1FIB-GestureServer....84
8.3.3.1.1.1GestureServer....85
8.3.3.1.1.2Group....85
8.3.4Model....86
8.3.4.1Informació sobre els Esdeveniments del Sistema....86
8.3.4.2Informació relativa als Components....87
8.3.4.2.1Blackboard....87
8.3.4.2.2Toolbar....87

8.3.4.2.3Object....88
8.3.4.3Classes Bàsiques....89

9.Pla de Desenvolupament del Projecte....90

9.1Organització del Projecte....90
9.1.1Rols i Responsabiltats....91
9.1.2Suport....91

9.2Planificació del Projecte....92
9.2.1Elements a Implementar....92
9.2.2Dates Límit i Mínims del Projecte....92
9.2.3Planificació....93
9.2.3.1Etapes de la Implementació del Software....94
9.2.3.2Etapes de la Construcció del Prototipus....95
9.2.4Càlcul dels Costos....95
9.2.4.1Cost de l'Aplicació....95
9.2.4.2Cost del Prototip....96
9.2.4.3Cost i Dedicació Total....96

10.Conclusions....97

10.1Posada en Funcionament i Resultats....97
10.1.1Versió 1.0....97
10.1.2Versió 1.1....98
10.1.3Versió 2.0....102
10.1.4Versió 2.1 (versió actual)....104

10.2Viabilitat del Projecte....104

10.3Metodologies de Treball....104

10.4Línia de Treball a Seguir....105
10.4.1Software....105
10.4.1.1Extensibilitat....105
10.4.2Hardware....105

11.Agraïments....106

12.Referències Bibliogràfiques....107

12.1Capítol 3. Anàlisi d'Antecedents i Alternatives....107

12.2Capítol 4. Eines Utilitzades....107

12.3Capítol 5. Metodologia de Treball....108

12.4Capítol 6. Tecnologia Utilitzada....109

12.5Capítol 7. Prototipus....109

12.6Capítol 8. Arquitectura del Software i Extensibilitat....109

12.7Capítol 10. Conclusions....110

13.Annex. Plantilles Utilitzades....111

1. Introducció

En aquest projecte, anomenat Pissarra Multitàctil, es dissenya un sistema que permet mitjançant una pantalla tàctil, simular la utilització d'una pissarra tradicional en la que s'accepten varis punts de contacte del/s usuari/s de forma simultània. Amb aquesta pissarra es pretén aprofitar les noves tecnologies multitàctils (que ja s'han vist en altres tipus de dispositius com els telèfons mòbils), per a facilitar la tasca d'ensenyament a les aules.

Creiem que la configuració d'una pissarra és el suport ideal per a una pantalla d'aquest tipus, ja que és una superfície en la que poden escriure una o vàries persones al mateix temps amb els dits sense que aquesta posició sigui incòmode o estranya. La motivació d'aquest projecte és demostrar la viabilitat d'aquesta idea mitjançant la demostració del correcte funcionament de l'aplicació i la construcció d'un prototipus que sigui viable econòmicament (que és realitzarà en un projecte en paral·lel a aquest).

Donat que aquest projecte ha estat realitzat per dos estudiants, s'han distingit les dues parts en què cadascun ha treballat més per a realitzar els documents de les dues memòries per separat, tot i que la separació més exacte d'hores i feines ve reflectida en el capítol *Pla de Desenvolupament del Projecte*. En aquesta memòria es descriu més en detall la part tècnica i de construcció del prototipus, mentre que la part d'aplicació i funcionalitats del projecte s'explica amb més detall a la memòria del meu company Edmon Martínez Martí.

2. Objectius

2.1 Descripció del Projecte

Es vol dissenyar una primera versió d'una aplicació que funcioni amb una pantalla tàctil que permeti la interacció de varis punts de contacte de forma simultània. L'objectiu és potenciar el funcionament de la pissarra de guix tradicional aprofitant les possibilitats que donen les noves tecnologies multitàctils (varis usuaris al mateix temps, utilitzar varis dits de la mà) i les clàssiques tecnologies multimèdia, oferint d'aquesta forma, una manera eficient de treballar sense la necessitat de requerir dispositius addicionals (com reproductors de vídeo, televisors, etc).

El projecte té una vessant Software, que inclou l'aplicació que permet actuar a partir dels gestos de l'usuari a la pissarra, i una vessant Hardware, que inclou el disseny i construcció d'un prototipus i els drivers necessaris per a permetre la interacció desitjada per a l'usuari.

2.2 Context del Projecte

L'aplicació s'ha pensat per fer-se servir en l'àmbit de l'educació (primària, secundària, superior, etc.), malgrat podria utilitzar-se en un gran ventall d'àmbits. Segons això, els usuaris seran principalment professors i alumnes. És important tenir present que els professors no tenen perquè tenir coneixements i/o habilitats informàtiques.

2.3 Definició d'Objectius

Els objectius que s'han marcat en aquest projecte són els següents:

- Facilitar la tasca dels docents.
  - El dispositiu funcionarà igual que una pissarra tradicional (una vegada executada l'aplicació).
  - Es podran obrir continguts multimèdia en la superfície de la pissarra, moure'ls i redimensionar-los com es vulgui en qualsevol moment.
  - Aconseguir major eficiència en la gestió del temps (no hi haurà necessitat d'altres dispositius).
  - Evitar la necessitat de coneixements previs avançats d'informàtica.
  - Permetre la utilització d'Internet i altres tipus d'aplicacions mitjançant accions tàctils sobre la pissarra.
  - Augmentar la interactivitat professor-alumne en les classes.

- Cost econòmic.
  - Estalviar en la compra de dispositius externs com podrien ser vídeos, reproductors de música, projectors, aules especials, etc.
  - Oferir una plataforma flexible que permeti la utilització d'aplicacions existents tàctils pensades per funcionar en qualsevol tipus de sistema operatiu i/o dispositiu tàctil diferent al nostre.
  - Estalviar en la formació dels usuaris del dispositiu.
  - Estalviar en el manteniment de l'aplicació. És important que el sistema sigui consistent per a que el manteniment sigui mínim, les actualitzacions hauran de fer-se de forma automàtica.

- Aconseguir la independència de la tecnologia i Software existent.
  - El Software ha de ser totalment independent de la tecnologia utilitzada i del sistema operatiu amb que funcioni la màquina on s'executi. Això vol dir que l'aplicació que es dissenyarà ha de tenir la possibilitat de poder-se executar sobre qualsevol sistema existent (per exemple, una pantalla que utilitzi un llapis d'infrarojos).

- Construcció del prototipus (paral·lelament a l'aplicació).
  - Estudiar la seva viabilitat.
  - Experimentar amb funcionament de la tecnologia utilitzada (FTIR - Frustrated Total Internal Reflection).
  - Comprovar la cohesió del sistema (Hardware i Software funcionant junts).

- Proves amb el prototipus.
  - Mesurar el temps de resposta.
  - Mesurar el rendiment de l'aplicació segons la quantitat de punts actius a la pantalla.

- Permetre extensibilitat.
  - L'aplicació a realitzar és un Software d'un prototipus, on es voldrà demostrar el potencial que podria tenir. Això implica que aquesta haurà de ser dissenyada tenint present que el producte final serà una base sobre la que treballar.

2.4 Abast del Projecte

Aquest projecte es podria arribar a estendre molt més enllà del que es requereix per a un treball de final de carrera, així que ja des de l'inici s'ha establert quins serien els límits del projecte per a poder realitzar-lo en el temps disponible, tenint en compte que en la construcció de l'aplicació només hi participem dos persones. Com s'ha explicat en l'apartat anterior, l'elecció dels casos d'ús que implementarem s'ha realitzat pensant en demostrar el potencial que podria tenir aquest Software.

L'abast d'aquest projecte es defineix per poder fer:

A. Una aplicació que permet a l'usuari poder fer les següents accions sobre el dispositiu

   I. Escriptura.

   II. Esborrar escriptura mitjançant dos dits.

   III. Permetre a varis usuaris escriure i/o esborrar al mateix moment.

   IV. Obrir una imatge a la pantalla.
      a. Moure-la mitjançant un moviment d'arrossegament amb un dit.
      b. Redimensionar-la mitjançant el moviment de dos dits (separant-los o ajuntant-los).

B. La construcció d'un prototipus que permeti comprovar la viabilitat del projecte.

3. Anàlisi d'Antecedents i Alternatives

En aquest capítol es fa un anàlisi dels antecedents al projecte i les actuals alternatives existents al mercat. A l'hora d'escollir els antecedents a explicar es decanta pel tipus de projectes que han influenciat més en aquest i a tots els relacionats amb superfícies multitàctils en general. En quan a les alternatives de mercat, ens centrem en els sistemes que s'estan implantant a Catalunya segons la informació obtinguda del Departament d'Ensenyament de la Generalitat de Catalunya [4].

3.1 Antecedents

Parlarem dels antecedents que han estat importants no només en aquest projecte, sinó també per la resta de projectes relacionats amb superfícies multitàctils, això vol dir aquelles superfícies que poden detectar varis punts de contacte al mateix temps.

En primer lloc, tenim al ReacTable, implementat per estudiants de la Universitat Pompeu i Fabra. Consisteix en un instrument de música sobre una superfície horitzontal que capta els diferents moviments i rotacions d'uns objectes *marcats* amb els quals generem les notes. No és un projecte pensat per a superfícies tàctils però si que incorpora elements similars com són el tema de varis usuaris a la vegada, tractament visual de la imatge rebuda a l'ordinador per a transmetre les posicions i coordenades a l'aplicació, etc.

Després tindrem a Jefferson Han, considerat el redescobridor del fenomen FTIR per a ser utilitzat en superfícies tàctils, per a implementar sistemes de captació de varis punts de contacte al mateix temps.

3.1.1 ReacTable

És un instrument musical electrònic col·laboratiu dotat d'una interfície tangible en forma de taula inspirat en els sintetitzadors modulars dels anys setanta. Va ser desenvolupat pel Grup de Tecnologia Musical de la Universitat Pompeu Fabra de Barcelona, per un equip d'investigació integrat per Sergi Jordà, Marcos Alonso, Günter Geiger i Martin KaltenBrunner. Múltiples usuaris simultanis comparteixen el control total de l'instrument movent i rotant objectes marcats sobre la superfície d'una taula circular lluminosa. Manipulant aquests objectes, els quals representen els components clàssics d'un sintetitzador modular, els usuaris poden crear tipologies sonores complexes i dinàmiques, mitjançant generadors, filtres i moduladors, en una classe de sintetitzador modular tangible.

Figura 1: ReacTable (versió d'exposició)

3.1.1.1 Funcionament

La ReacTable consta d'un tauler semitranslúcid il·luminat directament amb dos càmeres situades a sota que analitzen de tant en tant la superfície i segueix els moviments, la posició, i l'orientació dels diferents objectes físics i lògics que estan situats en el tauler per medi de la visió artificial. Varis músics simultanis comparteixen el control complet sobre l'instrument movent i rotant els objectes en el tauler. Al moure i relacionar els objectes per la superfície es modifica l'estructura i els paràmetres del sintetitzador de so. Aquestes objectes conformen els típics mòduls d'un sintetitzador modular. Simultàniament, el projecte mostra l'activitat i les característiques principals del so produït, donant-li d'aquesta forma la necessària retroalimentació a la persona que està executant els moviments. D'aquesta forma varis músics poden compartir el control desplaçant i rotant les fitxes transparents sobre la taula. Cada un dels usuaris pot crear una funció sonora diferent.

Entre altres coses, ReacTable intenta ser:

- Col·laboratiu: manipulat per varies persones a la vegada
- Intuïtiu: sense manual, sense instruccions
- Ser un instrument que emet una sonoritat interessant i desafiant
- Fàcil d'ensenyar i d'aprendre (inclús per nens)
- Apte per a novells i per a músics electrònics avançats

Entre les seves característiques:

- Pot arribar a fer el seguiment d'uns 20 objectes al mateix temps
- Funciona amb FTIR (Fustrated Internal Reflection) i DI (Diffused Ilumination) indistintament.
- Utilitza l'aplicació Reactivision per a fer el seguiment dels *blobs* per la pantalla.
- Utilitza per primera vegada el protocol TUIO per comunicar l'aplicació de seguiment amb el dispositiu tàctil.

3.1.1.2 Cost

El preu de la versió per a exposicions, que és la que seria més semblant a aquest projecte puja a 9.700 Euros

3.1.1.3 Especificacions

El seu format és en mode Taula (de forma circular), les seves mides són:

- 85 cm de diàmetre
- 90 cm d'alçada

Funciona amb el sistema operatiu MAC OS X 10.5 (o alguna versió més actual).

3.1.2 Jefferson Y. Han

Jefferson Han és un científic de recerca de la universitat de Nova York, i un dels principals desenvolupadors de *multi-touch sensing* (sensibilitat multitàctil), és a dir, superfícies capaces de detectar varis punts de contacte a l'hora. Va presentar el seu treball al febrer de 2006 a la conferència de TED (Technology Enternainment Design) a Monterey, California. TED va llençar sis mesos després un vídeo seu a Youtube que va tenir un alt seguiment. Han va ser el fundador de la companyia anomenada Perceptive Pixel[7] per desenvolupar tecnologia tàctil, i actualment ja ha instal·lat els seus dispositius en númerosos vehicles militars. Un dels seus majors èxits va ser l'anomenada *Magic Wall* (o MultiTouch Collaboration Wall) utilitzada per la CNN durant les eleccions del 2008 als EEUU.

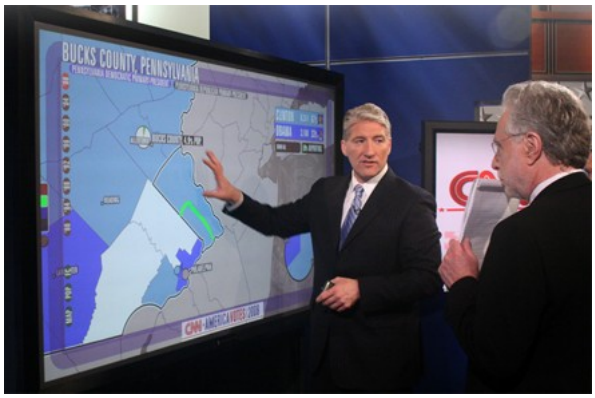

Figura 2: Magic Wall, CNN

3.1.2.1 Productes

	Multi-Touch LCD Display.	Multi-Touch Wall
Dispositiu	1920 x 1080 (Alta definició) 55 polzades de longitud diagonal LCD (IPS, In-Phase Switching)	1920 x 1080 (Alta definició) 88,5 polzades de longitud diagonal
Àrea tàctil activa	1210 x 680 mm	1960 x 1102,5 mm
Entrades tàctils possibles	Dits, llapis electrònic, guant especial	Dits, llapis electrònic, guant especial
Nombre de touchpoints	Il·limitat	Il·limitat
Precisió de cada contacte	Sub-pixel	Sub-pixel
Temps de resposta a cada contacte	<15ms (normalment)	<15ms (normalment)
Dimensions físiques	52 x 31.5 x 11.5 mm	2093 x 2083 x 1133 mm

3.1.2.1.1 Cost

Encara és un producte en fase experimental, només s'utilitza per esdeveniments especials, exposicions i casos molt específics, l'únic preu conegut és el preu de venta que va posar Neiman Marcus (centres comercials de luxe als EEUU) a un d'aquests dispositius:

Multi-Touch Wall: 100.000$

Els preus no estan publicats a la seva web ni s'ha pogut trobar cap informació a Internet, l'única forma de saber-ho és parlar directament amb ells i és d'imaginar que només faran pública aquesta informació en cas d'un clar interès de compra.

3.2 Alternatives de Mercat

Ens centrarem en la competència del nostre producte en el mercat, és a dir, els dispositius electrònics que són utilitzats en les aules d'ensenyament a Catalunya en l'actualitat, i quin és el seu cost i funcionalitats disponibles.

Ara mateix en les aules s'utilitza el que s'anomenen PDI (Pissarres Digitals Interactives). S'ha investigat quines són les marques contractades i s'ha fet un anàlisi del seu Hardware i Software (amb el seu cost) per tal de poder fer una comparativa amb el producte que estem intentant desenvolupar en aquest projecte.

3.2.1 Promethean

Companyia anglesa dedicada a la tecnologia interactiva per l'ensenyament, és a dir, ofereix el software i el dispositiu als seus clients. La seva base està situada a Blackburn, Lanchashire.

3.2.1.1 Tecnologia

El seu dispositiu es diu ActivBoard i en el seu model més avançat (ActivBoard 300 Pro) ofereix les següents possibilitats:

- Mode dos usuaris. S'utilitzen dos tipus de llapis electrònics especials.
- Mides: 78”, 87”, 95”
- Té dos ports USB
- Té dos altaveus i amplificadors integrats
- Opció de connexió sense cables (wireless)
- Garantia de 3 anys

Figura 3: Promethean ActivBoard 300 Pro

3.2.1.2 Software

- ActiveStudio. És un producte derivat de ActivePrimary
  - La principal característica és la de permetre a l'usuari preparar i presentar arxius mitjançant *flipcharts* (rotació de folis), que és un tipus de document que pot contenir una combinació de vectors i informació d'arrossegament de dades (línies, formes, text enriquit, vídeo, Flash Media, etc.)
  - Es poden fer anotacions en cada pàgina del *flipchart* mitjançant un llapis electrònic.
  - La versió 3 d'aquesta aplicació afegeix per primera vegada reconeixement gestual.

- ActiveInspire. És una actualització del ActiveStudio en resposta al Notebook v10 de Smart (veure *Apartat 3.3.2.2*), les seves característiques principals són:
  - És multiplataforma (Windows, Mac i Linux)
  - S'han afegit eines avançades com figures intel·ligents i reconeixement d'escriptura a mà
  - Millor facilitat per a la creació de lliçons mitjançant plantilles temàtiques
  - Compatibilitat total amb Flash per a arxius .flv i objectes Flash
  - Es pot importar arxius de SMART i PowerPoint
  - Es pot exportar preguntes i respostes a format Excel pel seguiment ràpid del procés de l'aprenentatge.
  - Integració a la comunitat docent via Web de Promethean (Promethean Planet)

- Altres.
  - ActivPrimary. És practicament idèntic al ActivStudio però pensat per classes de primària, la diferència recau en la interfície.
  - ActivStudio SE (Student Edition). Una versió amb funcionalitat reduïda.
  - ActivStudio Viewer. Només per visualitzar els flipcharts de l'ActivStudio.
  - ActivPrimary Viewer. Només per visualitzar els flipcharts de l'ActivPrimary.
  -

3.2.1.3 Cost

- Hardware. Els preus oscil·len entre els 1800$, 1400$ i 1200$ segons el tamany de la pantalla que escollim[9].

- Software. La versió més actual, ActiveInspire, es pot descarregar de forma gratuïta a la web oficial de Promethean[8].

3.2.2 Smart Technologies

Són els desenvolupadors de les SMART Boards, i són originaris de Calgari, Alberta, Canada. En aquest cas la pantalla és tàctil, tracta els *tocs* de la pantalla com si fossin esdeveniments de ratolí o de teclat. La majoria d'aquestes PDI només son capaces de treballar amb un punt cada vegada a la pantalla, però al Juny del 2009 es va introduir la capacitat de detectar-ne dos, això si, en llocs separats i definits de la superfície de la pissarra.

Aquests dispositius accepten *inputs* a partir dels dits, un llapis digital especial o un objecte sòlid. Cada contacte amb la PDI és interpretat com un clic amb el botó dret del ratolí. La projecció sol ser frontal malgrat hi ha versions amb LCD o plasma.

3.2.2.1 Hardware

- DVIT. És un Software de Detecció per Visió utilitzat per Smart Technologies (veure capítol sobre *Tecnologies Utilitzades*), és a dir, pertany a una altra companyia. Utilitza una càmera per a detectar i situar cada contacte realitzat sobre la superfície tàctil (sigui un dit o un dels llapis especials).
- SmartBoard Pen Tray. El programari relacionat amb aquestes PDIs està preparat per a l'escriptura manual. En aquest cas els llapis i/o l'esborrador no tenen cap component electrònic, la tecnologia es troba a la safata on desem aquests elements. Cada vegada que agafem un llapis, un sensor òptic reconeix la seva absència, el Software de la SmartBoard processarà el següent contacte amb superfície de la pissarra d'acord al llapis que correspon a l'slot buit. Cada slot es pot programar amb un color diferent al nostre gust. En cas d'haver-se agafat varis llapis, sempre escollirà l'últim extret.
- Tecnologia Resistiva. Les series 600, incorporen tecnologia resistiva (mirar capítol de *Tecnologies Utilitzades*).

Figura 4: SmartBoard 685ix

3.2.2.2 Software

Smart Notebook. Es pot fer servir en els sistemes operatius Windows, Mac i Linux i és una plataforma de visualització de continguts similar al Microsoft PowerPoint, és a dir, està dissenyada per a que els professors dissenyin les seves pròpies lliçons a partir de les eines i recursos disposats al programa. En el cas que ens toca, El Departament d'Ensenyament de la Generalitat té varis repositoris amb unitats didàctiques ja dissenyades per als professors.

3.2.2.3 Cost

El cost variarà entre els preus que hem posat a continuació, segons el model escollit i característiques:

- Hardware
  - SBD685ix. Versió més avançada, que és capaç de fer el seguiment de dos punts a la pantalla.
    Preu: 6199$ [11]
  - SB680. Smart Board més bàsica.
    Preu: 1999$ [11]

- Software
  - Es pot descarregar gratuïtament de la web[10] , però només una versió de prova que funcionarà durant 30 dies, per desbloquejar-lo caldrà utilitzar un codi que estarà escrit en la PDI que ens obligaran a comprar.

3.2.3 ebeam

És una PDI desenvolupada per Luidia que transforma qualsevol pissarra tradicional en una pissarra interactiva, cosa que la presenta com una de les opcions més econòmiques del mercat.

El sistema ha estat incorporat a altres fabricants com:

- 3M. Digital Board
- Hitachi. StarBoard
- Legamaster. Eboard

3.2.3.1 Hardware

Ebeam utilitza receptors d'infrarojos i ultrasons per fer un seguiment d'un llapis transmissor. Una unitat receptora acoblada a un dels costats de la pissarra, paret o superfície on vulguem escriure i determina la distància i la direcció d'aquest llapis utilitzant a la vegada els dos tipus d'informació que captarem (llum i so).

Podem comprar els sistemes ebeam en tres modalitats:

- ebeam projection: S'inclou el Hardware i Software per utilitzar en una imatge projectada de la pantalla de l'ordinador. El detector capta el que fem amb el llapis especial inclós i ho utilitza com si fos un ratolí convencional.
- ebeam whiteboard: S'inclou el Hardware i Software per utilitzar amb els *marcadors* (guixos, rotuladors, etc.) de les pissarres convencionals. En aquest cas no hi ha cap imatge projectada (aquest sistema només se centra en la detecció de l'escriptura). El detector captarà els moviments que fem amb els marcadors mitjançant un dispositiu electrònic que posarem al seu voltant, registrant d'aquesta forma tot el que s'ha escrit a la pissarra en el nostre ordinador.
- ebeam complete: Simplement els dos sistemes en el mateix *pack*.

El detector es pot connectar al nostre ordinador utilitzant USB o Bluetooth (opció que haurem d'escollir abans de comprar-ho).

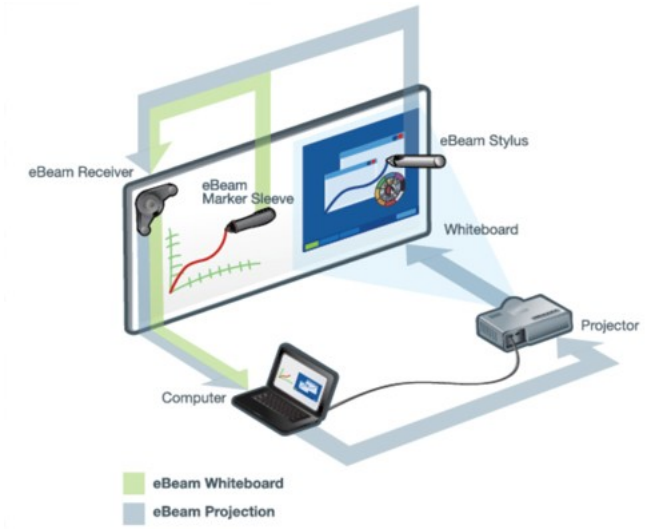

Figura 5: eBeam

3.2.3.2 Software

El Software relacionat amb aquest producte no està tan desenvolupat com en els casos anteriors, malgrat tot tenim:

- Ebeam scrapbook. És el Software que ens permetrà desar la nostra feina. Ens proporciona un fons on podrem dibuixar, escriure, importa imatges i tot el que sigui necessari. Noves pàgines podran ser afegides i al final de la lliçó podrem desar-ho tot i exportar-ho a format pdf, pàgina web, bmp, jpg, tif, emf, ppt, o pps.
- Software per permetre escriptura: EverNote
  S'ha de comprar a part, només pot escriure un usuari amb el llapis cada vegada.
- Altres. Existeixen eines específiques per a:
  - Ser utilitzades amb fitxers de tipus PowerPoint.
  - Fer una captura de la pantalla per a ser desada.
  - Fer una gravació de les accions realitzades i passar-ho a format .avi (útil per a mostrar la solució a un problema).

3.2.3.3 Cost

El principal avantatge d'aquest sistema és la facilitat per utilitza grans superfícies en contrast amb els sistemes que hem vist anteriorment.

Amb connexió USB [15]

- ebeam projection : 749.95$
- ebeam whiteboard: 749.95$
- ebeam Complete: 899.95$

Amb connexió Bluetooth [15]

- ebeam projection : 949.95$
- ebeam whiteboard: 949.95$
- ebeam Complete: 1099.95$

3.2.4 Mimio

Mimio és la marca associada a una sèrie de productes destinats al mercat de l'educació, entre ells, els que s'encarreguen de captar els moviments que realitzen els usuaris sobre una superfície a l'hora d'escriure. Quan aquests dispositius son utilitzats junt amb un projector es pot transformar una pissarra tradicional en una pissarra digital interactiva.

Els productes relacionats amb Mimio van ser originalment dissenyats per Virtual Ink, que era una companyia fundada pel MIT (Massachusetts Insitute of Technology) a l'any 1997, en resposta al concurs llençat per aquesta mateixa universitat "$50K Contest", un concurs pensat per potenciar l'esperit empresarial dels seus estudiants.

3.2.4.1 Hardware

La pissarra interactiva d'aquesta companyia utilitza una combinació d'ultrasons i rajos infrarojos per a situar els punt en el pla (cal utilitzar un llapis especial). És un sistema que utilitza la mateixa idea que l'ebeam però desenvolupat de forma totalment independent.

Al igual que amb l'ebeam, podem aconseguir aquest sistema de PDI en tres modalitats:

- Mimio interactive. Equivalent a "l'ebeam projection".
- Mimio capture. Equivalent a "l'ebeam whiteboard" però no s'inclou el hardware (el detector bàsicament). Una diferència amb el sistema ebeam és que podem registrar el que s'ha escrit a la pissarra a una memòria flash, sense la necessitat de tenir connectat un ordinador a la pantalla com en el cas de l'ebeam.
- Mimio interactive + capture. Els dos anteriors junts.

En aquest cas es pot connectar el detector via USB o Bluetooth. També podem fer-ho via wireless si comprem el mòdul de Hardware addicional.

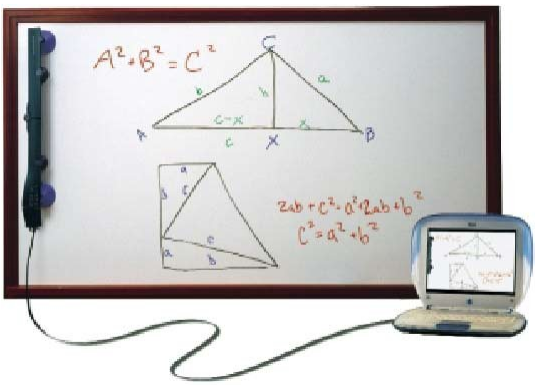

Figura 6: Mimio

3.2.4.2 Software

En el cas de l'ebeam vam comentar que el seu Software no estava tan desenvolupat com en els casos de SMART i Promethean (que compten amb molts recursos ja implementats), en el cas de Mimio la situació teòricament encara és pitjor, segons la documentació que hem pogut consultar (està en procés de desenvolupament, és possible que s'hagin pogut llençar actualment més recursos relacionats). En el cas de Mimio tenim:

- Mimio Notebook. Similar a la versió per ebeam.
  - Diferències:
    - Visualment és més pobre, té una estructura similar a la de Windows (el fa més fàcil de fer funcionar).
    - Es pot fer una captura de només una secció de la pissarra.
- Software de reconeixent d'escriptura: Recognise Ink
  Inclòs en el Notebook, s'ha d'escriure amb el llapis i només es pot una persona a la vegada.

3.2.4.3 Cost

- Mimio Interactive. 729$
- Mimio Capture. 249$
- Mimio Interactive + Capture. 699$

Dades obtingudes a partir de www.shoolmart.com[19]

3.2.5 Reflexió

Després de realitzar tota la investigació, no s'ha pogut trobar quin és el criteri del departament per triar el sistema de PDI a utilitzar, dóna sensació de aleatorietat. Això ha provocat que hi hagi presents en les aules nombrosos models de diferents marques, cosa que provoca que s'hagi d'utilitzar obligatòriament el seu Software i Hardware addicional, també afegeix més confusió i més hores de dedicació dels professors per a aprendre a utilitzar els dispositius amb tot el seu potencial, amb la qual cosa en alguns casos s'ha obtat per no utilitzar la PDI amb la qual cosa es perd totalment la inversió realitzada.

En general les pàgines web de les empreses que proporcionen aquest material són molt específiques amb el que fan els seus productes (en quan a Hardware i Software), a causa d'això s'ha hagut de consultar nombrosos blogs per tenir una opinió crítica del producte i obtenir una descripció acurada de que és el que aporta i quines són les seves mancances.

SmartBoard i Promethean tenen el Software més desenvolupat de tots els sistemes que s'han estudiat, i en contrast, es pot descarregar gratis des d'Internet. Basen els seus beneficis en la venta de dispositius electrònics (les seves PDI), cosa que contrasta amb la nostra idea de fer el Software sigui totalment independent del Hardware per a poder aprofitar qualsevol tipus de dispositiu preexistent (si és que aquests dispositius ho permeten, cosa que no està gens clara). Hi ha dos vessants clares en aquest mercat, un és el dispositiu tàctil (o interactiu) a utilitzar i l'altre és el Software, creiem que és un error voler fusionar les dos coses ja que limita molt les possibilitats.

La capacitat multiusuari és molt limitada o inexistent, depenent de la marca. En primer lloc no tots els dispositius permeten detectar varis contactes amb la superfície tàctil i els que ho permeten tenen limitada la quantitat a tres tocs simultanis o l'àrea en que poden treballar cada persona. Creiem que aquí hi ha un camp molt ampli d'on treure un clar avantatge empresarial respecte tots els competidors existents.

No totes les alternatives són tàctils, la major part d'elles tenen la necessitat d'utilitzar un llapis electrònic per a escriure o realitzar les accions que realitzaria una persona amb un ratolí al seu ordinador (l'opció d'escriptura manual ha estat introduïda recentment).

La majoria de dispositius emulen el ratolí, és important que es pugui fer això, però al poder treballar amb varis dits a l'hora o inclús amb les dos mans per fer accions a la pantalla es poden crear noves formes d'interaccionar amb els widgets més intuïtives. Hi ha moltes empreses punteres a l'estranger investigant sobre això, però de moment encara no s'ha llençat cap versió al mercat, només prototipus experimentals.

4. Eines Utilitzades

Si mirem la *Figura 1* podem deduir fàcilment totes les eines que ens caldran per a desenvolupar la nostra aplicació:

- Software per al *Processament d'imatges i Detecció de 'Blobs'* (Objecte brillant a la pantalla de dispositius multitàctils). Explicarem les possibilitats que teníem al nostre abast, quina hem escollit i perquè.
- Protocol de Transmissió de Dades. En aquest cas no hem hagut d'escollir entre vàries opcions ja que la majoria de sistemes similars utilitza el mateix, el protocol TUIO.
- Per a implementar la nostra aplicació necessitarem el següent:
  - Escollir el llenguatge de programació.
  - Escollir l'entorn de programació que utilitzarem.
  - Escollir el Software que utilitzarem per implementar la interfície gràfica per l'usuari.
  - Escollir el sistema de Reconeixement Gestual, per a interpretar el que volen fer usuaris.

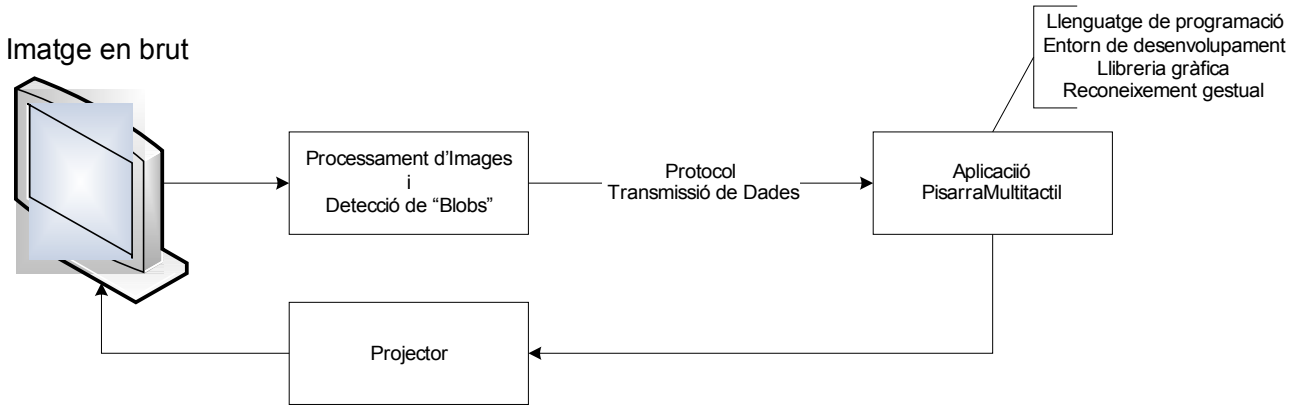

Figura 1: Estudi del procés realitzat al tocar la pantalla per trobar les eines necessàries.

Apart de les eines ja citades també ens caldrà:

- Triar un simulador TUIO que enviarà aquest tipus de missatges a la nostra aplicació simulant el comportament de la pantalla tàctil, per no dependre del dispositiu Hardware per a provar les funcionalitats que es vagin implementant (ja sigui perquè no estiguem al laboratori de treball o perquè no estigui finalitzat un prototipus funcional).
- Escollir l'editor de text que s'utilitzarà per escriure la documentació.
- Escollir una aplicació per a fer els diagrames de flux, d'UML, etc.

4.1 Software per al Processament d'Imatges i Detecció de *Blobs*

Per a poder fer una aplicació multitàctil basada en la visió d'una càmera, necessitarem un programa que agafi aquesta senyal de vídeo i ens doni coordenades dels punts de contacte dels dits sobre la pantalla tàctil:

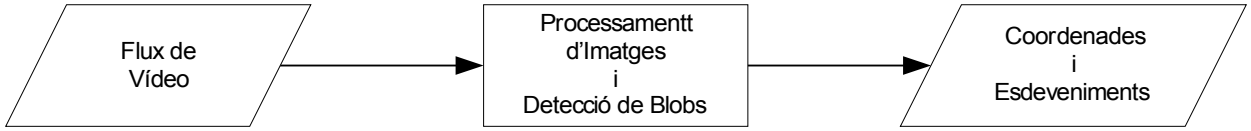

Figura 2: Entrades i sortides del sistema.

A partir de les coordenades i els seus esdeveniments corresponents (dit polsat, mogut o aixecat), la nostra aplicació interpretarà els gestos de l'usuari i farà les accions corresponents, això vol dir que es podrà utilitzar qualsevol aparell tàctil que ens aporti aquesta informació, l'aplicació és totalment independent de la tecnologia utilitzada. Arribats a aquest punt cal prendre decisions, ja que malgrat ser un camp molt verge, ja hi ha vàries opcions a triar:

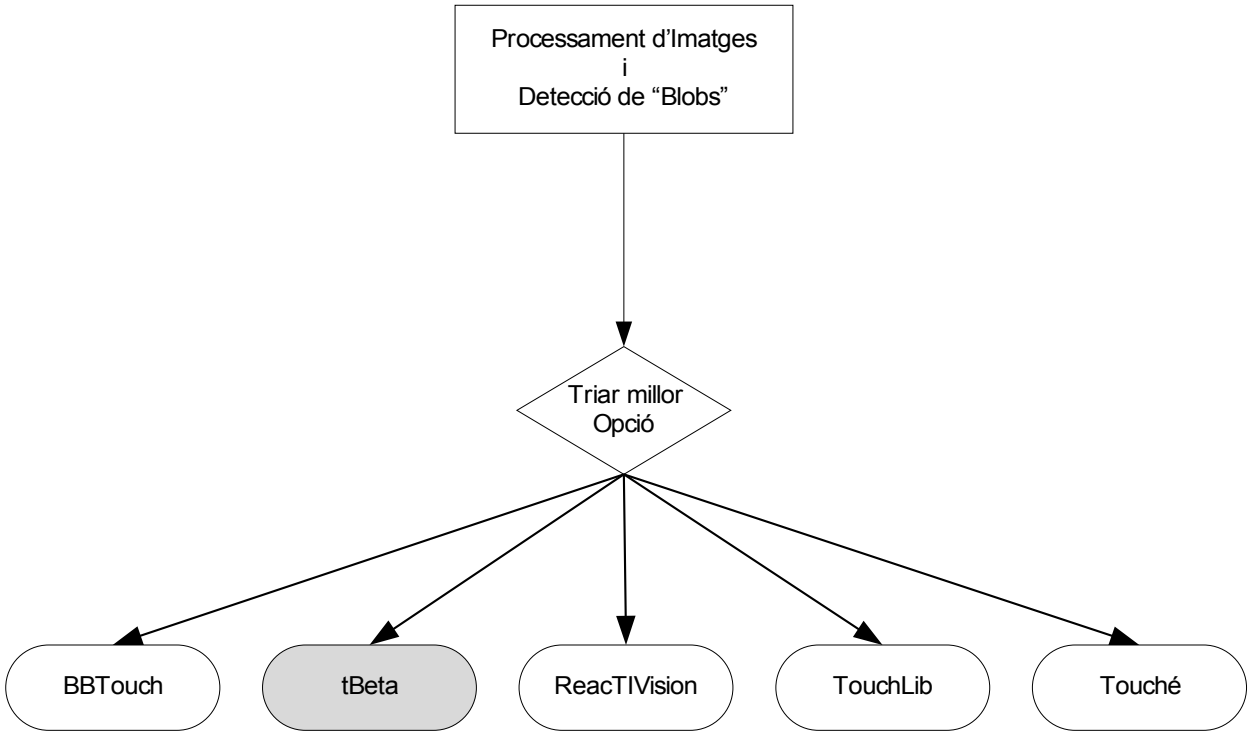

Figura 3: Diagrama de decisió sobre el Software de Processament d'Imatges i Detecció de *'Blobs'*

En la *Figura 3* hem fet un diagrama de flux amb la decisió amb la que ens hem hagut d'afrontar en aquell moment del transcurs del projecte, podem observar com hem marcat en vermell l'opció escollida, "tBeta", o com l'anomenarem durant el transcurs de la memòria CCV (abreviació de Community Core Vision). La idea principal d'aquest apartat de la documentació és veure com hem arribat a aquesta decisió. En els propers apartats s'expliquen les característiques de cada una de les opcions i el perquè s'ha escollit finalment el "tBeta" i s'ha descartat la resta.

4.1.1 BBTouch

Llenguatge de programació: Cocoa(Mac)
Plataforma: MAC OS X
Codi Font: Disponible
Llicència: GPL (General Public Licence)
Pàgina: http://benbritten.com/blog/bbtouch-quick-start/

Avantatges:

- És codi obert.
- Està molt ben documentat a la pàgina web indicada.
- Es poden enviar consultes al creador del sistema via mail.

Inconvenients:

- No estem familiaritzats ni amb el llenguatge de programació ni amb el sistema operatiu.
- No ens podem permetre treballar amb aquest tipus d'ordinadors (estan fora del nostre abast econòmic).
- Tot el que envolta les qüestions de Apple són molt tancades cap a un tipus de sistema operatiu, nosaltres volem que el nostre sistema sigui independent de la plataforma en que funciona i sigui utilitzable pel major nombre d'usuaris possibles.

DESCARTAT

4.1.2 CCV (tBeta) ***- Community Core Vision***

Llenguatge de programació: C++
Plataforma: Multiplataforma.
Codi Font: Disponible.
Llicència: MPL (Mozilla Public License) o MIT (Massachusetts Institute of Technology Lisence)
Pàgina: http://tbeta.nuigroup.com

Avantatges:

- Pot interaccionar amb diferents dispositius que transmetin dades mitjançant el protocol TUIO (això vol dir que podem rebre dades de diferents dispositius de captació d'imatges de vídeo).
- És independent del tipus de sistema d'il·luminació utilitzat (*FTIR*, DI, DSI o LLP).
- Actualment continua en desenvolupament per part dels membres d'una comunitat que treballa en tecnologies multitàctils (NUI Group).
- És multiplataforma (característica important tenint en compte els objectius del nostre projecte).
- Que estigui implementat en C++ i el codi estigui obert també és positiu de cara a possibles modificacions.

Inconvenients:

- Existeixen errades ja que és una versió “Beta” (no és un producte acabat), està implementada per membres de la comunitat NUI de forma totalment altruista (malgrat es van publicant noves versions cada vegada més perfeccionades).

OPCIÓ ESCOLLIDA

4.1.3 ReacTIVision

Llenguatge de programació: C++
Plataforma: Windows, MacOS X i Linux.
Codi Font: No disponible en la seva totalitat.
Llicència: GPL
Pàgina: http://reactivision.sourceforge.net/

Avantatges:

- Permet identificar la posició i rotació de patrons bidimensionals sobre una superfície. Va ser dissenyat dins el marc del projecte anomenat "reacTable" (veure capítol *Anàlisi d'Alternatives i Antecedents*).
- Està molt ben documentat a la pàgina web indicada.
- Pensada per treballar sota les tecnologies DI i FTIR.
- És multiplataforma.

Inconvenients:

- La seva principal característica no ens serveix, el rastreig segons els patrons d'un objecte (veure *Figura 9, Apartat 4.2.1*).
- Malgrat és molt positiu, no és tant complet com ho és el CCV.

DESCARTAT

4.1.4 TouchLib

Llenguatge de programació: C++
Plataforma: Windows
Codi Font: Disponible
Llicència: New BSD (Berkeley Software Distribution) License
Pàgina: http://nuigroup.com/touchlib/ http://code.google.com/p/touchlib/

Avantatges:

- És la llibreria més utilitzada pels que s'inicien en l'estudi de les tecnologies multitàctils.
- Inclou una aplicació de configuració amb algunes demos per ajudar a començar.
- Pot interactuar amb la majoria de Webcams i dispositius de vídeo.
- Actualment funciona només sota Windows però s'estan fent esforços per a què es pugui fer servir en altres plataformes.

Inconvenients:

- Només funciona sota el sistema operatiu Windows.
- L'aplicació CCV (o tBeta) és una versió millorada de la *TouchLib,* i per tant té més possibilitats i és més fàcil de fer servir.

DESCARTAT!

4.1.5 Touché

Llenguatge de programació: Cocoa (Mac)
Plataforma: MAC OS X
Codi Font: Disponible
Llicència: New BSD License
Pàgina: http://gkaindl.com/software/touche

Avantatges

- La qualitat del rastreig és molt alta gràcies a la utilització de les tecnologies: QuickTime, CoreAnimation, CoreImage i el framework Accelerate.

Inconvenients:

- No estem familiaritzats ni amb el llenguatge de programació ni amb el sistema operatiu.
- No ens podem permetre treballar amb aquest tipus d'ordinadors (estan fora del nostre abast econòmic).
- Tot el que envolta les qüestions de Apple són molt tancades cap a un tipus de sistema operatiu, nosaltres volem que el nostre sistema sigui independent de la plataforma en que funciona i sigui utilitzable pel major nombre d'usuaris possibles.

No s'ha investigat molt a fons degut a les limitacions trobades des d'un primer instant.

DESCARTAT

4.1.6 Funcionament del CCV

Com a opció escollida, a continuació coneixerem de forma resumida, les seves característiques. Una vegada s'executa l'aplicació tBeta, només hem de treballar amb 2 pantalles: la pantalla principal i la pantalla de calibratge.

4.1.6.1 Pantalla Principal

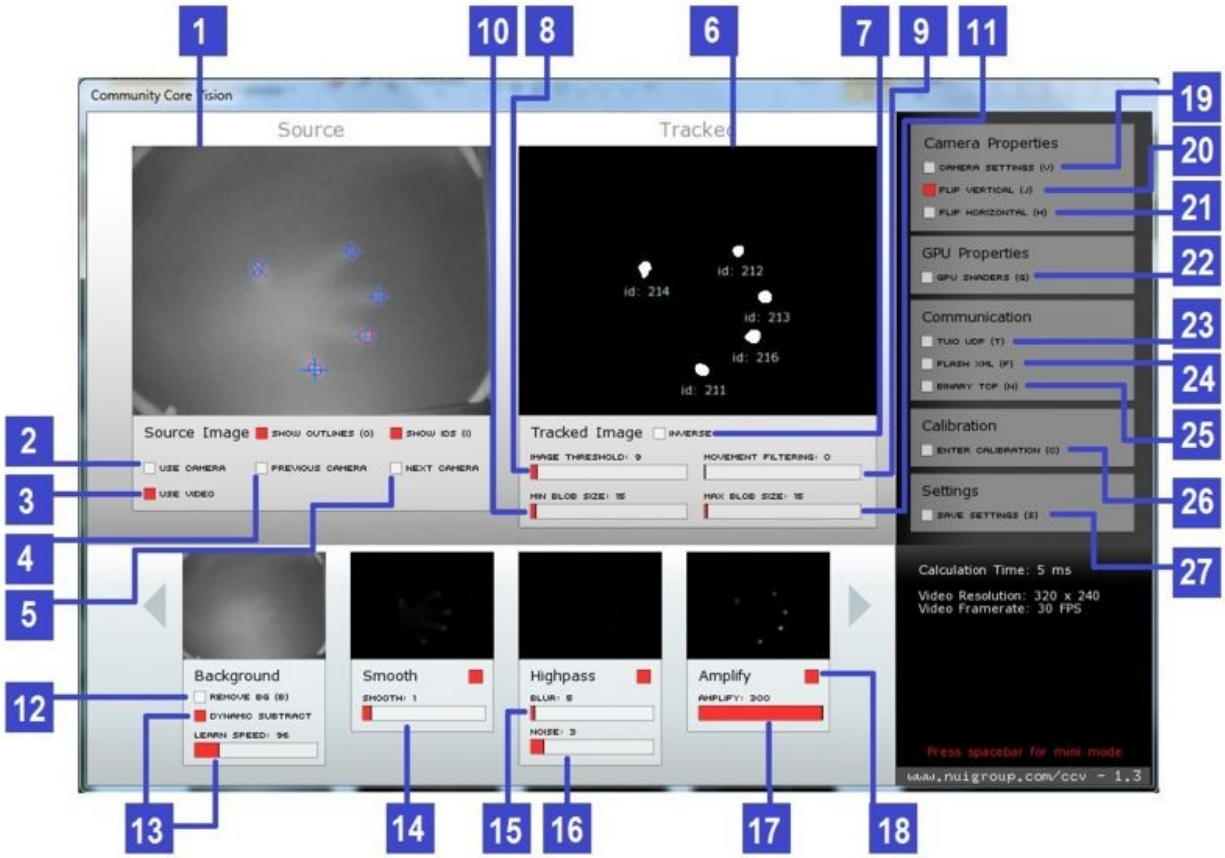

Figura 4: Pantalla princial del CVV (tBeta)

La pantalla principal d'aquesta aplicació és la que mostra la imatge que es captura amb la càmera i permet modificar paràmetres que facilitin la detecció dels blobs en el dispositiu. Els components d'aquesta pantalla, mostrats a la *Figura 4*, són els següents:

1. Font de la imatge - Mostra la imatge de vídeo en brut de qualsevol càmera o fitxer de vídeo.
2. Activar utilització de càmera - Estableix com la font d'entrada, la càmera.
3. Activar l'ús de vídeo - Estableix la font d'entrada de fotogrames d'un vídeo preexistent.
4. Càmera prèvia - Si tenim més d'una càmera, canvia a la càmera que es va se servir abans.
5. Càmera següent - Si tenim més d'una càmera, canvia a la càmera següent.

6. Seguiment de la imatge - Mostra la imatge final d'acord a la configuració establerta per a la detecció i el seguiment de *blobs*.
7. Invers - Mostra *blobs* negres en comptes de blanc.
8. Threshold Slider - Ajusta el nivell acceptable de píxels de seguiment. Com més gran sigui, més gran els *blobs*.
9. Moviment de filtrat - Ajusta el nivell de la distància acceptable (en píxels) abans que es detecti el

moviment d'un *blob*. Com més gran sigui, més s'ha de moure el dit per a registrar un moviment de *blob.*

10. Mida mínima d'un *Blob* - Ajusta la mida mínima acceptable d'un *blob*. Com més gran sigui, més gran serà el *blob* a qui s'assignarà un identificador.
11. Mida màxima de Blob - Ajusta la mida màxima acceptable d'un *blob*. Com més gran sigui, més gran serà el *blob* a qui s'assignarà un identificador.

12. Eliminar el fons - Captura la imatge d'origen i la utilitza com la imatge de fons estàtica. Escollir aquesta opció per recuperar una imatge de fons estàtica
13. Activar la substracció - Ajusta dinàmicament la imatge de fons. Activar aquesta opció quan tenim molts canvis d'il·luminació ambiental o falsos *blobs* a causa dels canvis ambientals.

14. "Suavitzar la imatge" - Suavitza la imatge i filtra el soroll (distorsió) de la imatge.

15. Highpass Blur Slider - Elimina les parts borroses de la imatge i deixa les parts més brillants.
16. Highpass Noise - Filtra el soroll (distorsió) de la imatge després d'aplicar Highpass Blur Slider.

17. Amplify Slider - Si els *blobs* són febles, això pot ser utilitzat per fer-los més forts.
18. Opció On/Off - S'utilitza en cada un dels filtres, serveix per activar o desactivar els filtres.

19. Configuració de la càmera - Obre una pantalla de configuració de la càmera.
20. Flip Vertical - Inverteix la imatge d'origen en sentit vertical.
21. Flip Horitzontal - Inverteix la imatge d'origen en sentit horitzontal.

22. Activar el mode de GPU - Activa l'acceleració de Hardware i utilitza la GPU. Aquesta és la millor opció en les noves targetes gràfiques només. Nota: El mode GPU es troba encara en desenvolupament i pot no funcionar en totes les màquines.

23. Activar envio UDP - Activa l'enviament de missatges TUIO.
24. Flash XML - Activa l'enviament de missatges XML Flash.
25. Binari TCP - Activa l'enviament de missatges de RAW (coordenades x, y).

26. Introduir Calibratge - Carrega la pantalla de calibratge

27. Desa configuració - Guarda tots els ajustes actuals en el fitxer de configuració XML.

4.1.6.1.1 Configuració

Per tal de rastrejar els dits, la pantalla principal ens permet configurar el CCV. L'objectiu final és tenir una imatge de seguiment de *blobs* clars procedents dels dits col·locats sobre la superfície tàctil (sense soroll de fons o falsos *blobs*). Per assolir aquest objectiu, s'han de ajustar diversos filtres d'imatge.

Consideracions FTIR:

- Desactivar els filtres *highpass* i *amplify*.
- Activar el filtre de "Suavitzar la imatge" i ajustar-ho per eliminar soroll.
- Només en el cas de tenir *blobs* febles, activar el filtre *amplify.*

Per a acabar, ajustar el Threshold Slider fins que només es faci el seguiment dels dits i no quedin falsos *blobs*. I si es vol, desar la configuració actual. (La configuració es guarda automàticament al sortir de l'aplicació).

4.1.6.2 Pantalla de Calibratge

Per tal de calibrar CCV per la càmera i el projector, es necessita executar el Procés de Calibratge. Aquest procés, permet calibrar els punts de contacte per alinear-se amb els elements a la pantalla. D'aquesta manera, quan es toca la pantalla, el toc queda registrat en el lloc correcte. Per fer això, CCV ha de traduir l'espai de la càmera en l'espai de pantalla, això es fa tocant punts de calibratge individuals. Per accedir al Procés de Calibratge s'ha de escollir l'opció Introduir *Calibratge* o pressionar la tecla 'c' del teclat.

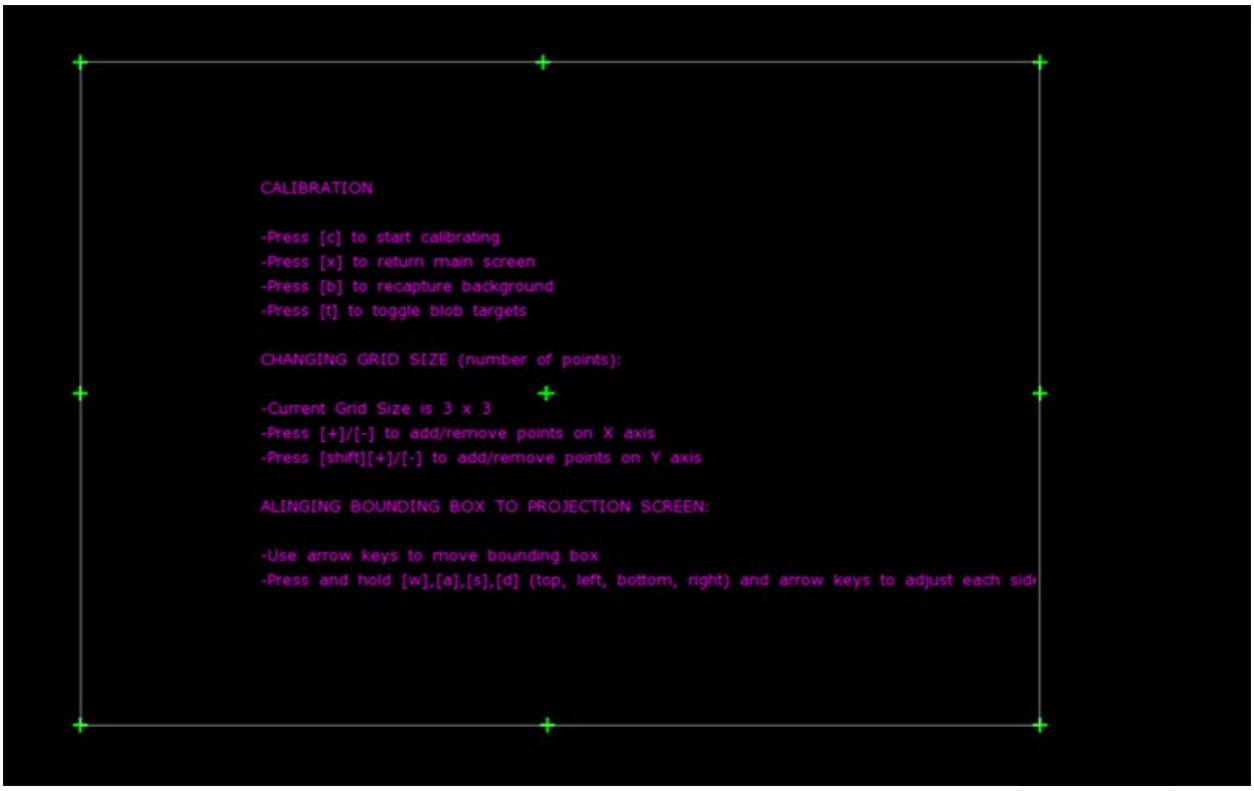

Figura 5: Configurar Calibratge

4.1.6.2.1 Configuració

La pantalla de Calibratge ens mostra una xarxa de punts verds per tota la pantalla. Aquests són punts de calibratge que s'utilitzaran una vegada comenci el calibratge. Es poden afegir/treure punts de calibratge segons la mida de la pantalla, la fiabilitat del filtre infrarojos i la il·luminació de l'ambient.

- Premeu "+" per afegir punts o "-" per treure'ls en l'eix X.
- Mantenir premut “shift”, Premeu "+" per afegir punts o "-" per treure'ls en l'eix Y.

Hi ha una caixa blanca (delimitador) que envolta els punts de calibratge. Si no coincideix amb l'àrea que volem fer servir de la pantalla, s'ha de modificar per arribar a tenir la posició que es desitja. Arribat a aquest punt, comencem el calibratge pressionant la tecla “c”.

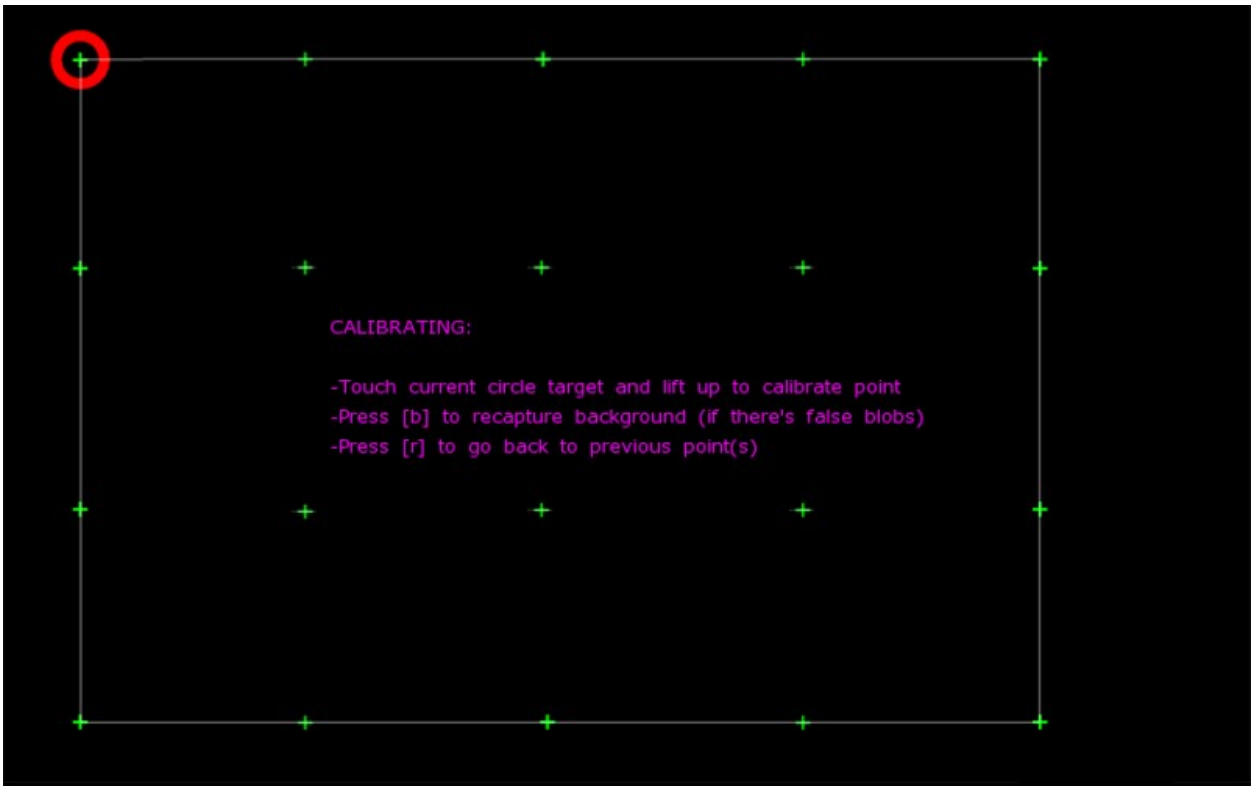

Figura 6: Calibratge

Un cercle vermell es mostrarà per sobre dels punts de calibratges (un punt a la vegada). Pressionem cada punt “seguint” al punt vermell fins que tots els punts de calibratge es pressionen. Si es comet un error, premeu "r" per tornar al últim punt de calibratge. Si hi ha *blobs* fals, premi "b" per recuperar el fons de pantalla. Un cop finalitzada el calibratge, la pantalla tornarà a la pantalla inicial (veure *Figura 4*) i es pot provar seva la precisió pressionant a l'àrea de toc que s'ha escollit en el Procés de Calibratge.

4.2 Protocol TUIO

La definició d'aquest protocol intenta donar una forma de comunicació versàtil entre un dispositiu tàctil i les capes de controlador i domini de les aplicacions que vulguem desenvolupar. Va ser dissenyat per treballar sobre superfícies multitàctils, on l'usuari es capaç de manipular un conjunt d'objectes i/o fer gestos utilitzant els dits [1]. En aquest projecte, TUIO és el pont de comunicació entre el CCV (tBeta) i l'aplicació desenvolupada. En resum, aquest protocol s'utilitza per comunicar la posició, la mida i la velocitat relativa dels *blobs*.

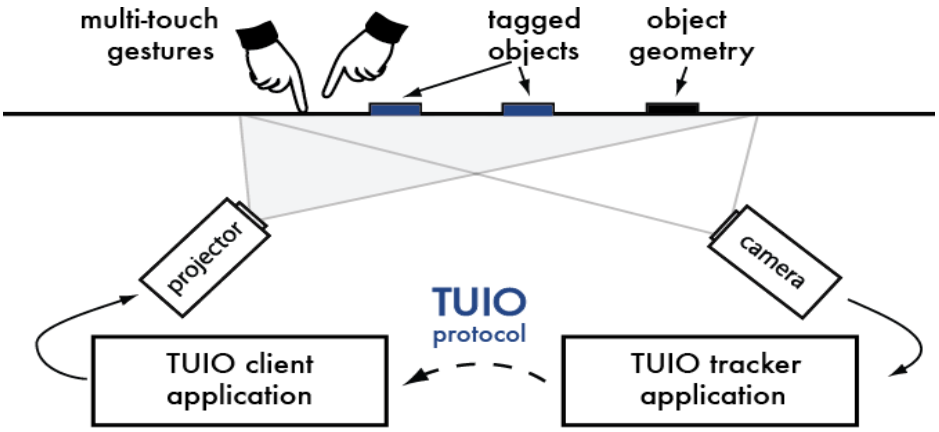

Figura 7: Diagrama sobre el procés realitzat per veure com s'utilitza el protocol TUIO

4.2.1 Conceptes bàsics

Per entendre el funcionament d'aquest protocol cal veure com es guarda la informació i quin serà el mecanisme de comunicació entre el dispositiu i l'aplicació. El diagrama de classes que utilitzarà el protocol per representar la informació és el següent:

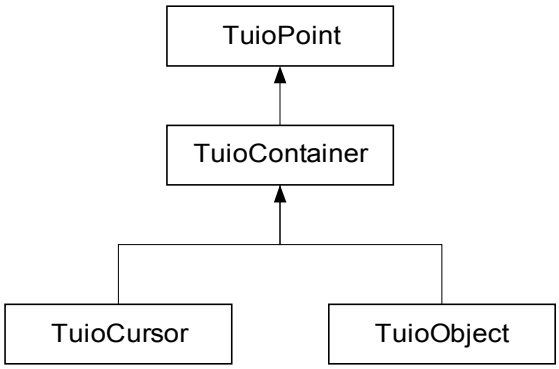

Figura 8: Diagrama de classes del protocol TUIO, extret de la seva API[2]

Aquestes classes seran les que continguin la informació que utilitzarem a l'aplicació:

- *TuioPoint:* És una classe abstracta que conté les funcions comuns de les classes *TuioCursor* i *TuioObject* que són les que contenen la informació més concreta dels objectes disposats sobre la superfície tàctil.
- *TuioContainer:* Conté els atributs comuns de les classes *TuioObject* i *TuioCursor.*
- *TuioObject:* És la classe que conté la informació sobre objectes amb fiducials (o marques amb patrons definits) col·locats sobre la superfície tàctil, per tant tindrem atributs referits a l'angle, velocitat de rotació, etc.

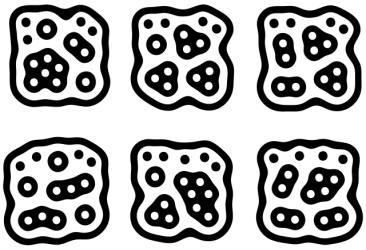

Figura 9: Patrons utilitzats per a fer el seguiment dels objectes amb fiducials.

- *TuioCursor:* És la classe que conté la informació de punts col·locats sobre la superfície tàctil, la utilitzarem per obtenir la informació de les coordenades dels nostres dits quan pitgem la pantalla. S'assignarà un únic ID a cada punt i no es proveirà informació sobre la rotació com en el cas del *TuioObject*.

Les classes principals que permeten la comunicació entre el dispositiu i l'aplicació són:

- *TuioServer:* Aquesta classe és l'encarregada de codificar i enviar els missatges TUIO, aquests, seran enviats via OSC (Open Source Control, veure *Apartat 4.2.2*) sobre UDP cap a l'adreça IP i ports especificats. En el nostre cas, aquest estarà definit al CCV.
- *TuioClient:* Serà l'encarregat de descodificar el missatge enviat pel *TuioServer*. La instància d'aquesta classe generarà esdeveniments TUIO que seran transmesos a totes les classes que implementaran la interfície *TuioListener*. Aquest *TuioListener* s'haurà d'afegir de la següent forma:

```
TuioClient *client = new TuioClient();
client->addTuioListener(myTuioListener);
client->connect();
```

- *TuioListener:* Consisteix en una interfície de C++, pels menys experimentats en aquest llenguatge podríem dir que seria una espècie de classe abstracta que ens dóna les capçaleres de les funcions que utilitzarà el TuioClient per enviar els esdeveniments rebuts del TuioServer.
  Per tant, caldrà definir un arxiu *.h* i un *.cpp* que implementi la classe *TuioListener* com es mostra en el següent exemple:

```
public class MyTuioListener implements TuioListener
...
```

Si es vol tenir més detalls de la implementació del "Listener" en la nostra aplicació és recomanable consultar les següents classes al codi font del projecte.

- `PMTuioListener.h`
- `PMTuioListener.cpp`
- `main.cpp`

4.2.2 OSC

Sempre que parlem del protocol TUIO escoltarem a parlar del OSC (Open Source Control) [3]. Podríem definir-lo com un format de codificació binària per a poder a comunicar els nostres ordinadors amb sintetitzadores de so, i/o altres dispositius multimèdia (en el nostre cas el dispositiu tàctil). Els missatges codificats en aquest format poden ser transportats per Internet utilitzant paquets UDP.

TUIO és un protocol de comunicació implementat amb aquest format de codificació, i el major avantatge que ens dóna OSC en aquest context és la seva simplicitat i velocitat de transmissió necessàries per a implementar dispositius com el que volem utilitzar.

4.2.3 TuioSimulator

Per a poder implementar la part Software del projecte ha estat imprescindible la utilització d'aquest simulador, ja que ens permetia comprovar el correcte funcionament de l'aplicació sense la necessitat de tenir llest el prototipus. En la *Figura 10* podem veure un exemple de com s'utilitza, en concret s'està comprovant el bon funcionament del moviment d'esborrar (en color vermell per poder veure els canvis per pantalla, en la versió final seria del mateix color que el fons).
La finestra petita és el nostre simulador, allà simularem amb el ratolí els moviments a comprovar, evidentment no podrem moure dos punts a la vegada però això no limita que es puguin fer proves totalment vàlides de les funcionalitats que s'estan fent. En aquest cas el moviment d'esborrar es realitza amb dos dits, aquest moviment calcula un punt mitjà entre els dos punts, així que el que farem serà generar el moviment d'esborrar amb dos punts (un després de l'altre) i després només mourem un per veure que es pinta a la pantalla amb la posició mitjana de tots dos. El que fa el simulador és enviar els mateixos missatges TUIO que enviaria el nostre dispositiu.

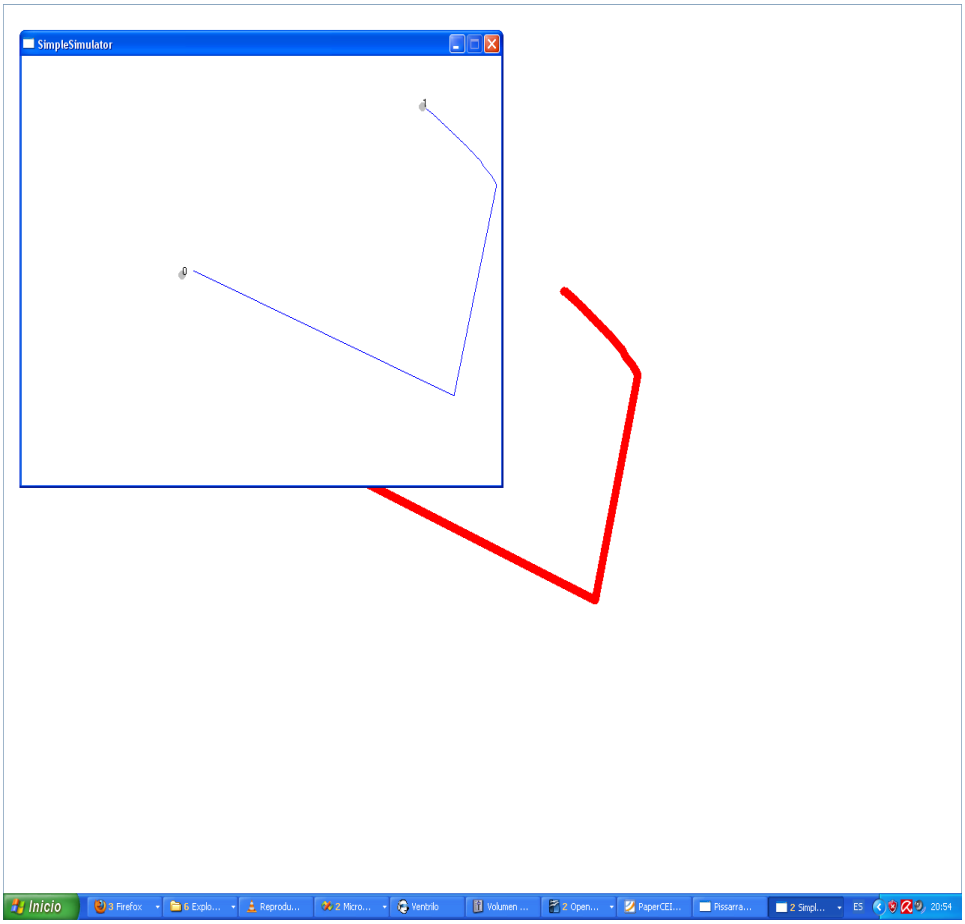

Figura 10: Exemple d'ús del simulador.
En aquest cas s'està provant l'acció "esborrar".

4.3 Llenguatge de programació

Tots els llenguatges de programació tenen els seus punts forts i els seus punts febles, així que realment no hi ha cap millor que l'altre, malgrat tot això, seguint els objectius marcats a l'inici del nostre projecte la nostra elecció ha estat el C++ [4]:

- És un llenguatge orientat a objectes.
  Característica imprescindible del qual només queda descartat el C (ja que gairebé tots els llenguatges de programació existents actualment suporten aquest paradigma). Necessitem que el llenguatge sigui OO per la facilitat que dóna per a la implementació al separar les funcionalitats a cada classe per separat.

- Les aplicacions implementades en aquest llenguatge poden anar més ràpid.
  En una aplicació Software MT on tenim varis punts simultanis dels que s'ha de fer seguiment e interpretació en temps real.

- És un llenguatge que treballa a més baix nivell que la majoria dels existents:
  - Ens permet fer una programació més sofisticada i flexible.
  - En teoria això fa que sigui més difícil d'utilitzar per la gent inexperta, però no suposa cap problema en el nostre cas.

- Pot funcionar sobre qualsevol tipus de plataforma (Windows, Linux, etc).
  No volem ser depenent de cap tipus de sistema operatiu (perquè l'aplicació tingui la màxima flexibilitat en quan a la seva utilització). En aquest punt va quedar descartat el C#, ja que és un llenguatge de programació estandarditzat per Microsoft com a part de la seva plataforma .NET pensada per a operar en el sistema operatiu de Windows.

- El servidor de reconeixement gestual que anàvem a utilitzar (Sparsh_UI) estava en aquest llenguatge.
  D'aquesta forma podríem estalviar molt de temps en entendre el contingut i funcionament d'aquest complement existent.

- És un llenguatge conegut per nosaltres.
  L'havíem fet servir anteriorment junt amb altres llenguatges molt semblants com són el Java o C#. En aquest punt vam descartar possibilitats com per exemple el Python o el ActionScript3.

4.4 Entorn de desenvolupament

Una vegada ja tenim decidit el llenguatge de programació que utilitzarem s'ha de decidir quin entorn de desenvolupament utilitzar per a programar la nostra aplicació.

Eclipse [5] va ser la primera opció inicialment, ja que l'havíem utilitzat en anteriors ocasions amb experiències molt positives (malgrat aquestes foren utilitzant Java i no C++). De fet, *Eclipse* és un dels entorns de desenvolupament més populars i utilitzats en el món de Java, per poder-lo utilitzar treballant en altres llenguatges (com en el nostre cas) cal estendre la seva funcionalitat mitjançant plugins.

Fins aquí cap problema, la veritable dificultat amb la que es va trobar va ser cercar i executar un compilador que generés els nostres fitxers binaris, en Linux, normalment solen estar instal·lades les comandes gcc i g++, en Windows la cosa és complica, ja que ens caldrà instal·lar aquestes comandes i per executar-les haurem d'utilitzar una espècie d'emuladors de Linux com són MinGW o Cygwin.

Després de dedicar moltes hores, es va presentar l'opció d'utilitzar el *Visual Studio*, amb la qual cosa es va acabar descartant l'*Eclipse* per la dificultat que suposava fer servir aquest entorn amb el C++ juntament amb Windows. D'aquest entorn hi destaquem la facilitat d'ús en Windows per a la programació en C++ (ja que en aquest cas no cal instal·lar cap compilador, sinó que ja ve integrat), i a més, els companys del Centre de Realitat Virtual de la UPC, que són els que ens van aconsellar fer servir aquest entorn i podien ajudar en cas de tenir cap dubte. Com a estudiants de la FIB tenim llicència per a poder-nos instal·lar i utilitzar aquest programa en els nostres terminals de treball, d'aquesta forma va quedar descartada una de les principals raons per no fer servir Software de Microsoft que és l'alt cost econòmic per a utilitzar les seves aplicacions.

4.5 Patrons de Disseny de Software

Els patrons de disseny són la base per a la cerca de solucions comunes a problemes en el desenvolupament de Software. En aquest projecte n'hem utilitzat dos tipus [6]: El Singleton , i l'Observador. En aquest apartat aprofundirem en cadascun d'ells.

4.5.1 Patró Singleton

És un patró dissenyat per a restringir la creació d'objectes que pertanyin a una classe. Es proporciona una única instància gràcies a:

- La pròpia classe és l'única responsable de crear una única instància.
- Permet l'accés global a aquesta instància mitjançant un mètode d'aquesta classe.
- Declara el constructor de classe com a privat per a que no sigui instanciable directament.

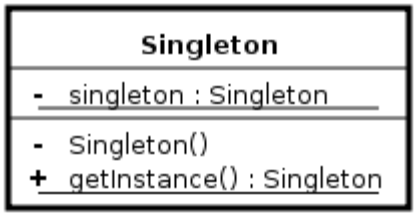

Figura 11: Patró Singleton [7]

4.5.2 Patró Observador

És un patró on un objecte, anomenat subjecte, manté una llista d'objectes dependents d'ell, anomenats observadors, i notifica automàticament qualsevol canvi en el seu estat a tots aquests. El que es busca és desacoblar classes que simplement estiguin a l'espera d'un canvi d'estat d'una altra, augmentat així la modularitat del llenguatge i al mateix temps evitant bucles d'actualització.

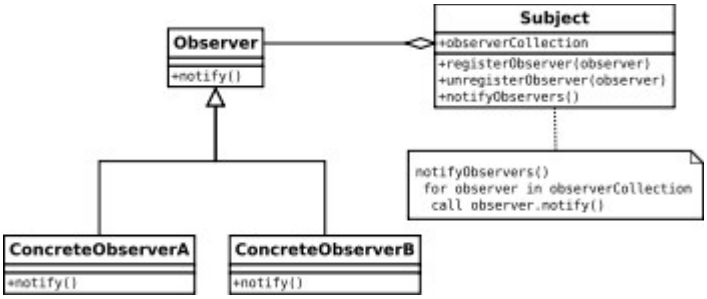

Figura 12: Patró Observador [8]

El patró observador ja està implementat en la majoria de llibreries de programació, aquest projecte, que fa servir Qt, utilitza el seu model de *Signal i Slots*.

4.6 Software per la Interfície Gràfica: Qt

En aquest punt no ha calgut fer cap estudi previ per fer una elecció entre varies candidates, la nostra elecció ha estat clara: QT [9]. És una biblioteca multiplataforma per a desenvolupar interfícies gràfiques d'usuari i programes sense interfície gràfica com eines de consola i servidors.

Funciona en totes les principals plataformes i té un ampli recolzament. L'API de la llibreria compta amb mètodes per accedir a les bases de dades mitjançant SQL, per utilitzar XML, gestió de "threads", suport en xarxa i una API multiplataforma unificada per a la manipulació d'arxius i una multitud d'altres mètodes per al manegament de fitxers a més d'estructures tradicionals de dades.

Es distribuïda sota termes de GNU Lesser General Public License i és un Software lliure i de codi obert. A nivell personal, ja havíem treballat amb aquesta llibreria cursant l'assignatura de VIG, sumat al consell de la nostra tutora, no va donar cap dubte sobre quina llibreria gràfica calia utilitzar.

És important mencionar i explicar dos eines importants de Qt en aquest apartat, un és el sistema de *Signals i Slots*, que és la implementació en aquesta llibreria del patró Observador (veure *Apartat 4.7, Patrons de Software*), que ens servirà per a fer les notificacions sobre canvis d'estat en les dades. L'altre és el sistema d'esdeveniments de Qt. En els apartats següents, ambdós eines estaran explicades en profunditat.

4.6.1 Signals i Slots

Aquest sistema és utilitzat per a la comunicació entre objectes [10]. És una de les parts més importants de les que forma Qt i probablement la més diferent respecte altres implementacions d'altres llibreries gràfiques o eines destinades al mateix fi.

Un Senyal *(Signal)* és emès quan es produeix un cert esdeveniment. Els widgets de Qt tenen molts senyals predefinits però sempre podem fer subclasses de widgets per definir els nostres propis senyals. Un Slot és una funció que es crida en resposta a un particular senyal, i de forma anàloga al que hem explicat amb els senyals, podem fer els nostres propis slots si no volem utilitzar els que ja estan redefinits (que serà el més comú ja que la nostra aplicació generarà esdeveniments de Qt creats per nosaltres).

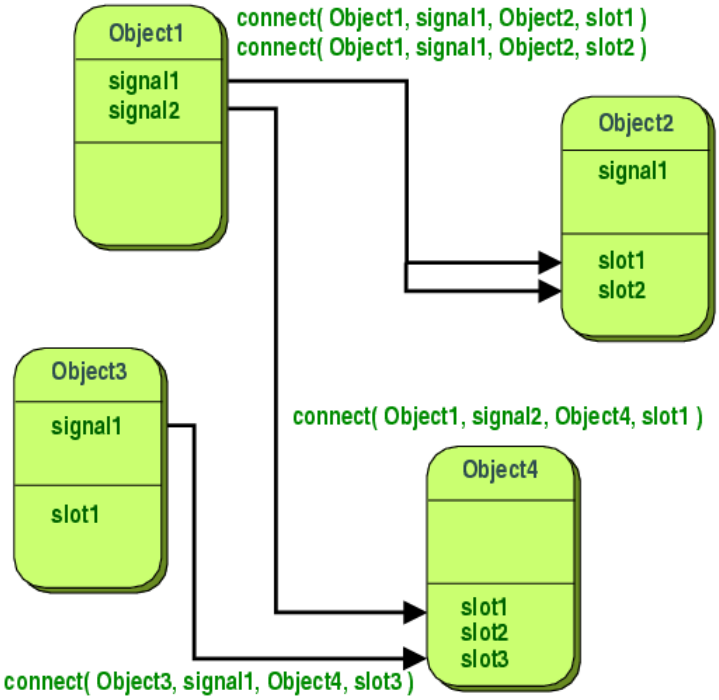

Figura 13: Exemple de funcionament del sistema de Signals i Slots

Totes les classes que heretin de QObject o alguna de les seves subclasses (per exemple QWidget) poden contenir Signals i Slots. Els senyals seran emesos per objectes quan canviï el seu estat i aquest canvi sigui interessant per a altres objectes.

En la *Figura 13*, podem observar el funcionament d'aquest sistema de comunicació entre objectes, mitjançant la funció "connect" podem connectar senyals a tants slots com desitgem o necessitem. Inclús seria possible connectar signals a altres signals (que emetria una segona senyal immediatament després de l'emissió de la primera).

4.6.2 Sistema d'Esdeveniments de Qt

En Qt, els esdeveniments són Objectes derivats de la classe abstracta Qevent, que representa el resultat d'una acció externa a l'aplicació que s'ha de notificar (en el cas que ens ocupa, els esdeveniments generats a partir de les accions dels usuaris). Aquests esdeveniments seran rebuts i tractats per qualsevol instància de QObject, normalment Widgets.

Com es reben

Una vegada creada la instància corresponent de la classe QEvent, aquesta és enviada a una altra instància de QObject (o alguna de les seves subclasses), que serà el destinatari, que rebrà l'esdeveniment mitjançant la crida de la funció "event()". Aquesta funció simplement mirarà el tipus de l'esdeveniment rebut i cridarà a la funció específica que s'encarregarà del seu tractament.

Com s'envien

Hi ha vàries opcions, de forma provisional hem escollit utilitzar "sendEvent()" que tracta l'esdeveniment immediatament. L'altra és "postEvent()" que encua els esdeveniments per després fer el tractament que toqui, l'avantatge d'aquest sistema és que si per exemple, tenim varis esdeveniments encuats per redimensionar un objecte aquests són comprimits en un. D'aquesta forma es podrien eliminar molts esdeveniments "inútils" augmentant l'eficiència de l'aplicació.

Donat que les possibilitats del sistema d'esdeveniments és molt potent en quan al manegament de widgets, pensant en una més que possible extensibilitat de les funcionalitats de l'aplicació és important utilitzar-lo en aquest projecte. Cal dir que l'explicació que s'ha donat aquí ha estat molt resumida, i que dins d'aquest sistema hi ha molts tipus de tractament d'esdeveniments que ja han estat implementats i que es poden utilitzar per afegir la funcionalitat que es desitgi, per saber-ne més detalls recomanem consultar la referència [11].

4.7 Software de Reconeixement Gestual

El Software de Reconeixement Gestual ens serveix per a poder interpretar els punts amb el seu estat (BIRTH, MOVE i DEATH) i obtenir quins gestos s'estan realitzant en la pantalla tàctil, com serien en el cas que ens pertoca: escriure, esborrar, click, moure imatge, etc.

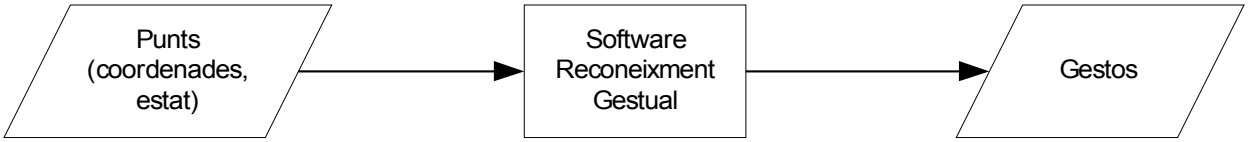

Figura 14: Dades amb les que treballarà el sistema de Reconeixement Gestual.

En aquest apartat explicarem l'estudi que vam fer per a intentar aprofitar la feina d'altra gent a l'hora d'afrontar aquest problema.

- Sparsh_UI
- Grafiti
- AME Patterns Library

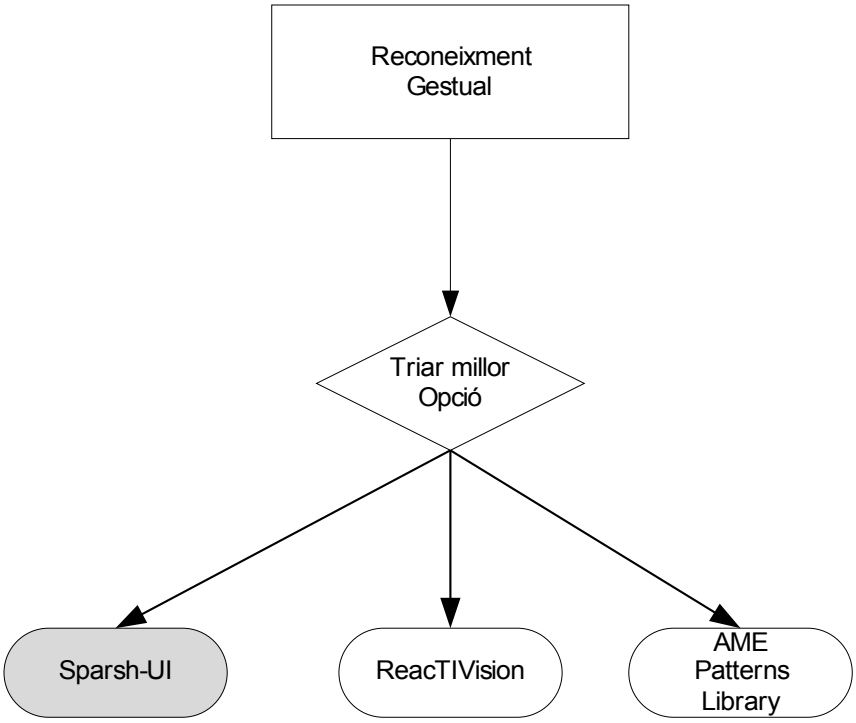

Figura 15: Diagrama sobre la decisió a triar en quan a Software de Reconeixement Gestual existent.

4.7.1 Grafiti

Aquest Software de reconeixement gestual [12] forma part del treball de final de carrera d'un estudiant italià, i és força interessant, a continuació comentem les seves característiques.

Avantatges:

- Està construït per treballar amb el protocol TUIO.
- Està contemplada la possibilitat d'utilització d'objectes.
- L'aplicació està preparada per a afegir gestos nous a detectar, juntament amb la documentació de com fer-ho.
- Té implementat el reconeixement d'uns gestos bàsics.

Inconvenients:

- Està implementat en C#, això fa que ens limiti a un sistema operatiu, cosa que no volem pel nostre projecte.
- Els gestos ja implementats no contemplaven tots els que necessitàvem, malgrat tot els podíem afegir, com hem explicat a dalt.
- És un Software pensat per treballar en mode taula, amb detecció d'objectes amb fiducials, cosa que nosaltres no anàvem a utilitzar.

DESCARTAT

4.7.2 AME Patterns Library

És un Software de reconeixement gestual basat en la detecció de patrons en temps real [13]. Utilitza el Model Ocult de Markov [14] per a distingir el gestos realitzats a la pantalla tàctil (també utilitza altres models més senzills de detecció de patrons). Explicant-ho de forma senzilla, el sistema "aprèn" a distingir els gestos que fem i d'aquesta forma és capaç d'interpretar-ho.

L'inconvenient principal és la quantitat d'incògnites que sorgeixen sobre aquesta idea i la poca informació disponible, ja que la documentació sobre el tema és una mica limitada o difícil d'entendre. També creiem que aquest tipus de sistema no és l'adequat pel nostre cas.

DESCARTAT

4.7.3 Sparsh_UI

L'Sparsh-UI és un Software de Reconeixement Gestual [15] que fa d'intermediari entre el dispositiu i la nostra aplicació tal com és mostra a la següent figura:

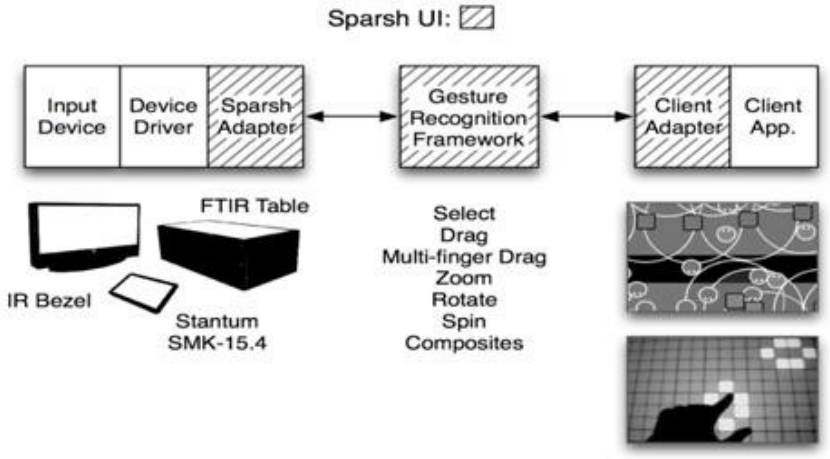

Figura 16: Funcionament de l'Sparsh_UI.

Avantatges:

- Es pot connectar a una àmplia varietat de dispositius MT.
- És multiplataforma.
- Nous gestos i algorismes de reconeixement poden ser afegits al paquet que s'ofereix inicialment.
- En la web els continguts estan molt ben explicats i organitzats.
- El codi és obert i està implementat en Java i C++.

Inconvenients:

- Inicialment no vam trobar cap, per això va ser l'opció escollida, més endavant en el desenvolupament del projecte ens vam adonar que no era del tot així (ho explicarem en al capítol *Arquitectura del Software i Extensibilitat*).

OPCIÓ ESCOLLIDA

4.8 Altres

4.8.1 Microsoft Visio 7.0

Programa utilitzat per fer els “dibuixos” del projecte, entrarien els diagrames de fluxe, esquemes, diagrama de classes en UML, etc.

Tenim dret com a estudiants de la FIB a descarregar-nos una versió d'aquesta aplicació, així que no hi ha problemes amb la llicència.

4.8.2 Open Office

Serà l'editor de text escollit.

5. Metodologia de Treball

En aquest apartat ens centrem en els passos que s'han seguit per a la implementació d'un projecte Software. Això vol dir com s'ha programat, de quina forma, amb quines pautes, etc. De tots és sabut que no es tracta d'agafar un ordinador i començar a picar codi, hi ha tot un procés de pensament darrera que ens pot estalviar molt temps, errors i sobretot molts maldecaps.

La nostra metodologia, és totalment personal i l'hem fet en base a les nostres experiències personals amb el RUP (Rational Unified Process) i l'SCRUM (és una abreviació del terme del Rugby "scrummage" que vol dir melé, metafòricament vol dir que tot l'equip intenta aconseguir distància treballant en equip), explicarem el que hem aplicat en el nostre cas concret.

En quan al codi de la programació explicarem breument en que consisteixen els Estàndards de C++ aplicats.

Creiem fermament en que l'aplicació d'aquest tipus de metodologies és el camí correcte pel desenvolupament de Software de qualitat, i aquesta és una de les raons que ens van impulsar a la seva utilització tenint en compte el "plus" de feina que comportaria (ja que la nostra experiència en la seva aplicació era molt reduïda).

5.1 Planificació

Durant el transcurs de la carrera hem cursat assignatures que tractaven el problema relacionat amb la forma més eficient de començar a programar sense acabar en un fracàs total, la més destacable podria ser PESBD. D'aquí ens va venir la idea d'utilitzar algunes de les pautes del RUP (Rational Unified Process) , que és una espècie de metodologia aconsellada per a la implementació d'una aplicació de Software i ho vam perfeccionar finalment amb algunes de les nocions que teníem del SCRUM.

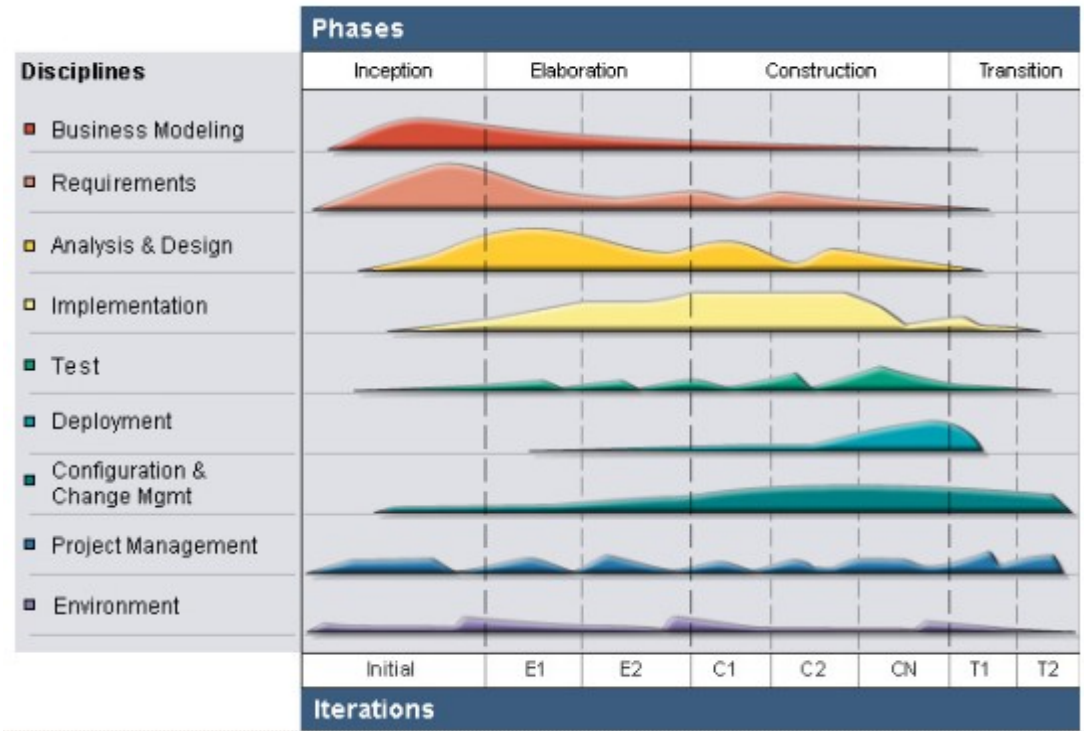

Figura 1. Fases del RUP

A la *Figura 1* es mostra un quadre que representa la metodologia de desenvolupament de Software RUP. Durant el transcurs d'aquest capítol s'explicaran els conceptes necessaris per entendre el contingut d'aquest quadre i entendre d'aquesta forma com funciona aquest sistema.

5.1.1 Qué és una iteració?

Iteració: Totes aquelles activitats destinades a desenvolupar alguna part tangible i/o executable d'un projecte Software.

En totes les definicions que hem trobat sobre que vol dir iterar, es referien a l'entrega (al final de cada iteració) d'una part "utilitzable" del que seria el resultat final. Això té cert sentit tenint en compte que el fet de passar per totes les fases de disseny del projecte farà sortir tots aquells problemes que poden romandre ocults durant les primeres fases de disseny, anàlisi i requeriments del projecte. Aquesta definició funciona a la perfecció excepte en les primeres fases del projecte, on és molt difícil obtenir quelcom que es pugui executar i que funcioni. En aquest cas vàrem intentar posar-nos objectius que, encara que no fossin executables i funcionals, ens servissin igualment per a avançar cap a l'objectiu principal.

Un exemple és millor que mil paraules, aquests són els objectius que ens vam posar a la nostra 11a iteració:

- Redactar Informe del nostre PFC.
- Compilar i executar "Sparsh-UI" (una part feta que volíem integrar dins el nostre projecte).
- Arreglar Software Arquitecture Document (SAD).
- Fer una primera implementació del domini dissenyat sobre el paper (sense executar).
- Fer revisió del document sobre el prototipus de la pantalla tàctil.

Podem observar que no hi ha cap resultat tangible en aquests objectius, però si que eren uns resultats assolibles que podíem aconseguir en aquest moment del projecte. S'ha de dir que dins aquest exemple hem omès el que són les "fases" de les que consta una iteració, que és un concepte que explicarem una mica més endavant en en l'apartat 2.2 d'aquest capítol.

Una pregunta important que un es pot fer després de llegir el que s'ha explicat fins ara és la següent: *Què es guanya fent iteracions?*

- Clarament ajuda molt a veure els progressos que es van fent cada cert període de temps, ja que sobretot al principi del projecte un té la sensació que es passa moltes hores sense aconseguir cap i molt pocs progressos a canvi. Quan un tanca cada iteració s'adona realment del que s'ha aconseguit.
- Tenir una millor noció de la capacitat de treball del grup. S'aconsegueix cada vegada ajustar millor els objectius de forma assolible i en el nostre cas veure com va el treball respecte el dia de la lectura del projecte.
- Es pot detectar fàcilment els casos de poc progrés en el treball, i anima a la gent a treballar dur en la seva tasca, ja que en tot moment és conscient de la feina que fan els demés i a l'inrevés.
- En el nostre cas, ens ha ajudat a tenir el tutor al dia de la feina que anàvem fent.

Tots aquests guanys han estat segons el nostre punt de vista, i creiem que malgrat vam perdre un temps a entendre el concepte i vam anar aplicant-ho mitjançant assaig-error, els resultats finals han estat molt positius.

En quan a la *duració* de cada iteració, el que diu el RUP és que cada iteració ha de durar fins que s'aconsegueixin objectius marcats, nosaltres en quest punt hem preferit estimar que cada iteració fos de *dos setmanes* tal i com és fa amb l' SCRUM (on a la iteració se l'anomena "sprint" però ve a ser el mateix concepte). De fet, seguint aquesta altra metodologia àgil, el que fèiem era informar-nos constantment de l'estat de treball de l'altre company de projecte per tal d'evitar que s'endarrerissin les tasques previstes. Tot i això a vegades sorgien imprevistos que provocaven que no s'assolissin els objectius, llavors el que vam fer va ser fer-ho constar en el document de tancament de la iteració i replantejàvem tots els objectius previstos per a la següent.

5.1.2 Cicle de vida del RUP

En aquest apartat descriurem el tipus fases de desenvolupament de les que costa un projecte, cada iteració contindrà varis tipus d'aquestes fases segons el moment en que es trobi el projecte.

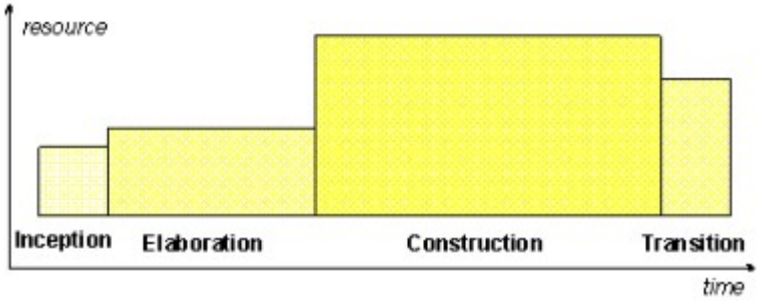

Figura 2. Cicle de vida del RUP

En la *Figura 2* podem veure les quatre fases de les que consta cada iteració, el gràfic simbolitza la dedicació total a cada una de les fases com a mitjana (malgrat pot variar segons el tipus de projecte), si suméssim la dedicació a cada una de totes les iteracions realitzades. Abans de continuar comentant el gràfic passarem a comentar que es fa en cada fase:

- **Inception** (Inici): Serà on es definirà el projecte pròpiament. Dins aquesta fase hem realitzat les activitats que corresponen a la motivació del projecte, sortida econòmica, i la descripció detallada de quin és l'objectiu i el cas d'estudi del projecte, etc. Activitats essencials:
  - Formular l'abast que tindrà el projecte. En aquest moment és on hem estudiat totes les possibilitats del projecte i hem decidit quines funcionalitats s'implementarien i quines no (tenint en compte la seva importància).
  - Sintetitzar un esborrador de l'arquitectura del sistema. Fer una primera aproximació ja que encara no està molt clar l'assumpte en aquest punt.
  - Preparar l'entorn del projecte. En aquest punt hem hagut de decidir les eines que hauríem d'utilitzar, per exemple: sistema operatiu, editor de programació, llenguatge de programació, etc. Tota aquesta feina de decisió es troba dins del capítol *Eines Utilitzades*.
- **Elaboration** (Elaboració): El principal objectiu d'aquesta fase serà refinar el model d'arquitectura del sistema proposada a l'apartat anterior. En el que és el RUP trobarem moltes referències a estudiar l'evolució de l'arquitectura per evitar errors i d'aquesta forma pèrdues econòmiques i si es farà o no en el temps estipulat. En el nostre cas aquestes coses no tenen tanta importància, però hem de ser conscients de la feina que fem quan hem d'afegir una nova funcionalitat al nostre sistema com podria ser "obrir la imatge a la pissarra" correspondrà a aquesta fase. Serà aquí on ens trencarem la closca per fer quadrar la nova "peça" dins el nostre gran "puzzle" que serà la nostra arquitectura.
  També correspon a aquesta fase la feina de confeccionar les plantilles que utilitzarem en la documentació (veure *Annex*).
- **Contruction** (Construcció): És la feina pròpia de desenvolupament del projecte, picar el codi i comprovar que funciona correctament. En el RUP es tenen en compte moltíssimes més coses i totes en referència al control, crític quan el temps i els diners marquen el destí del projecte, en el nostre cas per raons evidents aquests controls no faran falta (a més sent un projecte on només intervenen dues persones, sense grans equips de treball).

- **Transition** (Transició): En el nostre projecte només seran les feines de revisió de la feina feta amb el tutor, fer la última versió de l'aplicació, la confecció d'aquesta memòria i la preparació de l'aplicació. No afegim al projecte l'estudi d'usabilitat, per tant no ens caldrà obtenir les opinions dels usuaris, ho considerem, però és una tasca que estaria bé fer en un futur.

Una vegada sabem que vol dir i que és fa en cada fase podem tornar a mirar el gràfic de la *Figura 2* i entendre perquè normalment els projectes tenen aquesta distribució en hores de feina (on la major part del treball dur és fa a la fase de construcció). Ara bé, s'ha de dir que en el nostre cas ha estat una mica diferent, posant també gran pes d'hores en les fase d'elaboració i inici, en gran part a causa de la nostra inexperiència en la realització d'aquest tipus de projectes.

5.1.3 Reflexió

Després de llegir aquest apartat un es podria fer la següent pregunta, *i tot això de les fases quina utilitat té?* Un tendeix a pensar inicialment que seria més fàcil començar a barallar-se amb el codi directament (bé, és el que quasi tothom fa), nosaltres tampoc teníem gaire idea de la utilitat real d'aquesta metodologia però després d'haver-ho fet servir creiem que l'experiència ha estat molt positiva. Enumerem els avantatges a continuació:

- Ajuda a organitzar-se molt millor cada tipus de feina *sobretot* a l'inici del projecte que és quan un va més perdut.
- Al marcar uns passos en ordre dins de l'operativa evita errors que significarien una gran pèrdua de temps en el futur (o sigui, que en realitat estem estalviant temps en comptes de perdre'l).
- A mida que van passant les iteracions un es va adonant que cada vegada són menys les feines d'inici i elaboració i més les relacionades amb la construcció, d'aquesta forma es va essent conscient de l'estat en que es troba el projecte i s'és més capaç de predir i/o ajustar-se als terminis marcats.
- Ajuda molt a planificar les feines de cada iteració.

5.2 C++ Coding Standards

En un projecte és important marcar una pauta comuna de treballar amb el codi, és una forma de sentir que un està treballant en el mateix "camp de joc" que els seus companys. Els estàndards que hem utilitzat s'han desenvolupat a partir de les experiències en molts tipus de projectes Software, per part de moltes i diferents companyies i després de moltes setmanes de discussions entre els diferents grups de treball que els van confeccionar.

Tot el codi implementat segueix un seguit de normes que ens donen una guia sobre com anomenar a les funcions, als atributs de la classe, plantilles per redactar els fitxers ".h" i ".cpp" (veure *Annex*), etc. Tot està explicat en detall dins de la següent direcció web:
Http://possibility.com/Cpp/CppCodingStandard.hml#descriptive

Intentant respondre a la pregunta sobre què ens pot aportar el seguiment d'aquests estàndards a l'hora d'implementar qualsevol aplicació, podem comentar els seus pros i contres:

Avantages:

- Qualsevol programador pot investigar dins el codi i entendre en poc temps que s'està fent i quina és la situació.
- Es pot incorporar gent nova al projecte i agafar el ritme de treball dels més antics amb facilitat.
- La gent nova en el C++ deseguida pot agafar una pauta des d'on desenvolupar el seu propi estil de programació que podrà defensar fàcilment.
- La gent nova en el C++ s'estalviarà el fet de fer els mateixos errors una vegada rera una altra.
- Tot l'equip de programació té un "enemic comú", en el sentit de que tothom programa de la mateixa forma i el codi sempre estarà escrit en el mateix estil.

Inconvenients:

- Es redueix la creativitat.
- Si la gent sap que està fent són totalment innecessaris.
- La gent habitualment no els utilitza.
- Pot ser un obstacle a l'hora d'utilitzar codi ja implementat que no segueix la normativa que dicta l'estàndard.

Tots aquests avantatges i inconvenients estan un extrets de la pàgina web on estan definits els estàndards que hem utilitzat. Per la part que ens toca, era la primera vegada que ho fèiem servir i creiem que l'experiència ha estat molt positiva, malgrat la seva utilització no és element indispensable per l'èxit d'un projecte ajuda molt per al treball en equip, més si tenim en compte que aquest projecte està pensat com un inici cap a una aplicació molt més complexa que podríem no desenvolupar nosaltres.
Creiem que gràcies a haver utilitzat aquesta metodologia podrem aportar molt més valor al projecte, en el sentit que serà molt més fàcil d'estendre en quan a les funcionalitats que ofereix i animem de forma entusiasta a que qualsevol persona que treballi sobre el nostre codi segueixi els estàndards marcats.

5.3 Nom dels Fitxers:

En un projecte d'aquestes característiques és important mantenir una uniformitat en els noms dels fitxers, sobretot per mantenir un ordre i facilitar el treball de les persones implicades en el projecte (siguin una, dos o més).
Aquest sistema l'hem "importat" del que es fa servir a l'assignatura PESBD.

Els noms han d'indicar les següents característiques :

- F: La fase del cicle de la iteració.
- I: Número d'iteració en que ens trobem.
- V: Versió del fitxer.
- N: Nom del fitxer.
    - Separadors entre paraules: "_"
    - Cada paraula començant en Majúscula.

El format total seria de la següent forma:

- F-I-V-N

Aquesta metodologia és útil perquè a mida que un va treballant i es van acumulant els arxius i és necessari tenir un accés fàcil a la informació sobre cadascun d'ells. Sobretot quan un fa servir el subversion. Al final, un no sap quina és la última versió d'un document, a quina iteració va ser on es va modificar, etc. I no només parlem de la informació que conté el nom del fitxer, sinó també el fet que sense una metodologia clara per anomenar els fitxers cada persona ho faria a la seva manera i finalment es perdria molt de temps per a trobar el que es busca.

5.4 Confecció del Subversion

Dins del nostre repositori podem trobar les següent carpetes:

- **Trunk:** Aquí guardarem la última versió funcional de la nostra aplicació, seria com la versió principal del projecte. S'utilitzarà com a base per treballar en les noves funcionalitats.
- **Tag:** Aquí desarem les versions antigues del projecte per a tenir un "backup" disponible en qualsevol moment. Cada versió està marcada amb la data en que es va inserir.
- **Branch:** Aquí serà on desarem les versions amb les que estarem treballant en cada moment (en aquest cas els dos membres del grup). El motiu de la separació es que en un cert moment cada persona del grup estarà afegint una funcionalitat diferent, si un d'ells fa un commit d'una versió que no funciona afectarà a l'altre membre del grup que no podrà avançar a causa d'un error que no té a veure amb el que està treballant en aquell moment.

Al principi no ho vam fer així, però a mida que anàvem avançant en el projecte ens vam adonar de la seva necessitat, per seguretat i per comoditat, potser al principi és perd una mica de temps en comprendre com funciona aquesta disposició però després una vegada s'agafa la dinàmica no hi ha cap més problema i es fa simple.

6. Tecnologia Utilitzada

Una vegada hem tingut clar què és el que volíem fer o dit d'un altre forma, els nostres objectius, s'ha hagut de fer un estudi exhaustiu o recerca sobre les tecnologies tàctils disponibles actualment ja que a l'inici del projecte el nostre coneixement sobre aquest camp era pràcticament nul. Escollint així després de tot aquest estudi, la tecnologia que considerem, més adequada.

6.1 Tecnologies Tàctils

Tenim quatre grans grups de tecnologies tàctils diferents:

- Les Superfícies Tàctils Basades en Resistència.
- Les Superfícies Tàctils Basades en Capacitància.
- Les Superfícies Tàctils Basades en Superfície d'Ona.
- Les Superfícies Tàctils Òptiques.

Les tres primeres queden totalment descartades des d'un inici ja que necessiten d'instal·lacions que permetin una fabricació de qualitat industrial i per tant no són adequades per a la construcció d'un dispositiu assequible econòmicament i de fabricació pròpia. Malgrat tot s'ha cregut important incloure-les per una millor completesa de l'estudi realitzat i per conèixer i entendre com funcionen [1].

6.1.1 Superfícies Tàctils Basades en Resistència

Aquest tipus de superfícies es basen en dos capes conductores recobertes amb substàncies com ara l'òxid d'estany i indi. Aquestes capes estan separades per una altra capa d'aïllament normalment feta per petits punts de silicona (veure *Figura 1*). La part de davant del panell usualment està recobert d'una membrana flexible i dura mentre que el panell de darrera normalment està fet de qualsevol tipus de vidre. Un controlador s'alternarà entre les dos capes, donant una quantitat específica d'electricitat a una i mesurant la càrrega elèctrica actual de l'altre. Quan un usuari toca el dispositiu, les dos capes conductores es connecten, establint un corrent elèctric que es mesura horitzontal i verticalment pel controlador per determinar la posició exacte del contacte.

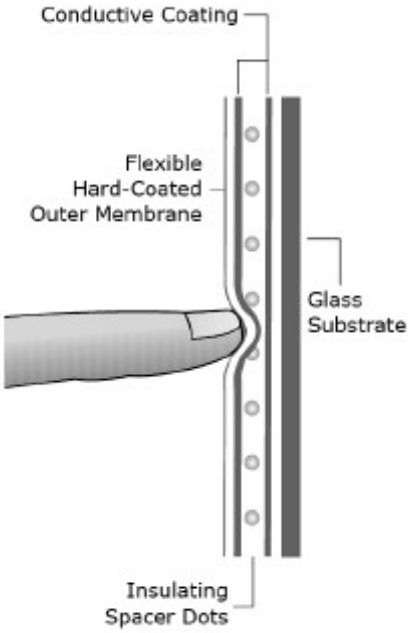

Figura 1. Construcció esquemàtica d'un panell tàctil basat en la tecnologia basada en resisténcia.

Aquestes superfícies tenen l'avantatge de tenir un baix consum d'energia i són utilitzades en dispositius com la Nintendo DS, PDAs i càmeres digitals. Com a contra, té poca precisió i no es pot afegir protecció addicional a la superfície de contacte sense alterar al seu funcionament.

6.1.2 Superfícies Tàctils Basades en Capacitància

En general les superfícies basades en capacitància estan subdividides en dos classes:

- Capacitància de Superfície.
- Capacitància de Projecció.

Les dos tècniques van ser dissenyades per a una interacció d'un sol toc (un sol punt de contacte en tot moment). Un avantatge d'aquest tipus de superfícies és la seva alta precisió, cosa que fa que siguin molt adequades per a una gran quantitat de dispositius tàctils que van més enllà d'un simple *touch pad*. Aquestes pantalles es poden fer servir amb qualsevol tipus de dispositiu conductor i per tant, no està limitat a la interacció basada amb els dits. No obstant, els panells capacitius són relativament cars de produir malgrat tenen una durabilitat i fiabilitat. Com a conseqüència, aquests sistemes són utilitzats en entorns "agressius" tals com llocs públics i en aplicacions industrials.

En teoria és possible utilitzar aquests sistemes per a dissenyar un sistema multitàctil, però normalment les empreses que fabriquen aquests dispositius limiten el nombre de "tocs" simultanis que es poden captar. La precisió disminueix en augmentar el nombre de tocs a detectar en comparació a la detecció d'un de sol. Havent descrit totes aquestes limitacions generals, cal dir que el MERL (Mitsubishi Electric Research Group) va superar moltes d'aquestes restriccions amb l'objectiu de permetre varis tocs simultanis, aquestes investigacions estan descrites en els següents apartats.

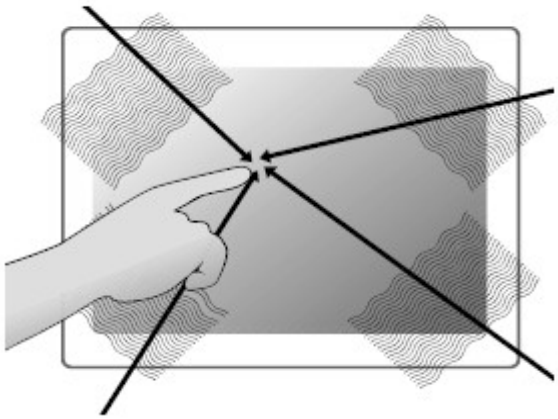

Figura 2. Superfícies Basada en Capacitància: Els elèctrodes al voltant de les vores distribueixen el voltatge a través de la superfície conductora creant un camp elèctric. Tocant el panell resulta en una corrent que va a cada cantonada que és mesurada per definir la posició on hi ha hagut el contacte.

6.1.2.1 Panell Tàctil Basat en Capacitància de Superfície

Aquest tipus de dispositius tàctils es basen en un recobriment conductor uniforme en una superfície de vidre. Comparat amb les superfícies resistives (*Apartat 6.1.1*), tenen molta més precisió i poden ser construïdes mitjançant òxid d'estany i indi com a material conductor (és transparent quan s'utilitza en capes primes). En cada costat del panell tàctil els elèctrodes mantenen un precís control dels electrons horitzontalment i verticalment cosa que crea un camp elèctric en tota la capa conductora. Els dits humans (o altres objectes conductors) també tenen dispositius elèctrics capaços de retenir càrrega elèctrica i d'emetre camps elèctrics, tocar el panell resulta en un petit transport de càrrega des del camp elèctric del panell cap al camp de l'objecte que fa contacte. La corrent generada és conduïda des de cada cantonada del panell, aquest procés és mesurat amb els sensors col·locats a aquestes cantonades i un microprocessador interpola la posició exacte on s'ha fet el toc (veure *Figura 2*). Aquest tipus de panells poden tenir una gran precisió.

6.1.2.2 Panell Tàctil Basat en Capacitància de Projecció

De totes les tecnologies estudiades, aquesta és la més costosa econòmicament, a més a més, el seu funcionament és pitjor. Peró no tot són inconvenients, aquests dispositius poden garantir una gran resistència mecànica que els fa ideals per a ambients durs com instal·lacions públiques.
Aquest tipus de superfícies poden ser cobertes per un material no conductor (amb un màxim d'amplitud al voltant de 20 mm) sense afectar negativament al seu funcionament.
Quan és utilitzat per dispositius multitàctils, una graella molt prima formada per cables de micròfons és instal·lada en mig de dos capes protectores de vidre (veure *Figura 3*). Quan és tocat, es forma capacitància entre el dit i la graella de sensors, i la localització del tocament es pot ubicar mitjançant un computador que mesura les característiques elèctriques de la graella. La precisió d'aquest tipus de tecnologia és similar a la basada en Capacitància de Superfície, malgrat la transmissió de llum en aquest cas és superior perquè la graella de sensors pot ser construïda d'una forma que la superfície en qüestió sembla transparent. Finalment, la detecció de múltiples tocs simultanis es pot interpretar més fàcilment amb la Capacitància de Projecció que amb la de Superfície.

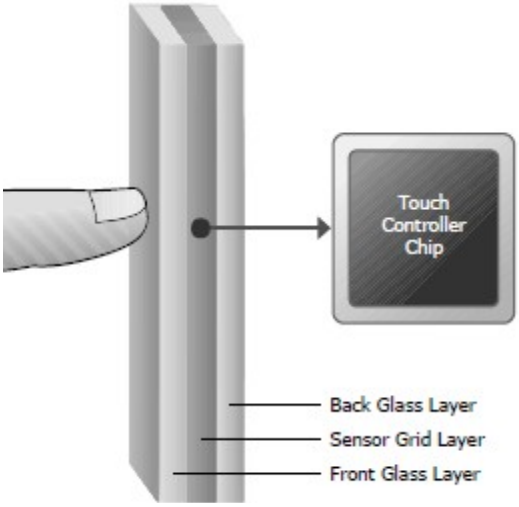

Figura 3. Una fina graella està protegida per dues capes de vidre. La capacitància es forma entre el dit i la superfície en el moment de contacte. Es mesura el canvi de propietats elèctriques per determinar la posició del contacte.

Exemples de dispositius construïts amb aquest tipus de tecnologies son el *SmartSkin* [2](desenvolupat per Jun Rekimoto al 2002, amb l'ajuda de Sony Computer Science Laboratories) i el *Diamond Touch* [3](desenvolupat pels laboratoris de Recerca Elèctrica de Mitsubishi – MERL: Mitsubishi Electric Research Group) estudi desenvolupat al 2003 a partir del treball de Rekimoto.

6.1.3 Superfícies Tàctils basades en Superfície d'Ona (SAW)

Aquest tipus de tecnologies processen ones d'alta freqüència que es propaguen sobre la superfície d'una pantalla de vidre. L'atenuació de les ones resultants pel contacte d'un dit sobre el vidre és utilitzada per trobar les coordenades on s'ha produït el contacte; en poques paraules, localitza o guia per soroll. SAW (Surface Acoustic Wave), normalment utilitza una tècnica de *temps de vol*, bàsicament es mesura el temps que triga la perturbació en arribar als sensors per localitzar les coordenades del tocament. Aquest enfocament és possible quan el medi es comporta d'una forma no dispersiva, o sigui, que la velocitat de propagació de les ones és la mateixa per a totes les freqüencies.

Aquesta tecnologia no és nova, i s'ha demostrat que els sistemes òptics són molt més robusts que els de SAW, ja que aquests tenen un alt nombre de limitacions i problemes bastant significatius. Ja hem explicat a l'inici que tots els sistemes, excepte els òptics, no estan dins del nostre abast, però seria interessant la idea de combinar aquest sistema amb els òptics per obtenir millors resultats (o almenys estudiar quins avantatges tindria).

6.1.4 Superfícies Tàctils Òptiques (Optical Based Touch Surfaces)

Aquest és el tipus de tecnologia triat per desenvolupar el nostre projecte, tecnologia que fa servir una webcam o càmera de vídeo per a la captació d'imatges (basada en la il·luminació per infrarojos). Té molta resolució i el sistema és molt robust, això junt amb el baix cost econòmic que comporta la fabricació d'aquest dispositiu són motius prou significatius per triar-la. El problema amb el que es troba aquesta tecnologia és que, degut a que fa servir la il·luminació infrarojos, pot patir interferències de la llum provinent del medi ambient. La recerca realitzada ha trobat els següents tipus de sistemes relacionats amb aquesta tecnologia [4].

6.1.4.1 Frustrated Total Internal Reflection (FTIR)

És el punt de partida pels sistemes MT, la tecnologia FTIR està basada en *reflexió total interna òptica* amb una superfície interactiva, es a dir, el seu funcionament consisteix en que la il·luminació infraroja és atrapada dins d'una superfície acrílica per reflexió interna. Quan es toca la superfície de l'acrílic aquesta reflexió interna de la llum es "frustrada", provocant que la llum es dispersi cap avall on és recollida per una càmera d'infrarojos.

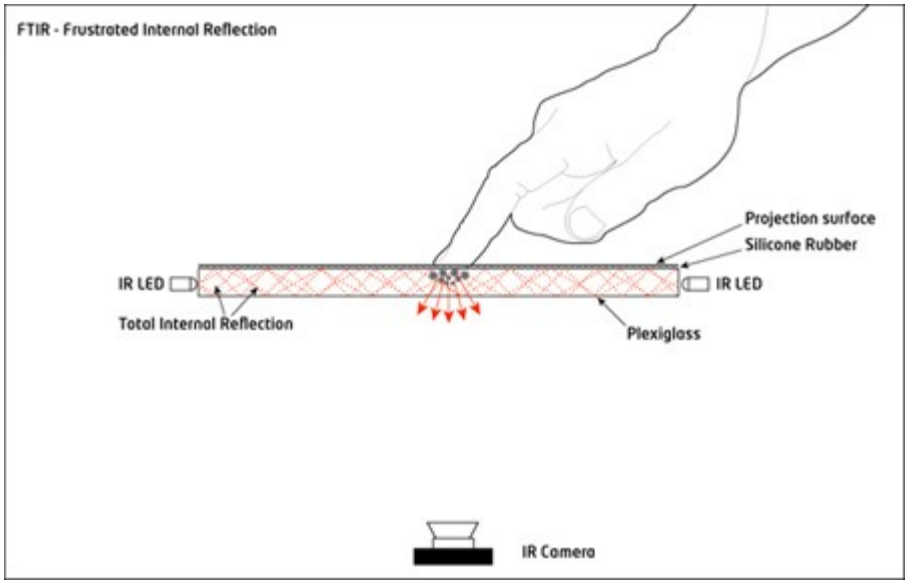

Figura 4. FTIR

Sovint es fa servir una capa de silicona com a *superfície compatible* per ajudar a millorar la sensibilitat del dispositiu (quan es fa un tocament o s'arrossega el dit per la superfície). Quan toquem una superfície acrílica cal pressionar fortament amb els dits per tenir l'efecte FTIR, amb una *superfície compatible* (com la silicona), es millora aquesta sensibilitat.

Pros:

- No cal una caixa tancada (per la llum).
- Blobs (punt de contacte amb la superfície) tenen un contrast fort.
- Permet variar la pressió del blob.
- Amb una *superfície compatible*, pot ser utilitzat amb quelcom tan petit com una punta de ploma.

Contres:

- Es requereix iipus específics de LEDs (infrarojos) i soldadura requerit.
- Requereix una 'superfície compatible' (silicona) per a millors resultats.
- No pot reconèixer objectes i marcadors distintius.
- No es pot utilitzar una superfície de vidre.

Tenir en compte els objectius i requeriments del nostre projecte triem aquest sistema per al desenvolupament del mateix. Considerem que les funcionalitats que ens ofereix compleixen perfectament amb els objectius marcats.

6.1.4.2 Diffuse Illumination (DI)

Té un hardware semblant al FTIR, tots 2 sistemes fan servir una càmera d'infrarojos però DI utilitza els infrarojos *per sota* de la superfície. Col·loca un difusor (material que distribueix la llum) per sobre o sota de la superfície de contacte, llavors, quan un objecte la toca, reflecteix més llum que el propi difusor o d'objectes en el fons, la llum extra és detectada per una càmera. Aquest sistema aprofita el fet que la superfície difumina la llum, esborrant les imatges dels objectes que es troben a una certa distància.

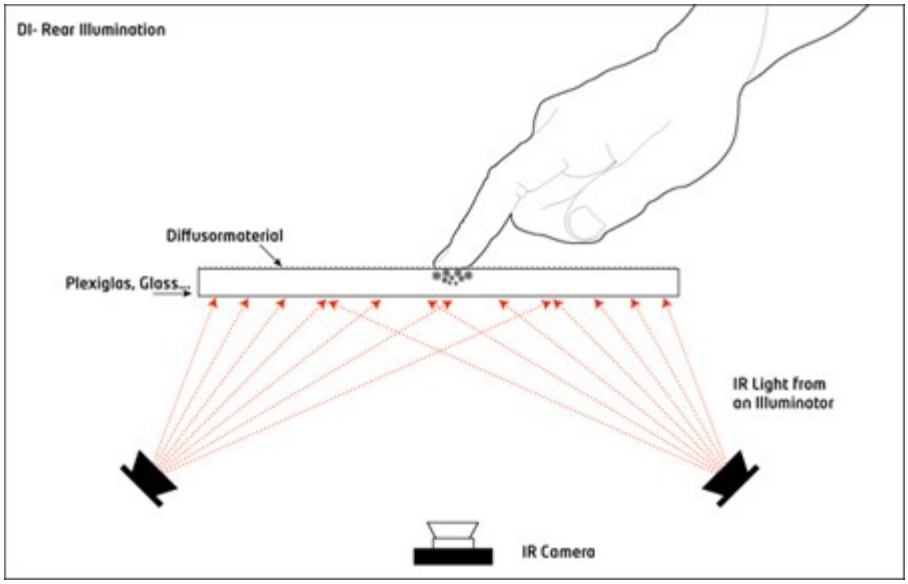

Figura 5. DI

Pros:

- Permet el seguiment i la identificació d'objectes a distància.
- No cal una superfície compatible, només un difusor per sobre o sota de la superfície.
- Es pot utilitzar qualsevol material transparent com el vidre (no només de l'acrílic).
- No requereix LEDs.

Contres:

- Difícil aconseguir una il·luminació uniforme.
- Blobs tenen baix contrast.
- Moltes possibilitats de detectar falsos blobs .
- Requereix una caixa tancada.

Descartem aquest sistema principalment pel tema de la distribució uniforme de la llum infraroja. Ja hem comentat en els contres que és difícil aconseguir-ho, i la dificultat augmenta com més gran és la superfície a utilitzar, si comptem que el que volem fer, ha de tenir les dimensions d'una pissarra, serà força gran. El tema del baix contrast dels Blobs i les possibilitats de detecar-ne de falsos no es compensen amb l'avantatge de poder captar objectes (funcionalitat adequada pels dispositius tipus *taula*, però no pel nostre que estarà penjat d'una paret horitzontalment).

6.1.4.3 Diffuse Surface Illumination (DSI)

Aquest sistema neix com a solució al problema de la distribució uniforme de la llum per la superfície explicat anteriorment amb el sistema de la DI. Bàsicament s'utilitza la configuració FTIR estàndard amb un marc de LEDs i afegeix una espècie de l'acrílic especial (compost internament de milers de partícules que actuen com petits miralls) que permet reflectir la llum per tota la part interna del material.

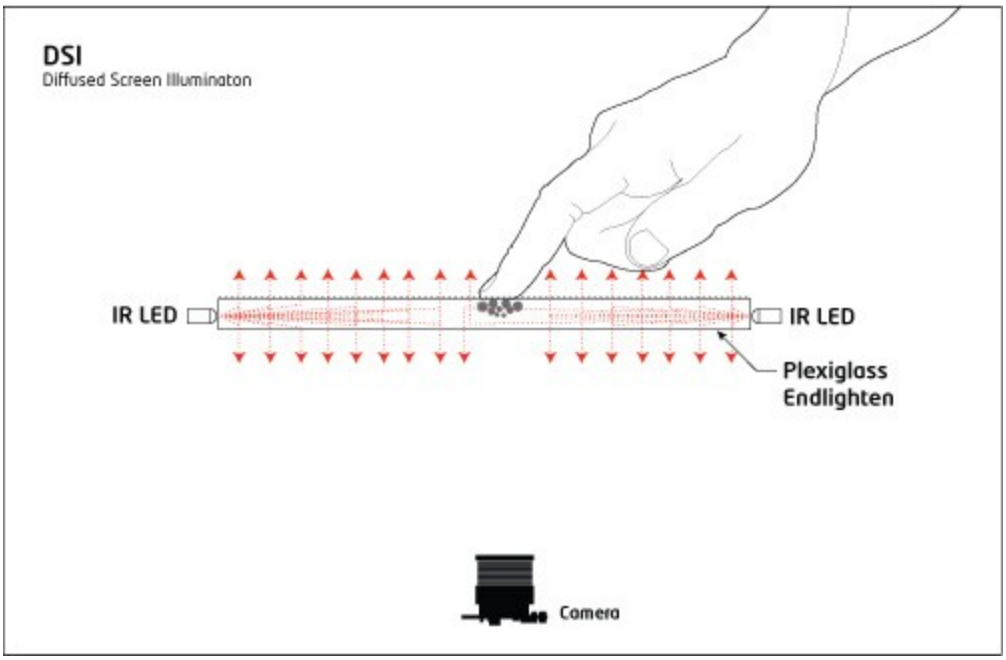

Figura 6. DSI

Pros:

- No té problemes en aconseguir una distribució uniforme de la llum, fàcil posada a punt del dispositiu.
- No cal una superfície compatible.
- Fàcil canvi a FTIR i viceversa.
- És sensible a la pressió.
- Pot detectar objectes i marques distintives.

Contres:

- Blobs tenen poc contrast comparat amb la resta de sistemes.
- Problemes amb l'ambient a causa d'aquest poc contrast.
- És possible que hi hagi restriccions de la mida a causa de la fragilitat del material.
- L'acrílic especial té un cost elevat.

Els majors avantatges d'aquest sistema són la facilitat de posta a punt i que té les mateixes característiques que el DI més la sensibilitat a la pressió. Malgrat tot, els problemes de contrast, amb l'ambient i les restriccions en la mida són massa grans com per adoptar aquesta tècnica en el nostre projecte.

6.1.4.4 Laser Light Plane (LLP)

La llum infraroja d'un o varis làsers estan brillant per sobre la superfície. El plànol de la llum d'aquest làser de llum infraroja és d'aproximadament 1mm de gruix i està situada just a sobre de la superfície quan el nostre dit toca la superfície la llum del làser és dispersada i la càmera l'enregistra com un 'blob'.

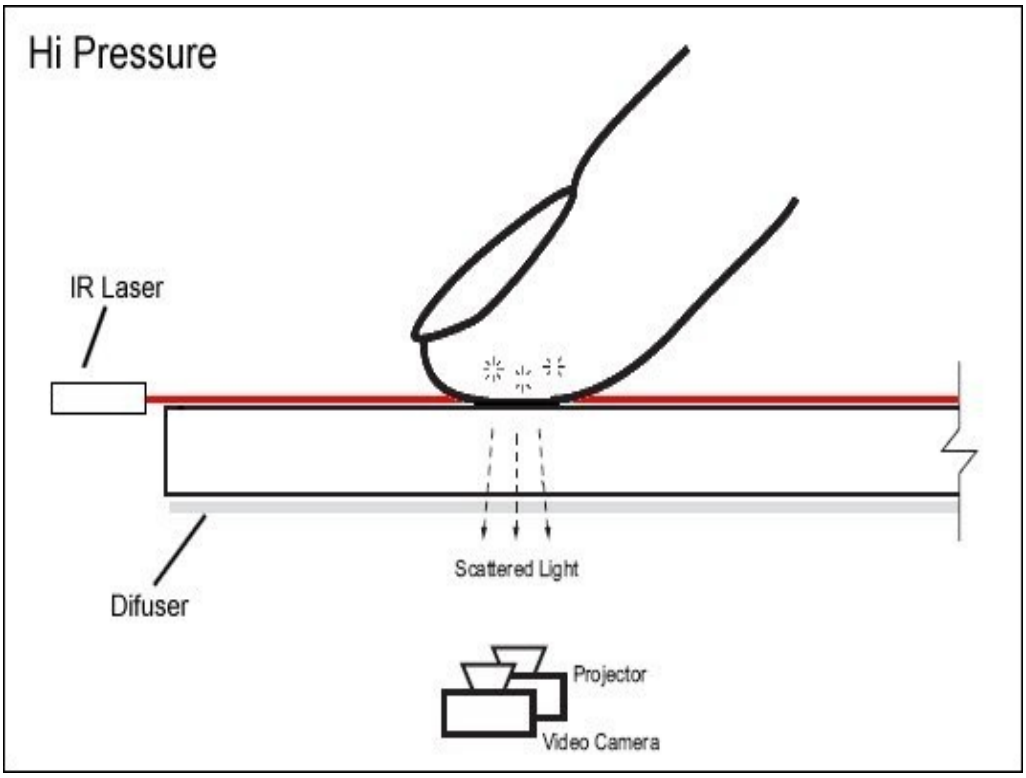

Figura 7. LLP

Pros:

- No cal una superfície compatible.
- Es pot fer servir qualsevol material transparent com el vidre
- No requereix LEDs.
- Pot ser significativament més barat que els altres.
- No cal una caixa tancada.

Contres:

- No és sensible a la pressió (ja que la intensitat de llum no canvia amb la pressió).
- Pot causar oclusió (si es posa un dit al costat d'un altre, no es crearà la llum difusa d'un d'ells)

Descartem aquest sistema perquè el problema de l'oclusió no ens permetia desenvolupar una de les funcionalitats desitjades pel nostre projecte: la capacitat d'escriure, al mateix temps, tantes persones com es vulgui, a la PM.

7. Prototipus

Quan parlem de treballar amb tecnologia multitàctil, ens referim al treball on actuen de forma simultània un conjunt de tècniques Software d'interacció home-ordinador i els dispositius de Harware necessaris. Per tant, ens caldrà una pantalla tàctil que sigui capaç de reconèixer de forma simultània múltiples punts de contacte i el Software necessari per treballar amb la informació rebuda d'aquests dispositius.

A continuació es presenta tota la informació sobre el dispositiu construït per a aquest projecte, descrivint amb tot detall els passos que s'han seguit per a fer-ho.

7.1 Disseny

Per al disseny del prototipus, com ja s'ha vist al capítol *Respecte la Tecnologia Multitàctil,* usarem la tecnologia FTIR (Frustrated Total Internal Reflection). En quest apartat recordarem una mica aquesta tecnologia i explicarem en detall tots i cadascun dels components del prototipus construït.

7.1.1 Tecnologia utilitzada

A partir de l'estudi previ de totes les tecnologies tàctils que tenim al nostre abast, el resultat va donar com a millor opció per les necessitats del projecte la tecnologia tàctil basada en la visió anomenada FTIR.

Es considera que la FTIR és el punt de partida per als dispositius Multitàctils a causa de la facilitat de muntatge i baix cost, està basada en *reflexió total interna òptica* amb una superfície interactiva, en resum, el seu funcionament consisteix en que la il·luminació infraroja és atrapada dins d'una superfície acrílica per reflexió interna. Quan es toca la superfície d'acrílic aquesta reflexió interna de la llum és *frustrada*, provocant que la llum es dispersi cap avall on és recollit per una càmera d'infrarojos.

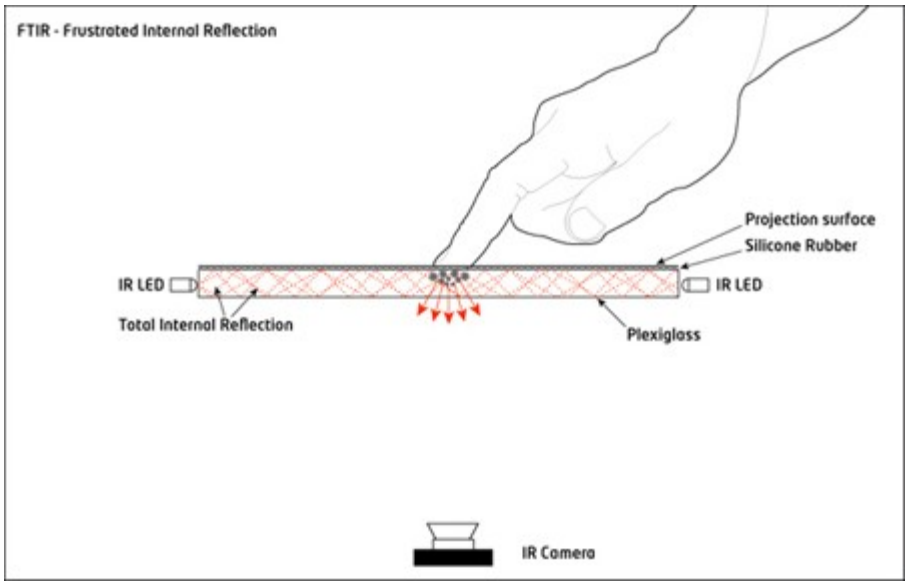

Figura 1: FTIR

Com mostra la figura, es fa servir una capa de silicona com a *superfície compatible* per ajudar a millorar la sensibilitat del dispositiu (quan es fa un tocament o s'arrossega el dit per la superfície). Quan toquem una superfície acrílica cal pressionar fortament els dits per tenir l'efecte FTIR, amb una *superfície compatible* (com la silicona), es millora aquesta sensibilitat.

Per a la superfície acrílica utilitzarem el metacrilat on haurem fet la instal·lació d'un circuit de LEDs (Light Emitting Diodes) de llum infraroja.

Una vegada això funciona, al posar en contacte el nostre dit amb la superfície del dispositiu, es produeix el fenomen abans explicat formant el que anomenarem *blobs* (les marques luminescents resultants). Per a captar aquests *blobs* ens caldrà una webcam amb un filtre d'infrarojos (perquè sinó no seria capaç de captar res) i finalment el Software analitzarà les imatges rebudes d'aquesta càmera.

7.1.2 La Reflexió Total Interna Òptica

7.1.2.1 La Reflexió de la Llum

La reflexió és el fenomen pel qual, quan una ona incideix sobre la superfície de separació entre dos medis, la seva energia no passa totalment del primer medi al segon i part de la seva energia surt rebotada cap al medi d'on venia. La relació entre la direcció del raig incident i la del raig reflectit és la següent: l'angle d'incidència del raig és igual a l'angle de reflexió del raig que surt rebotat de la superfície.

7.1.2.2 La Refracció de la Llum

La refracció és el fenomen pel qual, quan una ona incideix sobre una superfície de separació entre dos medis, la seva energia passa del primer medi al segon medi amb un angle diferent al del raig incident.

7.1.2.3 La Reflexió Total Interna [1]

La Reflexió Total Interna és el fenomen que es produeix quan un raig de llum, al intentar travessar d'un medi amb índex de refracció *n* més petit que el medi al que vol passar, no és capaç de travessar-lo i surt reflectit completament.

Aquest fenomen, es produeix només per a certs angles d'incidència a partir d'un cert angle anomenat angle crític. Aquest fenomen és utilitzat per exemple en la fibra òptica per enviar els feixos de llum al seu destí, que han de rebotar sobre les parets internes de la fibra per poder salvar les curvatures que es creen en el seu recorregut. L'aplicació més important per nosaltres és la dels dispositius multitàctils ja que utilitzarem aquesta propietat de la llum per a crear la nostra superfície tàctil i per enviar la informació al nostre receptor.

7.1.3 La Superfície de la Pantalla en Detall

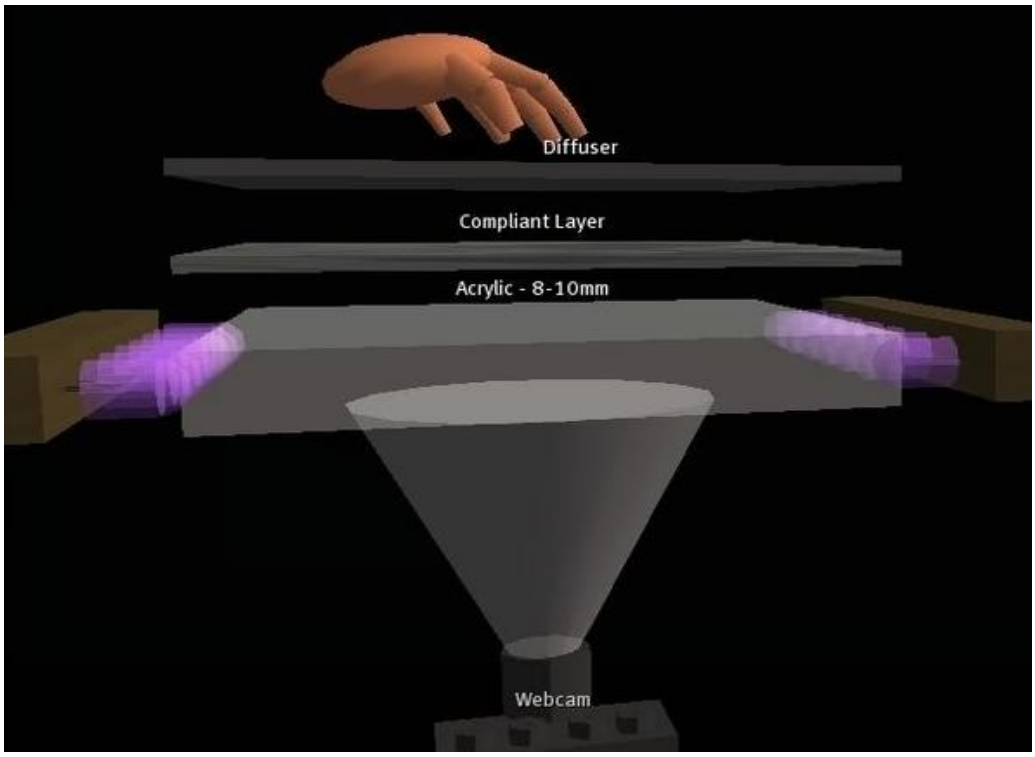

Figura 2: Arquitectura del prototipus

7.1.3.1 Difusor (pantalla de projecció posterior)

El difusor atura la imatge produïda pel projector, és a dir, no permet que la llum que arriba del projector es reflecteixi cap a ell mateix. El material difusor escollit és el Rosco Grey [2], ja que és el més assequible econòmicament i compleix amb els requisits que necessitem.

7.1.3.2 Capa compatible

S'utilitza per potenciar la brillantor produïda pels punts de contacte (*blobs*) en el moment d'acoblament amb l'acrílic. Aquesta capa es pot fer mitjançant varis tipus de materials, en aquest prototipus s'ha fet servir silicona.

7.1.3.3 Acrílic

Com a acrílic s'utilitza metacrilat (òpticament transparent), que ha de ser aproximadament de 8-15mm de gruix per millorar l'efecte FTIR.

7.1.3.4 Webcam

La webcam s'encarrega de capturar la llum infraroja frustrada que surt del metacrilat. A millors prestacions de la webcam, més claredat en la captació dels *blobs*.

7.1.3.5 Projector

El projector s'utilitza per projectar la imatge de l'ordinador a la pantalla del prototipus.

7.1.4 El Circuit Elèctric

7.1.4.1 LEDs Infrarojos

Per realitzar l'efecte de reflexió interna al metacrilat, utilitzarem LEDs IR. *Quants?*. En realitat no hi ha fórmules per calcular això. Aquest és un nombre que depèn de molts factors com: mida de la pantalla, potència dels LEDs, la sensibilitat de la càmera web, la qualitat del filtre IR, la llum ambiental, etc. Aquest prototipus té un LED cada 2.5cm del voltant de la superfície del metacrilat.

L'objectiu és tenir *blobs* visibles de llum infraroja reflectida pel toc dels dits. La paraula important aquí no és *blob* o llum infraroja, la paraula important aquí és *visible*. Això significa que aconseguim que els *blobs* siguin visibles i clars. En altres paraules, si no hi ha llum ambiental, si la càmera que s'està utilitzant és molt sensible i si s'està utilitzant un filtre de IR bo, les possibilitats d'aconseguir visualitzar correctament els *blobs* augmenten. D'aquesta forma si ens falta un d'aquests factors que milloren la visibilitat a la nostra configuració, s'ha d'equilibrar o bé posant més LEDs o utilitzant millors dispositius mencionats abans.

7.1.4.2 Electrònica i electricitat

Per a montar el circuit amb els LEDs, fem servir un circuit de LEDs en paral·lel [3].

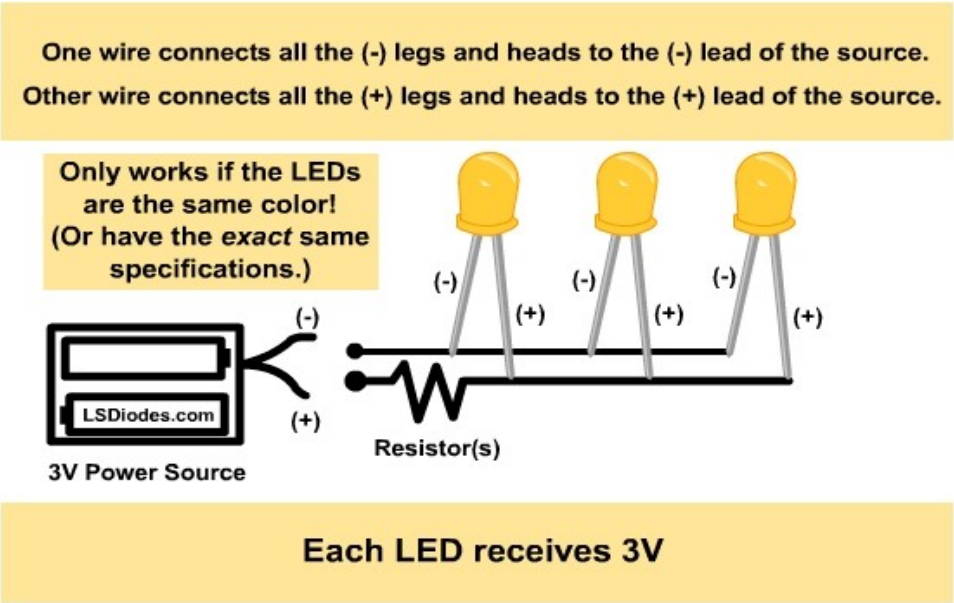

Figura 3: Circuit en paral·lel

7.1.5 Projector i Pantalla de projecció posterior (retroprojecció)

Per projectar la imatge de l'ordinador, necessitarem un projector i la pantalla de projecció.

Per tenir imatges clares a la pantalla, necessitarem utilitzar algun tipus de material especial que faci de superfície de projecció. Hi ha molts tipus de materials que poden fer aquest treball, i per descomptat hi ha uns millors que altres (també més cars).

El projector projectarà una imatge en la pantalla de projecció, aquesta pantalla es col·locarà a sobre del metacrilat.

S'utilitzarà una pantalla de projecció posterior (que implicarà que el projector estigui darrera de la pantalla) on es veurà la imatge, que és la millor forma de fer-ho utilitzant la tecnologia tàctil òptica basada en el FTIR.

Un avantatge de col·locar la pantalla de projecció per sobre del metacrilat és que podem interactuar directament amb la imatge. Per contra, no hi haurà imperfecció visual al col·locar la pantalla de projecció per sota del metacrilat. En aquest cas (sota del metacrilat) simplement tindrem la diferència òptica causada pel gruix del metacrilat.

Per aquest prototip farem servir Rosco Grey [4]. Rosco Grey és una pantalla de projecció molt bona, però que bloqueja la llum infraroja, i per això difon els *blobs* si es col·loca sota del metacrilat, i els fa inservibles per l'anàlisi del Software. Per tant, de forma obligatòria s'haurà de posar a la part superior del metacrilat, i això provocarà que els *blobs* siguin molt difusos, perquè no hi haurà un bon acoblament entre el metacrilat i el Rosco Grey, aquest problema tindrà solució mitjançant una espècie de capa compatible.

Aquest material ens permetrà a més, que la Pissarra Multitàctil pugui treballar en tot tipus d'entorns, ja que no volem cap tipus d'interferència de la llum infraroja provinent de l'ambient.

7.1.6 Capa Compatible (Compliant Layer)

El problema que causa la necessitat d'aquesta capa és que no hi ha un bon acoblament entre el metacrilat i el Rosco Grey (l'usuari ha de que pressionar amb força per aconseguir un *blob* decent). La capa de silicona ajuda amb aquest acoblament ja que les bombolles d'aire que n'hi han entre el metacrilat i la pantalla de projecció es veuen molt reduïdes. Permetent a l'usuari obtenir *blobs* clars fent menys força,

7.2 Muntatge

Descripció pas a pas del muntatge del prototipus.

3.1. Consideracions de Disseny

S'ha decidit que el prototipus tingui la pantalla en posició vertical tot i que és més fàcil construir un prototipus en posició horitzontal (taula).

La mida del prototipus, tot i la recomanació de tenir una pantalla amb una relació d'aspecte 4:3 [5], per aquest prototipus s'ha fet servir una placa de metacrilat de 38x46cm. S'ha de tenir en compte que la mida del prototipus creixerà directament proporcional a la mida de la pantalla perquè s'ha de considerar també la distància que el projector necessita per cobrir l'àrea de visualització així com el marc de la pantalla.

3.2. Materials utilitzats (el que necessitem)

- Ordinador
- Projector
- Acrílic (metacrilat), mínim 8mm de gruix
- Pantalla de projecció: ROSCO GREY
- Webcam (amb filtre IR)
- LED IR 880nm Osram SFH485
- Filferro i resistències
- Suport per la pantalla
- Silicona
- Toluè
- Marc de fusta
- Eines

7.2.1 Tractament de l'Acrílic

L'acrílic és un material que internament reflecteix la llum infraroja. Aquest material és fort i funciona molt bé com a guia d'ona òptica.

La superfície que ha de rebre la llum infraroja ha de ser transparent, per això, és millor si l'acrílic ha sigut tallat amb làser.

Figura 4: Metacrilat transparent

Està clar, que per tenir l'aspecte d'una Pissarra, s'ha de construir un marc que suporti a tot el prototipus. Per això, s'ha construït un marc de fusta que permeti tractar al prototipus com si fos una pissarra en petit.

7.2.2 Pantalla de Projecció i la Capa Compatible

El problema de fer servir una pantalla de projecció (Rosco Grey) sobre l'acrílic és que es generen *blobs* poc utilitzables, aquests *blobs* són febles degut al mal acoblament entre l'acrílic i la pantalla de projecció. S'ha de pressionar dur per tindre un *blob* capaç de ser utilitzat pel Software.

Tot aixo té que veure amb la física, es tracta de l'índex relatiu de refracció i tenim aquest fenomen degut a l'aire que hi ha entre la pantalla de projecció i l'acrílic. Per exemple, si toquem l'acrílic amb els nostres dits, tenim partícules minúscules d'aire entre l'acrílic i els nostres dits. Amb la capa compatible prevenim que les partícules d'aire siguin moltes, donant com a resultat *blobs* més clars i brillants.

La capa compatible ens permetrà mantenir els *blobs* a la pissarra quan arrosseguem els dits.

La capa compatible és el resultat de barrejar silicona amb Toluè (tipus de dissolvent).

Figura 5: Silicona i Toluè

La solució resultant d'aquesta barreja es posa sobre la pantalla de projecció mitjançant un 'rodillo' de pintar. Realitzem aquesta acció fins a aconseguir una capa d'1 mm de gruix (de 4 a 6 vegades).

Figura 6: Eines per crear la capa compatible

7.2.3 Cablejat del circuit electrònic

S'han utilitzat 62 LEDs de llum infraroja (LED IR 880nm Osram SFH485) col·locats al voltant de l'acrílic I en pral·lel.

La clau per tindre l'efecte de reflexió total interna, és posar els LEDs enganxats a l'acrílic.

Figura 8: Circuit de LEDs en Paral·lel

8. Arquitectura del Sotfware

8.1 Introducció:

Aquest document ofereix una espècie de *vista panoràmica* de tota l'arquitectura del sistema. L'objectiu de tot plegat, és capturar i comunicar les decisions importants que s'han fet a nivell estructural, tenint en compte a més, la seva futura expansibilitat.

8.2 Casos d'Ús Significatius

En aquesta secció es descriuen els casos d'ús que es van marcar en els objectius inicials del projecte:

- Paquet "Opcions d'Edició":
  - Escriure
  - Esborrar Escriptura

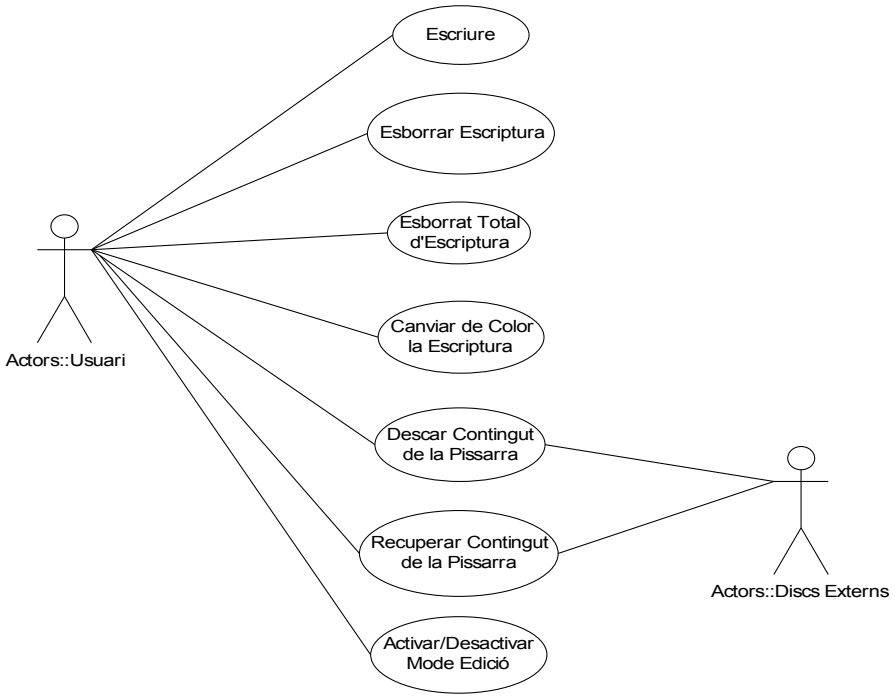

Figura 1: Paquet "Opcions d'Edició"

- Paquet “Gestos-Objectes”:
  - Moure Objecte Visible
  - Redimensionar Objecte Visible
  - Obrir Objecte

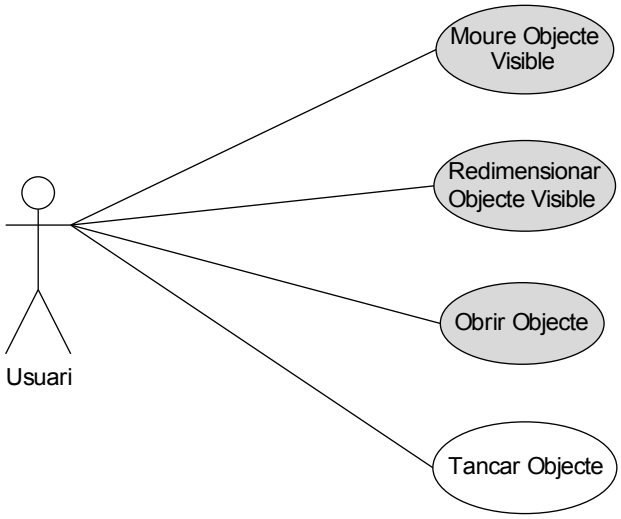

Figura 2: Paquet “Gestos-Objectes”

8.3 Vista Lògica

8.3.1 Arquitectura del Software

L'aplicació estarà implementada seguint el patró MVC (Model Vista Controlador):

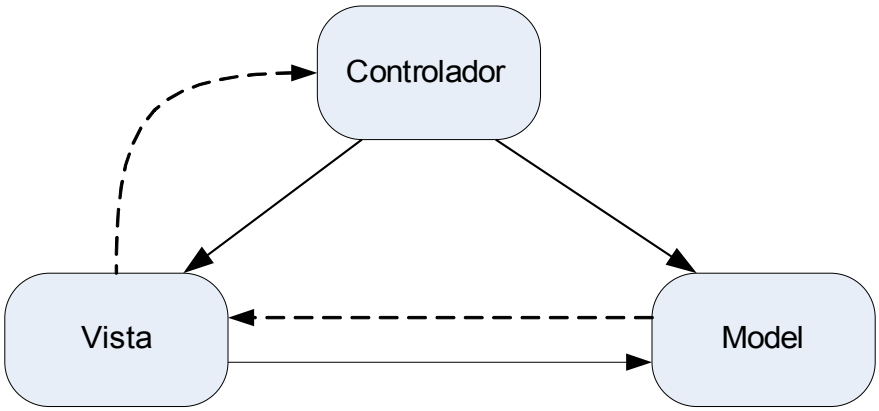

Figura 3: Model MVC
Nota: Les línies sòlides indiquen una associació directa, i les puntejades indirecta.
Les associacions indirectes se solen implementar amb el patró observador,
(veure el capítol *Eines Utilitzades*)

Aquest patró arquitectònic separa les dades de l'aplicació i la interfície de l'usuari (entrada de dades i presentació), d'aquesta forma es permet un desenvolupament, testeig i manteniment independent de cada part.

- *El Model*. S'encarrega de gestionar el comportament i les dades del domini de l'aplicació. Dins de les seves responsabilitats entra el respondre a les sol·licituds de la Vista (per conèixer l'estat de la informació), i respondre a les instruccions del Controlador (per a canviar l'estat de la informació). En el nostre sistema també haurà de notificar a la Vista els canvis produïts a la informació per a fer les accions que convinguin.
- *La Vista*. Converteix el model en una forma adequada per a la interacció amb els usuaris. Una de les raons per la qual és convenient tenir-la en un component per separat, és perquè pot haver diferents tipus de Vistes per a un mateix model per diferents propòsits.
- *El Controlador*. Rep esdeveniments i inicia la resposta fent crides als objectes del Model. Un Controlador accepta les entrades de l'usuari i dóna les instruccions al Model i a la Vista sobre què fer a partir d'aquestes entrades.

El *funcionament* d'aquest patró és el següent:

1. L'usuari interacciona amb la interfície d'alguna forma (en aquest cas, tocant la pantalla).
2. El Controlador capta l'esdeveniment creat (missatges enviats mitjançant el protocol TUIO), i converteix aquest esdeveniment en la seva apropiada acció, que sigui entesa pel Model (convertirem els punts i les coordenades que ens arribin en gestos).
3. El Controlador notifica al Model l'acció realitzada per l'usuari, cosa que possiblement resultarà en un canvi en la informació continguda pel Model (per exemple l'estat de la pissarra al escriure).
4. La Vista necessita el model per a generar l'apropiada interfície d'usuari (p.e. per saber el color de fons de l'àrea d'escriptura o el color amb que s'escriurà). En definitiva, la vista obtindrà la seva informació del Model.
5. La interfície esperarà per més accions de l'usuari i tornarà al pas número 1 (iniciant de nou el cicle).

8.3.2 Vista

Tenim dos vessants a estudiar. Per una banda aquesta capa haurà de captar els esdeveniments sorgits de les accions del/s usuari/s i enviar-les al Controlador (recepció de les dades). I per l'altre s'hauran de mostrar els canvis del Model a la pantalla, així que s'hauran de rebre els esdeveniments d'aquesta capa i fer els canvis corresponents als components gràfics que es controlaran a la Vista (emissió de les dades).

8.3.2.1 La Recepció de les Dades

En l'aplicació la recepció de les dades és farà amb un dispositiu tàctil. Per comunicar-nos amb el dispositiu s'utilitzaran missatges del protocol de comunicació TUIO, aquests missatges han de ser traduïts a un llenguatge que entengui la capa controlador per a ser interpretats.

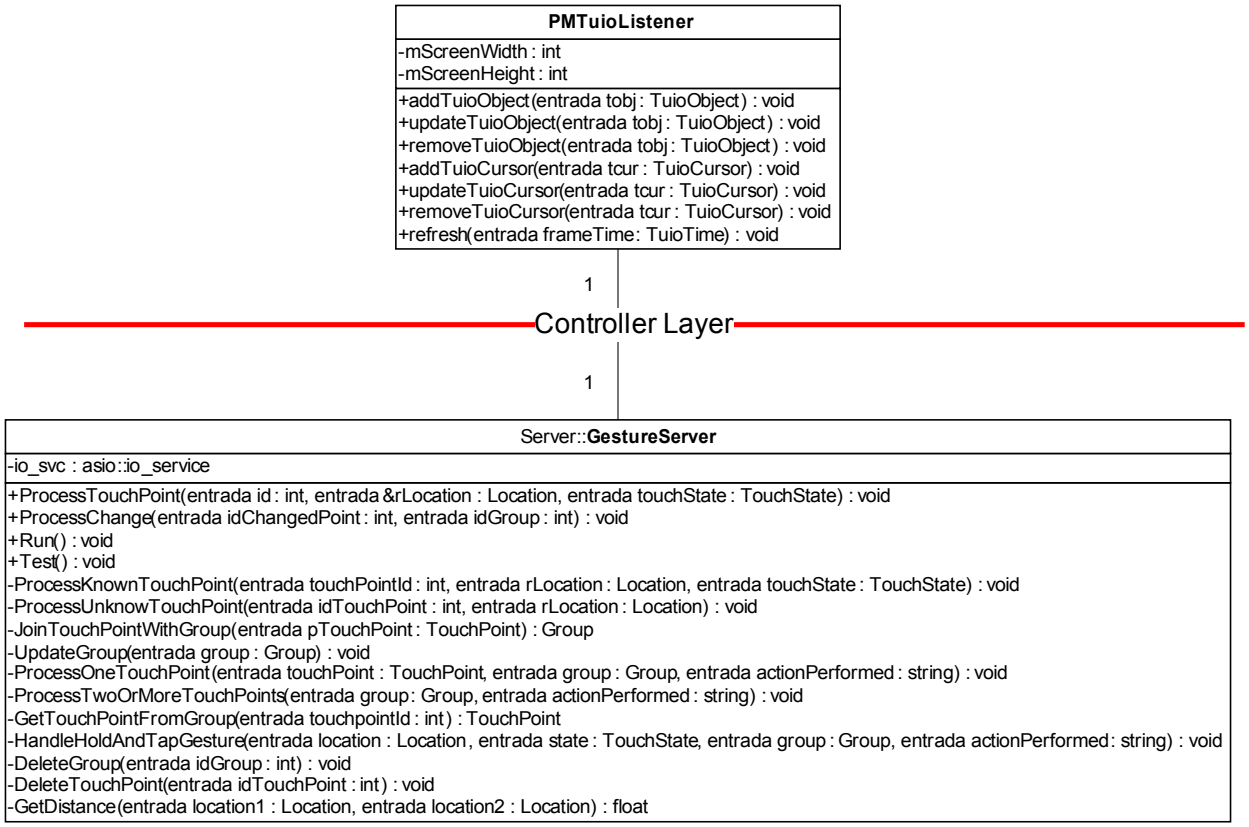

Figura 4: Capa Vista, recepció de dades

8.3.2.2 L'Emissió de les Dades

Per a mostrar les dades per pantalla s'utilitza la llibreria Qt. A continuació el diagrama de classes:

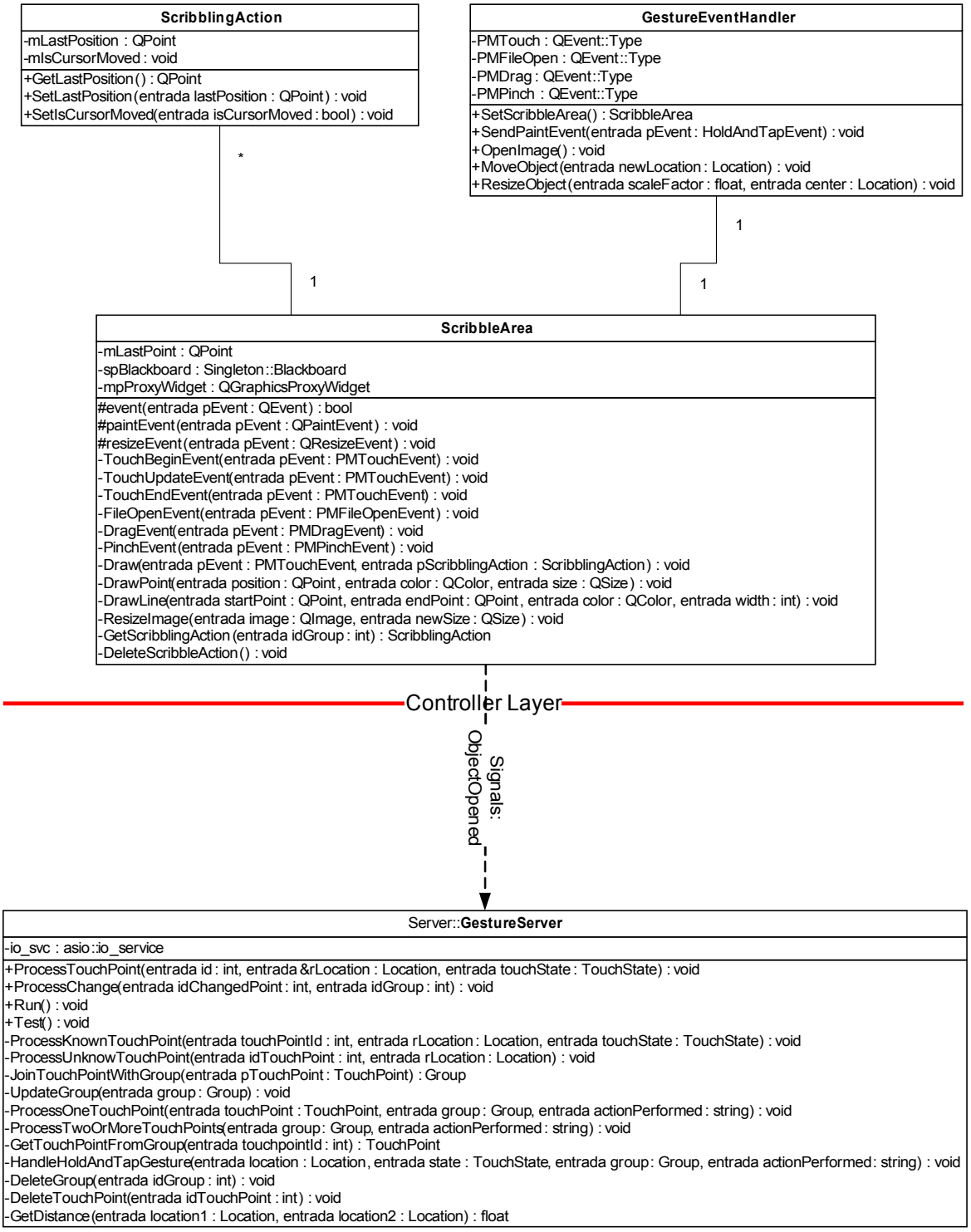

Figura 5: Capa Vista, diagrama de classes

8.3.2.2.1 GestureEventHandler

GestureEventHandler
-PMTouch : QEvent::Type -PMFileOpen : QEvent::Type -PMDrag : QEvent::Type -PMPinch : QEvent::Type
+SetScribbleArea() : ScribbleArea +SendPaintEvent(entrada pEvent : HoldAndTapEvent) : void +OpenImage() : void +MoveObject(entrada newLocation : Location) : void +ResizeObject(entrada scaleFactor : float, entrada center : Location) : void

Figura 6. Classe "GestureEventHandler"

La traducció literal seria "Manegador d'Esdeveniments Gestuals", i s'encarrega precisament d'això, gestionar els esdeveniments a mida que van arribant.

8.3.2.2.2 ScribbleArea

ScribbleArea
-mLastPoint : QPoint -spBlackboard : Singleton::Blackboard -mpProxyWidget : QGraphicsProxyWidget
#event(entrada pEvent : QEvent) : bool #paintEvent(entrada pEvent : QPaintEvent) : void #resizeEvent(entrada pEvent : QResizeEvent) : void -TouchBeginEvent(entrada pEvent : PMTouchEvent) : void -TouchUpdateEvent(entrada pEvent : PMTouchEvent) : void -TouchEndEvent(entrada pEvent : PMTouchEvent) : void -FileOpenEvent(entrada pEvent : PMFileOpenEvent) : void -DragEvent(entrada pEvent : PMDragEvent) : void -PinchEvent(entrada pEvent : PMPinchEvent) : void -Draw(entrada pEvent : PMTouchEvent, entrada pScribblingAction : ScribblingAction) : void -DrawPoint(entrada position : QPoint, entrada color : QColor, entrada size : QSize) : void -DrawLine(entrada startPoint : QPoint, entrada endPoint : QPoint, entrada color : QColor, entrada width : int) : void -ResizeImage(entrada image : QImage, entrada newSize : QSize) : void -GetScribblingAction(entrada idGroup : int) : ScribblingAction -DeleteScribbleAction() : void

Figura 7. Classe "ScribbleArea"

La traducció literal seria "Àrea Escrivible", és un widget de Qt que s'ha dissenyat especialment per a aquest projecte a causa de les seves característiques especials, ja que serà la zona on s'escriurà de forma manual i servirà de base per a la resta de widgets que es vagin obrint a la pantalla:

- És una adaptació del codi d'exemple penjat per QT anomenat "Scribble Example"[2] dissenyat per pintar amb ratolí. El que hem fet és canviar el disseny per pintar amb "PMTouchEvent" en comptes dels originals "QmouseEvent", que seran els esdeveniments que s'enviaran des del "GestureEventHandler" abans citats causats per un moviment d'escriptura a la pantalla tàctil.
  És important destacar que aquest tipus d'esdeveniments especials portaran informació addicional com seria un identificador del grup que s'està modificant i el tipus d'acció que s'està realitzant ja que nosaltres pintarem i esborrarem amb aquests esdeveniments.
- Aquesta classe serà la que ens donarà accés a tots els altres widgets que anem obrint a sobre, ja que tindran

una relació “pare-fill” entre ells. Bàsicament els esdeveniments passaran per “ScribbleArea” i si no corresponen a una acció d'aquest widget s'enviarà als fills per veure si corresponen.

- Contindrà informació sobre totes les accions d'escriptura actives a la pantalla. Per a fer-ho hem creat la classe anomenada “Scribbling Action” (veure *Figura 5*), tindrem una llista d'elements d'aquesta classe de forma que la informació compartida estarà separada, permetent d'aquesta forma l'escriptura de més d'un usuari simultàniament.

8.3.2.2.3 Scribbling Action

ScribblingAction
-mLastPosition : QPoint -mIsCursorMoved : void
+GetLastPosition() : QPoint +SetLastPosition(entrada lastPosition : QPoint) : void +SetIsCursorMoved(entrada isCursorMoved : bool) : void

Figura 8. Classe “ScribblingAction”

Aquesta classe existeix per permetre la escriptura simultània, ja que permet separar les dades de cada grup d'escriptura que estigui treballant en aquell moment a la pantalla. Per a cada acció d'escriptura creada a la pantalla tindrem un objecte d'aquesta classe.

8.3.2.2.4 Esdeveniments Implementats

Qt té funcionalitats implementades per a treballar amb gestos multitàctils, però no dóna cap eina a la seva llibreria per a generar aquests esdeveniments de forma manual, simplement aporta el necessari per implementar aplicacions que reben tipus d'esdeveniments.
Davant d'aquesta petita contrarietat (ja que nosaltres tenim un propi dispositiu generador d'esdeveniments) vam crear els nostres propis esdeveniments a partir de la classe QEvent de QT.

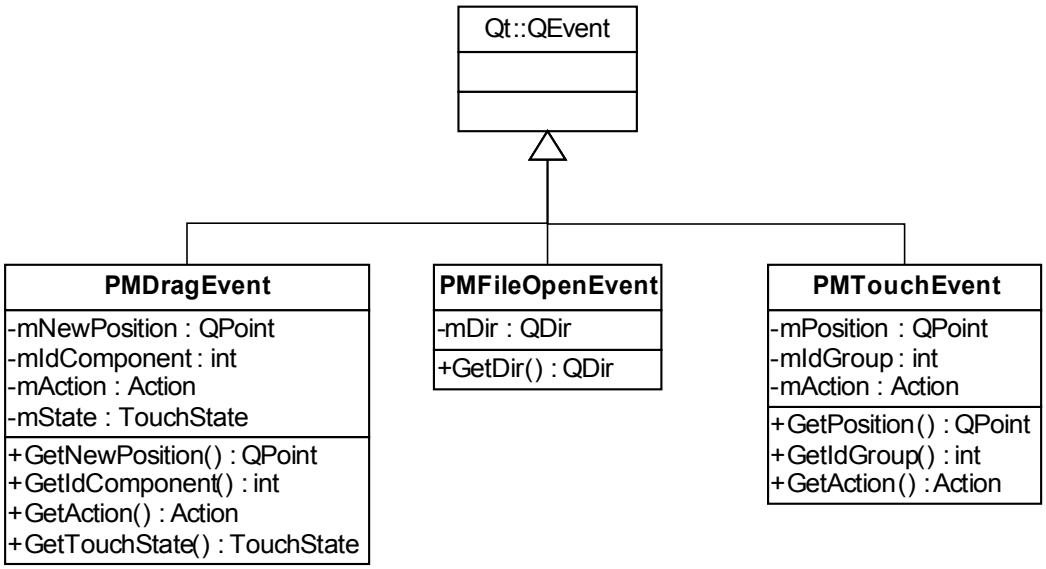

Figura 9. Esdeveniments Implementats

- *PMDragEvent*. S'encarrega dels moviments d'arrossegament d'un component en la pantalla.
- *PMFileOpenEvent*. És cridat al obrir un component a la pantalla.
- *PMTouchEvent*. S'utilitza normalment per a pintar els moviments d'escriptura que es fan a la pantalla, inicialment no s'havia implementat ja que s'usaven els esdeveniments de ratolí de Qt, però al final s'ha fet per permetre accions simultàniament (gràcies al "idGroup" podem distingir a qui correspon l'esdeveniment rebut).

8.3.3 Controlador

Habitualment aquesta capa s'encarrega de la recepció dels esdeveniments per executar les accions corresponents. En el nostre cas, la capa vista rebrà esdeveniments TUIO (que contindrà la informació sobre l'estat i les coordenades) i notificarà al Controlador l'acció realitzada per l'usuari, en aquest punt, seria del tipus "l'usuari està movent el punt "P" a les noves coordenades "(x,y)". Amb això és impossible saber que és el que realment es vol fer, per això ens caldrà fer un primer tractament de la informació per a que aquestes dades del tipus "Coordenades i Estat" passin a ser del tipus "Escriptura, Coordenades", "Arrossegament, Coordenades noves", "Ampliació, Escala", etc. En definitiva cal fer una *interpretació*, del que vol fer l'usuari.

Una vegada interpretada l'acció realitzada per l'usuari ja podem fer el que habitualment fa un controlador, rebre un esdeveniment i notificar els canvis a realitzar al Model.

8.3.3.1 Reconeixement Gestual

Aquesta capa ha estat especialment difícil de dissenyar en aquest projecte, la dificultat recau quan pensem com generar la resposta correcta a una entrada de l'usuari, sobretot tenint en compte que aquest escriurà sobre una superfície relativament gran, amb el nombre de dits que ell desitgi (si vol esborrar farà servir 2 dits o més), tenint en compte que ho pot fer simultàniament amb altres persones. A més, també haurem d'esbrinar sobre quin component s'està polsant, per decidir si generar un esdeveniment d'escriptura, d'arrossegament o de redimensió, etc. En definitiva, hem de fer un treball d'interpretació amb la informació disponible, que són els punts actius sobre la pissarra junt amb els seus estats i coordenades.

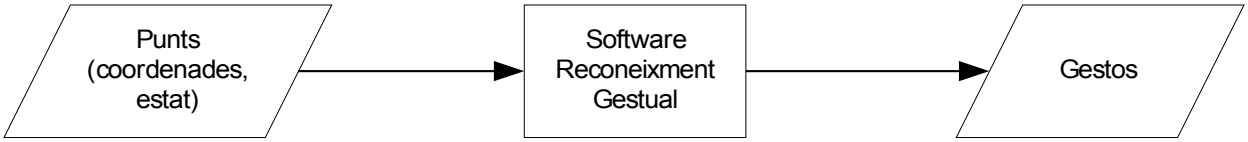

Figura 10. La problemàtica associada a la capa Controlador

8.3.3.1.1 FIB-GestureServer

Aquest és el mecanisme de reconeixement gestual definit per a aquest projecte, l'elecció del nom ha estat clara d'acord amb els objectius proposats pel projecte, entre els que està la possibilitat que altra gent (de la nostra facultat o d'altres) pugui aprofitar el nostre treball per desenvolupar les seves pròpies aplicacions multitàctils.

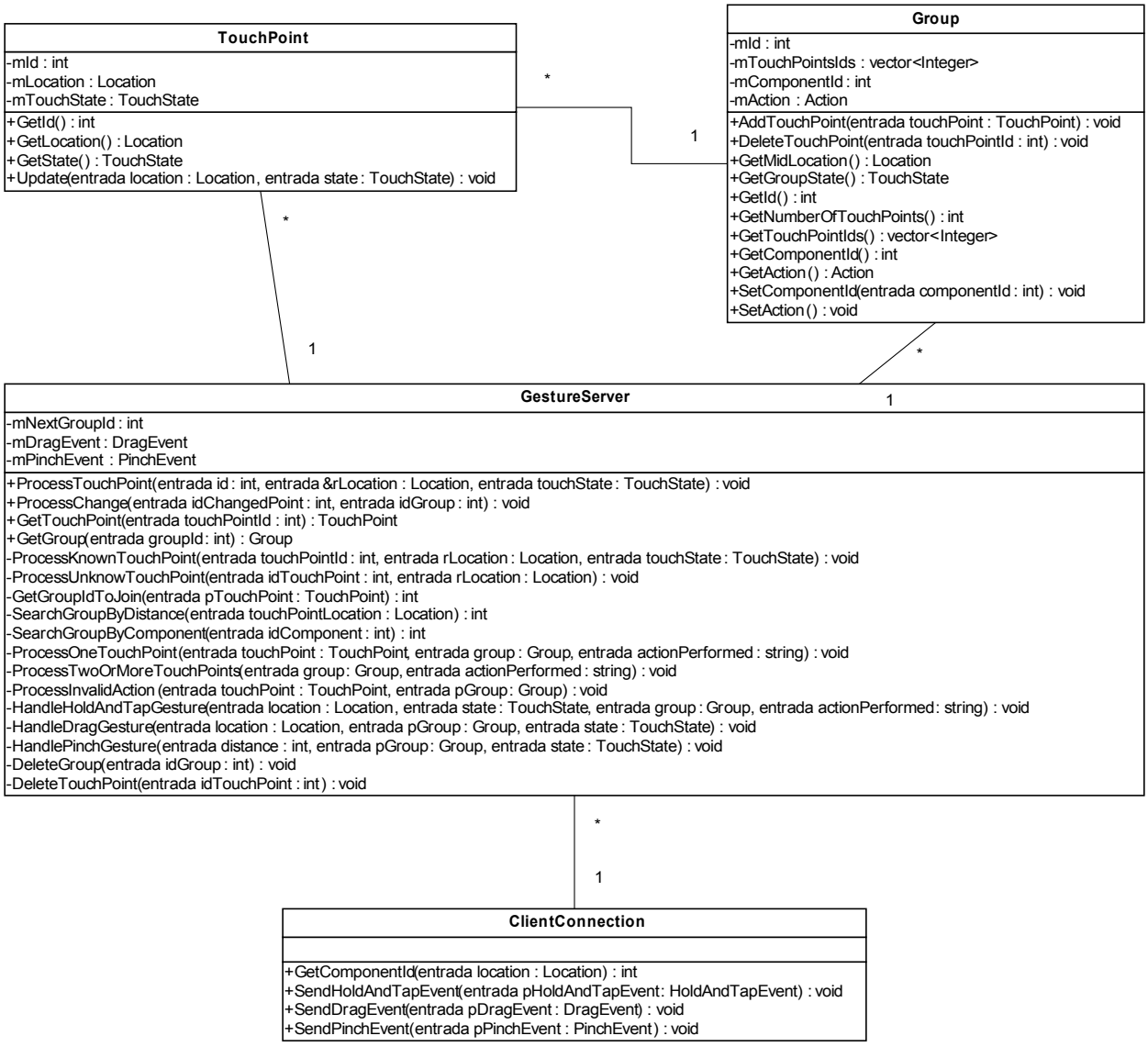

Figura 11. Diagrama de classes del FIB-GestureServer

8.3.3.1.1.1 GestureServer

Les funcions principals d'aquesta classe són:

- Contenir i gestionar tota la informació relativa als punts actius a la pantalla tàctil i com estan agrupats (per això les relacions amb "Group" i "TouchPoint", veure *Figura 14*).
- Formar els grups de punts actius. En aquests moments els grups es formen de dos formes:
  - Per distància. Només en el cas que aquests es trobin en la zona escrivible.
  - Per component. Cada component (imatge, vídeo, etc.) és un grup.
- Interpretació de les dades. Donat que el "GestureServer" té tota la informació sobre els punts (component on es troben, grup que formen, el seu estat, etc.) som capaços d'interpretar el que està realitzant l'usuari sobre la pantalla tàctil i per tant enviar els esdeveniments interpretats a la següent part d'aquesta capa Controlador. Actualment es poden interpretar els moviments d'escriure, esborrar, "click" sobre un botó, moure i redimensionar una imatge.
- Enviar els esdeveniments a la següent part de la capa Controlador. Ho farem mitjançant la classe "ClientConnection" (veure *Figura 14*). Podríem connectar directament amb la següent part de la capa en comptes d'utilitzar aquesta classe, però s'ha preferit deixar-ho així en cas que algú volgués aprofitar el mòdul de Reconeixement Gestual per fer la seva pròpia aplicació, aquesta classe seria la connexió amb la resta del programa.

8.3.3.1.1.2 Group

Aquesta classe contindrà la informació sobre els grups de punts actius sobre la pantalla, dins de les seves característiques cal destacar:

- Coordenades del grup (informació obtinguda mitjançant la funció "GetMidLocation"): De tots els punts que contindrà el grup ens donarà una posició mitjana de tots ells, de moment només funciona per grups d'un i dos punts.
- Estat del grup (informació obtinguda mitjançant la funció "GetGroupState"): En aquest cas el retorn d'aquesta funció serà en funció dels estats dels punts continguts pel grup de la següent forma:
  - Si tots els punts estan en estat de BIRTH (naixement) és retornarà BIRTH.
  - Només que hi hagi un punt en estat MOVE (en moviment), es retornarà MOVE.
  - Nomes que hi hagi un punt en estat de DEATH (mort del punt, o sigui, aixequem el dit), es retornarà DEATH.
- Atribut "mAction": Un grup tindrà associada una acció, això es farà una vegada s'hagi fet la feina d'interpretació, serà necessària aquesta informació per:
  - Generar els esdeveniments: Per exemple, un esdeveniment d'arrossegament pot estar associat a varies accions, moure una imatge o fer lliscar una barra d' "scroll". L'acció associada al grup serà la que s'associï a l'esdeveniment.
  - Invalidar accions. Tenir aquesta capacitat és important, ja que una vegada s'hagi invalidat l'acció d'un grup no farem cas de les actualitzacions rebudes per part dels punts associats del grup. El que farem serà esperar a que arribin les senyals de DEATH i anar esborrant tots els punts fins a que el grup es quedi sense cap i per tant el puguem esborrar també.
    Per exemple, invalidarem una acció d'esborrar quan estiguem esborrant amb dos dits i aixequem un.

8.3.4 Model

Aquesta capa contindrà tot el que siguin dades del sistema, hi haurà dos parts diferenciades. Una es centrarà en la informació centrada a els esdeveniments amb els que treballarà el sistema, això fa referència a tota la informació necessària per a que la capa Vista sigui capaç de fer els canvis. I l'altre estarà centrada en la informació relativa als components oberts a la pantalla, això voldrà dir que haurem de contenir la informació relativa a la seva mida i posició, i tot el relatiu a les seves característiques individuals (que depenen del tipus de component del que parlem)

8.3.4.1 Informació sobre els Esdeveniments del Sistema

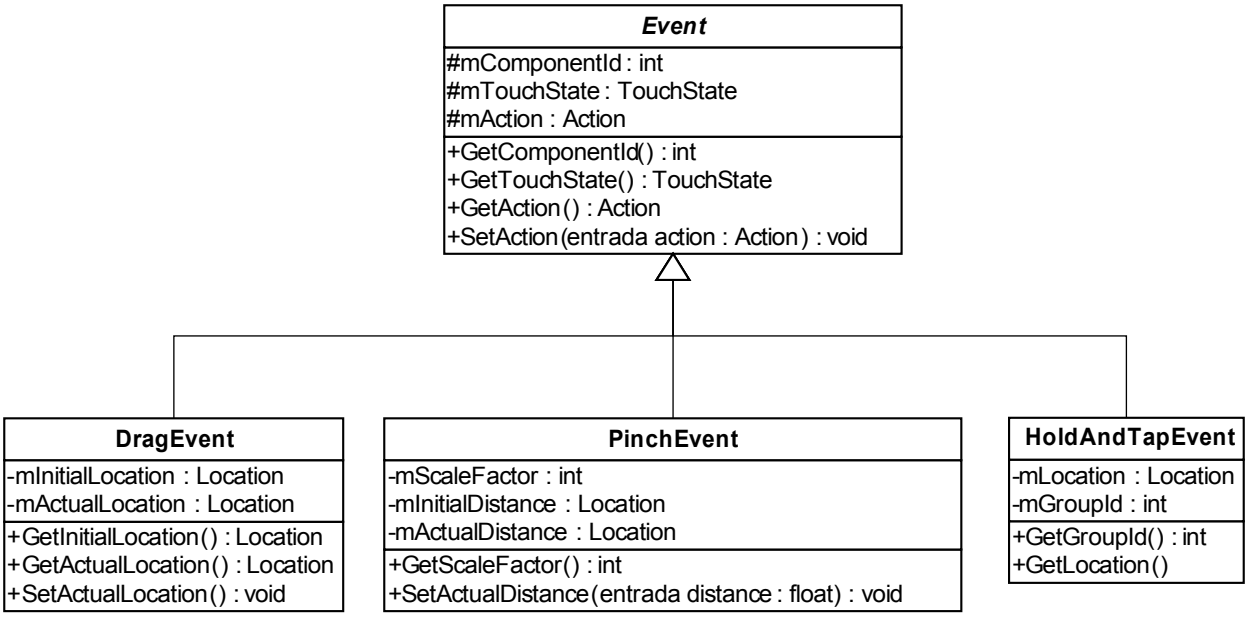

Figura 12. Informació sobre els esdeveniments en el Model

Tenim una classe pare en quan als esdeveniments, tots els que creem tindran una sèrie d'atributs comuns que seran compartits mitjançant aquesta classe abstracte "Event.h". En el nostre Model només tenim tres tipus d'esdeveniments d'acord amb les funcionalitats que hem implementat:

- HoldAndTapEvent. Son tots aquells moviments en que s'ha de fer un seguiment d'una posició d'un o varis dits sobre la pantalla tàctil, això el relaciona amb les accions d'escriptura i esborrat, així com les de simulació del ratolí (els "clicks").
- DragEvent. Relacionat amb les accions per moure o arrossegar un objecte per la pantalla (en el nostre cas, una imatge). A diferència de l'anterior moviment haurem de desar la posició inicial i la actual per fer els càlculs corresponents.
- PinchEvent. Relacionats amb els moviments de redimensió, actualment només funciona separant i unint dos dits a la pantalla.

Aquest sistema afavoreix la creació dels diferents tipus d'esdeveniments necessaris per a fer tota la interfície amb l'usuari (ja que només s'ha realitzat una part com a demostració), així com el perfeccionament dels esdeveniments ja creats.

8.3.4.2 Informació relativa als Components

8.3.4.2.1 Blackboard

Singleton::Blackboard
-mPenWidth : int -mPenColor : QColor -mEraseWidth : int -mEraseColor : QColor -mImage : QImage
+Write(entrada holdAndTapEvent) : void +Erase(entrada holdAndTapEvent) : void +GetPenColor() : QColor +GetEraserColor() : QColor +GetPenWidth() : int +GetEraserWidth() : int +GetImage() : QImage +SetPenColor(entrada color : QColor) : void +SetPenWidth(entrada width : int) : void +SetImage(entrada image : QImage) : void +Signal::Writed(entrada pHoldAndTapEvent : HoldAndTapEvent) : void +Signal::Erased(entrada pHoldAndTapEvent : HoldAndTapEvent) : void

Figura 13. Classe Blackboard

Aquesta classe constituïra el que és la superfície on hi escriurem i tindrà les següent característiques:

- Està implementada segons el patró Singleton (veure apartat d'*Eines Utilitzades*), ja que només tindrem una instància d'aquesta classe activa en l'aplicació en tot moment.
- Contindrà la informació sobre el color i l'amplada amb la que escriurem i esborrarem. També contindrem una variable que desarà la imatge que mostrada a la pantalla ("mImage", a la *Figura 13*)
- Enviarà informació sobre la posició on s'ha de modificar la imatge que mostrarem per pantalla.
- Contindrà la imatge que s'està mostrant actualment per pantalla.

8.3.4.2.2 Toolbar

Singleton::Toolbar
+OpenImageButtonClicked(entrada pHoldAndTapEvent: HoldAndTapEvent) : void +Signal::ButtonClicked() : void

Figura 14. Classe Toolbar

Serà la barra d'eines de l'aplicació, és una part que no ha estat molt desenvolupada en aquest projecte i és susceptible de canvis si es vol afegir funcionalitats al sistema. Com en el cas de la "Blackboard", ha estat implementada com a un Singleton per les mateixes raons, actualment només conté un botó que es utilitzat per a obrir la imatge amb les que farem proves d'arrossegament i redimensió. En la *Figura 14* es pot observar la senzillesa d'aquesta classe en el punt en que ha estat implementada.

8.3.4.2.3 Object

Object
-mPath : string -mCenter : Location
+Drag(entrada pDragEvent : DragEvent) : void +Zoom(entrada pPinchEvent : PinchEvent) : void +GetPath() : string +GetCenter() : Location +SetPath(entrada mPath : string) : void +UpdateCenter() : void +Signal::ObjectDragged(entrada newLocation : Location) : void +Signal::ObjectZoomed(entrada scaleFactor : float, entrada center : Location) : void

Figura 15. Classe Object

Aquesta classe fa referència a tots aquells components visibles oberts sobre la superfície escrivible que es poden moure i redimensionar, la idea és que es podrien obrir mitjançant aquesta classe fitxers multimèdia de tot tipus fent referència a imatges, vídeos, transparències... nosaltres dins dels objectius marcats ens hem limitat només a una imatge cosa que serveix com a demostració de que es pot estendre fàcilment.

- La informació que conté la classe, apart de les pròpies de la classe pare, són el directori on es troba el fitxer i el centre de la imatge. El centre serà important de cara a fer l'ampliació o reducció d'aquesta, ja que s'ha de fer tenint un punt fix amb el que s'han de fer els càlculs.
- La classe emetrà les senyals:
  - Objecte mogut. On enviarem la nova posició on es trobarà.
  - Objecte redimensionat. On enviarem la nova escala i el centre de l'objecte per a poder aplicar els mateixos canvis a la capa Vista.

8.3.4.3 Classes Bàsiques

Aquestes són les classes que fan referència a la informació sobre:

- La “posició”. Coordenades en dos dimensions, representada per la classe “Location”. Aquesta classe permet calcular la distància entre dos punts que s'utilitzarà en la interpretació i en el canvi geomètric dels Objectes.
- L' “acció”. Emmagatzema quina és l'acció realitzada per l'usuari, representada per la classe “Action”.
- L' “estat”. Del punt o del grup, representat per l'enumeració “TouchState”.

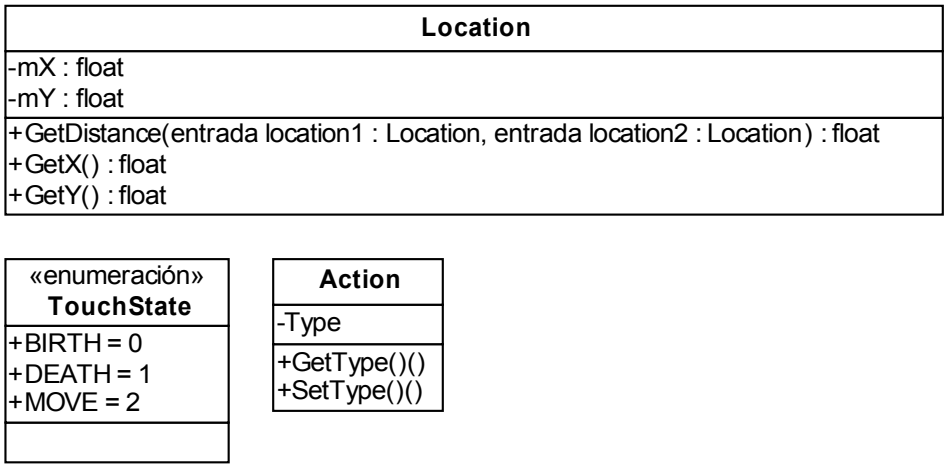

Figura 16. Classes Bàsiques

9. Pla de Desenvolupament del Projecte

En aquest capítol es defineixen les activitats realitzades per a la implementació de l'aplicació de la Pissarra Multitàctil en termes de fases i iteracions. Es descriu els rols i responsabilitats adoptats pels membres que formen el grup de treball i es fa un estudi de la feina realitzada per a fer una estimació aproximada del cost total centrat en el desenvolupament.

9.1 Organització del Projecte

Els rols i feines que han adoptat les persones que han participat en el desenvolupament d'aquest projecte son els següents:

Estudiant	*Rols*
Edmon Martínez Martí	Project Manager del Projecte Arquitecte del Software Dissenyador del Software Programador del Software Tester del Software
Jorge Novoa Portocarrero	Dissenyador del Prototipus Construcció del Prototipus Membre de Suport en el Disseny del Prototipus Programador del Software Tester del Prototipus Tester del Software

9.1.1 Rols i Responsabiltats

Rol	*Responsabilitat*
Project Manager	Aconseguir els objectius proposats. Organitzar la feina durant el transcurs del projecte (per iteració) Establir la metodologia de desenvolupament per assegurar la integritat i la qualitat del projecte.
Arquitecte del Software	Escollir el model d'arquitectura a utilitzar. Establir l'arquitectura de cadascuna de les capes que formen l'aplicació i les interaccions entre elles.
Dissenyador del Software	Definir les responsabilitats, operacions, atributs i relacions per cadascuna de les classes que s'implementaran.
Membre de Suport en el Disseny Softaware	Des del punt de vista del programador, aportar consells, idees i suggerències al disseny inicial d'una part de l'arquitectura del Software donada.
Programador del Software	Una vegada s'han definit totalment les classes aquest s'encarregarà de la seva implementació. Comprovar el correcte funcionament de les parts implementades dins del conjunt del projecte.
Dissenyador del Prototipus	Aconseguir el objectiu proposat. Escollir la tecnologia a utilitzar i el mètode de construcció a adoptar. Aconseguir el material necessari.
Constructor del Prototipus	Fer el muntatge del prototipus.
Tester del Prototipus	Comprovar el correcte funcionament d'aquest i fer les correccions que calguin per a corregir errors que vagin sorgint.
Tester del Software	Comprovar el correcte comportament de cada funcionalitat implementada dins del conjunt de l'aplicació.

9.1.2 Suport

Centre de Realitat Virtual (CRV)	El CRV ens va oferir un lloc on treballar i material que tenien disponible http://www.crvbcn.com/
Marta Fairen	Directora del projecte
Edmundo Martínez Herrera	Suport en el muntatge del prototipus

9.2 Planificació del Projecte

En la planificació del projecte es mostra l'organització de la feina respecte al temps i esforç durant el desenvolupament del projecte.

9.2.1 Elements a Implementar

Els següents elements hauran de ser produïts durant la realització del projecte:

- Confecció de l'Arquitectura del sistema
- Disseny i implementació del Servidor Gestual
- Disseny i implementació de cadascuna de les capes del sistema
- Disseny del Prototipus
- Construcció del Prototipus
- Test de funcionament i rendiment
- Redacció de la Memòria
- Preparació de la Presentació

9.2.2 Dates Límit i Mínims del Projecte

- Entrega de la Memòria: 21 de Gener del 2011
- Lectura del Projecte: 27 de Gener del 2011
- Hores dedicades mínimes per la Enginyeria Superior: 600 hores
- Hores dedicades mínimes per la Enginyeria Tècnica: 300 hores

9.2.3 Planificació

Planificació Software																										
Dates	Mes	Feb		Març		Abr		Mai			Jun		Jul		Ago		Sep		Oct		Nov		Des		Jan	
	Iteració	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	Inici	8-Feb	22-Feb	8-Mar	22-Mar	5-Abr	19-Abr	3-Mai	17-Mai	31-Mai	14-Jun	28-Jun	12-Jul	26-Jul	9-Ago	23-Ago	6-Sept	20-Sept	4-Oct	18-Oct	1-Nov	15-Nov	6-Des	20-Des	3-Jan	17-Jan
	Hores	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Tasques	Inception																									
	Elaboration																									
	Construction																									
	Transition																									

Figura 1. Seguiment de la feina realitzada durant la implementació de l'aplicació

S'han realitzat un total de 25 iteracions d'una duració de dos setmanes cadascuna (1 setmana, 20 hores), amb un total de 1000 hores dedicades. S'ha optat per aquest model flexible d'una dedicació de mitja jornada amb molta flexibilitat, donada la impossibilitat de dedicar jornada completa per part dels dos components del grup de treball. En el següent apartat veurem l'explicació detallada de la feina realitzada en cada etapa del projecte així com la dedicació invertida per cada estudiant en cadascuna d'elles. L'explicació sobre el tipus de tasques realitzades la podrem trobar en el capítol *Metodologia Aplicada al Projecte*.

Planificació Prototipus																										
Dates	Mes	Feb		Març		Abr		Mai			Jun		Jul		Ago		Sep		Oct		Nov		Des		Jan	
	Iteració	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	Inici	8-Feb	22-Feb	8-Mar	22-Mar	5-Abr	19-Abr	3-Mai	17-Mai	31-Mai	14-Jun	28-Jun	12-Jul	26-Jul	9-Ago	23-Ago	6-Sept	20-Sept	4-Oct	18-Oct	1-Nov	15-Nov	6-Des	20-Des	3-Jan	17-Jan
Tasques	Hores	5	5	5	5	40	40	40	20	5	5						5	5	10	10	5	5	10	10	20	40
	Disseny																									
	Obtenció de Materials																									
	Muntatge																									

Figura 2. Seguiment de la feina realitzada durant la construcció del prototipus

En aquest cas, el treball s'ha fet en paral·lel al de les tasques descrites a la *Figura 1*, el nombre d'hores dedicades per iteracions no és tan fixe com en el cas del Software ja que és un treball que s'ha anat fent en paral·lel i on es va donar una primera prioritat a la implementació de l'aplicació, al comparar les dos figures es pot entendre de certa forma com s'ha coordinat la feina (sobretot en l'etapa de construcció on es va aturar el procés durant unes quantes iteracions).

La data d'inici del projecte (8 de Febrer del 2010) va ser escollida en previsió al major dels riscos del projecte, no acabar en les dates límits. Com s'ha pogut comprovar va ser una decisió correcte ja que hem hagut d'utilitzar tot el temps disponible.

9.2.3.1 Etapes de la Implementació del Software

Inception	Estudi de la feina a realitzar i cerca d'antecedents (tant de software com de la tecnologia pel prototipus). Treball realitzat en equip, en aquest punt no hi va haver separació de treball. Total d'hores: 100 hores (Edmon i Jorge)
Elaboration	Disseny de l'arquitectura del sistema, disseny de les capes i les classes que les formen. Aquesta feina es va combinar amb la feina de construcció (el Jorge només va participar en la construcció, malgrat es van fer correccions al disseny a partir de les seves aportacions). Total d'hores: 380 hores (Edmon)
Construction	Implementació segons les pautes marcades a la fase d'El·laboració. Al final d'aquesta etapa es van començar a fer tasques pròpies de la etapa de transició. Total d'hores: 320 hores Divisió de feina: Edmon: 180 hores Jorge: 140 hores
Transition	Documentació de la feina realitzada i proves de cohesió del sistema (junt amb el prototipus). Cal recordar que aquesta feina està relacionada només en la part Software, però el comptatge d'aquestes hores estarà reflectit en la documentació del Jorge en concepte de participació de la part Software. Total d'hores: 200 hores Divisió de feina: Edmon: 140 hores Jorge: 60 hores

9.2.3.2 Etapes de la Construcció del Prototipus

Disseny	En l'etapa Inception de l'apartat anterior es va fer l'estudi sobre la tecnologia, però paral·lelament es va anar començant a dissenyar el prototipus (mides, configuració, etc). La feina en aquesta etapa es va incrementar una vegada feta la divisió de la feina entre els dos membres de l'equip de treball encarant ja l'obtenció dels materials i eines necessàries. Les hores que consten de L'Edmon compten com a hores de participació en el disseny del prototipus. Total d'hores: 160 hores Divisió de feina Jorge: 100 hores Edmon: 60 hores
Obtenció de Materials	En aquesta etapa es correspon al temps dedicat en la obtenció i investigació dels diferents materials i eines necessàries per a la construcció del prototipus (per exemple fustes per fer el suport, metacrilat, silicona, etc.) Jorge: 20 hores
Muntatge	Temps dedicat a la construcció del prototipus. Jorge: 80 hores Edmon: 30 hores

9.2.4 Càlcul dels Costos

Els càlculs s'han realitzant tenint en compte el rols realitzats per cadascú hem assignat els següents costos per hora segons els rols realitzats per cada estudiant (*Apartat 9.1.1*):

- Edmon Martínez: 15 Euros/Hora
- Jorge Novoa: 10 Euros/Hora

9.2.4.1 Cost de l'Aplicació

Edmon Martínez	760 hores
Jorge Novoa	300 hores

El cost de la realització de la part software del projecte serà:
“Cost_Hores_Edmon + Cost_Hores_Jorge”, és a dir 760*15 + 300*10 = 14.400 Euros

9.2.4.2 Cost del Prototip

Jorge Novoa	290 hores
Edmon Martínez	90 hores

El cost total dels materials ha estat de 245 Euros sense incloure els materials aportats pel CRV, com per exemple, la webcam i el projector, el càlcul s'ha realitzat tenint en compte els següents materials:

- Acrílic tallat amb làser (34cm x 46cm): 50 Euros
- Superfície compatible (Silicona i Toluè): 10 Euros
- LEDs infrarojos d'alta lluminositat: 35 Euros
- Rosco Grey: 10 Euros
- Peces de fusta pel suport: 20 Euros
- Altres: 20 Euros

Per tant, el cost total de la fabricació del prototipus és:
"Cost_Hores_Jorge + Cost_Hores_Edmon + Cost_Material", és a dir 290*10 + 90*15 + 245 = 4.495 Euros

9.2.4.3 Cost i Dedicació Total

Edmon Martínez	850 hores
Jorge Novoa	590 hores

Cost_Part_Software + Cost_Part_Hardware = 14.400 Euros + 4.495 Euros = 18.895 Euros

Si mirem l'*Apartat 9.2.2*, ens adonem que se sobrepassa àmpliament els mínims exigits, això ha estat degut a la magnitud del projecte i ja estava previst des del seu plantejament. El cost malgrat ser força elevat és un gast bastant assolible tenint en compte que és el resultat del sou de dos persones al llarg de tot el projecte i el preu que se sol moure en aquest tipus de dispositius/aplicacions que vam veure en el capítol *Anàlisi d'Antecedents i Alternatives*.

10. Conclusions

10.1 Posada en Funcionament i Resultats

Des de que és va aconseguir la primera funcionalitat a l'aplicació (amb les proves de simulació aprovades amb èxit) és va començar la construcció del prototipus, realitzant les proves adients. L'avaluació del funcionament de la funcionalitat implementada a l'aplicació va fer evolucionar la construcció del prototipus.

10.1.1 Versió 1.0

- ✔ Es va fer servir un acrílic de 15mm de gruix.
- ✔ La capa compatible (solució de Silicona amb Toluè), es va ficar sobre l'acrílic.
- ✔ Per tot el voltant de l'acrílic es va fer un forat per ficar els LEDs IR cada 2.5cm.

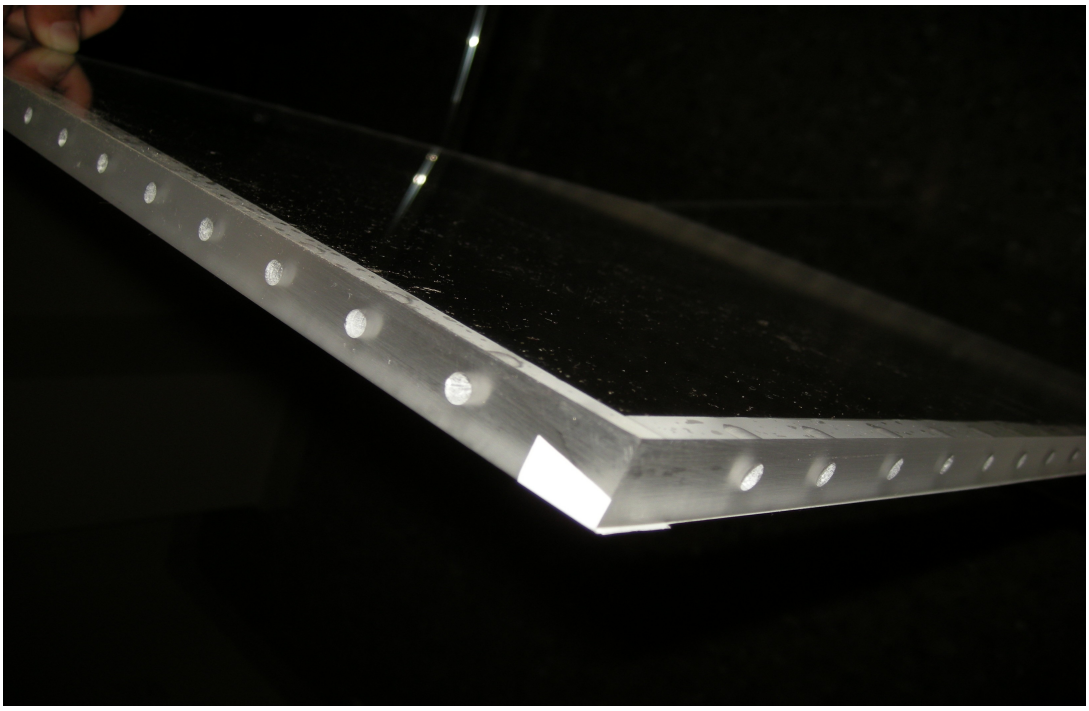

Figura 1: Forats a l'acrílic

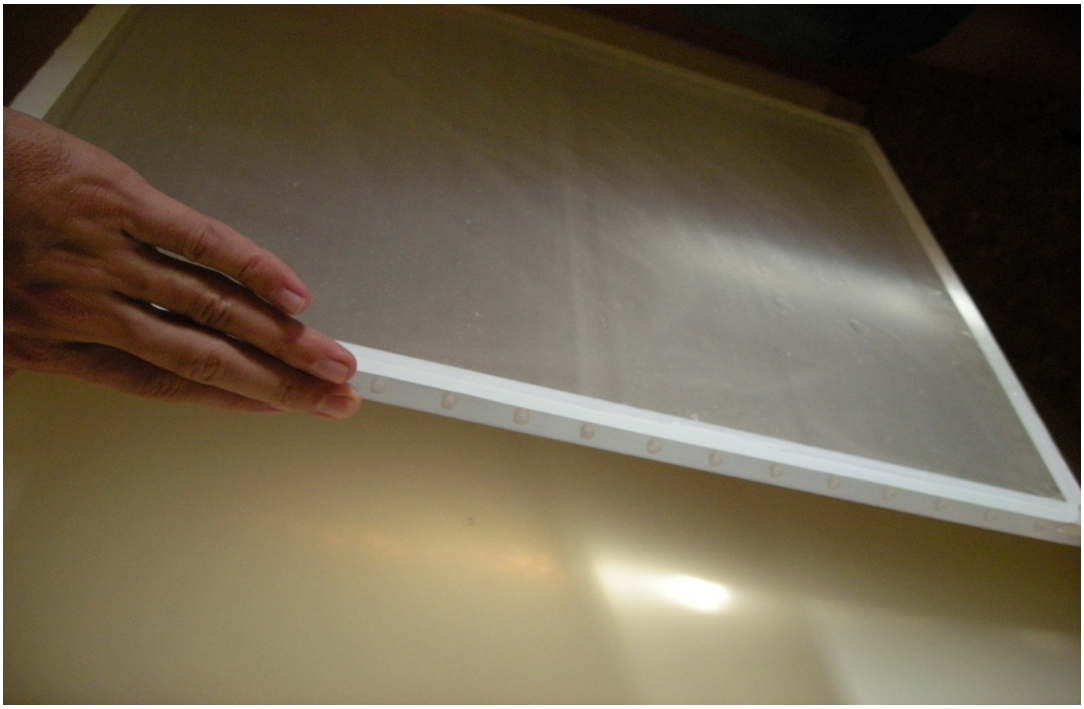

Figura 2: Capa Compatible sobre l'acrílic

A la posada en funcionament d'aquesta 1ra versió es va trobar que els blobs obtinguts, eren blobs fals o no havien blobs, després de fer l'anàlisi corresponent es va arribar a les següents conclusions:

- Que la pared dels forats on es van ficar els LEDs IR no era transparent, sortint així poca llum IR a l'acrílic impossibilitant el fenomen de la reflexió total interna.

- Que com les dimensions del prototipus son petites (48x36 cm), és complicat, si no es té algun mecanisme de fixació, aconseguir que tot el prototipus es quedi fixe.

10.1.2 Versió 1.1

- ✔ Es va procedir a tallar l'acrílic amb làser per tenir els costats transparents.
- ✔ Es va aprofitar les parts sobrants del tallat per fer una base per als LEDs IR (ens referim als forats de la primera versió).

Figura 3: Acrílic tallat amb làser i base pels LEDs

A la posada en funcionament de la 2na versió es va trobar que els blobs obtinguts, eren blobs fals o blobs febles, després de fer l'anàlisi corresponent es va arribar a les següents conclusions:

- Que no tota la llum emès pels LEDs IR incideix a l'acrílic, degut a que la base a on es van ficar no ho permetia.
- Que l'irregularitat del gruix de la capa compatible (1.5mm - 2mm), tenint en compta que es feia servir l'acrílic més gruix possible, era un factor d'error [1].
- Es manté el problema de la fixació del prototipus.

Arribat a aquest punt, es va decidir començar de 0, fent recerca de noves configuracions i tècniques per aconseguir el correcte funcionament de la tecnologia FTIR.

Degut a temes de pressupost es va mantenir el marc de fusta, la WebCam i el filtre IR. Llavors les possible millores només es podien realitzar a l'acrílic i en la forma de com fer arribar la major quantitat possible de llum IR a la superfície de l'acrílic.

La nova configuració a implementar es diferenciava de l'anterior en:

- Gruix de l'acrílic: ara 8mm, abans 15mm
- Posició dels LEDs respecte al costat de l'acrílic: a menys base, més incidència de la llum sobre l'acrílic.

Es va crear un mini-prototipus amb la nova configuració per comprovar que funcionava correctament, i ho feia.

Figura 4: Mini-prototipus

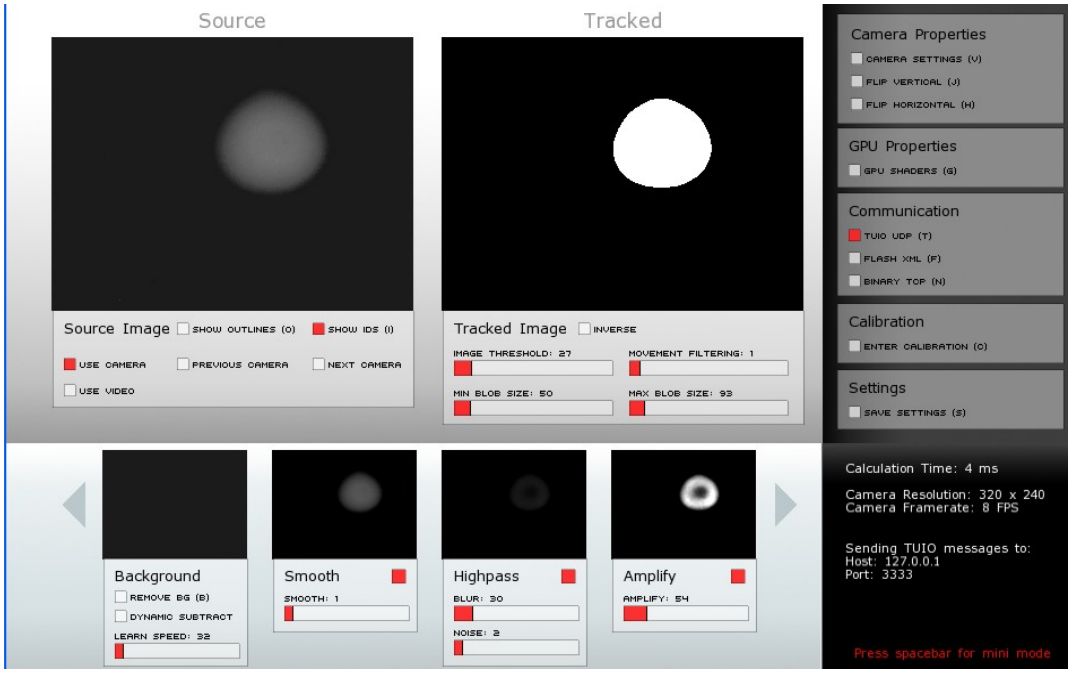

Figura 5: Blob clar al Mini-prototipus

Amb aquest resultat positiu es va tindre la seguretat del correcte funcionament de la WebCam i el Filtre IR. Es fa aquest comentari perquè a la comunitat de desenvolupament MT es recomana fer sevir com a WebCam la PS3 Eye [2] i un filtre industrial. Per motius de disponibilitat, en aquest projecte no es fan servir.

10.1.3 Versió 2.0

- ✔ Es va canviar l'acrílic de 15mm per un de 8mm.
- ✔ Es va canviar la posició dels LEDs (el cap del LED enganxats a l'acrílic)
- ✔ Es va canviar la tècnica de ficar la capa compatible, ara es va posar a sobre de la pantalla de projecció.

Figura 6: 'Cap' dels LEDs enganxats a l'acrílic

Figura 7: Capa Compatible sobre la Pantalla de Projecció

A la posada en funcionament de la 3ra versió es va trobar que es necessitava pressionar fort a la pantalla per a aconseguir mostrar els blobs i que aquests blobs eren blobs fals o blobs febles, després de fer moltes proves i revisió de la configuració FTIR es va arribar a les següents conclusions:

- Que encara es necessita estudiar com millorar la incidència de la llum amb l'acrílic, es deuria de fer servir fins a cert punt eines industrials per a la confecció del marc de la pantalla, així com per a la base del circuit de LEDs.

- Degut a la nostra confecció artesana del marc de la pantalla, l'acrílic es mou una mica, pertorbant el fenomen de reflexió interna total.

- Es considera que s'ha d'augmentar el gruix de la capa compatible.

- S'ha de millorar l'acoblament entre l'acrílic, la capa compatible i la pantalla de projecció ja que dona problemes per fer l'acció d'arrossegar.

- Es manté el problema de la fixació del prototipus.

10.1.4 Versió 2.1 (versió actual)

- ✔ Fixació del prototipus solucionat
- ✔ Gruix de la capa compatible correcte
- ✔ Se està treballant per mantenir la reflexió total interna sense pertorbació

Malgrat els problemes que impedeixen mostrar les funcionalitats de l'aplicació mitjançant el prototipus, és considera que la feina que se està realitzant ens portarà a obtenir la configuració vàlida per aquest prototipus. El camí es el correcte, s'ha de recorra fins arribar a la solució.

Amb la experiència obtinguda en els darrers mesos, la sensació de que aviat aconseguirem els resultats òptims és latent. Amb els peus al terra és considera que la feina feta donarà els seus fruits.

10.2 Viabilitat del Projecte

Segons l'anàlisi de la competència a Catalunya que s'ha realitzat en aquest projecte cap de les alternatives possibles no ha utilitza una interfície multitàctil que funcioni exactament com una pissarra tradicional. La dificultat recau principalment en el disseny de l'aplicació Software degut a la complexitat de la interpretació i concurrència d'esdeveniments que es produeixen.

És considera que per més costós que pugui resultar la introducció d'aquest dispositiu al sector de la ensenyança, els beneficis seran tals que no és parlarà del cost com alguna cosa negativa sinó com una inversió.

10.3 Metodologies de Treball

Metodologia aplicada durant el transcurs de la carrera, es podria dir que és part de l'ADN UPC. Sobre una bona base, es poden fer grans construccions.

L'estalvi de temps i organització que et dona el treballar amb una metodologia és important per mantenir la concentració i la claredat de a on es vol arribar.

És motivant ser part d'un projecte en el que és porta a terme tots el processos d'implementació per arribar al 'producte' final. Ha sigut gratificant treballar tant a l'aplicació i amb més profunditat amb el disseny i construcció del prototipus.

10.4 Línia de Treball a Seguir

10.4.1 Software

10.4.1.1 Extensibilitat

Component imprescindible en la realització d'un bon disseny. Tenint en compte això, en quan a les funcionalitats més “tangibles” s'hauria de tenir com a primer objectiu afegir les funcionalitats *obrir arxius d'àudio i vídeo*:

- S'hauria de poder obrir qualsevol format d'arxiu.
- S'hauria de prendre decisions importants de disseny en el que és la nostra barra d'eines.
- S'hauria de decidir com controlar la reproducció dels vídeos.

I això sense tenir en compte les modificacions que cal fer en el camp del reconeixement gestual, per poder realitzar moviments d'arrossegament i redimensió amb varis dits a la vegada.

10.4.2 Hardware

Fabricació d'un dispositiu que millori la sensibilitat a l'hora de detectar els blobs. Segons la quantitat d'experiments ja realitzats tenim dos vies per a fer-ho:

1. Estudiar la possibilitat de que el tipus de metacrilat no sigui l'adient per a transmetre la lluminositat pel seu interior, i en tal cas canviar-lo.
2. Estudiar la possibilitat de que el tipus de LED no sigui l'adequat (cosa que seria extranya d'acord amb l'estudi que s'ha realitzat abans de fer la seva compra).

11. Agraïments

A totes les persones que van a participar d'una o altra forma en el projecte, en especial a:

- ✔ Marta Fairen, per la seva guia.
- ✔ Centre de Realitat Virtual. Per donar-nos un lloc on treballar i facilitar-nos part del material que tenien disponible com el projector, el filtre, els ordinadors, etc.
- ✔ Edmundo Martínez Herrera (pare de Edmon Martínez Martí). Per ajudar-nos en el muntatge del prototipus.
- ✔ Edmon Martínez Martí. Company de projecte, gran amic i millor persona.

12. Referències Bibliogràfiques

12.1 Capítol 3. Anàlisi d'Antecedents i Alternatives

[1] Pissarra Digital Interactiva
http://blocs.xtec.cat/presentaciopdi/

[2] Pissarra Digital. Aspectos Generales
http://recursostic.educacion.es/observatorio/web/es/equipamiento-tecnologico/aulas_digitales/915-monografico-pizarrass_digitales-primera-parte?start=1

[3] Blog d'informació sobre pissarres digitals interractives i utilització de TIC en les aules
http://lapizarradigital.es

[4] Interactive Whiteboard
http://en.wikipedia.org/wiki/Interactive_whiteboard

[5] Reactable
http://es.wikipedia.org/wiki/Reactable

[6] Jeff Han
http://en.wikipedia.org/wiki/Jefferson_Han

[7] Perceptive Pixel
http://www.perceptivepixel.com

[8] Promethean
http://www.prometheanworld.com/

[9] Preus PDI de Promethean
http://www.peppm.org/Products/promethean/price.pdf

[10] Smart Technologies
http://smarttech.com/

[11] Preus SmartBoard
http://www.touchboards.com/smartboards/pricelist.asp

[12] ActivStudio
http://en.wikipedia.org/wiki/ActivStudio

[13] ActiveInspire
http://lastecnologiasenlaformacion.blogsport.com/2009_05_01_archive.html

[14] Comparativa entre ebeam i Mimio
http://gashed.net/?p=39

[15] Preus eBeam
http://www.adamsbiz.com/products/en/presentation/ebeam-educate.html

[16] Mimio
http://en.wikipedia.org/wiki/Mimio

[17] Massachusetts Institute of Technology
http://en.wikipedia.org/wiki/Massachusetts_Institute_of_Technology

[18] MIT $100K Enterpreunership Competition
http://en.wikipedia.org/wiki/MIT_$100K_Entrepreneurship_Competition

[19] Preus Mimio
http://www.schoolmart.com/mimio-interactiveteachingtechnologies.aspx

12.2 Capítol 4. Eines Utilitzades

[1] Protocol TUIO – Pàgina principal
http://www.tuio.org

[2] Protocol TUIO – API
http://www.tuio.org/?cpp

[3] Protocol OSC
http://en.wikipedia.org/wiki/Open_Sound_Control
[4] Sobre el llenguatge de programació:
http://nuigroup.com/forums/viewthread/2967/
C++
http://es.wikipedia.org/wiki/C%2B%2B
[5] Desenvolupament amb C/C++ en Eclipse
http://mundogeek.net/archivos/2007/01/17/desarrollo-con-cc-eclipse/
[6] Patrons de Disseny
http://es.wikipedia.org/wiki/Patrón_de_disseño
[7] Patró Singleton
http://es.wikipedia.org/wiki/Singleton
[8] Patro Observador
http://es.wikipedia.org/wiki/Observer_(patrón_de_diseño)
[9] Qt
http://es.wikipedia.org/wiki/Qt
[10] Signals & Slots
http://doc.qt.nokia.com/stable/signalsandslots.html
[11] The Event System
http://doc.qt.nokia.com/4.7-snapshot/eventsandfilters.html
[12] Grafiti
http://grafitiproject.worpress.com/
[13] AME Library Patterns
http://ame4.hc.asu.edu/amelia/patterns/
[14] Model Ocult de Markov
http://es.wikipedia.org/Modelo_oculto_de_Markov
[15] Sparsh-UI
http://code.google.com/p/sparsh-ui/

12.3 Capítol 5. Metodologia de Treball

[1] RUP Wikipedia
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
[2] SCRUM Wikipedia
http://es.wikipedia.org/wiki/Scrum
[3] Iteracio Wikipedia
http://es.wikipedia.org/wiki/Iteracion
[4] Rational Unified Proces
- Versió 7.0
- Copyright(C) IBM Corporation 1987, 2005.

[5] C++ Coding Standards
http://possibility.com/Cpp/CppCodingStandard.hml#descriptive
[6] PESBD
http://www.fib.upc.edu/fib/estudiar-enginyeria-informatica/enginyeries-pla-2003/assignatures/PESBD.html
[7] Guia del subversión de la FIB
http://www.fib.upc.edu/fib/serveis/guies/subversion.html

12.4 Capítol 6. Tecnologia Utilitzada

[1] Multi-Touch Surfaces: A Technical Guide
Technical Report TUM-I0833
Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, Ulrich von Zadow.

[2] SmartSkin: An Infrastructure for Freehand Manipulations on Interactive Surfaces.
http://www.sonycsl.co.jp/person/rekimoto/smartskin/

[3] Diamond Touch: Mitsubishi Electric Research Group.
http://www.merl.com/projects/DiamondTouch/

[4] Natural User Interface (NUI Group) Forums.
http://nuigroup.com/forums/viewthread/1982/

12.5 Capítol 7. Prototipus

[1] Reflexió Total Interna Òptica
http://en.wikipedia.org/wiki/Total_internal_reflection

[2] Rosco Grey
http://www.rosco.com/us/screens/roscoscreen.asp#Grey

[3] Circuit en Paral·lel
http://ca.wikipedia.org/wiki/Circuit_en_paral·lel

[4] Pantalla de projecció posterior (retroprojecció)
http://www.rosco-iberica.com/infotecnica/pantallas/technote_tbcs.asp

[5] Relació d'aspecte
http://es.wikipedia.org/wiki/Relaci%C3%B3n_de_aspecto

12.6 Capítol 8. Arquitectura del Software i Extensibilitat

[1] Model MVC
http://en.wikipedia.org/wiki/Model-View-Controller

[2] Scribble Example
http://doc.trolltech.com/4.5/widgets-scribble.html

[3] Patró observador:
http://es.wikipedia.org/wiki/Observer_(patrón_de_diseño)
http://en.wikipedia.org/wiki/Observer_pattern

[4] Nokia
http://es.wikipedia.org/wiki/Nokia

[5] Apunts de ES2
"Patró de disseny Controlador"

[6] Sparsh_UI
http://code.google.com/p/sparsh-ui

[7] Translació
http://es.wikipedia.orrg/wiki/Traslación_(física)

12.7 Capítol 10. Conclusions

[1] Tècniques per crear la Capa Compatible sobre l'acrílic.
http://www.youtube.com/watch?v=bnjeWbvVCCo
[2] WebCam PS3 Eye
http://es.playstation.com/ps3/peripherals/detail/item78899/PlayStation%C2%AEEye/

13. Annex. Plantilles Utilitzades

Plantilla pels Fitxers .h

```
/**  A one line description of the class.
 *
 * #include "XX.h" <BR>
 * -llib
 *
 * A longer description.
 *
 * @see something
 */

#ifndef XX_h
#define XX_h

// SYSTEM INCLUDES
//

// PROJECT INCLUDES
//

// LOCAL INCLUDES
//

// FORWARD REFERENCES
//


class XX
{
public:
// LIFECYCLE

   /** Default constructor.
    */
   XX(void);


   /** Copy constructor.
    *
    * @param from The value to copy to this object.
    */
   XX(const XX& from);


   /** Destructor.
    */
   ~XX(void);


// OPERATORS
```

```
   /** Assignment operator.
    *
    * @param from THe value to assign to this object.
    *
    * @return A reference to this object.
    */
   XX&                     operator=(const XX& from);

// OPERATIONS
// ACCESS
// INQUIRY

protected:
private:
};

// INLINE METHODS
//

// EXTERNAL REFERENCES
//

#endif  // _XX_h_
```

Plantilla pels Fitxers .cpp

```
#include "XX.h"  // class implemented


//////////////////////////////// PUBLIC ////////////////////////////////////////

//============================ LIFECYCLE ====================================

XX::XX()
{
}// XX

XX::XX(const XX&)
{
}// XX

XX::~XX()
{
}// ~XX


//============================ OPERATORS ====================================

XX&
XX::operator=(const XX&);
{
   return *this;

}// =

//============================ OPERATIONS ===================================
//============================ ACESS      ===================================
//============================ INQUIRY    ===================================
//////////////////////////////// PROTECTED  ///////////////////////////////////

//////////////////////////////// PRIVATE    ///////////////////////////////////
```