DIARIO DI BORDO – progetto DSB

1. 20.9-1.10

Incontro con Michele:

discussione preliminare sul progetto: realizzazione GUI che andrà a sostituire quella attualmente in uso per gli esperimenti del macchinario la quale risulta funzionale unicamente se l’utilizzatore è lo sviluppatore della stessa, C#, Visual Studio 2019 Enterprise.

Visti gli use cases. Discussione sulla scelta fra WPF, Winforms, o eventuali altre librerie grafiche.

Sottoscritto:

Lettura documentazione su C# (*Programmare con C#8 | Guida Completa*, Antonio Pelleriti, ISBN\_978-88-6895-769-8), realizzazione Hello World grafico con C# e WPF. Realizzazione primi mockup (in seguito risultati troppo graficamente indipendenti fra loro, nel senso che ogni mockup era una finestra separata dalle altre; ma poiché erano stati inizialmente preparati sulla base del singolo use case anziché accorpati in un’unica interfaccia grafica).

1. 1.10-8.10

Incontro:  
il prodotto dovrà gestire la parte offline, ossia visione dei dati salvati sul file system delle precedenti deposizioni, osservare gli attuali setup dei dispositivi associati a un dato esperimento/deposizione; e la parte online, ossia la creazione di un nuovo esperimento con i relativi dispositivi e setup associati.

Il setup dei dispositivi è delegato alle applicazioni dedicate ai dispositivi.

Possibilità di usare il database anziché il file system?

Completare la demo Hello World con una simulazione di errore (con message on top, principale inconveniente riscontrato da Michele per quanto concerne Winforms).

Sistemare i mockup.

Sottoscritto:

Implementazione Thread nell’Hello World: il messaggio appare on top in momento relativamente casuali. Sistemazione dei mockup. Continuazione della lettura della documentazione su C#.

1. 8.10-15.10

Incontro:  
decisione finale sull’uso della libreria grafica: WPF. In .NET 5 e.NET 6 è ancora presente, inoltre è stato valutato che l’applicazione rimarrà dedicata all’ambiente desktop; difficilmente avrà sviluppo in ambiente mobile.  


Fonte: <https://www.winfxitalia.com/articoli/dotnet6/introduzione-dotnet6-csharp10.aspx>

1. 15.10-22.10

Incontro:

1. - deep dive in wpf (documentazione e video) -> cosa va in XAML, cosa in controller, ecc
2. - completare un primo scheletro del mockup in WPF
3. - sistemare mockup:

* led status da togliere (i bottoni start e feeder on possono già indicare lo status),
* aggiungere GCODE e DSB software nella parte status,
* sugg: l'aggiunta di una tab (segno +) potrebbe copiarmi nella nuova tab un esperimento attualmente aperto in modo da eseguire dei confronti,
* un'immagine sola nei record e no ingrandimento,
* separare mockup nuovi e vecchi in due documenti

     4.- focus di coerenza: mostrare i bottoni di experiment o deposition quando essi sono selezionati, altrimenti no (mostrare le informazioni necessarie all'attuale focus).

!!Sono stati passati un esperimento con delle deposizioni per i test!!

Sottoscritto:  
 inizio implementazione dell’applicativo basato sul mockup.

1. 22.10-29.10  
     
   Incontro:

**Vacanze morti  
Mariano assente/limitato per il progetto: corso di formazione 1-6 novembre "full immersion"!**

Mockup  
Start --> automatic start (sniff start)  
Feeder On --> MANUAL start  
--> ambedue spostabili a seconda di dove sono comodi e smart!

Use case  
compress experiment (dapprima comprime tutte le deposizioni singolarmente e poi l'esperimento).

Editor immagini  
Michele: progetti ingegneria del software 1 --> editor di immagini  
Mariano: guardare il progetto di imagik di De Sanctis, **per ispirazione**  
Verificare: Open source wpf image viewer

Image viewer  
Setting come contrasto automatico, etc, o embedded o tramite xnviewer (pulsante, mouse destro, etc)  
Preview delle immagini embedded --> pulsante di ordine per immagini più potenti o altri filtri.  
Utile all'utente per verificare che il setup della camera sia corretto senza scorrere tutte le immagini  
  
Immagine di due camere diverse affiancate

Apertura setup device  
SensorTools v1.18  
IDS software suite (Melt Pool)  
Text CN (già in possesso Mariano)  
DSB software --> selezionare un file  
Smartekvision\_CameraSuite\_v3

Sottoscritto:

Cercati alcuni image viewer in WPF, ma non sono stati implementati

1. 29.10-12.11

Incontro:

Directory  
Possibilità di directory: aprire la directory dell'esperimento selezionato o aprire la directory degli esperimenti.

Tab esperimenti  
Non aprire tab identiche dallo stesso experiment item!!

Demo finale  
Immagini live (Smartek) e immagini catturate/registrate precedentemente (vedi quelle nelle deposizioni)

XAML  
Integrazione file xaml esterni (lavorarci su un esempio esistente e piccolo)

Ricevuto  
Da Michele: setup di smartek e nuova deposizione rar

Sottoscritto:  
corso G+S calcio bambini dal 1.11 al 6.11 durante il quale ho potuto mettere poco le mani sul progetto. Test tramite gli user control nell’Hello World: funzionante! Lo user control permette di separare fra loro diverse strutture WPF costruite in file xaml separati. È sufficiente istanziare il cs degli user control.

Settimana di infortunio dal 8.11. Sono riuscito ad aprire le tab per gli esperimenti e navigare fra le immagini della deposizione aperta dall’esperimento stesso.

Bottoni dell’apertura delle directory non ancora implementati. Progettati e scritti alcuni user control.

Cambio mockup e implementazione: si apriva una singola deposizione per ogni esperimento, anzi una deposizione sola in totale, quindi l’applicativo è stato ridisegnato affinché ogni tab degli esperimenti abbia

1. 12.11-19.11  
     
   Incontro:  
     
   Mariano

1-Diario sul percorso: ogni 2-4 h un commento da mettere in github o in un text file (x non perderselo)

2-Creare le issue per le cose da fare

3-Mostrare le immagini e slider (possibilità d'inserire un numero di millisecondo)

4-Trovare un modo di gestire visione di dettaglio e

globale

5-Visione di dettaglio, basata sul millisecondi, e

6-visualizza l'immagine, LaserOn, PowerFeedback e

temperatura

7-Visione globale: aprire un editor notepad++ da un

explorer di files del folder della deposition

Nota

* I protocolli del CN possono essere piu' di uno per deposizione
* Online: create experiment => DB dependencies
* visualizza l'immagine MP, CamEsterna, LaserOn, PowerFeedback e temperatura



Sottoscritto:

15.11 🡪 creazione di issues e milestones su github, creazione di un file Microsoft Projects con le deadlines.

Bug: errore nello spostamento dei TabItem degli esperimenti (nullPointerException) 🡪 depoTabControlController.setPath((string)tabItem.Header);

🡪risolto il 15.11:  
Fixed:

TabItem tabItem = (TabItem)mainTabControl.SelectedItem;

if(tabItem != null)

{

depoTabControlController.setPath((string)tabItem.Header);

}

16.11 🡪

file explorer trovato al link:

* <https://github.com/mikependon/RepoDB.Tutorials>
* <https://github.com/AngryCarrot789/FileExplorer> 🡪 … insomma
* <https://www.youtube.com/watch?v=aonRoEokQeY> 🡪 soluzione usata



1. 19.11-26.11  
   Incontro:  
     
   0) *Adattare il tuo file system al nuovo formato: pf inserire la foto in diario di bordo*

1*) Xn view ti dà il mio stesso errore, on next image crasha?*

2) Cliccando sull'immagine a destra nel file explorer, carica quella immagine a sinistra

3) *Close Tabs non fa nulla: ok => Mettilo fra parentesi (Close Tabs)*

*Fare cosi dappertutto e mantenerlo*

4) Collegare il commento dell'experiment al file dell'experiment, che c'è sempre, ev. vuoto

5) Usare i ms della prima e ultima immagine per estrarre i ms significativi e usare quelli per tutto il displaying dei ms (immagini e dati ecc)

6) Nei dati inserire: temperatura, LaserOn, PowerFeedback,

Visualizzare il dato relativo al ms esatto, oppure una media degli ultimi 3 valori piu' vicini

7) Poter inserire un numero di millisecondo e visualizzare i dati relativi

8) Header del file explorer fondato su web browser: per ora lasciarlo cosi

Problematica: è molto lento

9) Il file .mpp è bello ma lo integriamo completamente in github>https://github.com/MarianoFasano/DSB/milestones

Nota: Le milestones sono sotto le issues!

Abituiamoci ad avere tutto nel kanban: https://github.com/MarianoFasano/DSB/projects/1

Agganciare OGNI ISSUE ad una milestone, così abbiamo il controllo della rilevanza agganciato alla priorità (scadenza)

10) Settimana prox abbiamo il MS1

Sottoscritto:

21.11: adattato il file system, sia sul fisso sia sull’hard disk con i dati usati sul portatile.

XnView dà lo stesso problema che riscontra Michele.

Il titolo dei tasti non ancora implementati è stato messo fra parentesi.

Le immagini, così come i dati del pirometro e del CN, non sono consecutivi di ms in ms, spesso e volentieri saltano di qualche ms.

22.11: usare sempre lo stesso oggetto, senza istanziarne dei nuovi ogni volta, per myDepoData. Assegnare di volta in volta il valore del dictionary all’istanza già esistente sovrascrivendola (SSD riscrive tutto il blocco di memoria…)

Mostrare un messaggio quando i ms selezionati non coincidono con quelli dell’immagine? Meglio avere i dati che non collimano con i ms passati dell’utente o meglio non avere l’immagine (che salta subito all’occhio!).

23.11: implementare un binary search a seconda della lunghezza del valore passato al textbox!?

Rinominato initButtonsAction in initControlsAction in DepoTabControlController, poiché non si assegnano unicamente i bottoni ma anche textbox e labels (ciò che serve).

Search textbox 🡪 aggiunto metodo, non implementato, quando si preme enter this.searchMs.KeyDown += SearchMs\_KeyDown;

24.11: implementato un binary search per ms “piccoli” (ossia che vanno da 0 al valore massimo riportato dalla label) 🡪 durante la fase di refactoring valutare un pattern per il dispatcher!

Implementata anche la versione con i ms “lunghi”.