Technology Baseline

Informazioni Documento

Versione	1.0.0
Data approvazione	12 Gennaio 2018
Responsabile	Samuele Modena
Redattori	Marco Focchiatti, Giulio Rossetti, Kevin Silvestri, Manfredi Smaniotto, Cristiano Tessarolo
Verificatori	Matteo Rizzo, Samuele Modena
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Graphite
Uso	Esterno
Recapito	graphite.swe@gmail.com

Indice

Elenco delle figure

Elenco delle tabelle

1. Introduzione

1.1 Scopo del Documento

Il presente documento ha l'obiettivo di trattare in modo esaustivo l'esposizione e la motivazione delle tecnologie, dei framework e delle librerie selezionate per lo sviluppo del prodotto *DeSpeect*, nonché di dimostrarne l'adeguatezza e il grado di integrazione tramite *Proof of Concept* correlato agli obiettivi di progetto. Il documento analizza inoltre possibili criticità di sviluppo, progettuali o organizzative, proponendo delle soluzioni a supporto della bontà delle scelte tecnologiche intraprese.

1.2 Scopo del Prodotto

Lo scopo del *prodotto*$_{\text{G}}$ è quello di fornire un *interfaccia grafica*$_{\text{G}}$ utilizzabile come strumento di supporto all'utilizzo di *plugin*$_{\text{G}}$ sulla piattaforma Speect. L'utente avrà anche la possibilità di salvare i grafi generati a schermo dall'applicazione.
Il funzionamento dell'applicazione sarà garantito su un sistema *Linux Ubuntu*$_{\text{G}}$ versione 16.04 o superiore.

1.3 Ambiguità

Per evitare ogni tipo di incomprensione riguardo al linguaggio presente nei documenti viene fornito il *Glossario v1.0.0* contenente la definizione dei termini in corsivo marcati con una G al pedice.

1.4 Riferimenti

1.4.1 Normativi

- *Norme di Progetto v2.0.0*;
- Capitolato: http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C3.pdf

1.4.2 Informativi

- Presentazione capitolato d'appalto: http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C3.pdf
- Slide del corso "Ingegneria del Software" riguardanti le regole di progetto: http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/P01.pdf

2. Tecnologie selezionate

2.1 Introduzione

In questo capitolo vengono esposte e motivate le tecnologie, i framework e le librerie selezionate per lo sviluppo e la progettazione del prodotto *DeSpeect*.

2.2 Classificazione delle tecnologie

Le tecnologie di seguito esposte vengono catalogate secondo il seguente codice:

T[Categoria][ID]

Dove:

- **T**: indica che si tratta di una tecnologia;
- **Categoria**: indica la categoria a cui appartiene la tecnologia in esame. Può assumere i seguenti valori:
  - **S**: indica che la tecnologia è inerente lo *sviluppo* del prodotto;
  - **P**: indica che la tecnologia è inerente la *progettazione* del prodotto;
  - **O**: indica che la tecnologia è inerente l'*organizzazione* del gruppo.
- **ID**: rappresenta un codice numerico incrementale atto all'identificazione univoca della tecnologia.

Le tecnologie sono presentate secondo il seguente schema:

- Nome, versione e codice;
- Codice identificativo;

- Descrizione;
- Tecnologie concorrenziali;
- Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto:
  - Considerazioni generali;
  - Competenza del gruppo;
  - Usabilità;
  - Costo economico;
  - Consumo di risorse;
  - Sviluppi possibili della tecnologia.
- *Proof of Concept* (laddove lo si ritenga necessario);
- Aspetti negativi.

2.3 Tecnologie inerenti lo sviluppo e la progettazione del prodotto

2.3.1 Speect v1.1.0-69-g65f4 - TS01

2.3.1.1 Descrizione

Speect è un sistema di *Text To Speech* (TTS) multilingua. Esso offre un sistema TTS completo (analisi e decodifica del testo e sintesi vocale) con annesse varie API, nonché un ambiente per la ricerca e lo sviluppo di sistemi e voci TTS. Speect è scritto in linguaggio C, con una stretta conformità allo standard ISO / IEC 9899: 1990, consentendo così la massima portabilità su diverse piattaforme di calcolo. Le chiamate di sistema specifiche della piattaforma sono astratte per consentire le porte a nuove piattaforme. Speect rappresenta il cuore del prodotto che il gruppo intende sviluppare ed in particolare è l'applicazione per la quale si vuole realizzare un'interfaccia grafica atta a semplificarne l'utilizzo e il debug.

2.3.1.2 Tecnologie concorrenziali

La tecnologia in esame è vincolata dall'obbligo di utilizzo esplicitato dalla Proponente. Si riportano per completezza alcune tecnologie concorrenziali:

- **OpenMary**: sistema di *Text To Speech* (TTS) multilingua opensource scritto in Java e reperibile al seguente link:

  https://github.com/marytts

- **Idlak**: recente sistema di *Text To Speech* (TTS) multilingua opensource basato su tecnologie moderne e reperibile al seguente link:

  https://sourceforge.net/p/kaldi/code/HEAD/tree/sandbox/idlak/

2.3.1.3 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: Nonostante la tecnologia in esame sia vincolata dall'obbligo di utilizzo esplicitato dalla Proponente, da un confronto con sistemi concorrenti emerge il pregio di Speect di essere scritto in un linguaggio familiare alla totalità dei membri del gruppo. Speect è inoltre dichiaratamente conforme allo standard ISO / IEC 9899: 1990, che seppur datato ne assicura una soglia minima di qualità, estendibilità e portabilità. La tecnologia è dotata di una buona documentazione, che coadiuvata dal materiale e dal supporto fornito dalla Proponente permette di superare le difficoltà rappresentate dalla configurazione e dal linguaggio datato;

- **Competenza del gruppo e facilità di utilizzo**: Benché la tecnologia risulti ostica in termini di configurazione, la familiarità del linguaggio su cui si basa e la bontà della documentazione fornita ha permesso al gruppo di manipolarla con un certo agio;

- **Costo economico**: La tecnologia è opensource e reperibile gratuitamente al seguente link:

  http://speect.sourceforge.net/

- **Consumo di risorse**: Speect è compatibile solo con sistemi Linux e richiede l'utilizzo di CMAKE e di un compilatore conforme ad ANSI C/ISO C90. Ciò non rappresenta un problema in termini di risorse per quanto riguarda la dotazione tecnologica dei membri del gruppo, né un ostacolo per la corretta configurazione del software;

- **Supporto e sviluppi possibili della tecnologia**: Speect non gode di ampio supporto da parte dei suoi sviluppatori, tuttavia la Proponete si è offre al gruppo sostegno in caso di problemi o necessità di modifiche per la portabilità della libreria stessa. Non appare plausibile che la tecnologia subisca aggiornamenti o modifiche di rilievo nel futuro prossimo.

2.3.1.4 Proof of Concept

Lo studio della tecnologia si concretizza in un *Proof of Concept* che ne dimostra la compatibilità con gli obiettivi di prodotto evidenziando nel contempo lo stato delle competenze del gruppo in relazione alla stessa.

2.3.1.5 Aspetti negativi

Speect è scritto in C, linguaggio familiare al gruppo ma che pecca in leggibilità del codice e nell'implementazione completa del paradigma a oggetti. Ciò implica la necessità da parte del gruppo di lavorare in prima persona sulla portabilità della libreria verso il C++. Inoltre, Speect necessita di non banali procedure di configurazione per un corretto utilizzo. Per entrambi i problemi evidenziati la Proponente ha offerto al gruppo supporto attivo per giungere ad una soluzione soddisfacente in caso di necessità.

2.3.2 QT v5.9 LTS - TS02

2.3.2.1 Descrizione

QT è una libreria multipiattaforma per lo sviluppo di programmi con interfaccia grafica tramite l'uso di widget (congegni o elementi grafici). La libreria è scritta in C++ e gode di ampia diffusione e supporto. Il gruppo intende utilizzare questa tecnologia per lo sviluppo dell'interfaccia grafica del prodotto.

2.3.2.2 Tecnologie concorrenziali

La tecnologia in esame è parzialmente vincolata dalla richiesta della Proponente, che nel capitolato suggerisce una rosa di librerie in cui compare QT. Si è scelto di utilizzate QT in virtù della familiarità del gruppo con la stessa, della sua semplicità d'uso e del suo ampio utilizzo in ambito aziendale, ed in particolare la versione 5.9 LTS per la sua stabilità e garanzia di supporto. Seguono le tecnologie analoghe valutate ma ritenute meno performanti e conformi agli obiettivi di prodotto:

- **GTK+**: un toolkit multipiattaforma per la creazione di interfacce grafiche consigliato dalla Proponente, reperibile gratuitamente al seguente link:

  https://www.gtk.org/

- **wxWidgets**: una libreria multipiattaforma ed estendibile per la creazione di semplici interfacce grafiche, reperibile gratuitamente al seguente link:

  http://www.wxwidgets.org/

2.3.2.3 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: La tecnologia in esame è supportata da una peculiare semplicità d'uso, favorita tra l'altro da una vasta ed esplicativa documentazione. QT include inoltre QT Creator, software specificatamente pensato per il rapido sviluppo di interfacce grafiche in C++ e che gode già di una certa familiarità da parte del gruppo. Quest'ultimo inoltre è particolarmente robusto e ampiamente testato, come sinteticamente illustrato al link seguente:

  https://wiki.qt.io/Category:Tools::QtCreator::QualityAssurance

  Infine, QT offre librerie dedicate per lo sviluppo di test su più livelli;

- **Competenza del gruppo e facilità di utilizzo**: I componenti del gruppo possiedono conoscenze relativamente profonde del framework ed hanno già avuto modo di utilizzarlo per lo sviluppo di una basilare interfaccia grafica. Ciò, unito alla semplicità e ampia documentazione dello stesso, permettono una profonda manipolazione della tecnologia;

- **Costo economico**: La tecnologia è concessa secondo una licenza commerciale oppure open source (il gruppo predilige quest'ultima), ed è reperibile al seguente link;

  https://www.qt.io/

- **Consumo di risorse e compatibilità**: QT è compatibile con tutti i maggiori sistemi, ma il gruppo lo installerà su Ubuntu v16.04.3 LTS piattaforma su cui è richiesta la presenza di compilatore, debugger e

make per C++. QT non richiede particolari requisiti hardware ma necessità di una rilevante quantità di memoria per la sua istallazione, tuttavia ciò non rappresenta un problema per quanto riguarda la dotazione tecnologica del gruppo;

- **Supporto e sviluppi possibili della tecnologia**: QT gode di ampio supporto e viene costantemente aggiornato. La sceta di una versione LTS, tuttavia, garantisce il massimo supporto, stabilità e compatibilità attualmente disponibili.

2.3.2.4 Proof of Concept

Lo studio della tecnologia si concretizza in un *Proof of Concept* che ne dimostra la compatibilità con gli obiettivi di prodotto evidenziando nel contempo lo stato delle competenze del gruppo in relazione alla stessa.

2.3.2.5 Aspetti negativi

QT risulta meno performante di alcune tecnologie concorrenti e prevede la necessità di una notevole quantità di spazio per la sua installazione.

2.3.3 CMAKE v3.10.2 - TS03

2.3.3.1 Descrizione

CMake è una famiglia di strumenti open source e multipiattaforma progettati per creare, testare e pacchettizzare software. CMake viene utilizzato per controllare il processo di compilazione del software utilizzando semplici file di configurazione indipendenti dalla piattaforma e dal compilatore e generare makefile e aree di lavoro nativi che possono essere utilizzati nell'ambiente del compilatore di propria scelta. Il gruppo intende utilizzare questa tecnologia per l'automazione della compilazione del prodotto.

2.3.3.2 Tecnologie concorrenziali

La tecnologia in esame è parzialmente vincolata dalla richiesta della Proponente, che nel capitolato ne suggerisce l'utilizzo in quanto già usata da Speect e alcune sue dipendenze. Si riportano per completezza alcune tecnologie concorrenziali:

- **GNU Makefile**: strumento che controlla la generazione di file eseguibili e altri file non di origine di un programma dai file sorgente dello stesso, reperibile gratuitamente al seguente link:

https://www.gnu.org/software/make/manual/make.html

- **Qmake**: strumento che aiuta a semplificare il processo di creazione di progetti di sviluppo su piattaforme diverse. Esso automatizza la generazione di Makefile ed è utilizzabile per qualsiasi progetto software, indipendentemente dal fatto che sia scritto con Qt o meno. Si noti che *qmake* sarà parzialmente utilizzato dal gruppo contestualmente a CMAKE per l'integrazione di Speect e QT. Il software è reperibile al seguente link:

  http://doc.qt.io/qt-5/qmake-manual.html

2.3.3.3 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: CMAKE è una tecnologia necessaria alla corretta configurazione di Speect e si adatta bene all'integrazione tra quest'ultimo e QT, nonché allo sviluppo di automazioni tramite Travis. Pur inizialmente quasi completamente sconosciuta al gruppo, è ben documentata e supportata e ciò ne agevola sensibilmente l'apprendimento;

- **Competenza del gruppo e facilità di utilizzo**: La bontà della documentazione e il supporto della Proponente nel suo apprendimento (anche tramite seminario con relativo materiale pubblicato al link: https://github.com/giuliopaci/cmake-tutorial) ne ha notevolmente agevolato l'apprendimento, permettendo al gruppo di padroneggiarla entro i confini dello scopo del progetto;

- **Costo economico**: La tecnologia è open source e reperibile gratuitamente al seguente link:

  https://cmake.org/

- **Consumo di risorse**: CMAKE è una tecnologia multipiattaforma compatibile con i maggiori sistemi operativi e non presenta particolari requisiti harware;

- **Supporto e sviluppi possibili della tecnologia**: La tecnologia in esame è in costante aggiornamento e supportata da una community particolarmente attiva, tuttavia la scelta di adottarne una versione stabile offre le massime garanzie di supporto e stabilità attualmente disponibili.

2.3.3.4 Proof of Concept

Lo studio della tecnologia si concretizza in un *Proof of Concept* che ne dimostra la compatibilità con gli obiettivi di prodotto evidenziando nel contempo lo stato delle competenze del gruppo in relazione alla stessa.

2.3.3.5 Aspetti negativi

CMAKE è una tecnologia non standardizzata e caratterizzata da lacune in termini di:

- Build incremntale;
- Personalizzazione a build-time;
- Building parallelo;
- Organizzazione dei file prodotti.

2.3.4 Ubuntu v16.04.3 LTS - TS04

2.3.4.1 Descrizione

Ubuntu è un sistema operativo focalizzato sulla facilità di utilizzo. È prevalentemente composto da software libero proveniente dal ramo instabile di Debian GNU/Linux, ma contiene anche software proprietario, ed è distribuito liberamente con licenza GNU GPL. È orientato all'utilizzo sui computer desktop, ma presenta delle varianti per altri dispositivi, ponendo grande attenzione al supporto hardware. Il gruppo intende utilizzare Ubuntu come sistema operativo di riferimento per lo sviluppo del prodotto, offrendone garanzia di corretto funzionamento sullo stesso.

2.3.4.2 Tecnologie concorrenziali

La tecnologia in esame è parzialmente vincolata dalla richiesta della Proponente, che richiede garanzia di funzionamento del prodotto su questo specifico sistema operativo. Oltre a tale richiesta, il gruppo ha deciso di fare uso di questa tecnologia per la maggiore compatibilità e facilità di integrazione della stessa nei confronti degli altri strumenti selezionati. In particolare, si è optato per una versione LTS a garanzia di supporto e stabilità. Si riportano di seguito i sistemi operativi concorrenti reputati inadeguati al conseguimento degli obiettivi di prodotto:

- **Microsoft Windows**: sistema operativo commerciale reperibile al seguente link:

  https://www.microsoft.com/it-it/windows/get-windows-10

  Si noti che alcuni membri del gruppo faranno uso di tale sistema operativo per operazioni organizzative o comunque non strettamente correlate allo sviluppo del prodotto;

- **Apple MacOs**: sistema operativo commerciale reperibile al seguente link:

  https://support.apple.com/en-us/HT201475

2.3.4.3 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: Ubunto è un sistema operativo contraddistinto da una significativa semplicità di utilizzo affiancata da un buon supporto e un buon grado di libertà garantito all'utente. L'utilizzo dello stesso per quanto concerne lo sviluppo, sistema già familiare alla totalità dei componenti del gruppo, sarà parte della garanzia di piena compatibilità del prodotto con la piattaforma;

- **Competenza del gruppo e facilità di utilizzo**: Il sistema è molto familiare ai membri del gruppo e gode di per sè di una notevole semplicità d'uso;

- **Costo economico**: La tecnologia è open source e reperibile gratuitamente al seguente link:

  https://www.ubuntu-it.org/

- **Consumo di risorse e compatibilità**: Il sistema presenta modesti requisiti hardware (reperibili al seguente link: https://wiki.ubuntu-it.org/Installazione/RequisitiDiSistema) che non rappresentano un problema per la dotazione tecnologica del gruppo;

- **Supporto e sviluppi possibili della tecnologia**: La tecnologia in esame è in costante aggiornamento e supportata da una community particolarmente attiva, tuttavia la scelta di adottarne una versione LTS offre le massime garanzie di supporto e stabilità attualmente disponibili.

2.3.4.4 Aspetti negativi

Ubuntu non supporta (o non supporta completamente) alcuni software di utilizzo comune o selezionati dal gruppo per fini organizzativi. Ciò non rappresenta tuttavia un ostacolo significativo al suo utilizzo dato che ogni membro del gruppo disporrà parallelamente di un altro sistema operativo pronto a supplire ad eventuali mancanze di Ubuntu.

2.3.5 Travis CI - TS05

2.3.5.1 Descrizione

Travis CI è un servizio di integrazione continua distribuito utilizzato per costruire e testare progetti software ospitati su GitHub. I progetti open source possono essere testati gratuitamente attraverso travis-ci.org. Il gruppo intende utilizzare questa tecnologia per l'esecuzione di test automatici a seguito del caricamento di codice sulla repository relativa al progetto, così da garantirne la correttezza.

2.3.5.2 Tecnologie concorrenziali

La tecnologia in esame è stata scelta in virtù della sua diffusione, semplicità d'uso e perfetta integrazione con lo strumento di versionamento Github e la tecnologia CMAKE. Vengono di seguito proposte alcune tecnologie analoghe non reputate altrettanto valide:

- **Circleci**: Servizio commerciale di integrazione continua reperibile al seguente link:

  http://circleci.com/

- **Wercker**: Recente servizio di integrazione continua che propone un piano gratuito, reperibile al seguente link:

  http://www.wercker.com/

2.3.5.3 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: Travis CI permette di installare gratuitamente e semplicemente un sistema di integrazione continua per progetti open source. Il suo utilizzo, padroneggiato discretamente (nei confini degli obiettivi del progetto) dal gruppo, fornisce garanzia di codice corretto (o quantomeno testato) nella repository dedicata al progetto;

- **Competenza del gruppo e facilità di utilizzo**: L'ampia documentazione e facilità d'uso della tecnologia ha permesso di colmare in tempi brevi la mancata conoscenza del funzionamento dello strumento, nei confini degli obiettivi del progetto;
- **Costo economico**: Travis CI è disponibile gratuitamente per progetti open source e reperibile gratuitamente al seguente link:

  https://travis-ci.org/

- **Consumo di risorse e compatibilità**: Il servizio è pienamente compatibile con le tecnologie correlate selezionate e non presenta particolari requisiti per il suo funzionamento;
- **Supportoo e sviluppi possibili della tecnologia**: La tecnologia è in costante sviluppo e supportata da una buona documentazione e da una community attiva.

2.3.5.4 Proof of Concept

Lo studio della tecnologia si concretizza in un *Proof of Concept* che ne dimostra la compatibilità con gli obiettivi di prodotto evidenziando nel contempo lo stato delle competenze del gruppo in relazione alla stessa.

2.3.5.5 Aspetti negativi

Travis CI necessità di software di terze parti per personalizzazioni avanzate, tuttavia ciò non rappresenta un'impedimento in relazione agli scopi del gruppo.

2.4 Tecnologie inerenti l'organizzazione e la documentazione

2.4.1 Google Drive

2.4.1.1 Codice identificativo

2.4.1.2 Descrizione

2.4.1.3 Tecnologie concorrenziali

2.4.1.4 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: ;
- **Competenza del gruppo**: ;
- **Usabilità**: ;
- **Costo economico**: ;
- **Consumo di risorse**: ;
- **Sviluppi possibili della tecnologia**: .

2.4.1.5 Proof of Concept

2.4.1.6 Aspetti negativi

2.4.2 Hangout

2.4.2.1 Codice identificativo

2.4.2.2 Descrizione

2.4.2.3 Tecnologie concorrenziali

2.4.2.4 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: ;
- **Competenza del gruppo**: ;
- **Usabilità**: ;

- **Costo economico**: ;
- **Consumo di risorse**: ;
- **Sviluppi possibili della tecnologia**: .

2.4.2.5 Proof of Concept

2.4.2.6 Aspetti negativi

2.4.3 Wrike

2.4.3.1 Codice identificativo

2.4.3.2 Descrizione

2.4.3.3 Tecnologie concorrenziali

2.4.3.4 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: ;
- **Competenza del gruppo**: ;
- **Usabilità**: ;
- **Costo economico**: ;
- **Consumo di risorse**: ;
- **Sviluppi possibili della tecnologia**: .

2.4.3.5 Proof of Concept

2.4.3.6 Aspetti negativi

2.4.4 LaTex

2.4.4.1 Codice identificativo

2.4.4.2 Descrizione

2.4.4.3 Tecnologie concorrenziali

2.4.4.4 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: ;

- **Competenza del gruppo**: ;
- **Usabilità**: ;
- **Costo economico**: ;
- **Consumo di risorse**: ;
- **Sviluppi possibili della tecnologia**: .

2.4.4.5 Proof of Concept

2.4.4.6 Aspetti negativi

2.4.5 Git

2.4.5.1 Codice identificativo

2.4.5.2 Descrizione

2.4.5.3 Tecnologie concorrenziali

2.4.5.4 Dimostrazione di adeguatezza rispetto agli obiettivi di prodotto

- **Considerazioni generali**: ;
- **Competenza del gruppo**: ;
- **Usabilità**: ;
- **Costo economico**: ;
- **Consumo di risorse**: ;
- **Sviluppi possibili della tecnologia**: .

2.4.5.5 Proof of Concept

2.4.5.6 Aspetti negativi

2.5 Tabella riepilogativa delle tecnologie

3. Possibili criticità di sviluppo legate alle tecnologie

3.1 Classificazione delle criticità

Le criticità di seguito esposte vengono catalogate secondo il seguente codice:

C[Ambito][Priorità][ID]

Dove:

- **C**: indica che si tratta di una criticità;
- **Ambito**: indica l'ambito a cui appartiene la criticità in esame. Può assumere i seguenti valori:
  - **S**: indica che la criticità è inerente lo *sviluppo* del prodotto;
  - **P**: indica che la criticità è inerente la *progettazione* del prodotto;
  - **O**: indica che la criticità è inerente l'*organizzazione* del gruppo.
- **Priorità**: indica la priorità di risoluzione della criticità. Può assumere i seguenti valori:
  - **B**: priorità bassa;
  - **M**: priorità media;
  - **A**: priorità alta.
- **ID**: rappresenta un codice numerico incrementale atto all'identificazione univoca della criticità.

Le criticità sono presentate secondo il seguente schema:

- Nome;

- Codice;
- Descrizione;
- Tecnologia risolutiva proposta;
- Tecnologie alternative scartate;
- *Proof of Concept* (laddove lo si ritenesse necessario).

3.2 Dettaglio delle criticità

3.2.1 Compilazione in C++ di QT e Speect via CMAKE

3.2.1.1 Codice

3.2.1.2 Descrizione

3.2.1.3 Tecnologia risolutiva proposta

3.2.1.4 Tecnologie alternative scartate

3.2.1.5 Proof of Concept

3.2.2 Configurazione di Speect

3.2.2.1 Codice

3.2.2.2 Descrizione

3.2.2.3 Tecnologia risolutiva proposta

3.2.2.4 Tecnologie alternative scartate

3.2.2.5 Proof of Concept

3.2.3 Manipolazione della voice configurata

3.2.3.1 Codice

3.2.3.2 Descrizione

3.2.3.3 Tecnologia risolutiva proposta

3.2.3.4 Tecnologie alternative scartate

3.2.3.5 Proof of Concept

3.2.4 Disegno e manipolazione di elementi grafici attraverso il cursore

3.2.4.1 Codice

3.2.4.2 Descrizione

3.2.4.3 Tecnologia risolutiva proposta

3.2.4.4 Tecnologie alternative scartate

3.2.4.5 Proof of Concept

3.2.5 Efficienza delle operazioni di salvataggio e ripristino di uno stato di Speect

4. Conclusioni

4.1 Compatibilità tra le varie tecnologie

Le tecnologie proposte possono coesistere e con che grado di semplicità le collego?

4.2 Considerazioni finali sulle tecnologie

Considerazioni sui problemi risolti rispetto a quelli mancanti.

4.2.1 Considerazioni importanza tecnologie mancanti

Considerazioni sui punti non ancora risolti. Sono molto importanti? (Ovviamente quelli non risolti devono essere cavolate)