

## ***Robot Discovery - ddr***

*Elaborato di progetto per il corso Sviluppo di Sistemi Software*

<https://bitbucket.org/chiara-volonnino/iss-18-ddr>

Chiara Volonnino<sup>1</sup>

Email ids: <sup>1</sup>chiara.volonnino@studio.unibo.it

A.A. 2018/2019

# Indice

<b>1</b>	<b>Sprint 1</b>	<b>2</b>
1.1	Requisiti . . . . .	2
1.2	Analisi dei requisiti . . . . .	3
1.3	Analisi del problema . . . . .	4
<b>2</b>	<b>Test planning</b>	<b>6</b>

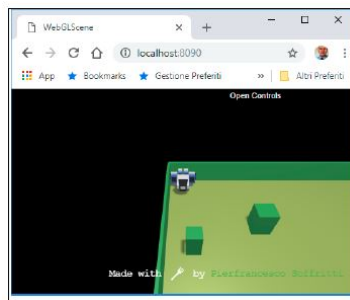
# Capitolo 1

## Sprint 1

### 1.1 Requisiti

Un *ddr* robot (discovery) viene usato per verificare l'esistenza di una bomba all'interno della hall di un aeroporto. La suddetta hall deve essere evacuata ma devono rimanere i bagagli sul pavimento. Il discovery robot è controllato in modo remoto da uno smart device pilotato da un operatore umano che opera in una zona protetta. Il robot può partire solo quando sono verificate due condizioni:

- l'operatore umano invia un comando di Explore ([R-startExplore](#)) tramite l'ausilio di una interfaccia grafica funzionante sullo smart device;
- il valore della temperatura della hall non è più alta di un valore prefissato ([R-tempOk](#)).



Il sistema software che funziona sul robot e sul device dell'operatore deve fornire le suddette funzionalità:

1. il robot deve poter esplorare la hall in modo autonomo ([R-explore](#)) con l'obiettivo di raggiungere ogni borsa presente sul pavimento;
2. durante l'esplorazione l'operatore può fermare ([R-stopExplore](#)) il robot e inviare comandi per farlo tornare nella sua posizione iniziale ([R-backHome](#)) o per fargli continuare la fase esplorativa ([R-continueExplore](#));
3. se il robot scopre qualcosa durante l'esplorazione deve far lampeggiare un led ([R-blinkLed](#)) su di lui inserito e inviare/aggiornare i dati riguardanti lo stato della scoperta e lo stato del robot sulla console che possiede l'operatore umano;
4. quando è in prossimità di una borsa il robot deve:
  - (a) stopparsi ([R-stopAtBag](#));
  - (b) scattare una foto ([R-takePhoto](#)) della borsa;
  - (c) inviare la foto al device dell'operatore ([R-sendPhoto](#)).

5. quando il device dell'operatore riceve la foto esegue un tool per controllare e quindi capire se la borsa contiene effettivamente una bomba. In caso di esito positivo (quindi in caso che la borsa contenga una bomba):
  - (a) il device deve notificare la scoperta all'operatore (**R-alert**);
  - (b) memorizzare la foto (**R-storePhoto**) su un dispositivo di memorizzazione permanente insieme alle informazioni contestuali (es. data e ora, posizione, etc.);
  - (c) inviare al robot il comando di tornare alla posizione iniziale (**R-backHomeSinceBomb**).
- In caso contrario (quindi non si è riscontrata una bomba in quella borsa) l'operatore invia al robot il comando per continuare l'esplorazione (**R-continueExploreAfterPhoto**);
6. l'operatore quando viene allertato per possibile bomba:
  - (a) deve aspettare che il robot sia tornato nella sua posizione iniziale (**R-waitForHome**);
  - (b) quando il robot è tornato deve notificare ad un altro robot (dotato di strumenti adeguati) di raggiungere la borsa sospetta (**R-reachBag**) per mettere la borsa in un posto sicuro e trasportarla fino alla posizione iniziale (**R-bagHome**).

## Obiettivo di business

Implementare un sistema che permetta ad un utente umano di controllare un robot remoto capace, fra le altre, di esplorare l'ambiente in cerca di bombe e che sia capace allo stesso tempo di tracciare dinamicamente una mappa di "navigazione" dell'ambiente a lui circostante.

## 1.2 Analisi dei requisiti

Dopo una attenta analisi e un'accurata discussione con il committente si sono delineate le seguenti osservazioni.

Il sistema software, dopo una prima revisione, risulta essere un sistema distribuito composto da due entità principali: il **robot** e la **console**. Nello specifico:

- il robot è un'entità remota (reale o virtuale) autonoma e reattiva che riceve comandi da un utente umano attraverso la console;
- la console ha il compito di catturare i comandi attuati dall'operatore umano e inviare questi comandi al robot remoto. Questa è un'entità autonoma che può essere in esecuzione su un qualsiasi dispositivo Smart-Divice o su un Personal Computer.

Il robot per poter iniziare l'esplorazione deve verificare che alcune condizioni ambientali siano favorevoli alla sua inizializzazione. Nello specifico, il robot deve restare in ascolto per ricevere dall'utente il comando di start exploration e deve verificare che la temperatura nella hall non superi un limite prefissato. Per controllare la temperatura, si è deciso con il committente di escludere la possibilità di avere un qualsivoglia dispositivo di controllo temperatura sulla console, in quanto rischierebbe di non essere uniforme con la reale temperatura della hall in cui lavorerà il robot, essendo la postazione remota. Inoltre tale punto può essere interpretato come servizio aggiuntivo che offre il nostro sistema.

Il robot è dotato di un sonar, posto sul suo fronte, per permettergli di essere sensibile all'ambiente. Questa tecnologia quindi rimane indispensabile per il suo funzionamento; altre tecnologie sarebbero sicuramente di supporto, ma per il momento, dati i costi, si è deciso con il committente di escluderle. Inoltre il robot reale deve includere nel suo equipaggiamento dei led, indispensabili per notificare all'utente umano che ha scoperto qualcosa e un dispositivo che gli permetta di catturare una foto delle borse, in modo da inviarla alla console e permettere un'analisi in grado di intendere se tale borsa contiene o meno una bomba.

Il software sulla console deve permettere all'utente finale di interagire con il robot in modo chiaro e semplice e quindi deve includere dei comandi che permettano tale scopo (es. Start, Stop, BackHome,

ContinueExploration). A tal fine il software sul robot deve interpretare i comandi inviati dall'utente tramite la console e inviarli al robot che deve poterli attuare in modo completamente autonomo.

Come già preannunciato il robot deve essere sensibile all'ambiente, con particolare riferimento alla temperatura e ai possibili ostacoli presenti sul piano della hall. In questo contesto quindi faremo riferimento a tali condizioni come `enviromentCondition`. Nello specifico delineiamo che: gli ostacoli posti nella hall devono essere rilevati dinamicamente dal robot tramite l'ausilio dei sonar; mentre, per quanto concerne la temperatura, deve essere acquisita in qualche modo.

Il robot deve avere la capacità di costruire in modo dinamico una mappa del territorio, capace di mostrare anche gli ostacoli incontrati. Quindi risulta fondamentale una fase di *planning* dell'ambiente circostante al robot.

Il device console deve inoltre includere un tool che permetta di controllare in qualche modo, per ogni foto ricevuta dal robot, se la borsa contiene o meno una bomba.

Inoltre, se il tool di ricerca bombe produce un esito positivo, il device deve includere un qualche sistema di alerting per notificare all'operatore finale la scoperta, e proseguire con il salvataggio della foto e delle relative informazioni (es. data e ora, posizione, etc) su un qualche dispositivo di memorizzazione permanente. Infine, deve notificare al robot di tornare alla posizione di partenza (*home*).

Infine, a fronte di una scoperta, il robot deve ricevere il comando di ritornare nella sua posizione iniziale e, una volta giunto, deve avere la capacità di notificare in qualche modo, ad un altro robot la posizione della borsa.

Il secondo robot, deve poter raggiungere con facilità la borsa contenente la bomba, quindi deve conoscere il piano di navigazione creato dal primo robot. Una volta raggiunta tale bomba, deve, in qualche modo raccoglierla e portarla in un posto sicuro per disinnescarla.

### 1.3 Analisi del problema

Dopo aver analizzato e discusso con il committente i requisiti da lui posti, procediamo con l'analisi del problema.

In tale accezione, ci accorgiamo che la logica di business del sistema è principalmente legata al software che si occupa di implementare il comportamento del robot. Diamo così il nome **Robot Discovery** a tale parte che deve essere in grado di creare un robot autonomo capace di perlustrare una stanza e che sia reattivo agli ostacoli e agli eventi.

Risulta inoltre utile dividere il concetto di *discovery robot* in due parti:

- *Robot Mind*, che avrà il compito di implementare la strategia necessaria a risolvere l'obiettivo di business (implementerà la business logic del sistema software);
- *Robot Actuator*, che eseguirà i comandi per muovere il robot (attività passiva di semplice attuatore).

Quindi il robot-mind avrà il compito di monitorare l'ambiente e gli eventi, oppure si può optare di delegare tali compiti ad una terza entità, chiamata *Word-Observer*.

Da subito rimane indispensabile definire un piano di *Planning*, utile per permettere al robot di: valutare in modo corretto tutto il territorio che lo circonda, indispensabile per catturare e implementare la business core di tale sistema, l'esplorazione; e definire anche i punti in cui sono presenti degli ostacoli. Questa mappa del territorio inoltre, risulta indispensabile per permettere al secondo robot di raggiungere in completa autonomia e semplicità la borsa contenente la bomba e portarla al sicuro.

Quindi d'ora in poi definiremo:

- *Explorer-Robot* il primo robot, ovvero colui che ha il compito di esplorare la hall e valutare le diverse borse in cerca della bomba;
- *Defusing-Robot* il secondo compito, ovvero colui che ha il compito di andare a prelevare la borsa che ha lanciato l'allarme, seguendo il percorso indicatogli sulla mappa dal explorer-robot.

Definire la mappa del territorio in questi termini ci permette di definire che il sistema deve mantenere informazioni, in quanto tale mappa deve essere trasferita in qualche modo al defusing-robot.

Quando parte il defusing-robot dobbiamo assumere che l'ambiente non sia cambiato.