

# Libreria LEGO | Diario di lavoro - 18.01.2019

Gabriele Alessi, Giulio Bosco

Canobbio, 18.01.2019

## Lavori svolti

Durante questa giornata abbiamo continuato con l'implementazione e la documentazione. Oggi era pianificato di terminare il capitolo, ma ci serviranno ancora alcuni giorni per perfezionare il tutto e procedere ai test. Nella documentazione stanno venendo descritti nel dettaglio tutte le classi con i relativi metodi.

Ad esempio oggi è stato implementato il LineFollower proporzionale e questa è la base del metodo principale:

```
public void run() {
    waitLightSensor.setValue((byte) 50);
    waitLightSensor.setBigger(this.isLineOnLeft());

    this.startNavigation();
    while (!isAlive()) {
        waitLightSensor.waiter();

        if (this.isLineOnLeft()) {
            this.right(TURNING);
        } else {
            this.left(TURNING);
        }

        waitLightSensor.setBigger(!waitLightSensor.isBigger());
        waitLightSensor.waiter();

        if (this.isLineOnLeft()) {
            this.left(TURNING);
        } else {
            this.right(TURNING);
        }
        waitLightSensor.setBigger(!waitLightSensor.isBigger());
    }
}
```

Dopo una rapida revisione generale, è stato creato un file in cui sono elencate le cose che dovranno venire perfezionate:

- Javadoc
- versions + authors
- use a default layout:
  - Constants
  - Fields
  - Getters
  - Setters
  - Constructors
  - Help Methods (private methods)
  - General Methods
- Rename some fields and methods

Questa è la documentazione della classe Wait, cioè la classe principale dei metodi di attesa rappresentati dal blocchetto arancione in Mindstorms:

Classe `Thread` usata per generalizzare tutte le classi `Wait`. Queste classi sono rappresentate tramite il blocco arancione in LEGO® Mindstorms e servono per aspettare che succeda qualcosa prima di continuare la sequenza di azioni (ad esempio aspettare del tempo o un sensore che legga un certo valore o che cambi valore).

```

```

- `WAIT_TIME`: Costante che rappresenta il tempo (in millisecondi) da aspettare.
- `finished`: Attributo che indica se l'attesa è finita.
- `isFinished()`: Metodo utile per ottenere lo stato dell'attesa (finito/non finito).
- `setFinished()`: Metodo che serve per impostare lo stato dell'attesa.
- Costruttore: Istanza una nuova attesa.
- `beginWait()`: Imposta il valore dello stato dell'attesa a `true`.
- `waiter()`: Metodo principale utile per iniziare l'attesa.

Orario	Lavoro svolto
13:15 - 16:30	Implementazione e documentazione

## Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

## Punto della situazione rispetto alla pianificazione

Leggermente in ritardo con la pianificazione riguardo l'implementazione.

## Programma di massima per la prossima giornata di lavoro

Messa a punto implementazione (Javadoc e revisione), documentazione e, se possibile, test.