

# Mondo Robot

**Progetto scelto:** MONDO ROBOT

**Studenti partecipanti al progetto:**

Palmieri Matteo, matricola: 20038775

Magi Davin, matricola: 20039230

## DESCRIZIONE CLASSI:

**Package:** mondo\_robot.Model

- Classe **Casa:**  
Classe in cui è presente l'intera struttura dati del gioco "**MondoRobot**"
- Interfaccia **Moveable:**  
Interfaccia che permette alle varie caselle di muoversi nella Casa.
- Interfaccia **Statable:**  
Interfaccia per implementare le funzioni utili alla gestione dello stato.
- Classe **Casella:**  
È la superclasse di tutte le celle, sono presenti le proprietà generali dei vari elementi/celle.
- Classe **CasellaMobile:**  
Sottoclasse di Casella e superclasse di caselle mobili come Drone e Animale.  
Assegna le coordinate alle caselle interessate.
- Classe **CasellaStato:**  
Sottoclasse di Casella e superclasse di caselle mobili come Lavatrice, Fornello e Rubinetto.  
Permette di impostare le coordinate della casella e di valutarne lo stato (riparato/rotto) e di effettuare riparazioni.
- Enum **CasellaTipo:**  
Elenco che contiene tutti i tipi di caselle esistenti nel gioco.
- Enum **ColoreNodo:**  
Elenco riguardante lo stato di ricerca BFS del metodo "allagaPavimenti" nella classe Thread "RompiElementiThread".
- Enum **Direzioni:**  
Enumerativo che indica tutte le possibili direzioni del Drone.
- Classe **Animale:**

Classe che estende **"CasellaMobile"** e implementa l'interfaccia **"Statable"** che rappresenta le proprietà dell'animale, tra cui le coordinate, l'immagine e lo stato del pavimento quando ci passa sopra (per cambiarne lo stato).

- Classe **GameMode:**

Enumerativo che indica se il gioco si trova in modalità **"debug"** (all'avvio del gioco si vedrà direttamente la mappa completa) o in **"game"** (modalità giocatore normale).

- Classe **Fornello:**

Classe che estende **"CasellaStato"** e indica le caratteristiche del fornello.

- Classe **Fornello:**

Classe che estende **"CasellaStato"** e indica le caratteristiche del fornello.

- Classe **Lavatrice:**

Classe che estende **"CasellaStato"** e indica le caratteristiche della lavatrice.

- Classe **Rubinetto:**

Classe che estende **"CasellaStato"** e indica le caratteristiche del rubinetto.

- Classe **Muro:**

Classe che estende **"Casella"** dove sono presenti le proprietà del muro (immagine, coordinate).

- Classe **Pavimento:**

Classe che estende **"CasellaStato"** e implementa l'interfaccia **"Moveable"** e indica le caratteristiche del pavimento.

- Classe **Nodo:**

Classe wrapper utilizzata per la ricerca BFS del primo pavimento asciutto da bagnare, contiene le proprietà del nodo.

- Enum **Svolta:**

Enumerativo che indica le azioni eseguibili dal drone per cambiare direzione (svoltare)

## Package: **mondo\_robot.View**

- Classe **Frame\_Game:**

Classe della View che crea la mappa di gioco.

- Classe **Frame\_Menu:**

Classe che contiene le caratteristiche e proprietà del menù principale.

- Classe ***MondoRobot\_Frame:***

Classe astratta, al suo interno sono contenute le proprietà in comune dei frame.

- Classe ***Panel\_Game:***

Classe che contiene le immagini della mappa di gioco.

#### Package: **mondo\_robot.Controller:**

- Classe ***Controller\_Game:***

Classe del controller, implementa l'*actionListener* e il *KeyListener*.

Il game controller ha il compito di lanciare e che gestisce la logica del gioco "***MondoRobot***".

- Classe ***Controller\_Menu:***

Gestisce gli input dell'utente del menù principale presente in "***Frame\_Menu***".

#### Package: **mondo\_robot**

- Classe ***Main:***

Il *Main* crea il controller che a sua volta lancia il gioco.

## FUNZIONALITÀ PROGRAMMA

All'avvio dell'applicazione viene generata una GUI che contiene:

- Un pulsante "***Aiuto***" che, se cliccato, crea un file di testo con al suo interno la guida sul gioco MondoRobot.
- Un pulsante "***Nuova Partita***" che fa apparire un'altra schermata per scegliere la dimensione della mappa desiderata, una volta scelta verrà formata la mappa pronta per incominciare a giocare.
- Un pulsante "***Carica Partita***" che permette di scegliere, seguendo un determinato path, un file di testo con all'interno la mappa generata in una partita precedente per continuare a giocare.
- Un'opzione "***Modalità sviluppatore***" che, se selezionata, quando verrà caricata o generata una nuova partita verrà mostrata non solo la mappa di gioco iniziale ma anche un'altra mappa senza foschia in modo da poter controllare la posizione di ogni elemento (animali e oggetti da riparare). Quest'ultima mappa è possibile usarla in gioco anche premendo il tasto "***V***".
- Durante una partita è possibile, tramite le opzioni in alto a sinistra della schermata di gioco, selezionare le opzioni:
  - **File** → Per salvare la partita o per uscire dal gioco.
  - **Aiuto** → Per accedere alla guida del gioco o per aprire la schermata con i comandi utilizzabili.

Una volta avviato il gioco sarà possibile muoversi con il Robot, che potrà: interrogare le celle, chiudere l'acqua e spegnere i fornelli.

Ogni volta che verrà eseguita un'azione o si cercherà di eseguire un'azione non consentita, verrà aperta una finestra con il messaggio opportuno, solo alla chiusura della finestra si potrà continuare il gioco.

**TABELLA FUNZIONALITÀ**

<b>Test delle classi del modello</b>	Fatto
<b>Javadoc per componenti pubblici del progetto</b>	Fatto
<b>Il modello non fa alcuna letture/scrittura da console o altra interazione con l'utente</b>	Fatto

È stata usata l'ereditarietà ovunque appropriato	Fatto
È stato riutilizzato il codice ovunque appropriato	Fatto
Sono state utilizzate visibilità appropriate	Fatto
Sono state lanciate eccezioni ove appropriato	Fatto
Specificare dove/come è stata fatta l'estensione di una classe o classe astratta	<ul style="list-style-type: none"> <li>• java.lang.<b>Object</b> <ul style="list-style-type: none"> <li>○ mondo_robot.Model.<b>Casella</b> (classe astratta) <ul style="list-style-type: none"> <li>▪ mondo_robot.Model.<b>CasellaMobile</b> <ul style="list-style-type: none"> <li>• mondo_robot.Model.<b>Animale</b></li> <li>• mondo_robot.Model.<b>Drone</b></li> </ul> </li> <li>▪ mondo_robot.Model.<b>CasellaStato</b> <ul style="list-style-type: none"> <li>• mondo_robot.Model.<b>Fornello</b></li> <li>• mondo_robot.Model.<b>Lavatrice</b></li> <li>• mondo_robot.Model.<b>Rubinetto</b></li> <li>• mondo_robot.Model.<b>Pavimento</b></li> </ul> </li> <li>▪ mondo_robot.Model.<b>Muro</b></li> </ul> </li> <li>○ mondo_robot.Model.<b>Nodo</b></li> <li>○ mondo_robot.Model.<b>Casa</b></li> </ul> </li> </ul>
Specificare dove/come è stato usato il lancio/gestione delle eccezioni	<p>Le eccezioni principali sono:</p> <ul style="list-style-type: none"> <li>• Nella classe <b>Controller_game</b> viene lanciata: <b>FileNotFoundException</b>, se non viene trovato il file e <b>IOException</b>.</li> </ul> <p>Nel caso in cui non venga trovato il file, viene effettuata la creazione di quest'ultimo</p> <p>Un'altra eccezione principale è stata posta sul thread in attesa per la rottura degli elementi.</p> <p>In questo caso verrebbe lanciata l'eccezione: <b>InterruptedException</b> se non è stato possibile completare la sleep.</p>
Specificare dove è stata usata una classe annidata	<p>Nella classe <b>Casa</b>, è stata implementata una classe annidata <b>RompitElementiThread</b>, la quale permette la gestione della rottura degli elementi tramite sleep di thread.</p>