

RESPONSABILITA

CONTROLLER:

Controller: Gestisce le relazioni tra gioco ed interfaccia grafica o interfaccia prompt, operando sul model dagli input dell' interfaccia

VIEW:

JavaFxView: Gestisce tutta il disegno del gioco e dei suoi elementi

PromptView: Gestisce tutti i messaggi di output ad ogni aggiornamento del gioco

MODEL:

RacetrackGame: Gestisce il gioco, ovvero l'interazione tra il circuito e i giocatori che ci giocano sopra

RacetrackCircuit: Gestisce il circuito, costituito da più checkpoint

CircuitCheckPoint: Gestisce un checkpoint, che può essere di diversi tipi e ha differenti funzionalità

CircuitLocation: Gestisce una location sulla mappa formata da un x e un y

GamePlayer: Gestisce un giocatore del gioco, e i suoi metodi per spostarsi e cambiare il suo stato

RaceCar: Gestisce la macchina che appartiene ad ogni Bot

Vector: Gestisce un vettore formato da due CircuitLocation la testa e la coda del vettore, serve per muoversi

JsonConfigChecker: controlla che il file di configurazione sia adeguato rispettando degli standard

JsonGameLoader: carica e (salva) il gioco da un file json

PlayerRuleChecker: controlla ad ogni turno le regole del gioco e le applica al player

PromptInformation: classe di support che contiene le string da usare per printare i risultati dalle interfacce

VectorEngine: gestisce la creazione di vettori, calcolando il trasposto e il vettore intelligente di un bot

App: main class