UNIVERSITÀ DEGLI STUDI DI NAPOLI “PARTHENOPE”

FACOLTÀ DI SCIENZE E TECNOLOGIE

CORSO DI LAUREA IN INFORMATICA



RELAZIONE PROGETTO PROGRAMMAZIONE iii

Gioco Auto

DOCENTE CANDIDATO

Angelo Ciaramella Salvatore Capobianco

Anno Accademico 2020-2021

Indice

[Requisiti 2](#_Toc64474162)

[Diagramma UML delle classi 3](#_Toc64474163)

[Esecuzione 4](#_Toc64474164)

[Codice 8](#_Toc64474165)

[*INTERFACCIA STRATEGY* 8](#_Toc64474166)

[*METODO CHECK* 11](#_Toc64474167)

[*CLASSE CRASHMANAGER* 12](#_Toc64474168)

[*METODO CREAOSTACOLO* 13](#_Toc64474169)

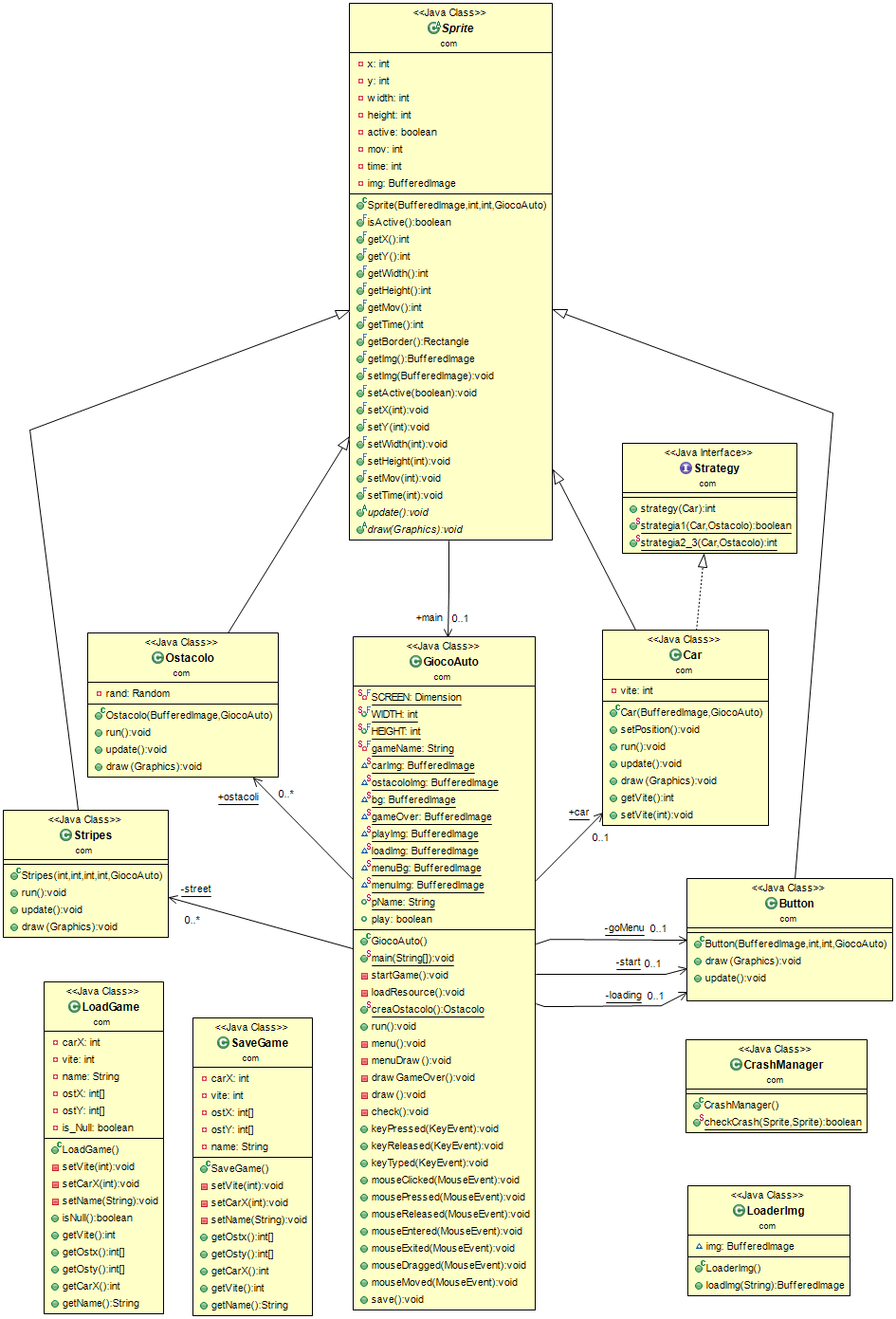
# Requisiti

Si vuole realizzare un mini-gioco autonomo, il cui protagonista è un’auto che si muove lungo un percorso. Lungo di esso sono disposti casualmente e automaticamente degli ostacoli, che l’auto tenterà di evitare mettendo in pratica tre strategie di movimento. Ogni qualvolta l’auto scontra un ostacolo, cambia la strategia.

La partita termina quando l’auto “esaurisce le vite” scontrando tre ostacoli.

In qualsiasi momento è possibile mettere in pausa il gioco, salvare il suo stato e riprendere la partita successivamente, dal punto in cui era stata interrotta.

# Diagramma UML delle classi



# Esecuzione

All’avvio del gioco, viene visualizzato il menu principale che permette il caricamento di una partita precedente o l’inizio di una nuova:

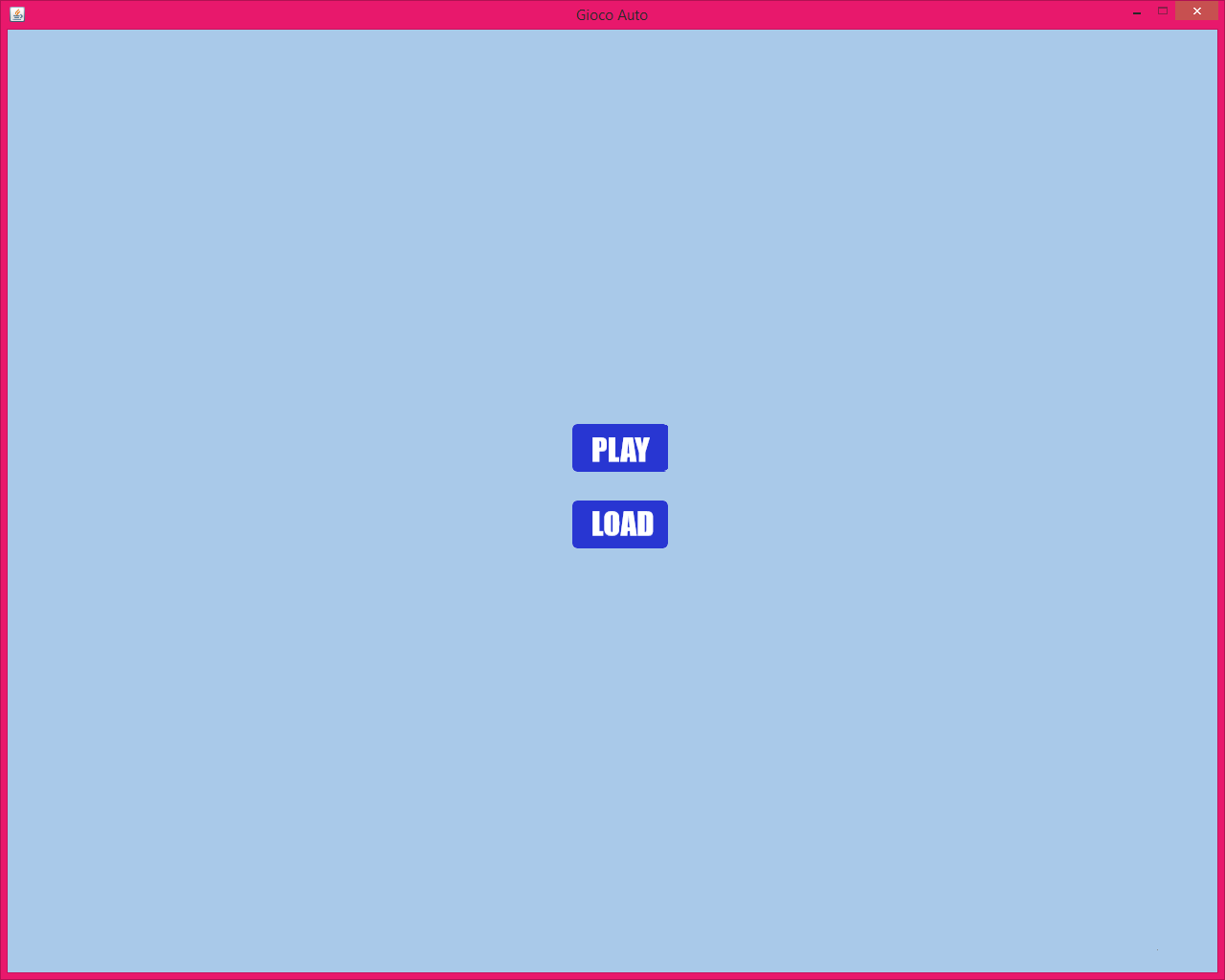


Figura 1: Menu iniziale

Cliccando sul bottone “PLAY”, verrà richiesto il nome del giocatore (“PLAYER 1” di default se non viene specificato):

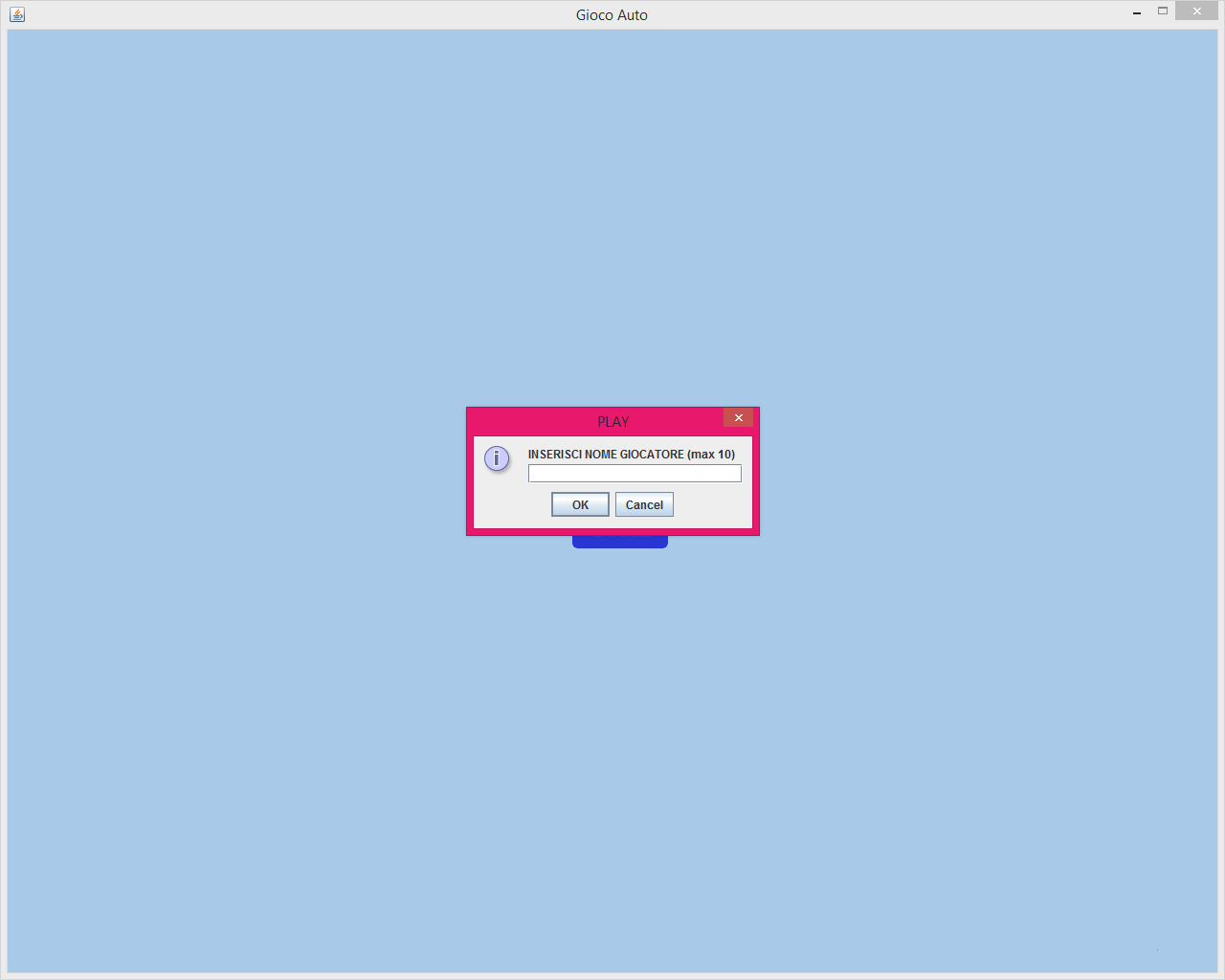


Figura 2: Scelta nome giocatore

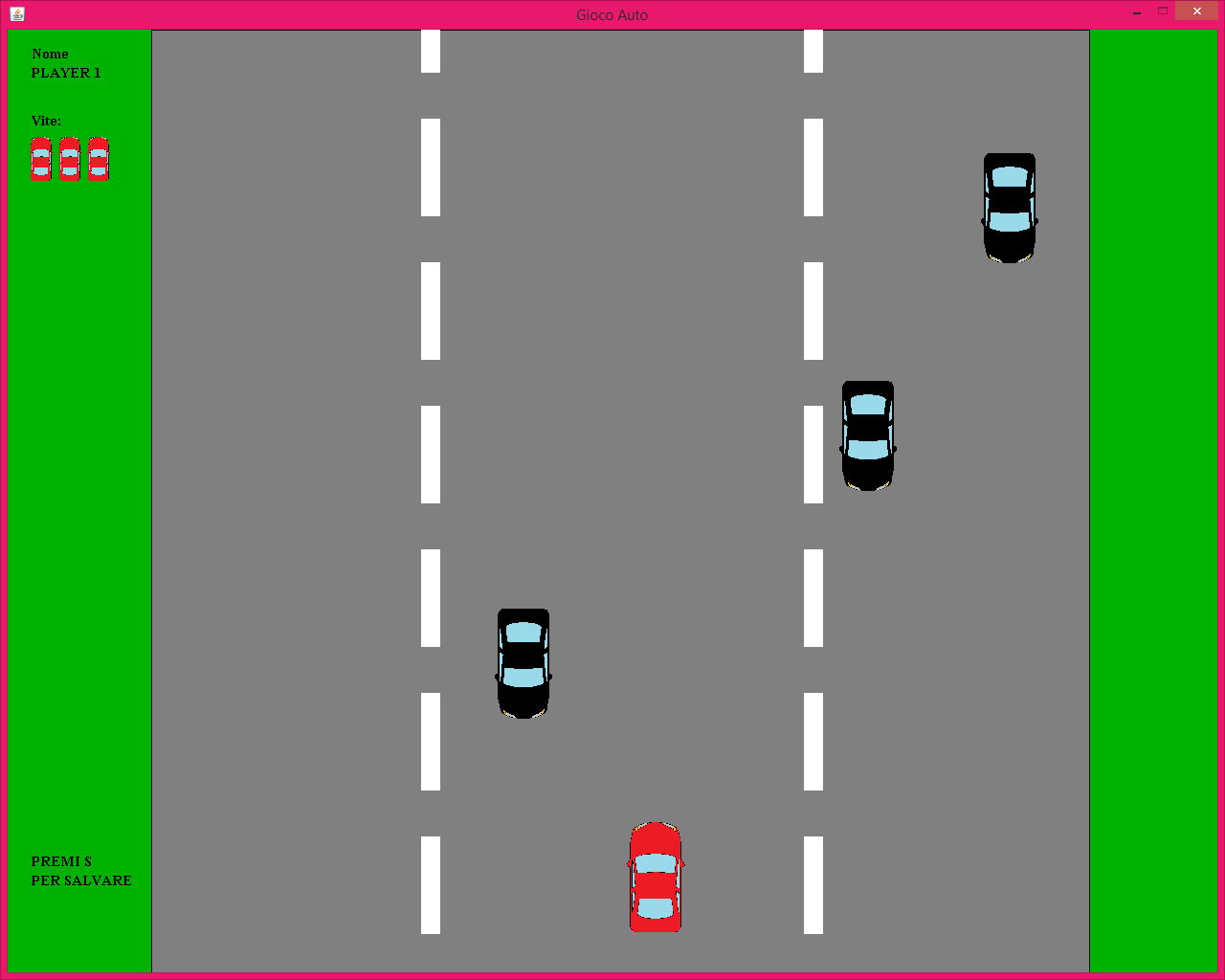


Figura 3: Partita avviata

Ad intervalli regolari, vengono creati ostacoli che compaiono dalla parte superiore della finestra; secondo la strategia corrente (indicata dal colore) l’auto metterà in atto dei comportamenti per evitarli.

Il gioco permette di salvare il suo stato, in qualsiasi momento della partita, semplicemente premendo il tasto “S” su tastiera. Un messaggio a schermo mostrerà l’esito del salvataggio, dopodiché, si ritorna al menu principale.



*Figura 4: Salvataggio stato*

Ogni qualvolta un ostacolo colpisce il giocatore, il gioco si riavvia e una vita viene sottratta; l’auto cambia colore, nonché comportamento (strategia) in base alle vite rimanenti.

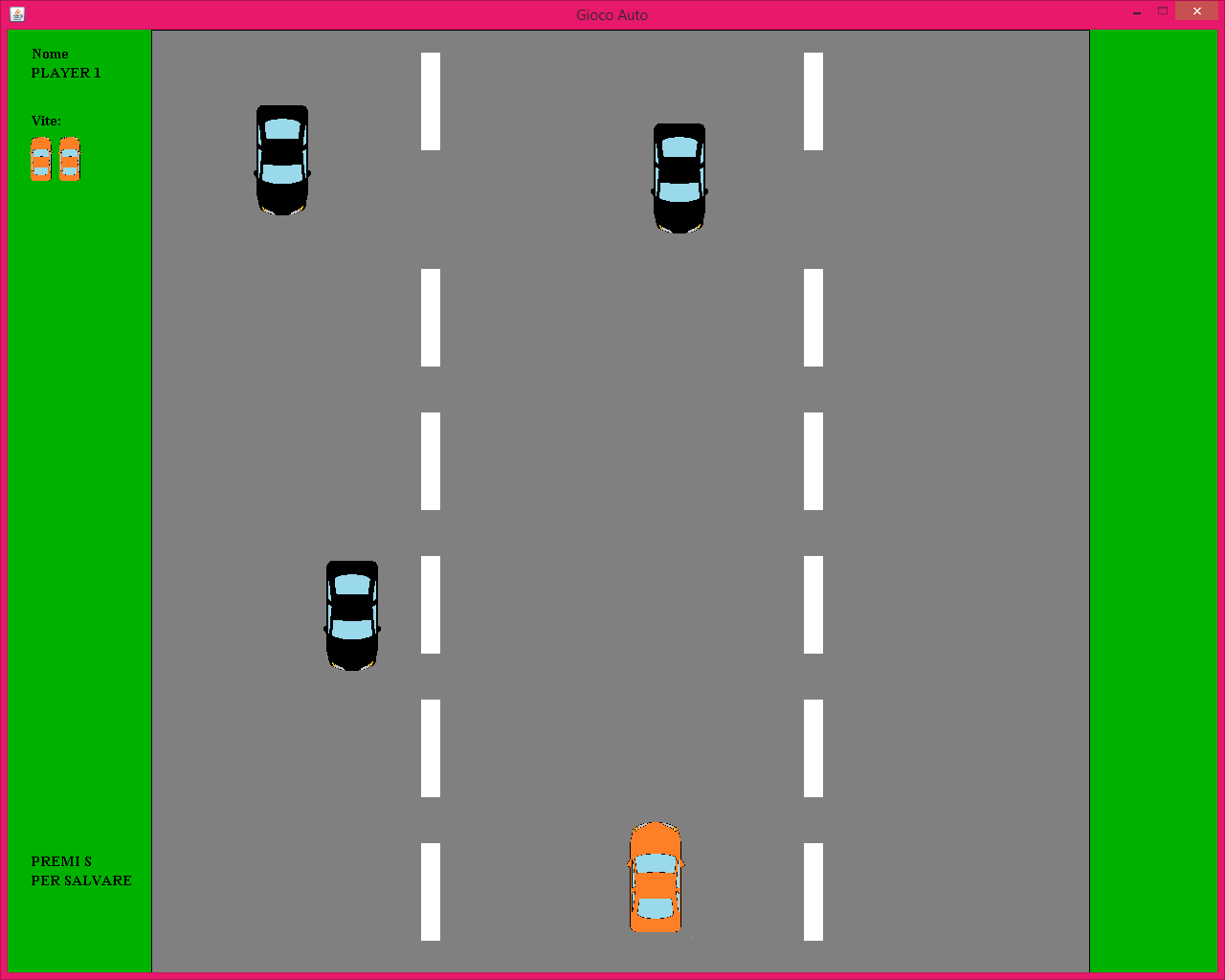


Figura 5: Due vite

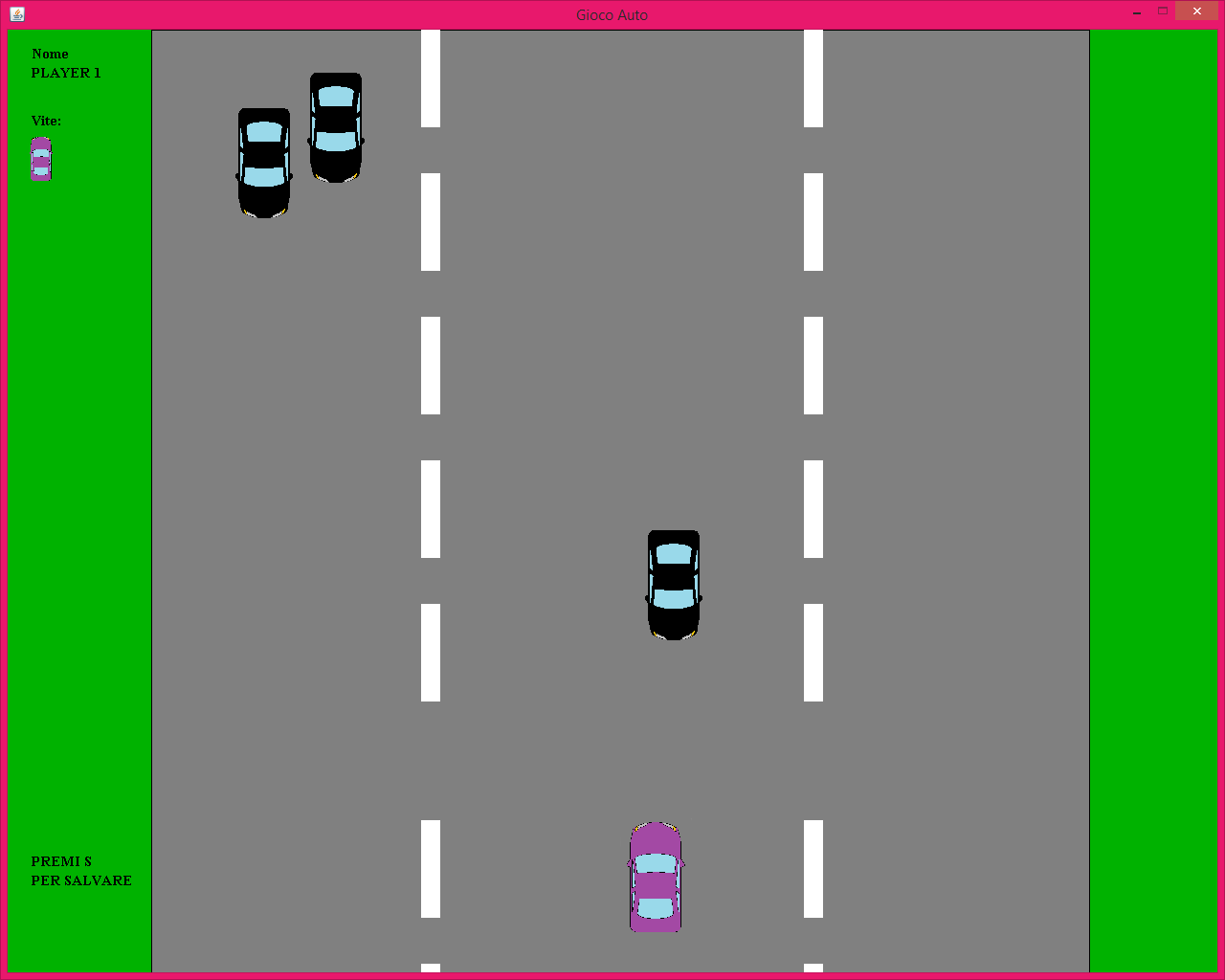


Figura 6: Una vita

Quando si esauriscono le vite, la partita termina e si può tornare al menu principale.



Figura 7: Game over

È possibile caricare, ripristinare lo stato precedente di una partita cliccando il pulsante “LOAD” nel menu principale.

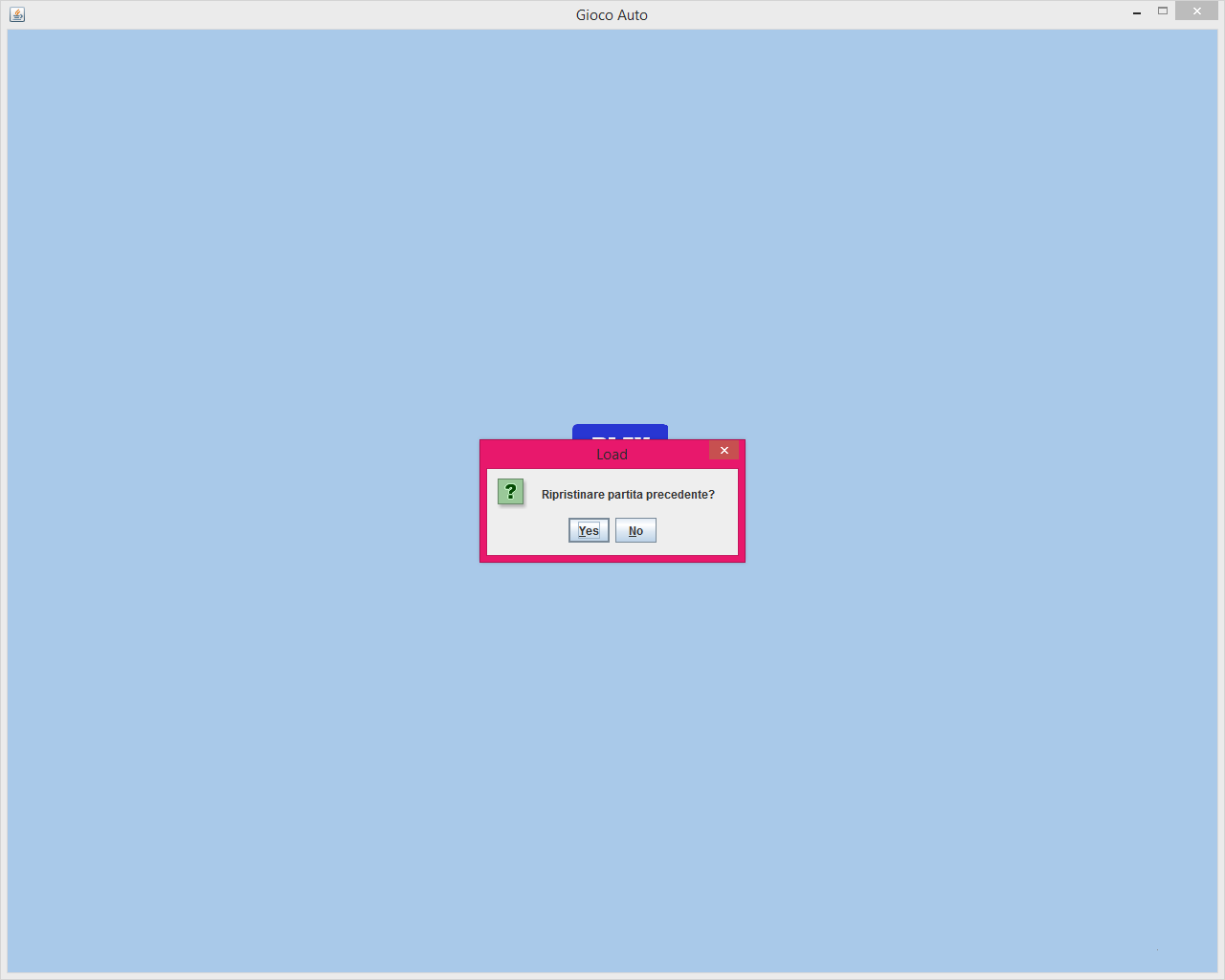


Figura 8: Caricamento

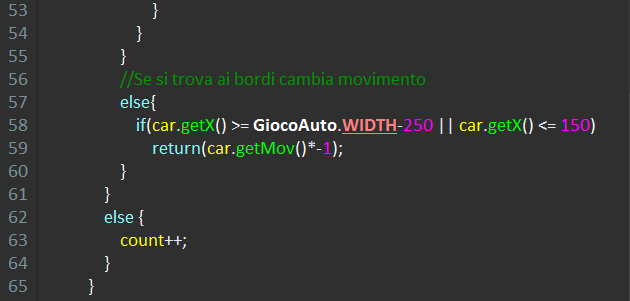
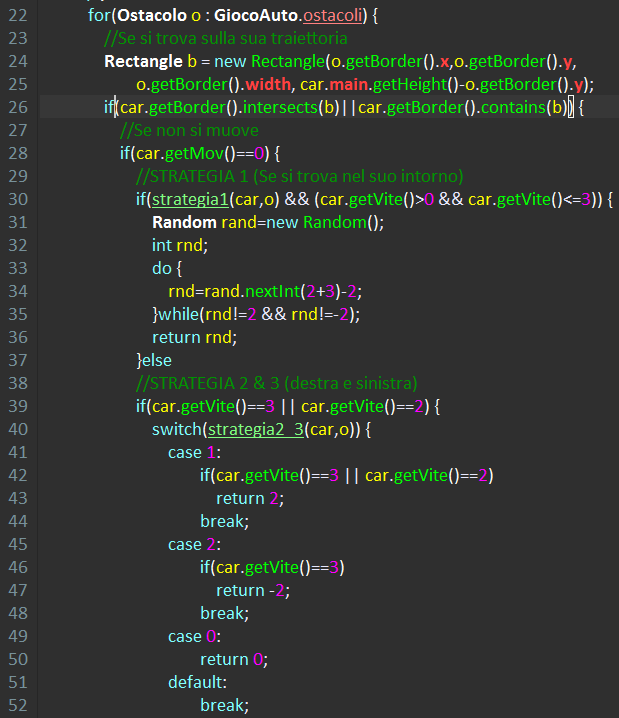
Confermando l’operazione, il gioco riprenderà dalla partita salvata in precedenza.

# Codice

## *INTERFACCIA STRATEGY*

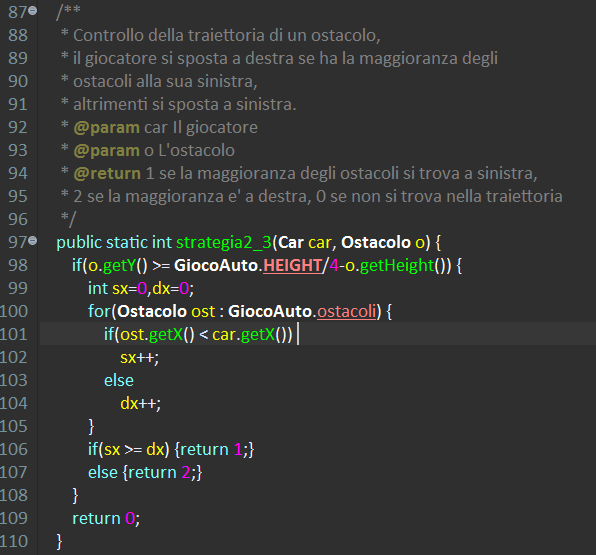
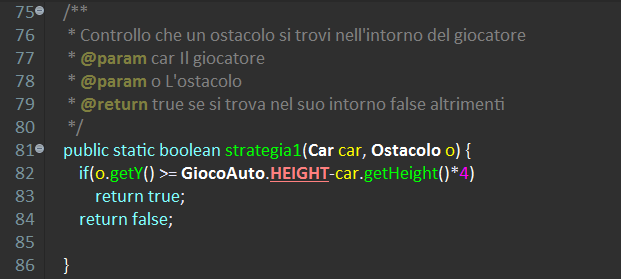
L’interfaccia Strategy è la parte fondamentale e funzionale del giocatore che, permette di applicare dei comportamenti, al fine di evitare gli ostacoli autonomamente.

Il cuore di quest’interfaccia è il ciclo for (nel metodo di default strategy) che controlla se c’è almeno un ostacolo che si trova nella traiettoria del giocatore.



I metodi statici, “strategia1” e “strategia2\_3” dell’interfaccia Strategy, sono le varie strategie da intraprendere, che sono:

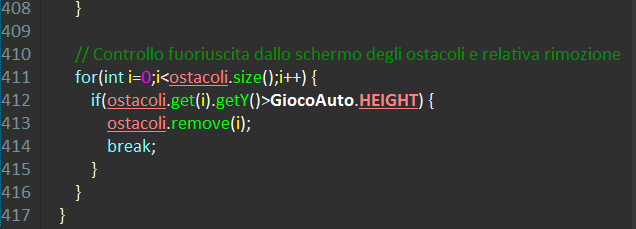
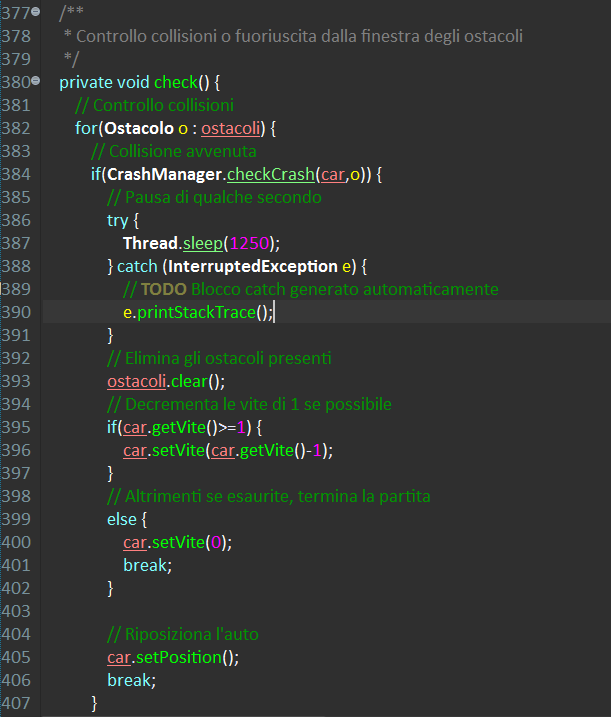
1. Casualmente nel caso un ostacolo si trovi nel suo intorno
2. A sinistra se la maggioranza degli ostacoli si trovano alla sua destra
3. A destra se la maggioranza degli ostacoli si trovano alla sua sinistra



## *METODO CHECK*

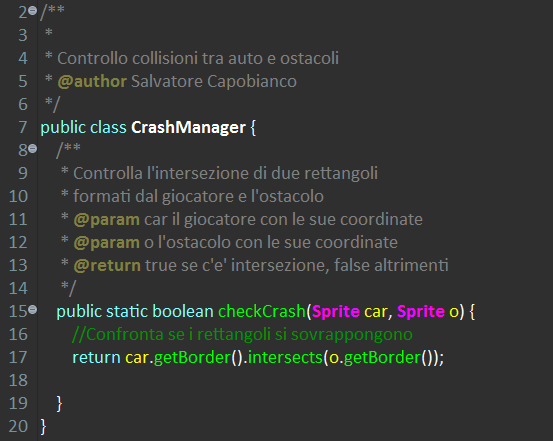
Questo metodo privato (invocabile solo dal principale GiocoAuto) controlla eventuali collisioni tra l’auto e un ostacolo; in caso affermativo riduce le vite a disposizione e riavvia lo scenario.

Inoltre controlla la fuoriuscita degli ostacoli dallo schermo e nel caso, provvede a rimuoverli dal gioco.



## *CLASSE CRASHMANAGER*

Questa classe viene utilizzata durante l’esecuzione quando viene invocato il metodo “check”. Ad essa è infatti delegato il controllo delle collisioni a basso livello: restituisce un valore booleano che indica se i due oggetti (Sprite) collidono oppure no.



## *METODO CREAOSTACOLO*

Questo metodo statico, crea un oggetto di tipo Ostacolo controllando che non si sovrapponga con gli altri già presenti.

