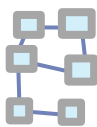
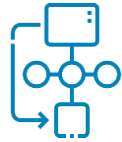




Tecnología de Programación



Modelando un videojuego – parte 3

Modelado de Road Fighter.

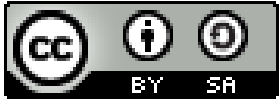
Diagrama de clases reducido.

Moviendo al auto del jugador aplicando hilos.

Dr. Federico Joaquín 
federico.joaquin@cs.uns.edu.ar

Algunos derechos reservados

Diseño OO - Modelado de un videojuego - parte 3



© Octubre 2025 por Federico Joaquín

Esta obra está bajo una Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional

Continuemos programando

REFACTORIZANDO UNA IMPLEMENTACIÓN PARCIAL
CASO DE ESTUDIO: ROAD FIGHTER
INCORPORACIÓN DE CAMBIO DE DIRECCIÓN DEL JUGADOR

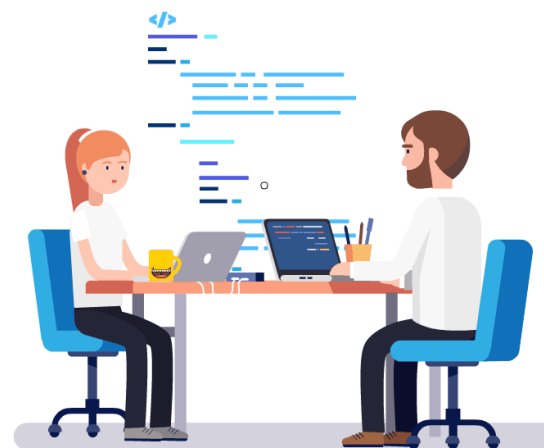


A partir de la implementación alcanzada durante la aplicación de las nociones del patrón de diseño basado en 3 capas, se **extenderá** dicha funcionalidad **refactorizando** el código para incorporar la posibilidad de que el usuario cambio de dirección el vehículo del jugador.

viernes, 10 de octubre de 2025

Implementación cambio dirección jugador

- Refactorización de la implementación **resuelta en vivo** durante la clase.



- **Analice** los detalles de la refactorización en los **siguientes enlaces**.
- **Captura eventos de teclado.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-concurrencia/commit-correspondiente>
- **Cambio de dirección desde la lógica y gráfica.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-concurrencia/commit-correspondiente>

Continuemos programando

REFACTORIZANDO UNA IMPLEMENTACIÓN PARCIAL
CASO DE ESTUDIO: ROAD FIGHTER
MOVIENDO EL JUGADOR DE FORMA MANUAL

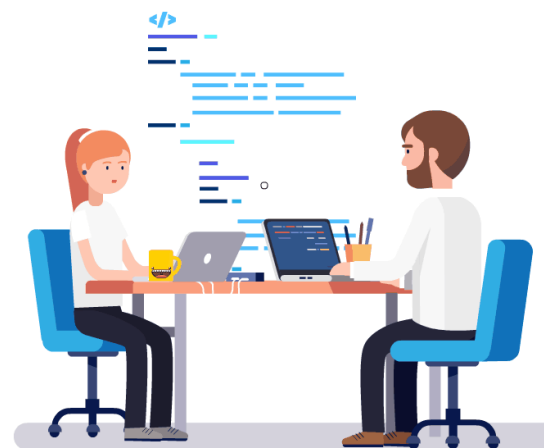


*A partir de la implementación alcanzada durante la refactorización previa, se **extenderá** dicha funcionalidad refactorizando el código para incorporar la posibilidad de que el usuario pueda mover el auto del jugador de forma manual presionando la barra espaciadora.*

viernes, 10 de octubre de 2025

Implementación movimiento manual

- Refactorización **resuelta en vivo** durante la clase.



- **Analice los detalles** de la segunda refactorización en clases en el **siguiente enlace**.
- **Mover auto mediante barra espaciadora.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-concurrencia/commit-correspondiente>

Continuemos programando

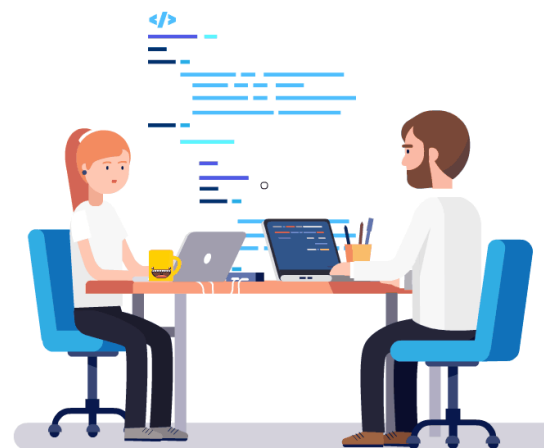
REFACTORIZANDO UNA IMPLEMENTACIÓN PARCIAL
CASO DE ESTUDIO: ROAD FIGHTER
MOVIENDO EL JUGADOR DE FORMA AUTOMÁTICA CON THREAD



*A partir de la implementación alcanzada durante la refactorización previa, se **extenderá** dicha funcionalidad refactorizando el código para incorporar la posibilidad de que el usuario pueda mover el auto del jugador de forma automática usando Threads.*

Implementación movimiento con Thread

- Refactorización **resuelta en vivo** durante la clase.



- **Analice los detalles** de la tercera refactorización en clases en el **siguiente enlace**.
- **Mover auto mediante hilos**. Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-concurrencia/commit-correspondiente>



Fin de la presentación.



Referencias

- **Primer Diagrama de clases para Road Fighter.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter/>
- **Refactorización según Modelo 3 Capas del Diagrama de clases para Road Fighter.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-3-capas/>
- **Refactorización de implementación parcial.** Repositorio TdP en GitHub.
<https://github.com/tecnologiaprogramacion/road-fighter-concurrencia>