

Universidad Complutense de Madrid  
Facultad de Informática  
Grado en Ingeniería Informática, Grupo DG

# Tecnología de la Programación I

## Práctica 2 – Parte II

Curso 2021-22

pasar los 4 primeros test del 00 al 04. mirar q no aparezca una moneda de mas o de menos, sería por el orden de ejecución hay q actualizar primero los objetos del juego y luego al player.

empezar con el muro, el mas sencillo, luego las instant actions y el shoot. Luego la granada y dejar para un poquito mas adelante el truck y el pedestrian

# Recordatorios Parte I

- `Command#parse(String[] words)`
  - Se encarga de procesar la entrada del usuario (separada en palabras)
  - Genera mensajes de error si los parámetros no son los esperados (número o tipo)
  - Devuelve `this` o `null` dependiendo de si el procesamiento es correcto o no.
- `GameObjectGenerator`
  - Métodos factoría para generar los objetos tanto en el momento de inicialización / reset como en tiempo de ejecución.
- `Collider#doCollision()`
  - Gestiona si ha habido o no colisión (ahora mismo en `Player`)
- `Collider#receiveCollision(Player)`
  - Gestiona la acción / resultado de la colisión con el objeto
- `ResetCommand`
  - **Hay que tener en cuenta que cambia visibilidad y carriles !**

comprobar con  
codigo actual

# Principios generales de la Parte II

- Los objetos **se van solapar en una casilla.**
  - **En caso de duda se tendrá en cuenta el orden en el que se han añadido al juego.**
- El jugador siempre se debe pintar

mirar q nuestro object in position se para al encontrar uno

los walls y todo esto sol objects

no hay q implementar nada, solo q en el object in position devuelva el primero de la lista q fue el primero q se añadió

# Objetos del juego nuevos

- Wall
  - hereda de la clase obstaculo (asi como pedestrian)?
  - Obstáculo con dureza
- SuperCoin
  - Proporciona más monedas al jugador
- Turbo
  - “Empuja” al jugador varias casillas
  - **No colisiona como resultado de empujar al jugador**
    - Al “siguiente” turno si podría colisionar.
- Truck
  - Se mueve de derecha a izquierda sin cambiar de carril.
  - Solo colisiona con el player (ver principios generales)
  - Hay que comprobar dos veces las colisiones para que no se nos pase ningún choque

si hay algun obstaculo lo pasa por encima, si pasas volando por obtaculos no te chocas y por encima de monedas no las coges, solo mover al coche

el turbo solo te mueve hacia delante y el turbo es un objeto si luego haces un movimiento hacia arriba o lo q sea

derecha a izq de la pantalla

se pinta la primera q se haya añadido al juego

no hace falta eliminar el camion aunque no choque, se queda avanzando fuera de la carretera hasta el infinito

la logica de la colision es en dos pasos:

actualizar y gestionar todos los objetos del juego (verificar colision) y luego actualizar y gestionar al jugador (verificar colision), porque puede ser que se intercambien las posiciones el jugador y el camion

mirar en los test que pasa cuando el jugador esta delante del camion (le de a lo que le de se choca no?) ● 4

● todos los objetos se crean al principio

# Objetos del juego nuevo

la granada no esta implementada como instantAction pq tiene un temporizador, pero la lanzamos

- Grenade

- Tiene un tiempo de vida limitado el tiempo son ciclos, turnos
- La explosión sólo afecta a Obstáculos y muros tb mata a los pedestrian

- Pedestrian

- Sólo cambia de carril si matas a un pedestrian con una accion que sea tu culpa (no un camion) pierdes todas las monedas (recuerda q el camion no hace docollision eso es solo el player)
- Comienza en el carril superior
- Cuando llega al inferior cambia de dirección
- Solo colisiona con el player (ver principios generales)
- Hay que comprobar dos veces las colisiones para que no se nos pase ningún choque

igual hacen algun cambio en los test del thunder: quieren poner en vez de "Obstacle hit" el nombre del obstacle al q hitteas

## Nuevas acciones

hay un test en la pagina 9 q está mal pq se pinta el tablero entre medias entre q el jugador y el camión se choca dice que en los test está bien puesto. Al actualizar el jugador te aparecería muerto y ya está, los movimientos se hacen en 1 turno no no en 2

un paquete separado dentro de logic con las acciones

- **InstantAction** las acciones se ejecutan como un comando pero no pasa ciclo, es como help
  - Se ejecutan en el momento => no hay que esperar a que se vayan a actualizar jugador / objetos de juego.
  - El resultado de la acción se implementa en cada acción instantánea
    - Shoot, Wave, Thunder
  - NOTA: no mezclar responsabilidades entre comandos y acciones
- **Shoot**
  - El jugador dispara al primer obstáculo que tenga el carril
  - Si hay solapamientos ... (ver principios generales)
- **Wave**
  - Desplaza los objetos dentro del rango "visible"
  - Sólo si es posible desplazarlos
    - Pensar un poco cómo lo podéis implementar para maximizar los objetos a mover
- **Thunder**
  - Mata instantáneamente el obstáculo o muro.

otra cosa es q tu como resultado del comando tengas que ejecutar la accion

las clases q implementan la interfaz instantAction son shoot, wave, thunder...

que esté inicialmente en visibility!! Puede q esté en la última casilla y q se mueva 1 fuera de la visible

por como deberíamos implementar la logica si hay objetos solapados se mueve el primero q encuentres, el q está antes añadido (no preocuparse mucho)

pensar si hay q desplazar primero los de la derecha o los de la izquierda. se mueven todos menos el coche. cada objeto lo desplaza 1 posición.

aunq tenga varias vidas el obstaculo

la unica runtime action q tenemos ahora es el thunder

el rayo no le puede caer encima al jugador. El thunder no es comando. Si hay varios objetos se carga al primero de la

# Nuevas acciones

interfaz, si se puede comprar o no un objeto/accion (shoot o granade por ej)

- **Buyable**
  - Hay objetos / acciones que se deben comprar
  - Sólo se puede ejecutar / usar el objeto si el jugador tiene monedas suficientes

una vez q compras algo se usa,  
no tenemos inventarios

en los comandos en los q tengas q implementar logica de comprar monedas  
tienes q implementar la interfaz buyable para ver si se puede ejecutar.  
Si no se puede ejecutar se lanza un mensaje de error, si se puede se  
decrementan las monedas

no se guarda el numero de monedas de una partida a otra