

GEOMAGO

Proyecto Final
Programación Orientada a Objetos
2014

Marco Benzi - René Pozo - Andrés Ulloa

DESCRIPCIÓN DEL PROBLEMA

- Deseamos crear un juego de mesa
 - Inspirado en el ajedrez
 - Para 2 a 4 jugadores

DESCRIPCIÓN DEL PROBLEMA

- **GeoMago**

- Cada jugador parte con 10 piezas
- Existen tres tipos de pieza. Cada una tiene una cantidad máxima de movimientos posibles en un turno
- El objetivo es eliminar (“comer”) las piezas de todos los oponentes
- Todas las piezas ganan uno de movimiento al comenzar el turno

DESCRIPCIÓN DEL PROBLEMA

- **GeoMago**
 - Los movimientos se realizan mediante un tablero cuadriculado
 - Cada movimiento se define por cantidad de cuadros desplazados

DESCRIPCIÓN DEL PROBLEMA

- Sin embargo, un juego de mesa tiene limitaciones inherentes a que es un objeto físico:
 - Hay que construirlo
 - Las piezas se pueden perder
 - Los jugadores deben hacer regir las reglas (por ejemplo contar los movimientos)

CASO DE USO

JUGADOR VERSUS AI

JUGADOR VS AI

- Para todo evento, el usuario en este caso es una persona que desea jugar. También se considera actor la AI
- El humano debe haber sido capaz de configurar y ejecutar el juego

JUGADOR VS AI

- La aplicación en este punto debe ser capaz de generar un estado nuevo de juego, que incluye una configuración diferente de obstáculos. También debe asignar correctamente piezas nuevas a todos los jugadores

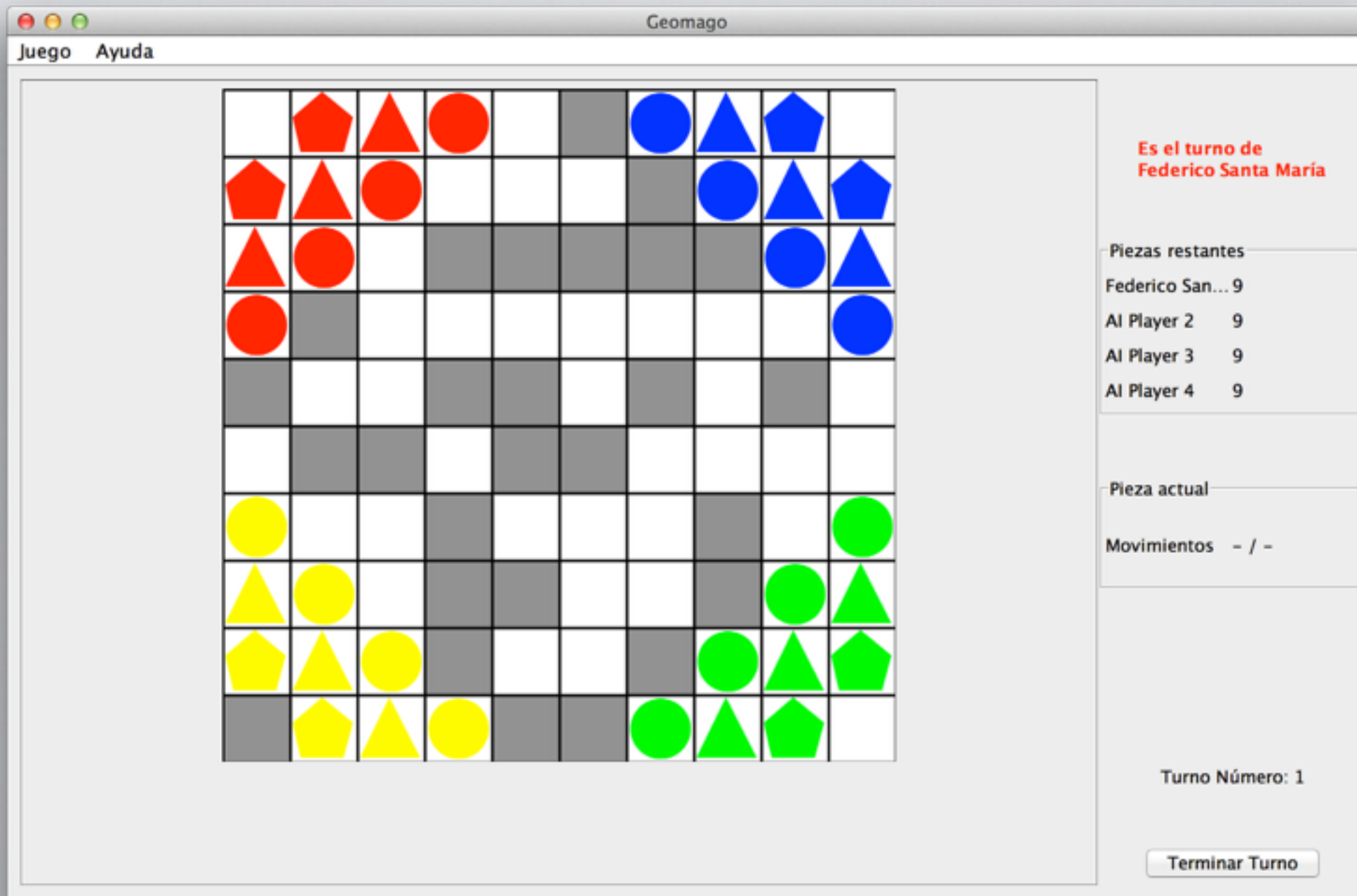
JUGADOR VS AI

- Al terminar el juego, el sistema le debe preguntar al usuario humano si desea jugar de nuevo, abriendo inmediatamente el dialogo para configurar una nueva partida si dice que si

SOLUCIÓN

- Generar una aplicación Java usando Swing que cumpla los requisitos mencionados anteriormente.

SOLUCIÓN

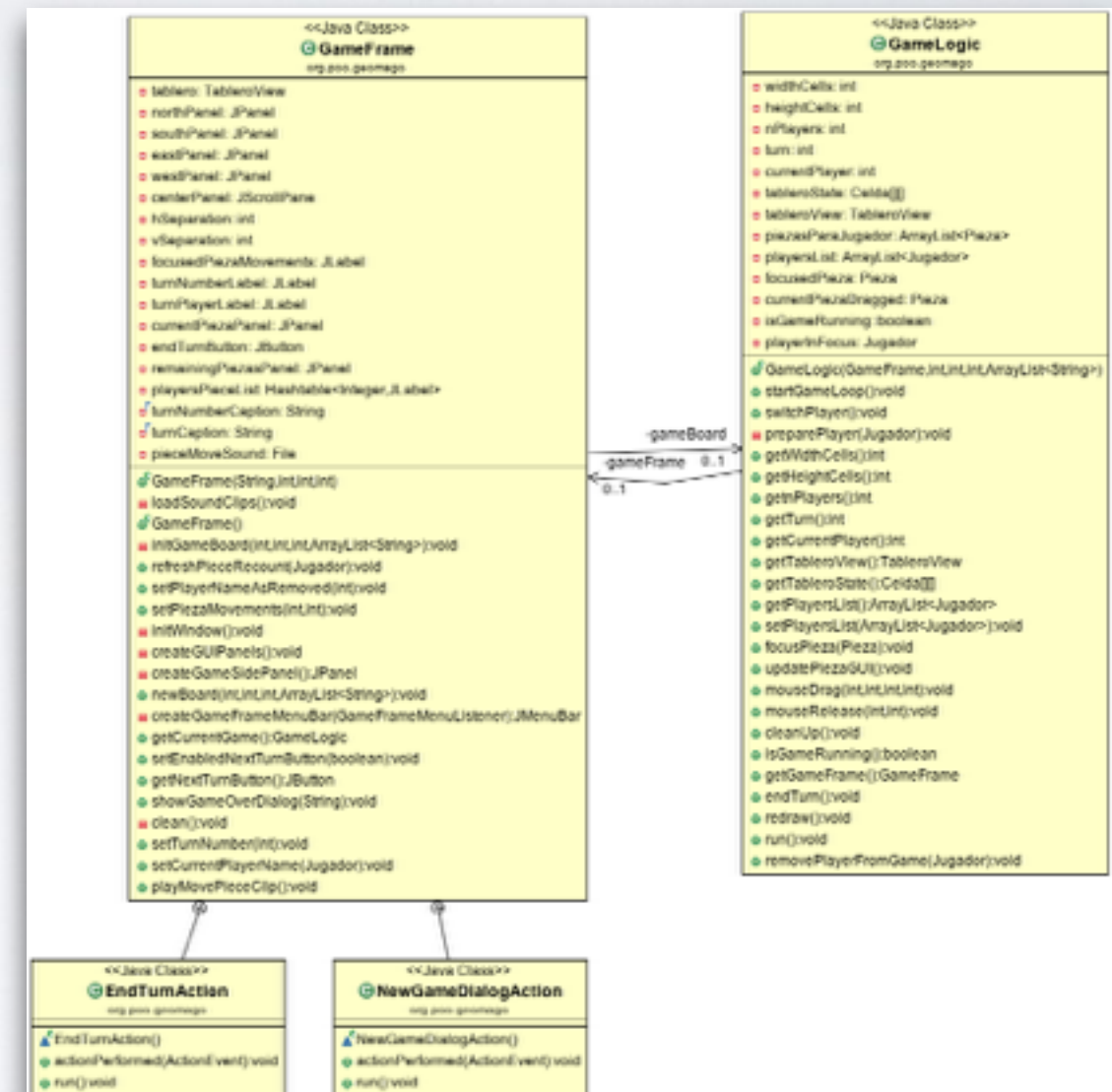


ASPECTOS GENERALES DE LA SOLUCIÓN

- Uso de Modelo - Vista - Controlador como base del diseño del programa
- Trabajo Multi-hilo para evitar que procesos bloqueen la GUI
- Esquema de herencia para las piezas y los jugadores

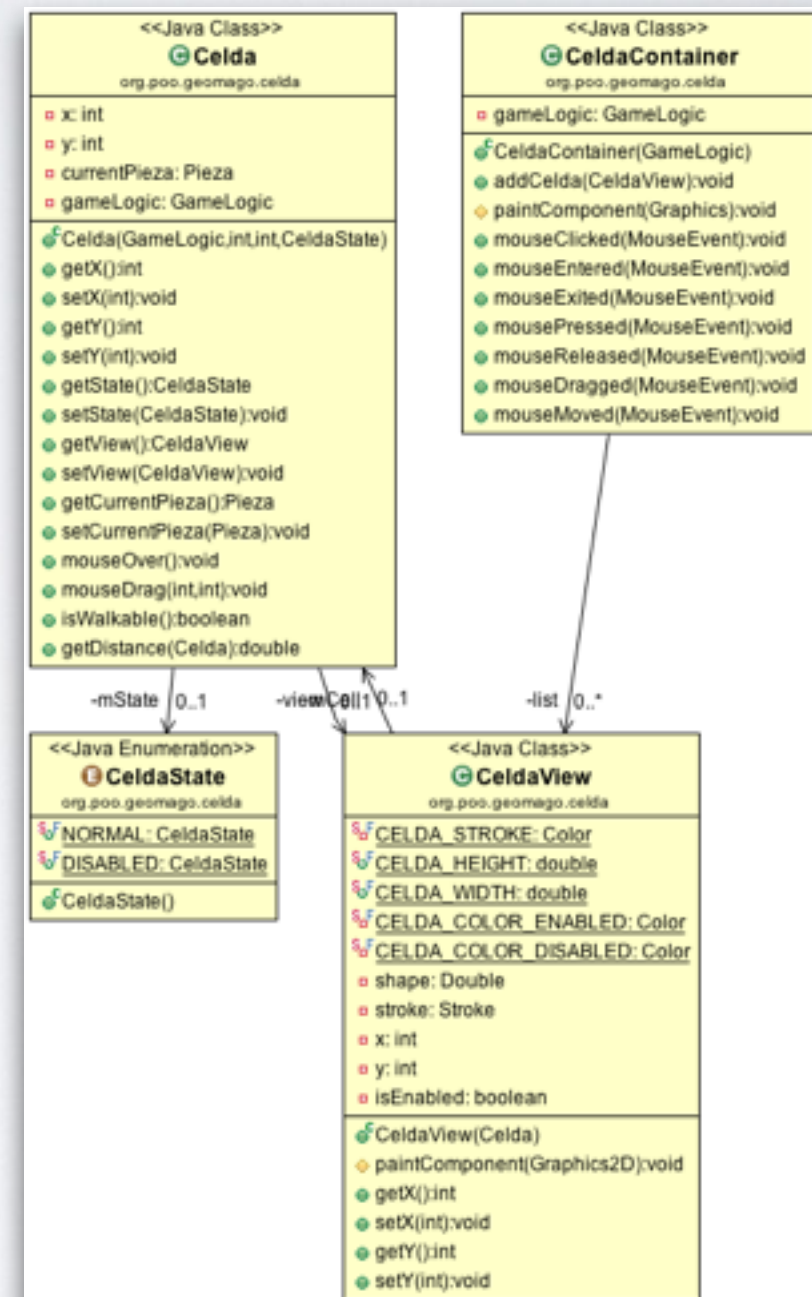
ASPECTOS GENERALES DE LA SOLUCIÓN

- Uso de Modelo - Vista - Controlador como base del diseño del programa



ASPECTOS GENERALES DE LA SOLUCIÓN

- Uso de Modelo - Vista - Controlador como base del diseño del programa

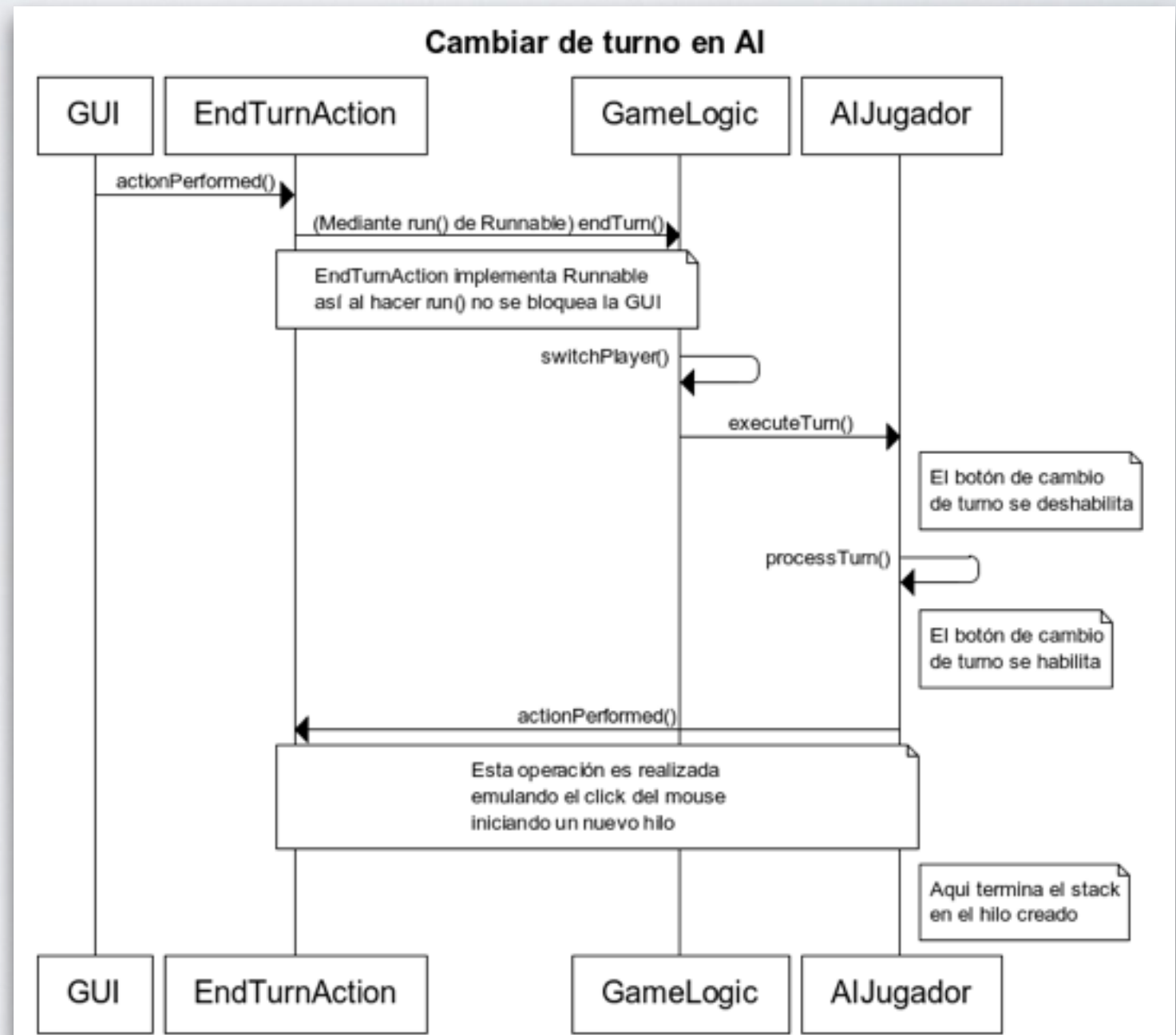


ASPECTOS GENERALES DE LA SOLUCIÓN

- Uso de Modelo - Vista - Controlador como base del diseño del programa
- Trabajo Multi-hilo para evitar que procesos bloqueen la GUI
- Esquema de herencia para las piezas y los jugadores

ASPECTOS GENERALES DE LA SOLUCIÓN

- Trabajo Multi-hilo para evitar que procesos bloqueen la GUI

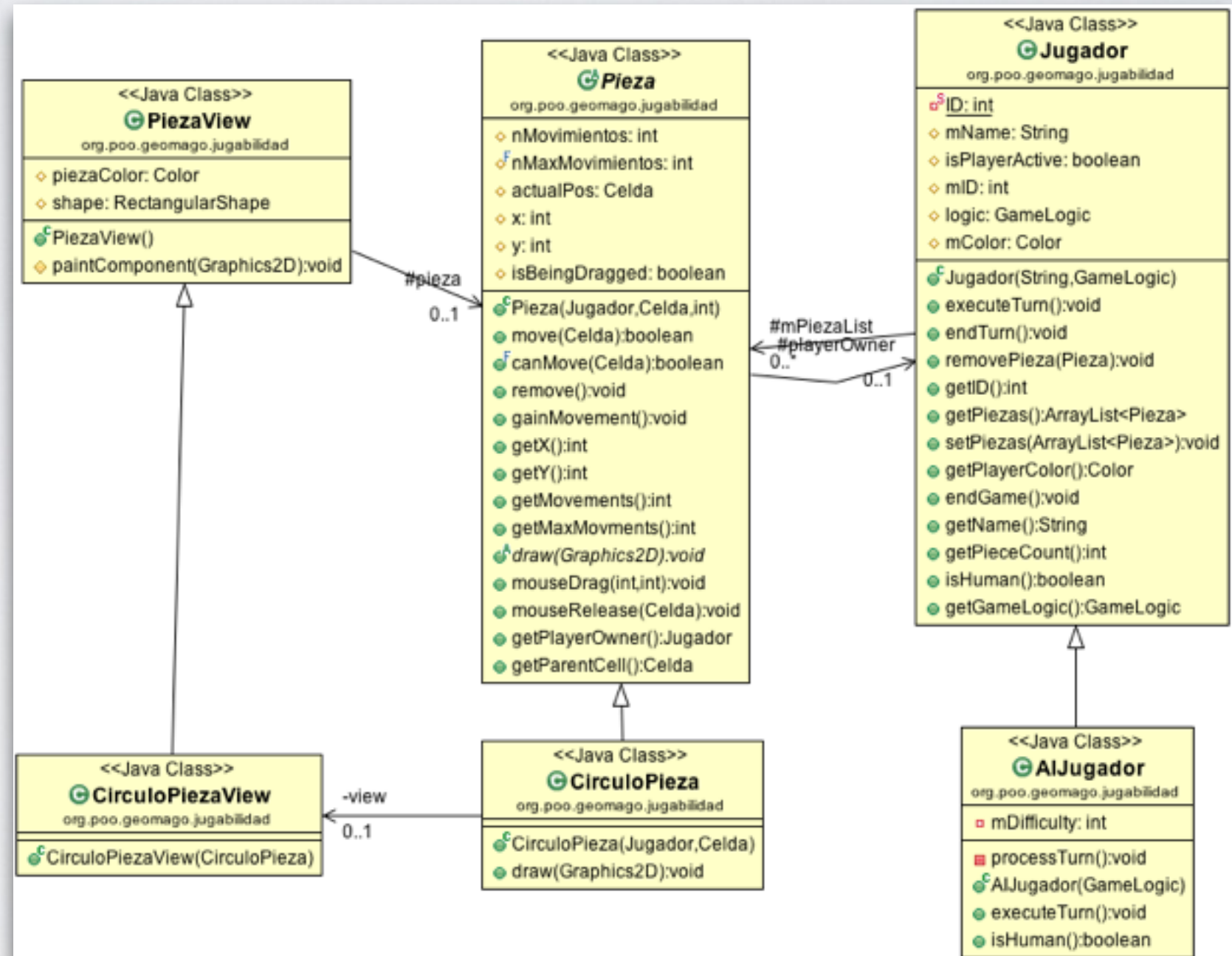


ASPECTOS GENERALES DE LA SOLUCIÓN

- Uso de Modelo - Vista - Controlador como base del diseño del programa
- Trabajo Multi-hilo para evitar que procesos bloqueen la GUI
- Esquema de herencia para las piezas y los jugadores

ASPECTOS GENERALES DE LA SOLUCIÓN

- Esquema de herencia para las piezas y los jugadores



ASPECTOS GENERALES DE LA SOLUCIÓN

- Usamos Eclipse como IDE y un plugin para generar los UML directamente
- Ayudó mucho en el diseño iterativo de las piezas

DIFICULTADES

- Como evitar bloqueos de GUI
- Desarrollo de una IA
 - *iStack Overflows!*
- Dibujar figuras que no vienen por defecto (triángulos y pentágonos)
- Reproducir sonidos más de una vez

“Do, don’t tell”

DEMO