



undo/redo



Qué guardar

- Se guarda el estado del juego
- El estado del juego es un objeto de la clase **GameState**
- Para conocer el estado del juego antes de una jugada J se guarda una matriz de enteros, donde cada componente contiene el valor de la celda correspondiente antes de hacer la jugada J, y se guarda el *score* que hubiera antes de hacer la jugada J
- El valor *highest* es calculable a partir de la matriz de enteros



Cómo guardar el estado

- Se utilizan arrays de estados del juego para guardar las últimas jugadas
- Los objetos de la clase **GameStateStack** son los que se usan para guardar las últimas jugadas
- Los objetos de esta clase tienen, como atributos, un array de **GameState**'s, más un contador para indicar el nivel de ocupación
- El array se comporta como una pila. Las operaciones **push(state)** y **pop()**, apilan y desapilan estados del array, respectivamente



undo/redo de Game

- **Game** mantiene dos pilas (**ues** y **res**), una para implementar **undo()** y otra para implementar **redo()**
- **undo()**. En condiciones sin excepción:
 - Desapila de **ues** en el estado **state**
 - Apila el estado actual en **res**
 - Establece **state** como estado actual del juego
- **redo()**. En condiciones sin excepción:
 - Desapila de **res** en el estado **state**
 - Apila el estado actual en **ues**
 - Establece **state** como estado actual del juego