

Implementación del algoritmo de búsqueda

Para implementar el algoritmo de búsqueda de palabras hemos necesitado crear una nueva clase auxiliar `PlayableWord` que contiene como atributos una `String` que contiene la posible palabra a colocar en el tablero y sus coordenadas de inicio y final de posicionamiento dentro del tablero.

La función principal del algoritmo `generateValidWords` devolverá una lista justo de elementos de esta clase que contendrá todas las palabras posibles a crear en un tablero con un diccionario y un atril de letras específicos, esta busca palabras a partir de casillas calculadas llamadas “anchors” (anclas) que no son más que las casillas adyacentes a fichas ya colocadas por arriba y por la izquierda, a partir de aquí se separa la búsqueda en posibles palabras en horizontal y en vertical, y para buscar cada una de estas se separa el algoritmo también en dos búsquedas, primero se buscan las partes izquierdas de las posibles palabras usando siempre como sufijo de estas medias palabras el contenido de la ficha a la que es adyacente el ancla del que se está buscando, guardando todas las posibles combinaciones que se pueden hacer con las fichas del rack que existan como prefijos de palabras en el diccionario usando la técnica del `backtracking`.

Una vez hecho esto se pasarán a calcular las palabras enteras calculando las posibles partes derechas, usando también `backtracking` y comprobando que todas las palabras existen en el diccionario y se pueden poner en el tablero, una vez hecho esto se devuelven y se gestionan.

String extendRight(String word, int x, int y, Board board, Dictionary dictionary)

Esta función extiende la palabra “word” hacia la derecha iniciando en la casilla pivote con las coordenadas x e y.

int getLeftPartLimit(Board board, Box anchor, boolean hor)

Esta función indica la cantidad de casillas que hay a la izquierda de la casilla pivote “anchor” hasta que encuentre otra ficha o se salga del tablero.

List<String> generateLeftParts(Board board, Box anchor, Rack rack, Dictionary dictionary, int limit, List<Integer> nSpecialCharactersInLeftParts)

Esta función invoca a la función recursiva del mismo nombre para generar las partes izquierdas de las posibles palabras que se pueden formar con las fichas del rack. El atributo nSpecialCharactersInLeftParts