

Tarea del Ajedrez

Alfredo Paz Valderrama

11 de junio de 2019

1. Introducción

No es poco común utilizar al ajedrez como inspiración para ilustrar o practicar soluciones y técnicas para problemas computacionales, tal como lo hizo Dijkstra en 1972 con el problema de las 8 reynas. En esta primera tarea, usted usará una biblioteca de código para dibujar posiciones sobre un tablero de ajedrez.

Aunque la biblioteca hace uso de OpenGL, usted no tendrá que saber nada sobre programación de gráficos. En esta tarea, sólo tendrá que realizar sus propias versiones del método `display()`, las que deberá ubicar en distintos archivos denominados `ejer01.c`, `ejer02.c`, etc. Por ejemplo para dibujar un caballo bastará definir la siguiente función `display()`.

```
#include "chess.h"
#include "figures.h"

void display(){
    char** blackKnight = reverse(knight);
    interpreter(blackKnight);
}
```

Para compilar los archivos usted dispone de un archivo `makefile` ya listo, al que deberá invocar de la siguiente manera desde la línea de comandos.

```
$ make HW=ejemplo
```

En este caso el archivo a compilar se deberá llamar `ejemplo.c` y el ejecutable generado será `ejemplo`. Al ejecutar el programa podrá visualizar la figura en una ventana, la que podrá cerrar presionando `ctrl+c` en la terminal o consola.

Tener cuidado de haber instalado la biblioteca *glut*.

```
$ sudo apt-get install freeglut3-dev
```

Al ejecutar el programa

2. Las figuras

El archivo `figures.h` contiene las figuras que usted usará para pintar el tablero, las figuras están descritas como arreglos de *strings*, arreglos que contienen como último elemento a 0. Note que los propios *strings* usan el valor de 0 para marcar su fin.

whiteSquare		bishop		king		knight	
		pawn		queen		rook	

3. Funciones

El archivo `chess.h` contiene las declaraciones de las funciones abajo descritas. Tenga en cuenta que estas funciones están reservando nueva memoria (`malloc`), para las figuras que devuelven, por lo tanto podría ser necesario que libere la memoria de los arreglos de strings usando `free`.

<code>flipV</code>	crea una imagen que es un espejo vertical de la imagen original
<code>flipH</code>	crea una imagen que es un espejo horizontal de la imagen original
<code>rotateL</code>	crea una imagen producto de rotar hacia la izquierda a la imagen original
<code>rotateR</code>	crea una imagen producto de rotar hacia la derecha a la imagen original
<code>reverse</code>	crea una imagen que representa un negativo de la imagen original
<code>superImpose</code>	crea una imagen producto de poner una imagen sobre otra
<code>join</code>	crea una imagen producto de poner una imagen al lado de otra
<code>up</code>	crea una imagen producto de poner una imagen sobre otra
<code>repeatH</code>	crea una imagen producto de repetir horizontalmente n veces la imagen original
<code>repeatV</code>	crea una imagen producto de repetir verticalmente n veces la imagen original

4. Ejercicios

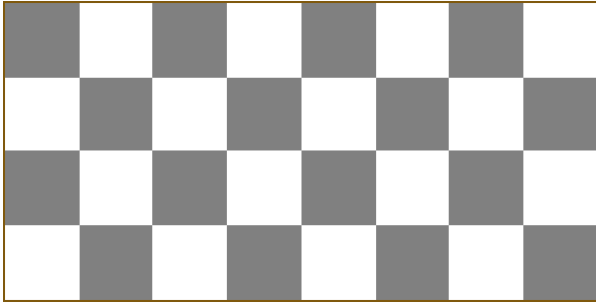
1. Utilizando la función `repeatH` cree la siguiente figura, tenga en cuenta que se inicia con un casillero blanco.



2. Usando libremente las funciones de la biblioteca, cree la siguiente figura:



3. Usando libremente las funciones de la biblioteca, cree la siguiente figura:



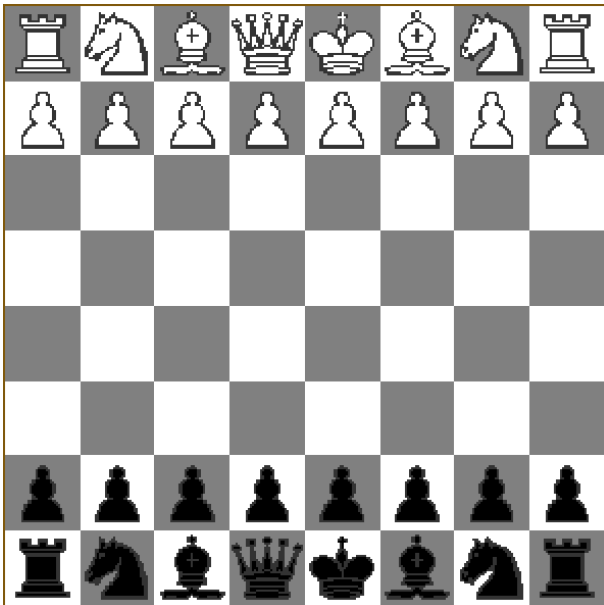
4. Usando libremente las funciones de la biblioteca, cree la siguiente figura:



5. Usando libremente las funciones de la biblioteca, cree la siguiente figura:



6. Usando libremente las funciones de la biblioteca, cree un tablero de ajedrez completo:



7. Dibuje una solución para el problema de las n-reynas (menor a 8).
8. Libremente haga un dibujo simétrico usando las piezas del ajedrez de la biblioteca del profesor y al menos las funciones `flipV`, `flipH`, `rotateL`, `rotateR`.