



**FACULTAD
DE INGENIERIA**
Universidad de Buenos Aires

Algoritmos y Programación I
Curso Mendez

TP2 - Snitch Dorada



| | |
|--------------------|------------|
| Fecha Presentación | 01/01/2018 |
| Fecha Entrega | 01/01/2018 |

Introducción

El **Quidditch** es el deporte más popular en la comunidad mágica. Es una especie de fútbol-baloncesto aéreo que se juega volando sobre escobas.

Entre las escobas las hay de muy diversa calidad, desde las Barredora 5, pasando por la Nimbus 2000 y Nimbus 2001, hasta llegar a la Saeta de Fuego, escoba prácticamente exclusiva para profesionales y de la que Harry Potter tiene un modelo que le regaló su padrino Sirius Black.

Se juega en un estadio en forma ovalada (150 metros de semieje mayor y 55 de semieje menor), cada uno de los lados está dotado de tres aros ubicados a un distinto nivel de altura.

Para el juego se utilizan tres tipos de bolas distintas: *Quaffle*, *Bludger* y la *Snitch Dorada*.

La bola más importante durante el encuentro, es la *Quaffle*, que se utiliza para marcar tantos haciéndola pasar a través de los aros del equipo contrario, al pasarlos por los aros se ganan 10 puntos, con ella juegan los Cazadores.

Las *Bludgers* son bolas que atacan a los equipos. Son pequeñas pero pesadas y vuelan con fuerza y son receptadas por los Bateadores, quienes utilizan bates pequeños para evitar que golpeen a algún jugador de su equipo.

Y la *Snitch Dorada* es una pequeña bola dorada y alada muy rápida y difícil de atrapar, la cuál debe ser atrapada por los Buscadores, al atraparla se ganan 150 puntos y el partido se termina.

“Los Dementores están entre las criaturas más nauseabundas del mundo. Infestan los lugares más oscuros y más sucios. Disfrutan con la desesperación y la destrucción ajenas, se llevan la paz, la esperanza y la alegría de cuanto los rodea... Si alguien se acerca mucho a un Dementor, este le quitará hasta el último sentimiento positivo y hasta el último recuerdo dichoso. Si puede, el Dementor se alimentará de él hasta convertirlo en su semejante: un ser desalmado y maligno. Lo dejará sin otra cosa que las peores experiencias de su vida.” Le dijo Remus Lupin a Harry Potter.

En una oportunidad, un Dementor persiguió a Harry Potter mientras éste perseguía a la Snitch Dorada, haciéndolo caer de su escoba desde muchos metros al vacío.

Objetivo

El presente trabajo práctico tiene como objetivo evaluar a los alumnos en aspectos básicos de la programación.

Se evaluará la interacción con el usuario, la forma de informar errores en sus ingresos usando una interfaz gráfica rudimentaria. También se hará foco en el correcto uso de estructuras de control y tipos de dato compuestos: matrices, vectores y registros.

Por supuesto, se requiere que el trabajo cumpla con las buenas prácticas de programación profesadas por la cátedra. Se considerarán críticos la modularización y reutilización de código, la definición y el uso de constantes y la claridad del código.

Enunciado

En esta oportunidad se quiere desarrollar una expansión al trabajo práctico Sombrero Seleccionador, logrando una versión reducida del juego **Quidditch**.

Dicho juego debe permitir a un jugador perseguir la Snitch Dorada a través de un estadio, en el cual habrá Bludgers y Dementores con el objetivo de que el jugador no logre atrapar la Snitch.

A continuación se explicará cada elemento que debe estar presente en el trabajo práctico, cuáles serán sus características y su rol dentro del juego.

Jugador

Es el personaje que se moverá dentro del estadio con el objetivo de atrapar la Snitch Dorada, deberá esquivar Bludgers y Dementores para lograr su objetivo.

- El jugador debe ser manejado por el usuario con las teclas *wasd*: a (izquierda), d (derecha), w (arriba) y s (abajo).
- Se considerará que el jugador atrapó la Snitch si ambos están en la misma posición del estadio.
- Se contará con una resistencia inicial de 150 puntos para atrapar la Snitch. Cada movimiento de la partida le restará 1 punto, en caso de

que se termine la resistencia y no se haya atrapado la Snitch, el jugador ha perdido.

- No están permitidos los movimientos en diagonal.
- Si el jugador atrapa la Snitch antes que se terminen sus movimientos, habrá ganado el juego.
- Cada movimiento del jugador dará lugar a un turno.
- El jugador *moverá primero*, de modo que si tiene la snitch al lado, y se le mueve encima, habrá ganado, y la snitch no deberá moverse más.
- Cada jugador pertenece a una casa y dicha casa le otorga una habilidad especial dentro del juego.
 - Gryffindor: Mediante el uso del hechizo *expecto patronum*, puede eliminar a los dementores que se le acercan. En este sentido, cuando un dementor ocupa la misma posición que un jugador de la casa Gryffindor, el dementor desaparece.
 - Slytherin: Gracias a su ingenio, pueden esquivar las Bludgers cuando estas los intentan golpear.
 - Ravenclaw: Debido a su gran inteligencia y estudio, lograron aprender a atrapar la Snitch no solo si estan en la misma posición, sino también si esta se encuentra en una posición aledaña a la suya (submatriz de 3x3).
 - Hufflepuff: La querida Hufflepuff, dentro de todo lo que sabía, también les enseñó a manejar la escoba con una extraordinaria rapidez, por lo que ellos pueden moverse dos veces por cada movimiento de la Snitch, Dementores y Bludgers.
- Para determinar a qué casa pertenece, adaptar el TP1 como una biblioteca con una función *obtener_casa* que pida los atributos y determine la casa a la que el jugador pertenece.

Snitch Dorada

Tiene un movimiento errático, su dirección será aleatoria respecto de la posición actual. Sin embargo, esta esquivia construcción mágica, tiene la facultad de teletransportarse luego de cargar alguna energía, según los factores climáticos que enmarquen la partida. Por esto, cada 5 turnos, debe teletransportarse a una ubicación aleatoria. Si se teletransportara a la posición del jugador, este ganaría la partida inmediatamente.

No es afectada ni por los Dementores ni por las Bludgers, pero no puede salir del campo. No mueve en las diagonales, salvo que se trate de un teletransporte.

Sugerencia

Para lograr el movimiento en dirección aleatoria use la función **random**.

Dementores

En estado normal tiene un movimiento errático, su dirección será aleatoria respecto de la posición actual. Sin embargo, cuando un jugador está a menos de 3 casillas de distancia de su posición, se dirigirá directamente hacia su posición, comenzando a perseguir al jugador. Dado que no se puede mover en diagonal, en caso de necesitar hacerlo, se moverán siempre primero en el sentido vertical.

Sugerencia

Para lograr el movimiento en dirección aleatoria use la función **random**.

Estas escurridizas criaturas tienen la costumbre de congregarse entre ellos para hacer su trabajo más eficientemente, por ello es que en el campo de juego, se irán sumando. El juego comienza con sólo uno de ellos en una posición aleatoria, y con el pasar de los turnos, uno a uno se irán sumando. Considere en un principio que aparecerá un nuevo dementor en una casilla libre, aleatoria, cada 15 turnos. Este número debe ser fácilmente modificable. Los dementores pueden aparecer en cualquier lugar del campo a excepción de la posición exacta del jugador.

Bludgers

Las Bludgers nunca se quedan quietas, en el campo habrán 5 filas por las que pasará un Bludger por cada una y se mueven horizontalmente un casillero por turno. Si una Bludger se encuentra en la misma posición que el jugador, éste pierde 10 movimientos y la Bludger desaparece.

Campo de Juego

El campo de juego se representará con una matriz de 25 filas por 15 columnas, dentro del cual transcurrirá toda la acción del juego.



Ninguno de los elementos del juego puede salir nunca del tablero de juego.

Estado Inicial

El juego comenzará con una distribución de elementos prefijada y se detalla a continuación.

- **Jugador:** Comienza en el centro del campo (12, 7).
- **Snitch:** Comenzará en un punto aleatorio del campo, cuidando que la distancia al jugador sea mayor o igual a 10.
- **Bludgers:** Se encontrarán en las filas 2, 7, 12, 17 y 22, todas partirán desde la columna 0 y se moverán de izquierda a derecha y al llegar al borde derecho, volverán de derecha a izquierda.
- **Dementores:** Se dispondrán 4 en el campo, en las posiciones (5,3), (5,11), (19, 3) y (19, 11).

Resultado Esperado

El programa creado debe:

- Solicitar al usuario que ingrese un valor para cada una de las habilidades necesarias para determinar a qué casa pertenece el jugador. El comportamiento para esta parte debe ser exactamente igual al del trabajo práctico 1.
- Tomando los valores ingresados por el usuario, calcular la casa a la que pertenece y determinar su comportamiento particular.
- El juego deberá determinar de forma aleatoria, una posición para cada elemento, dentro del campo, teniendo en cuenta que el jugador comenzará en el centro del mismo, y ningún objeto puede comenzar en esa misma posición, ni en los casilleros lindantes.
- A cada *paso* del jugador, el programa deberá mover todos los elementos involucrados, según el criterio indicado para cada tipo en la sección anterior.
- Cuando el jugador logre atrapar la Snitch dorada, o se le terminen los movimientos, el juego finalizará, y deberá mostrarse un resultado para indicar si ganó o perdió.

Requerimientos complementarios

El juego se representará sobre una matriz de 25 filas x 15 columnas. Se requiere que se presente una visualización del tablero de juego similar a la que se muestra en la siguiente figura.

```

anibal@anibal-latitude: ~/fiuba/algol/2018-1C/tp2/resolution
0 |                                     P
1 |                                     P P P P P P P P
2 |                                 P P P P P P P P P
3 |                            P P P P P P P P P P
4 |                        B B B B B B B B B B B
5 |                    B B B B B B B B B B B B
6 |                B B B B B B B B B B B B
7 | B B B B B B B B B B B B B B
8 | B B B B B B B B B B B B B B
9 | - - - - - - - - - - - -
10| - - - - - - - - - - - -
11| - - - - - - - - - - - -
12| - - - - - - - - - - - -
13| D - - - - - - - - - - - -
14| - - - - - - - - - - - -
15| - - - - - - - - - - - -
16| B B B B B B B B B B B B B B
17| B B B B B B B B B B B B B B
18| B B B B B B X B B B B B
19| B B B B B B B B B B B B
20| B B B B B B B B B B B
21| E E E E E E E E E E E
22| E E E E E E E E E
23| E E E E E E E
24| E

*****
* INSTRUCCIONES *
* w: Arriba *
* a: Izquierda *
* s: Abajo *
* d: Derecha *
* q: Salir *
*****

¡Que te diviertas! ^^

*****
* Resistencia: 150 | Casa: Gryffindor *
* *
*****

```

En este caso, el espacio debajo de la Resistencia del jugador sirve para mostrar cualquier mensaje que sea necesario al usuario.

El entregable que deberá presentar deberá constar de 5 archivos fuente.

- sombrero.h
- sombrero.c
- quidditch.h
- quidditch.c
- control.c

Biblioteca Sombrero

Simplemente se deberá modificar el código del tp1 aprobado para que en vez de tener un programa principal (main), tenga una función

```
char obtener_casa();
```

además de su respectivo *guard-block*, que lea por teclado los atributos del jugador y determine la casa a la que pertenece.

Sugerencia:

Codifique las casas como letras para facilitar su comparación y permitir su uso en estructuras como *switch*, independientemente de que luego para mostrarla al usuario muestre el texto completo.

Biblioteca Quidditch

Esta biblioteca deberá implementar el funcionamiento del juego a partir de 4 funciones. Estas 4 funciones son las que deberán ir en el archivo de cabeceras *.h* que será público. Use la siguiente plantilla para su código.

quidditch.h

```
#ifndef __QUIDDITCH_H__
#define __QUIDDITCH_H__

#include <stdbool.h>

typedef struct juego {
    /* definición a cargo del alumno */
} juego_t;

typedef struct estado {
    bool juego_en_progreso;

    int snitch_fila;
    int snitch_columna;

    bool buscador_vivo;
    int buscador_fila;
    int buscador_columna;
    int buscador_resistencia;
```



```
    int cantidad_dementores;  
    int coordenadas_dementores[200][2];  
} estado_t;  
  
void inicializar_juego(juego_t *juego);  
bool mover(juego_t *juego, char input, char* mensaje);  
void mostrar_tablero(juego_t *juego, char* mensaje);  
void presentar_estado(juego_t *juego, estado_t *estado);  
  
#endif
```

`void inicializar_juego(juego_t *juego);`

Inicializará la estructura `juego_t` sobre la que se basará el comportamiento del juego. Dado que la estructura deberá definirla el alumno, lo que haga esta función queda a su mismo criterio. Sea coherente.

`bool mover(juego_t *juego, char input, char* mensaje);`

Deberá *actuar* un turno completo de juego. El parámetro **input** será un char leído de la entrada estándar. El usuario puede haber ingresado cualquier valor. Deberá validarlo. En caso de que haya algún problema (input inválido, por ej), deberá cargar el mensaje que lo advierta en la variable **mensaje** destinada a dicho fin.

`void mostrar_tablero(juego_t *juego, char* mensaje);`

Esta función deberá mostrar el estado del juego en el momento de su invocación. Utilice como guía la imagen incluida más arriba. Es imperativo que en caso de recibir un mensaje, sea mostrado.

`void presentar_estado(juego_t *juego, estado_t *estado);`

Esta función deberá traducir la estructura que haya elegido para **juego_t**, a la estructura de **estado_t** definida por enunciado. Cargue los campos con la información correspondiente a cada campo.

IMPORTANTE

No modifique la estructura **estado_t** por ningún motivo

Dentro del archivo de implementación de cada biblioteca, puede crear todas las funciones necesarias para que su código sea simple, completo, modular y consistente, pero no incluya dichas funciones en el archivo de encabezados (.h) ya que no formarán parte del *contrato* establecido para la biblioteca.

Compilación y Entrega

El trabajo práctico debe ser realizado en un archivo llamado juego.c, y la biblioteca de funciones para jugarlo quidditch.c y quidditch.h, y debe poder ser compilado sin errores con el comando:

```
gcc -o quidditch -std=c99 -Wall -Wconversion -Werror sombrero.c control.c  
quidditch.c -lm
```

Por último debe ser entregado en la plataforma de corrección de trabajos prácticos Kwyjibo en la cual deberá aparecer con la etiqueta successful.

Para la entrega en Kwyjibo, recuerde que deberá subir un archivo zip conteniendo únicamente los 5 archivos antes mencionados, sin carpetas internas ni otros archivos. De lo contrario, la entrega no será validada por la plataforma.

Bibliografía