



**FACULTAD  
DE INGENIERIA**  
Universidad de Buenos Aires

**Algoritmos y Programación I**  
**Curso Mendez**

## **TP2 – Snitch Dorada**



<b>Fecha Presentacion</b>	<b>26/04/2018</b>
<b>Fecha Entrega</b>	<b>24/05/2018</b>

## Introduccion

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 distintos aspectos 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, además de la creación de bibliotecas.

Por supuesto, se requiere que el trabajo cumpla con las buenas prácticas de programación profesadas por la cátedra y 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 que intentarán 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 campo.
- Se contará con una resistencia inicial de 50 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, sin la necesidad de mover los demás elementos del campo.
- 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 jugador de Gryffindor está en la misma posición que un dementor, el dementor desaparece.
- **Slytherin:** Gracias a su ingenio, pueden esquivar las Bludgers cuando éstas los intentan golpear, esto quiere decir que, si un jugador de Slytherin y una Bludger están en la misma posición, el jugador no se ve afectado por ella.
- **Ravenclaw:** Debido a su gran inteligencia y estudio, lograron aprender a atrapar la Snitch no solo si están en la misma posición, sino también si ésta 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 los demás elementos.
- Para determinar a qué casa pertenece el jugador, se debe adaptar el TP1 para ser utilizado como una biblioteca (*sombrero.h*) con una función llamada **obtener\_casa** que pida los atributos y determine la casa a la que el jugador pertenece, ésta no recibirá parámetros y deberá devolver la inicial en mayúsculas de la casa a la que pertenece el jugador.

## Snitch Dorada

Tiene un movimiento predecible, siempre se aleja del jugador hacia la posición que la deje lo más lejos posible, sus movimientos sólo serán en diagonal, y ante igualdad de distancia entre el punto final del jugador, priorizará, izquierda antes que derecha y arriba antes que abajo.

Si la Snitch se encuentra a una distancia mayor a 10, no se moverá ya que se siente segura, cabe aclarar que, si la distancia es menor o igual a 10, se moverá siempre, siempre y cuando el movimiento no haga que la atrapen, en ese caso, se quedará quieta.

Si la Snitch se encuentra en los bordes del campo, elegirá el movimiento que la aleje, si es que existe, tomando como prioridad izquierda antes que derecha y arriba antes que abajo y si no, se quedará quieta, nunca se moverá para posarse en la misma posición que el jugador.

Si la posición que más aleja a la Snitch del jugador está ocupada por otro elemento, la Snitch priorizará siempre ir hacia arriba.

La Snitch no se ve afectada por los dementores ni por las Bludgers ya que los esquivará.

## Dementores

Los dementores tienen la capacidad de robarle toda la energía al jugador, por la que si un jugador se posa sobre un dementor, pierde el juego.

Estos no tendrán movimiento y se encontrarán en puntos específicos del campo.

## Bludgers

Las Bludgers nunca se quedan quietas, en el campo habrá 5 filas por las que pasará una 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 ultimo pierde 10 movimientos y la Bludger desaparece.

Si en la posición que debe ocupar la Bludger está la Snitch, lo ocupará, pero no será visible en la matriz del campo (tiene prioridad la Snitch), sí deberán ser actualizadas las coordenadas de la Bludger en el registro de estado del juego.

## Campo de Juego

Será representado como una matriz de 25 filas x 15 columnas.

Ningún elemento puede moverse fuera de los límites del campo.

## 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 5.

**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).

## Orden de Movimientos

El primero en moverse será el jugador, luego las Bludgers y por último la Snitch.

Luego del movimiento de cada elemento, debe actualizarse el estado del juego.

Esto significa que, si el jugador se mueve y se posa sobre la Snitch, gana el juego, sin importar hacia donde se iba a mover la Snitch o si un Bludger en su próximo movimiento hacía perder la resistencia al jugador.

Así mismo, si una Bludger se posa sobre el jugador y éste se queda sin movimientos, pierde el juego, sin necesidad de mover la Snitch.

## 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.
- Se deberá mostrar el campo de juego con todos los elementos distribuidos como se explica en la sección anterior.
- 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 o se choque con un dementor, el juego finalizará, y deberá mostrarse un resultado para indicar si ganó o perdió.

## Aclaraciones complementarias

### Sobre la representación de los elementos:

Deberá representarse al jugador con una 'J', a la Snitch con una 'S', a los dementores con una 'D', a las Bludgers con una 'B' y los casilleros sin elementos con un espacio en blanco ' ' dentro de la matriz del campo de juego.

### Sobre el campo de juego:

El campo de juego tendrá su posición inicial (0,0) en la posición más arriba a la izquierda de la pantalla, por lo que, en una misma columna, las posiciones que están 'más arriba' son las de menor valor, y las que están 'más abajo' son las de mayor valor. Del mismo modo, para una misma fila, las posiciones 'más a la izquierda' son las de menor valor y las de 'más a la derecha' son las de mayor valor.

### Sobre el posicionamiento de los elementos en campo de juego:

Si el jugador y la Snitch quedan en el mismo lugar, deberá mostrarse el jugador y no la Snitch, así mismo con las Bludgers ya que éstas, al golpear al jugador desaparecen y el jugador sigue buscando la Snitch. En caso de ocupar el mismo lugar el jugador y un dementor, quien prevalece es el dementor ya que en ese caso el que pierde es el jugador.

Sobre el movimiento de la Snitch:

La Snitch nunca podrá posarse sobre otro elemento, siempre los esquivará y descartará ese movimiento aunque sea el que más lo aleje del jugador.

La distancia entre la Snitch y el jugador se calcula como la suma de los módulos de las restas entre filas y columnas de ambos elementos (también llamada Distancia Manhattan o Distancia del Taxista, abajo encontrarán la referencia):

**Ejemplo 1:**

La Snitch se encuentra en la posición (2,2) y el jugador en la posición (1,1)

La Snitch podría moverse, en diagonal a las posiciones (1,1), (1,3), (3,1) y (3,3).

La distancia del jugador a cada una de esas posibles posiciones es:

$$\text{De (1,1) a (1,1)} = |1 - 1| + |1 - 1| = 0$$

$$\text{De (1,3) a (1,1)} = |1 - 1| + |3 - 1| = 2$$

$$\text{De (3,1) a (1,1)} = |3 - 1| + |1 - 1| = 2$$

$$\text{De (3,3) a (1,1)} = |3 - 1| + |3 - 1| = 4$$

*La Snitch debe elegir moverse a la posición (3,3).*

**Ejemplo 2:**

La Snitch se encuentra en la posición (1,2) y el jugador en la posición (14,10)

La Snitch podría moverse, en diagonal a las posiciones (0,1), (0,3), (2,1) y (2,3), pero la distancia actual de la Snitch al jugador es:

$$\text{De (1,2) a (14,11)} = |1 - 14| + |2 - 10| = 21$$

*La Snitch debe elegir quedarse en el lugar porque se siente segura.*

**Ejemplo 3:**

La Snitch se encuentra en la posición (5,8) y el jugador en la posición (5,10)

La Snitch podría moverse, en diagonal a las posiciones (4,7), (4,9), (6,7) y (6,9).

La distancia del jugador a cada una de esas posibles posiciones es:

$$\text{De (4,7) a (5,10)} = |4 - 5| + |7 - 10| = 4$$

$$\text{De (4,9) a (5,10)} = |4 - 5| + |9 - 10| = 2$$

$$\text{De (6,7) a (5,10)} = |6 - 5| + |7 - 10| = 4$$

De (6,9) a (5,10) =  $|6 - 5| + |9 - 10| = 2$

La Snitch debe tener 2 posiciones a la que puede ir, en este caso, en ambas opciones conserva la columna y se mueve sobre la fila y por prioridad elige ir hacia arriba, por lo que la posición a la que se mueve es la (4,7).

## Entrega

Importante: En el presente enunciado hemos puesto los números en las longitudes de los vectores y matrices, éstos deben ser reemplazados por constantes en el trabajo.

El entregable deberá constar de 5 archivos fuente.

- sombrero.h
- sombrero.c
- quidditch.h
- quidditch.c
- juego.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.

Como se explicó más arriba, la función deberá devolver la inicial de la casa correspondiente (G – Gryffindor, S – Slytherin, R – Ravenclaw, H - Hufflepuff).

## Registros Predefinidos

Se deberán utilizar indefectiblemente los siguientes registros:

```
typedef struct coordenada{
    int fila;
    int columna;
}coordenada_t

typedef struct juego{
    coordenada_t posicion_jugador;
    char casa;
    int resistencia;
```



```
coordenada_t posicion_snitch;  
coordenada_t posicion_bludgers[5];  
char direccion; // 'D': derecha - 'I': izquierda  
coordenada_t posicion_dementores[4];  
int estado; // 0: en ejecución - 1: jugador ganó - 2:  
jugador perdió  
}juego_t
```

## Biblioteca Quidditch

Deberán implementarse las siguientes funciones:

```
void inicializar_juego(char campo[25][15], char casa, juego_t  
*juego);
```

Este procedimiento inicializará el campo de juego, posicionando todos los elementos en la matriz y cargando los datos en el registro del juego.

```
char pedir_movimiento(coordenada_t posicion_jugador);
```

Deberá solicitar al usuario un movimiento y validar que sea correcto, como resultado de la función se tendrá un caracter válido (w, s, a o d).

Recibirá las coordenadas del jugador para evitar que si está en un borde, tome como válido un movimiento que lo lleve fuera del campo, es decir, si esta en el borde inferior y el usuario presiona 's', no es un movimiento válido.

```
void mover_jugador(char campo[25][15], char direccion, juego_t  
*juego);
```

El procedimiento actualizará el campo de juego, moviendo al jugador y todos los elementos que sea necesario actualizar en la matriz del campo. También deberá actualizarse el registro con los datos del estado del juego.

La dirección a moverse será un movimiento válido.

```
void mover_snitch(char campo[25][15], juego_t *juego);
```

El procedimiento actualizará el campo de juego, moviendo la Snitch. También deberá actualizarse el registro con los datos del estado del juego.

```
void mover_bludgers(char campo[25][15], juego_t *juego);
```

El procedimiento actualizará el campo de juego, moviendo todas las Bludgers. También deberá actualizarse el registro con los datos del estado del juego.

```
coordenada_t posicion_inicial_snitch();
```

Devolverá las coordenadas donde se posicionará la Snitch al comenzar la partida. Esas coordenadas deben cumplir con el requisito de ser aleatorias y a una distancia mayor o igual a 5 del centro del campo.

```
bool finalizo_juego(juego_t *juego);
```

Devolverá true o false si la partida terminó, ya sea porque el jugador ganó o perdió. Así mismo actualizará el estado del juego, seteando las variables que indican a que se debió la finalización de la partida.

```
void mostrar_resultado(juego_t juego);
```

Mostrará por pantalla el resultado del juego.

## Programa Principal: juego.c

El programa principal manejará el flujo del juego, invocando a las funciones de las bibliotecas en el orden necesario para el correcto funcionamiento del juego.

## Preentrega

Todos los alumnos estarán habilitados, y recomendamos que así lo hagan, a realizar una preentrega el día **10/05/2018** de los procedimientos **mover\_jugador**, **mover\_snitch** y **mover\_bludgers**.

## Compilacion y Entrega

El trabajo práctico debe ser realizado en un archivo llamado **juego.c**, y las bibliotecas de funciones para jugarlo Quidditch en **quidditch.c** y **quidditch.h** y Sombrero en **sombrero.c** y **sombrero.h**, es imprescindible respetar el nombrado de archivos.

El trabajo práctico consiste en esos 5 archivos y debe ser compilado con la siguiente línea **sin errores**:

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

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

## Bibliografía

<http://es.harrypotter.wikia.com/wiki/Quidditch>

<http://es.harrypotter.wikia.com/wiki/Dementor>

[https://es.wikipedia.org/wiki/Geometr%C3%ADa\\_del\\_taxista](https://es.wikipedia.org/wiki/Geometr%C3%ADa_del_taxista)