

Universidad de Puerto Rico
Recinto Universitario de Mayagüez
Departamento de Ingeniería Eléctrica y de Computadoras

ICOM 4015: Programación Avanzada

**Programa 4 – Dominó
(Vence 16-mayo-2001)**

En este programa complementarás la implementación de un juego de dominó computadorizado. La implementación utiliza diseño “object oriented” y clases para representar los componentes principales del juego tales como: los jugadores, las fichas en mano y las fichas jugadas. Los jugadores pueden ser computadorizados o externos al programa, en cuyo caso pueden ser humanos u otros programas. En la versión que recibirás del programa, cada jugador computadorizado mueve las fichas utilizando una estrategia extremadamente simple; jugar la ficha con el mayor valor numérico o pasar si no puede jugar ninguna ficha. El jugador externo mueve las fichas según sea indicado en el input provisto por el usuario.

Al inicio del juego es necesario determinar cuantos jugadores habrá de cada clase. Lo jugadores siempre tienen un número o posición asociado que va desde el 0 al 3. El programa inicia el juego repartiendo 7 fichas al azar a cada jugador. El jugador con el doble 6 comienza cada juego. Los demás jugadores le siguen en orden numérico modulo 4. Esto es 1->2->3->0->1 ...

Para entender las siguientes instrucciones será indispensable leer y entender los source files de los módulos que componen el programa (Ver sección sobre detalles logísticos mas adelante).

Parte A –Programador Ilustre (33%)

Modifica la estrategia del jugador computadorizado (método **ComputerPlayer::move**) para que éste juegue el doble de mayor valor numérico de tener doble. De lo contrario el jugador debe jugar la ficha de mayor valor numérico.

Parte B –Programador Mostro (33%)

Completa la definición de **ComputerPlayer::apply** para que todo jugador computadorizado registre, para cada uno de los demás jugadores (computadorizados o humanos) los números a los cuales este último jugador ha pasado (arreglo passed de **PlayerInfo**) así como la cantidad de fichas de cada número que ha jugado (arreglo **frequency** de **PlayerInfo**) y el número de fichas que el jugador a tirado.

Modifica el módulo principal para que cada jugador se informe sobre que fichas se van jugando invocando el método **Player::apply**. Nota que los jugadores externos guardan su información externamente de manera que no hay necesidad que de **HumanPlayer::apply** realice ningún trabajo.

Parte C –Programador de Titanio (34%)

Declara una nueva clase de jugador, llamada **TitaniumPlayer**, derivada de **ComputerPlayer** que juegue las fichas de forma que cuadre el juego a un numero X siempre que tenga la oportunidad y el siguiente jugador haya pasado a X anteriormente. No debe cuadrar el juego si su compañero ha pasado a X anteriormente. Si no puede cuadrar el juego, el jugador debe jugar como juega un **ComputerPlayer** modificado según la parte A.

Para probar tu programa

Muestra el resultado de varios juegos de dominó que ilustren que tus estrategias funcionan correctamente. Por favor, trata de economizar papel editando el archivo de output que vas a entregar removiendo detalles irrelevantes. No más de 5 páginas de output por favor. Utiliza la hoja de evaluación incluida como cubierta para tu programa.

Detalles logísticos

Todos los archivos que necesitas para completar esta asignación están accesibles desde la página de web del curso

<http://www.ece.uprm.edu/~bvelez/courses/spring2001/icom4015/icom4015.htm>

Utiliza el comando make para compilar tu programa en un módulo ejecutable llamado **dominoes**.

```
> g++ ListOfDomino.cc main.cc Move.cc Table.cc ComputerPlayer.cc HumanPlayer.cc  
    utils.cc -o dominoes
```

Recuerda, tu trabajo tiene que ser independiente y original. Puedes consultar con tus compañeros de clase, pero evita mostrar tu código así como mirar el código de tus compañeros. Al comienzo de la clase en la fecha de vencimiento debes presentar una copia de todos los archivos que modificaste, así como copia de un archivo que muestre varios escenarios de interacción con tu programa. Antes de pensar en entregar tarde tu programa, lee cuidadosamente la política de entregas tardías en el prontuario del curso. Esta política se ejercerá estrictamente en este tercer programa.

**Universidad de Puerto Rico
Recinto Universitario de Mayagüez**

ICOM 4015 - Programación Avanzada

**Programa 4 – Dominó
Hoja de Evaluación**

Nombre: _____

Sección: _____

Parte A. ComputerPlayer::move method

	Valor	Obtuvo
Funcionalidad	10	+ _____
Modularidad	10	+ _____
Eficiencia	7	+ _____
Estilo	6	+ _____
	SUBTOTAL	_____/33

Parte B. ComputerPlayer::apply method

	Valor	Obtuvo
Funcionalidad	10	+ _____
Modularidad	10	+ _____
Eficiencia	7	+ _____
Estilo	6	+ _____
	SUBTOTAL	_____/33

Parte C. Titanium player class

	Valor	Obtuvo
Funcionalidad	10	+ _____
Modularidad	10	+ _____
Eficiencia	7	+ _____
Estilo	7	+ _____
	SUBTOTAL	_____/34

Penalizaciones:

TOTAL _____/100