TEL-102 Sem. de Programación
2º Semestre 2021
Control 2
27 de Octubre de 2021

Instrucciones

- Lea con detenimiento y desarrolle **individualmente** cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con compreso **.rar**, **.zip** o **.tar.gz** con lo desarrollado. El nombre del archivo debe tener el siguiente formato: **TEL102_C2_Nombre_Apellido.tar.gz** sin incluir tíldes. Este archivo debe incluir una imagen con formato **.png o .jpg** y un código **.cpp** siguiendo la misma nomenclatura: **TEL102_C2_Nombre_Apellido.png** y **TEL102_C2_Nombre_Apellido.cpp**. (Ej. TEL102_C2_Nicolas_Galvez.tar.gz).
- Enviar el archivo a través de la página de aula del ramo, sección "Control 2" hasta las 19:55:00 del día de **hoy**, Miércoles 27/10/2020, Hora continental de Chile (UTC-3).
- Cada minuto de atraso tendrá un descuento siguiendo la serie de Fibonacci.
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online**.
- Sea riguroso con las instrucciones de desarrollo.

PukaLOL

El videojuego del momento en *PlusPlusCity* es **PukaLOL**, un MOBA (Multiplayer Online Battle Arena), en el cual dos (2) equipos se enfrentan por la victoria. Este videojuego cuenta con un sistema de creación de partidas o **MatchMaking** en el cual a cada equipo se le asignan cinco (5) jugadores y estos eligen a su Pukamon favorito para la batalla.

La empresa desarrolladora del juego, **PIMI**, lo contrata a ud para implementar una versión básica del juego en C++. Por esto le entrega una porción de código, ya implementado, llamado **matchmaking.cpp**.

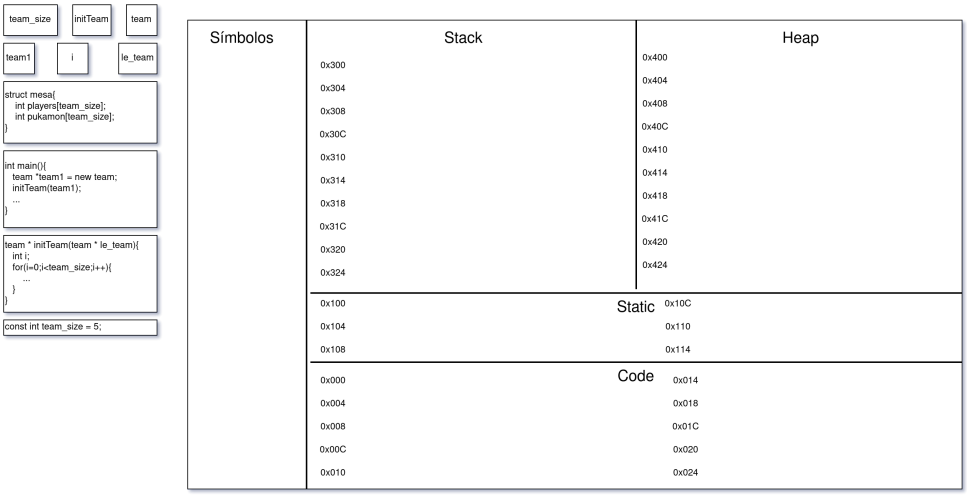
En este código, a través de structs y punteros, se inicializan ambos equipos. Sin embargo, resta implementar la asignación de jugadores y la selección de Pukamon. Existe una estructura llamada `team` que contiene dos atributos: `int players[team_size]`, un arreglo de enteros que contiene los identificadores de los cinco miembros del equipo, e `int pukamon[team_size]`, un arreglo de enteros que contiene los ids de los pukamones escogidos por cada uno de los identificadores. Note que los identificadores de jugador están en el rango $[1, 100000000]$ y que los identificadores de pukamon están en el rango $[1, 898]$. Asuma que nunca se ingresaran dos identificadores de usuario iguales, por lo tanto NO DEBE verificar aquello.

matchmaking.cpp

```
#include <iostream>

const int team_size = 5;

struct team{
        int players[team_size];
        int pukamon[team_size];
};

void initTeam(team *le_team){
        int i;
        for(i=0;i<team_size;i++){
                (*le_team).players[i] = -1;
                le_team->pukamon[i] = -1;
        }

}

//void pickUp(team *le_team){}//descomentar e implementar

//void printTeam(team *le_team, const char *nom){} //descomentar e implementar


int main(){
        team *team1 = new team;
        initTeam(team1); //dibujar este momento

        team *team2 = new team;
        initTeam(team2);

//      pickUp(team1); //descomentar para probar
//      pickUp(team2);

//      printTeam(team1, "OverHeat"); //descomentar para probar
//      printTeam(team2, "Elite Five");

        delete team1;
        delete team2;
        return 0;
}
```

Su nuevo empleador le solicita:

(a) (50pts) Dibujar la asignación de memoria: code, static, stack y heap; justo despues de ejecutar la linea `initTeam(team1);`. Use como base el siguiente esquema de memoria:



Tendrá dos opciones para realizar esto:

1. Podrá utilizar el archivo `memory-scheme.drawio`, disponible en AULA USM, y editarlo digitalmente en la plataforma https://draw.io.
2. Podrá dibujar a mano el esquema de memoria y digitalizarlo a través de una foto.

Deberá diferenciar claramente entre elementos que están siendo utilizados en memoria de elementos utilizados pero ya liberados. Para el primer tipo, debe usar **bordes y flechas continuas** y, para el segundo tipo, **bordes y flechas segmentadas**.

(b) (30pts) Implemente la función `void pickUp(team *le_team())`, la cual recibe como parámetro un arreglo dinámico del tipo `team` llamado `le_team` que representa a un equipo, y no retorna valores. La funcion debe solicitar por entrada estándar los datos de identificador de usuario e identificador de pukamon de cada miembro del equipo en cuestión y asignarlos al arreglo entregado por parámetro.

(c) (20pts) Implemente la función `void printTeam(team *le_team, const char *nom)`, la cual recibe como parámetros un arreglo dinámico del tipo `team` llamado `le_team` que representa a un equipo, y un string `nom`. La función debe imprimir el nombre del equipo y como está compuesto.

Para mayor referencía de las funciones solicitadas, guíese por el siguiente ejemplo.

Salida (consola)

```
[elprofe@tel102 control2]$ ./MATCHMAKING
Escogiendo equipo...
Ingrese su id de usuario PukaLOL: 10
Ingrese el id de su Pukamon escogido: 1
Ingrese su id de usuario PukaLOL: 20
Ingrese el id de su Pukamon escogido: 2
Ingrese su id de usuario PukaLOL: 30
Ingrese el id de su Pukamon escogido: 3
Ingrese su id de usuario PukaLOL: 40
Ingrese el id de su Pukamon escogido: 4
Ingrese su id de usuario PukaLOL: 50
Ingrese el id de su Pukamon escogido: 5
Equipo listo...
Escogiendo equipo...
Ingrese su id de usuario PukaLOL: 777
Ingrese el id de su Pukamon escogido: 555
Ingrese su id de usuario PukaLOL: 666
Ingrese el id de su Pukamon escogido: 444
Ingrese su id de usuario PukaLOL: 555
Ingrese el id de su Pukamon escogido: 333
Ingrese su id de usuario PukaLOL: 444
Ingrese el id de su Pukamon escogido: 150
Ingrese su id de usuario PukaLOL: 333
Ingrese el id de su Pukamon escogido: 151
Equipo listo...
El equipo OverHeat está conformado por:
Player: 10. Pukamon: 1
Player: 20. Pukamon: 2
Player: 30. Pukamon: 3
Player: 40. Pukamon: 4
Player: 50. Pukamon: 5
El equipo Elite Five está conformado por:
Player: 777. Pukamon: 555
Player: 666. Pukamon: 444
Player: 555. Pukamon: 333
Player: 444. Pukamon: 150
Player: 333. Pukamon: 151
```

(d) (10pts) Bono: Explique brevemente la siguiente imagen. Sea explícito con las caracterísitcas de lo que ud. cree que sucede. Responda a éste desafio como un comentario despues de la función main del código a entregar.