Projeto Prático – 2023/2024
Gestão de Casinos

OBS: Este tipo de problema pode ser usado em várias situações (Controlo de ESCOLAS, Centralina de um automóvel, Sistema de segurança/gestão de uma fábrica, etc.)

Pretende-se um programa que faça a Simulação da gestão de um casino. Um casino é frequentado por pessoas (Jogadores) que podem jogar nas diversas máquinas de diversão.

Cada máquina do casino tem estados típicos: OFF, ON, AVARIADA.

Para tornar o problema mais próximo da realidade, devem ter um ficheiro em XML com a configuração do casino e com todo o conteúdo a gerir.

Uma conversa sobre o programa!!!!

O programa deve fazer a simulação com vários jogadores na sala e com várias máquinas em funcionamento, os jogadores devem ir entrando na sala de um modo aleatório com uma dada cadência.

Entrando um cliente no casino, este vai escolher uma máquina para jogar (de entre as que não estejam em utilização – estado OFF). Senta-se e joga o jogo dessa máquina. Cada máquina tem um jogo específico.

O Gestor do casino a todo o momento pode carregar na tecla 'M' para ir ao menu e fazer alguma das operações abaixo definidas!

Quando um jogador está a jogar o programa não deve ficar parado para os outros jogadores!... Digamos que o Run está num ciclo infinito onde passa por cada jogador e o faz avançar no jogo!, **Não se pretende que os alunos implementem com Threads**!

Os alunos devem implementar no mínimo 4 tipos de máquinas de jogos, o casino pode ter mais do que uma máquina do mesmo tipo a funcionar.

Fazendo uma pesquisa na Net… encontraram-se vários tipos de máquinas de um casino

Nome do jogo	**Breve descrição**
Classic Slots	Têm três rolos e símbolos como frutas, barras, sinos e o número sete
Slots temáticos	são projetados em torno de um tema específico, como Egito Antigo ou Sorte irlandesa
Slots de recursos	incluem ações de jogos paralelos, como jogos de adivinhação ou rodadas grátis com multiplicadores de ganhos
Jogos de caça-níqueis licenciados	são baseados em franquias de mídia populares, como filmes ou programas de televisão
Blackjack	Um jogo de cartas popular onde o objetivo é ter um valor de mão próximo de 21 sem ultrapassar
Roleta	Os jogadores apostam onde a bola irá parar em uma roda giratória
Craps	Um jogo de dados em ritmo acelerado onde os jogadores apostam no resultado do lançamento de dois dados
Bacará	Um jogo de cartas simples, popular na Europa e na América Latina.
Variantes de Poker	Variantes de pôquer (por exemplo, Omaha, Texas Hold'em): os jogadores recebem uma mão de cartas e miram para criar a mão de classificação mais alta

Considerações:

- O Casino tem uma hora de abertura e hora de encerramento;
- Quando existe algum alerta/aviso (da parte das máquinas) o gestor do Casino deve ser logo informado, para desse modo atuar!; Fazer função para reparar maquina
- Cada Máquina de jogo está numa dada posição (X, Y) do Casino (não pode estar duas ou mais máquinas na mesma posição);
- Se uma dada máquina está com uma percentagem muito grande de "dar" prémios, deve ser enviado um alerta ao Gestor do Casino, para este alterar as configurações.
- As máquinas devem ter um sensor de temperatura, pois podem começar a aquecer!, no caso de aquecimento deve ser alertado o Gestor do Casino. Meter a temperatura como deve ser
- Todas as Máquinas têm uma probabilidade de avaria!
- Qualquer alteração do estado de uma máquina deve ser comunicado ao Gestor do Casino.

Meter horas a funcionar por hora:minuto:segundo

Funcionalidades Gerais:

- Construtor do Casino, sendo dado o nome do casino;.
  Casino::Casino (string nome)
  ~~O adicionar não esta a verificar bem o y~~ e talvez ver a parte de adicionar do ficheiro xml

- As configurações do Casino dadas em ficheiro XML, com todas as informações.
  bool Casino::Load(const string &ficheiro)
  Adiciona users random da lista user.txt
  Jogador entra no casino
  Falta: Adicionar maquina ao jogador jogar maquina

- Adicionar Utilizadores/Jogadores
  bool Casino::Add(User *ut)

- Adicionar Máquina
  bool Casino::Add(Maquina *m)
  Crud - Maquina
  -Add maquina
  -Remove maquina
  -Edit Maquina
  Criar função mover máquina

- Listar o estado atual do Casino;
  void Casino::Listar(ostream &f = std::cout)
  Completar todas as maquinas e seus estados numero total de jogadores no casino

- Desligar uma dada máquina, dado o seu ID;
  void Casino::Desligar(int id_maq)
  ~~Se maquina avariar, jogadores saltam da lista de espera, e caso alguém esteja a jogar sai~~

- Saber o estado de uma dada Máquina, dado o seu ID;
  ESTADO_MAQUINA Casino::Get_Estado(int id_maq)

- Calcular a memória total ocupada pela estrutura de dados;
  int Casino::Memoria_Total()
  Devolver trocar fichas por dinheiro quando sai
  Caso calhe o user que já tenha entrada no casino ele digamos que atualiza

- Listar e devolver todas as máquinas de um dado Tipo;
  list<Maquina *> *Casino::Listar_Tipo(string Tipo, ostream &f = std::cout)

- Quais as Maquinas que mais avariam, deve devolver uma lista (ordenada) da que mais avaria para a mais fiável;
  list<string> * Casino::Ranking_Dos_Fracos() ver melhor o sort

- Quais máquinas mais usadas (que mais trabalham), deve devolver uma lista (ordenada);
  list<Maquina *> * Casino::Ranking_Das_Mais_Trabalhadores()
  variavel uso, que incrementa sempre que inicia uma rodada

- Quais os jogadores que mais TEMPO passaram no casino a jogar, deve devolver uma lista (ordenada);
  list<User *> * Casino::Jogadores_Mais_Frequentes ()
  variavel tempo, que guarda o tempo quando joga e quando saiu faz conta para ir buscar o tempo total que lá passou

- Quais os jogadores que mais PRÉMIOS ganharam no casino, deve devolver uma lista (ordenada);
  list<User *> * Casino::Jogadores_Mais_Ganhos ()

- Enviar um relatório em XML, do estado do Casino; O relatório deverá ter a informação do estado atual de cada máquina nesse dia;
  void Casino::Relatorio(string fich_xml)
  Quando casino fecha gerar também relatório

- Quando uma máquina tem um prémio, devem ser aumentada a probabilidade de ganho das outras máquinas que estão em redor dela, à distância máxima de R. O método deve devolver (por parâmetro) a lista das máquinas onde foi feita a alteração da probabilidade de ganhar;

void Casino::SubirProbabilidadeVizinhas(Maquina *M_ganhou, float R, List<Maquina *> &lmvizinhas)

Melhorar codigo

- Listar todas as máquinas onde a probabilidade de ganhar é superior a X. Esta lista pode ser mostrada no monitor ou escrita em ficheiro;

- **void Casino::Listar(float X, ostream &f = std::cout)**

- O Casino tem um método Run, que coloca todo o processo em funcionamento! O Simulador deve estar sempre a correr e quando se pretende introduzir alterações, deve-se carregar numa tecla 'M' de modo a aparecer um menu (nesse instante o simulador deve estar parado, até a opção ser executada!)

- **void Casino::Run(bool Debug = true)**

Adicionar vizinho
Remover vizinho
Editar vizinho

Observações:

- Podem e devem criar novos métodos auxiliares (privados) com os nomes que acharem por bem! E chamar nos métodos principais (públicos);
- Sempre na metodologia orientada a objetos;

Avaliação / Observações:

- Se a gestão da memória, não estiver correta, haverá uma penalização de 5 valores;
- Se for detetado "copianço", trabalho anulado!
- Os grupos são constituídos até 3 alunos;
- O código deve estar comentado;
- Eventuais dúvidas serão esclarecidas pelos docentes da disciplina.

Entrega:

- **16/12/2023** para avaliação em época normal (a avaliação ocorre no período de compensação)
- **02/02/2024** para avaliação em época recurso;