



UNIVERSIDADE DE SÃO PAULO - ICMC
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0220 “Lab. ICC2” - 2ºSem/2020

Exercício 05 - Pokebox

Entrega: Olhar run.codes

Tarefa:

Você deve criar um programa que leia uma série de inteiros e, para cada entrada, realizar diferentes operações sobre uma recriação da famosa box de PC do jogo Pokémon!

Para isso, você deve contar com a ajuda da biblioteca de Pokémon que está disponível no GitHub (e como arquivos auxiliares nos projetos) <https://github.com/LeonardoTPereira/PokeCClone>

Você deverá fazer uma estrutura de dados do tipo **lista duplamente encadeada circular** para armazenar uma lista de “boxes” (caixas/telas do PC do pokémon, que armazenam diversos pokémon).

Quanto ao conteúdo de cada caixa, já está definido no código. Podem usar um vetor mesmo, conforme definido nos arquivos “PokeBox.c” e “PokeBox.h”. As caixas são vetores de tamanho 9, mas com uma estrutura interna já preparada para imprimir seu conteúdo como uma matriz 3x3

Em “Pokebox.h”:

```
#define BOXLINES 3  
#define BOXCOLLS 3  
#define BOXSIZE BOXLINES*BOXCOLLS
```

Para detalhes da impressão, conferir a função *printBoxLine* em “SpriteManager.c”

Para manipular o conteúdo das caixas e a própria estrutura da lista de caixas, você deverá ler diversas entradas que representam diferentes operações.

Primeiramente, deve-se iniciar um índice da caixa atual como a caixa 0 (primeira da lista) e um índice que indica o Pokémon atual dentro da caixa também como 0 (primeiro do vetor). Sempre mantenha esse valor atualizado dentro do loop de leitura de entradas! É como se fosse o cursor do jogador durante o jogo :)

Também deve-se criar previamente a lista de caixas com 3 caixas. Ela pode ser expandida posteriormente ou ter caixas removidas!

Todas as entradas são números inteiros (“%d”). As entradas consistem na adição ou remoção de Pokémon, adição de caixas, mover a caixa atual para a da direita ou esquerda, imprimir a caixa atual, e terminar a leitura.

Cada ação tem os seguintes códigos mapeados (lembre de fazer uma estrutura robusta para manter esses códigos, como uma sequência de defines):

- 1 - Ir para a caixa à **direita**
- 1 - Ir para a caixa à **esquerda**
- 2 - Adicionar uma nova caixa VAZIA ao final da lista
- 2 - Remover a caixa atual
- 3 - Adicionar Pokémon na posição 0 da caixa atual
- 4 - Adicionar Pokémon na posição indicada
- 4 - Remover Pokémon na posição indicada
- 5 - Imprimir caixa atual*
- 0 - Parar a leitura de entradas e terminar o programa

Movimentação da caixa (-1 ou 1):

Você deve simplesmente ir para o próximo elemento da lista no caso de mover à direita, ou ao anterior, caso seja mover para a esquerda. Lembre-se que estamos usando uma lista duplamente encadeada circular. Lembre-se de atualizar o índice atual da caixa, pois é o que simula a visão/cursor do jogador.

Adição de caixa (2):

A caixa deverá ser adicionada ao FIM da lista. Lembre de guardar o tamanho atual da lista para saber seu começo e fim. Lembre-se de SEMPRE inicializar as caixas com todos os Pokémon sendo o valor NULL

Remoção de caixa (-2):

A remoção será da caixa ATUAL. Após isso, a próxima caixa a ser apontada deve ser a próxima da lista! Não precisam tratar erros de remover mais caixas que existem ou ficar sem caixas. Não altere o índice do pokémon atual!

Impressão de caixa (5):

Para imprimir a caixa, apenas passe a caixa atual para a função *printBox*, definida em "PokeBox.c" e "PokeBox.h". Ela cuidará de toda a leitura dos arquivos .asc contendo os sprites dos Pokémon e os espaços em branco. Só lembre de SEMPRE inicializar suas caixas com NULL :)

Adição de Pokémon (3 ou 4):

Para adicionar um Pokémon, temos 2 operações diferentes, mas BEM similares. Para ambas você deverá ler um inteiro ("%d") que é o número do Pokémon na National Dex (caso você não saiba o que é isso, tudo bem, o programa vai tratar disso para você). Também para ambas o processo de adição é relativamente simples: use a variável "pkdex" declarada na *main.c*, que já terá carregado as informações de TODOS os Pokémon de acordo com seu número na dex e copie o ponteiro praquela Pokémon para o índice atual da caixa :) Um exemplo disso pode ser visto (e rodado caso use o make debug) na própria *main.c*

Em ambas as opções, o índice atual do pokemon na caixa deve ser mantido como o índice do Pokémon que foi inserido.

Caso seja a opção 3, para adicionar no índice 0 da caixa atual, apenas tente adicionar o Pokémon na posição 0 da caixa atual. Na opção 4, para adicionar em uma

posição específica, deve-se ler a posição como um inteiro ("%d"). Não precisa verificar se ela vai fugir do tamanho da caixa, não vai. E vai ser na caixa atual, sempre! Então, chame a função para adicionar na posição passada ao invés do índice 0 :)

CUIDADO!

Caso você tente adicionar um Pokémon em um índice da caixa já ocupado, você deverá ir para o próximo Pokémon da caixa, até achar um local livre. Caso a caixa toda esteja lotada A PARTIR DO ÍNDICE QUE VOCÊ COMEÇOU (ou seja, não circule de volta para o início da caixa), vá para a próxima caixa, MAS NÃO CIRCULE DE VOLTA PARA O COMEÇO! Caso você chegue na última caixa, crie uma nova e coloque o Pokémon novo na primeira posição dela :)

Remoção de Pokémon (-4):

Você deverá ler um inteiro ("%d") que é o índice do Pokémon na caixa atual. Caso exista um Pokémon lá, remova ele (atribua NULL à posição!). Caso não exista, não faça nada. Deixe como está.

Por fim, você deve submeter no run.codes um **.zip** com os **códigos utilizados** e um arquivo **makefile**

IMPORTANTE: Não altere os valores dos ponteiros do pokémon em si, apenas qual endereço a caixa está guardando! Estamos apenas apontando para índices de dentro da dex para evitar cópias de valores idênticos pela caixa. Se você alterar o valor que o ponteiro está apontando, vai sobrescrever o valor do Pokémon e ter erros!

OBS: As entradas serão índices dos Pokémon e não o valor em si na National Dex, logo, a entrada 0 se refere ao Bulbasaur e assim por diante :)

OBS2: Na aula explico como fiz tudo isso e tento montar um tutorial.

Exemplo:

ENTRADA:

1
1
3
95
3
101
3
145
4
560
6
5
0

Movemos a caixa 2 vezes para a direita, adicionamos o Pokémon de índice 95, depois do de índice 101, depois o de 145, depois o de 560 no índice 6 da caixa. Imprimimos tudo da caixa atual e saímos.

SAÍDA (Em forma de imagem pois o documento destruiria a formatação):



