

TRABALHO 2: DRAGON-BALL EVOLUTION

ALGORÍTMO GENÉTICO PARA CRIAÇÃO DE POPULAÇÕES

Ao início de tudo, é criada a estrutura que comporta toda a população do programa. Essa estrutura é uma lista duplamente encadeada. Para que essa população cresça, os personagens se cruzam entre si, respeitando algumas regras. Uma função que prevê todas as combinações dos cruzamentos dos personagens é feita antes do início da população. Essa função retorna um ponteiro duplo que contém os identificadores dos personagens que cruzarão.

Em seguida, os 4 personagens Primordiais são gerados e são atribuídos valores para cada uma de suas características através de uma implementação de uma função dos alunos. Com esses 4, são gerados mais 4 utilizando as combinações citadas no início. Basta apenas receber uma combinação de uma determinada posição do ponteiro duplo 'i' e buscar os pais na lista que vão gerar um filho.

Com esses pais, se faz o cruzamento deles andando na árvore e verificando cada gene do pai e da mãe, a fim de passar isso para o filho. Essa escolha e depois passagem para o filho é feita através de uma série de funções que escolhem a menor dominância e calcula a probabilidade de mutação do gene.

Ao final da geração desses 16 personagens, a lista deve ser liberada de modo que não se tenha dados na memória, nem dados que estão "soltos", que estão vazando na memória. Entretanto, se tentou fazer o uso da função "Tree* tree_free(Tree* mangueira)", que tem o propósito de liberar a árvore(ou o personagem). Porém, durante o tempo de execução do programa, se depara com um erro de duplo pedido de free, como se um ponteiro que tivesse sido alocado memória para ele estivesse sendo liberado duas vezes, o que dá erro.

Acreditasse que o problema pode ser nas funções que o programa já usa, e fica complicado de saber pois não se tem acesso ao código fonte dessas funções. Logo, a parte de liberação da memória foi toda comentada a fim de não se ter problema quando for rodar o programa por parte do usuário.

DA EXECUÇÃO

O programa possui uma particularidade relacionada ao uso das funções de aleatoriedade: ao ser rodado de uma só vez e criando e mostrando todos os personagens, caso haja alguma mutação, quase todos os próximos personagens possuirão o mesmo tipo de mutação e a mesma característica. Isso se dá pela velocidade de execução do programa que usa o tempo como Seed para a função rand(). Por isso uma solução plausível e que melhora a experiência para o usuário é a utilização do comando getch() que aguarda um enter do usuário para mostrar a próxima árvore/personagem.

A saída do programa vem de percorrer a árvore e imprimir a característica dos nós folhas de cada personagem, considerando a possibilidade de mutação.

Seguem abaixo prints da execução do programa mostrando somente árvores de personagens e também as imagens deles:

PRIMEIROS DOIS PERSONAGENS PRIMORDIAIS

```
./trabalho
→ code git:(fillype) make
g++ -o trabalho main.c preenchimento.c lista.c -g -L. -l personagem -l playAPC -I.
→ code git:(fillype) ./trabalho
-Identificacao: 1--Pai: 0--Mae: 0-----
olhos cor: castanho
formato: triangular
cabelos cor: azul
tipo: goku
membros calcas: azul
botas: azul
tronco raca: humano
camisa: treino

-Identificacao: 2--Pai: 0--Mae: 0-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: namekusei
camisa: treino
```

PERSONAGENS FILHOS COM PARENTESCO ALEATÓRIO

```
./trabalho
membros calcas: laranja
botas: azul
tronco raca: saiyajin
camisa: treino

-Identificacao: 5--Pai: 2--Mae: 3-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: MUTANTE saiyajin
camisa: treino

-Identificacao: 6--Pai: 2--Mae: 4-----
olhos cor: MUTANTE azul
formato: circular
cabelos cor: MUTANTE azul
tipo: careca
membros calcas: MUTANTE laranja
botas: azul
tronco raca: namekusei
camisa: treino

-Identificacao: 7--Pai: 3--Mae: 1-----
olhos cor: castanho
formato: circular
cabelos cor: azul
tipo: careca
membros calcas: azul
botas: azul
tronco raca: humano
camisa: treino
```

ULTIMOS PERSONAGENS E ENCERRAMENTO DO PROGRAMA

```
fillype@phelps: ~/Documentos/ed2017/dragon_ball/code
camisa: treino

-Identificacao: 14--Pai: 5--Mae: 3-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: MUTANTE laranja
botas: azul
tronco raca: saiyajin
camisa: treino


-Identificacao: 15--Pai: 7--Mae: 3-----
olhos cor: castanho
formato: circular
cabelos cor: azul
tipo: careca
membros calcas: azul
botas: MUTANTE azul
tronco raca: humano
camisa: treino

-Identificacao: 16--Pai: 3--Mae: 4-----
olhos cor: azul
formato: circular
cabelos cor: azul
tipo: MUTANTE trunks
membros calcas: MUTANTE laranja
botas: azul
tronco raca: saiyajin
camisa: treino

→ code git:(fillype)
```

INICIO DO PROGRAMA COM IMAGENS DOS PERSONAGENS

```
Montagem
→ code git:(fillype) make
g++ -o trabalho main.c preenchimento.c lista.c -g -L. -l personagem -l playAPC -I.
→ code git:(fillype) X ./trabalho
-Identificacao: 1--Pai: 0--Mae: 0-----
olhos cor: azul
formato: triangular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: namekusei
camisa: treino
█
```



Montagem

-Identificacao: 5--Pai: 1--Mae: 3-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: namekusei
camisa: MUTANTE armadura

-Identificacao: 6--Pai: 2--Mae: 3-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: laranja
botas: azul
tronco raca: humano
camisa: armadura

-Identificacao: 7--Pai: 1--Mae: 2-----
olhos cor: MUTANTE castanho
formato: MUTANTE triangular
cabelos cor: MUTANTE preto
tipo: careca
membros calcas: MUTANTE laranja
botas: MUTANTE azul
tronco raca: humano
camisa: armadura

-Identificacao: 8--Pai: 4--Mae: 2-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: humano
camisa: MUTANTE semcamisa

□



Montagem

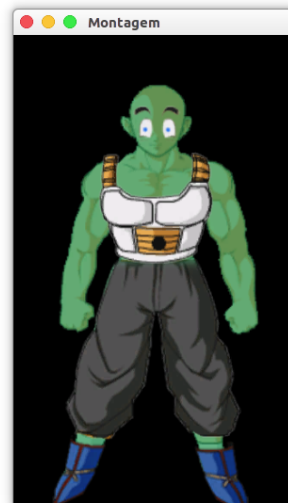
-Identificacao: 11--Pai: 8--Mae: 6-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: humano
camisa: armadura

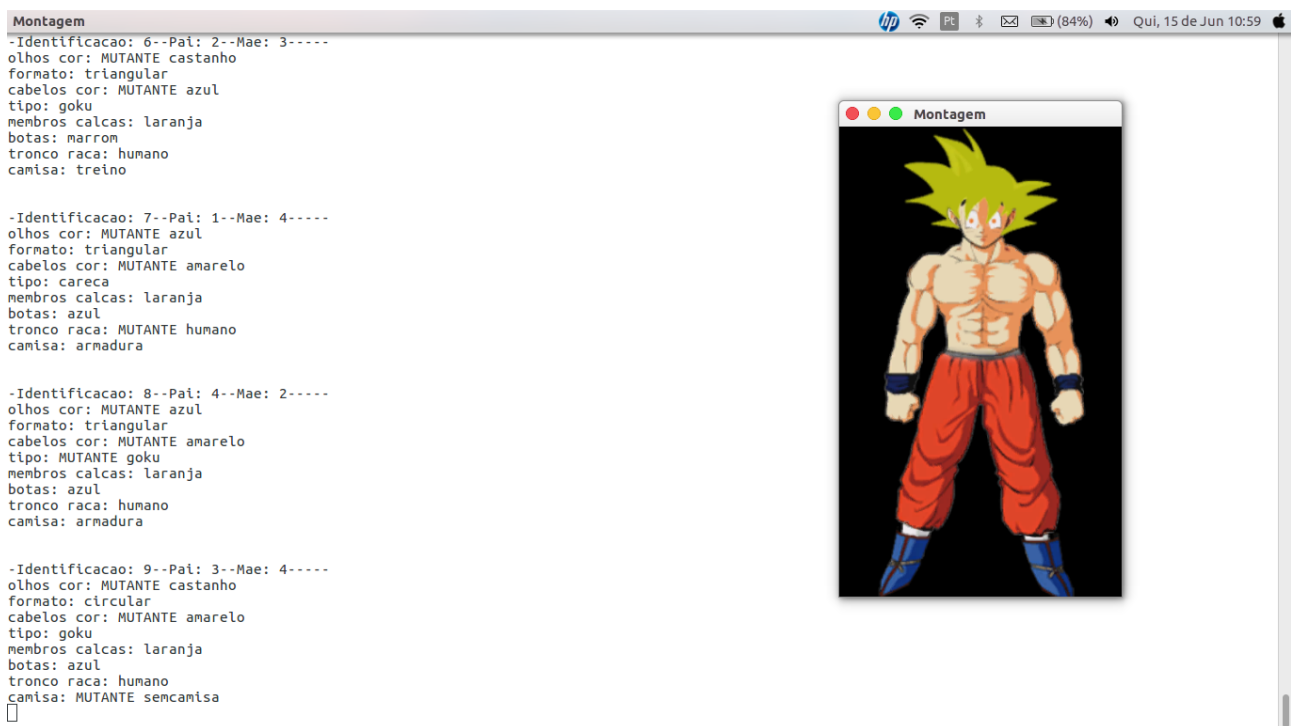
-Identificacao: 12--Pai: 5--Mae: 8-----
olhos cor: azul
formato: circular
cabelos cor: amarelo
tipo: careca
membros calcas: azul
botas: azul
tronco raca: humano
camisa: armadura

-Identificacao: 13--Pai: 6--Mae: 4-----
olhos cor: MUTANTE verde
formato: circular
cabelos cor: MUTANTE preto
tipo: careca
membros calcas: azul
botas: MUTANTE azul
tronco raca: humano
camisa: armadura

-Identificacao: 14--Pai: 5--Mae: 1-----
olhos cor: MUTANTE azul
formato: circular
cabelos cor: MUTANTE amarelo
tipo: careca
membros calcas: MUTANTE preta
botas: azul
tronco raca: namekusei
camisa: armadura

□





Pedro Kelvin de Castro Moreira Batista – 13/0129674
Fillype Alves do Nascimento – 16/0070431