Picareta

Tiago mendes da silva

MAIN

```cpp
#include <iostream>
#include <string>
#include "Picareta.h"
#include "Bloco.h"
#include "Mapa.h"
int main()
{
    string str;
    char op;
    cout<<"insira uma cadeia de blocos (ouro=0,pedra=p,ferro=f,diamante=d,inicio=i)"<<endl;
    cin>>str;
    Mapa  mundo(str);
    cout<<"insira uma material para a picareta (pedra,diamante,ouro,ferro,madeira)"<<endl;
    cin>>str;
    Picareta ferramenta(str);
    ferramenta.mostrarDurabilidade();
    while (op!='x')
    {
    system("cls");
    mundo.refresh();
    cout<<"=escolha uma acao="<<endl;
    cout<<"d-atacar direito"<<endl;
    cout<<"a-atacar esquerdo"<<endl;
    cout<<"x-sair"<<endl;
    cin>>op;
    if (op=='d')
    ferramenta.atacar(mundo.bloco(+1));
    else if (op=='a')
        ferramenta.atacar(mundo.bloco(-1));
    system("pause");
    }
    cout<<"tchau!!!"<<endl;
    return 0;
```

Declara uma string e um char que serão usados no main

Instancia um mundo do tipo mapa

Construtor de mapa

```
Mapa::Mapa(const string & str)
{
 int i;
 playerX=0;
 tamanho=str.length();
 if (tamanho>=20) tamanho=20;
 grid=new Bloco[tamanho];
 grid[0].init("ar");
 for (i=0;i<tamanho;i++)
 {
     if (str[i]=='f')
        grid[i].init("ferro");
     else if (str[i]=='o')
             grid[i].init("ouro");
     else if (str[i]=='d')
             grid[i].init("diamante");
     else if (str[i]=='i')
             {
             grid[i].init("ar");//se houver mais de um i todos seram vazio mas o ultimo sera o inicio
             playerX=i;
             }
     else
          grid[i].init("pedra");
 }
 grid[playerX].init("ar");//garante que nao haja bloco no inicio caso ele nao seja especificado;
}
```

```
Mapa::Mapa()
{
 int i;
 tamanho=20;
 grid=new Bloco[20];
 for (i=1;i<20;i++)
     grid[i].init("pedra");
 grid[0].init("ar");
 playerX=0;
}
```

```
Mapa::~Mapa()
{
 delete[](grid);
}
```

Basicamente o construtor cria os blocos baseado em uma string ou 20 blocos de pedra caso não haja string

No destrutor libera a memoria do array

```
void Bloco::init(const string & mtr)
{
      if (mtr=="ouro")
      {
        material=mtr;
        resistencia= 40;
        cor=6;
      }
      if (mtr=="ar")
      {
        material=mtr;
        resistencia= 0;
        cor=15;
      }
      if (mtr=="pedra")
      { material=mtr;
        resistencia=10;
        cor=8;
      }
      if (mtr=="ferro")
      { material=mtr;
        resistencia= 20;
        cor=12;
      }
      if (mtr=="diamante")
      { material=mtr;
        resistencia= 80;
        cor=9;
      }
}
```

Inicializar Bloco

No metodo anterior ele chamava esse metodo para todos os blocos, ele que define os atributos do bloco

MAIN

```cpp
#include <iostream>
#include <string>
#include "Picareta.h"
#include "Bloco.h"
#include "Mapa.h"
int main()
{
    string str;
    char op;
    cout<<"insira uma cadeia de blocos (ouro=0,pedra=p,ferro=f,diamante=d,inicio=i)"<<endl;
    cin>>str;
    Mapa  mundo(str);
    cout<<"insira uma material para a picareta (pedra,diamante,ouro,ferro,madeira)"<<endl;
    cin>>str;
    Picareta ferramenta(str);
    ferramenta.mostrarDurabilidade();
    while (op!='x')
    {
    system("cls");
    mundo.refresh();
    cout<<"=escolha uma acao="<<endl;
    cout<<"d-atacar direito"<<endl;
    cout<<"a-atacar esquerdo"<<endl;
    cout<<"x-sair"<<endl;
    cin>>op;
    if (op=='d')
    ferramenta.atacar(mundo.bloco(+1));
    else if (op=='a')
        ferramenta.atacar(mundo.bloco(-1));
    system("pause");
    }
    cout<<"tchau!!!"<<endl;
    return 0;
```

Instanciando ferramenta do tipo picareta, depois mostra a durabilidade

Construtor de picareta

```
Picareta::Picareta()
{
    material="madeira";
    initDurabilidade();
    cout<<"voce ganhou uma picareta de "<<material<<endl;
}
```

Se não receber string ou receber uma string não aceita considera a string madeira
Depois inicializa os valores

```
Picareta::Picareta(const string & str)
{
    if (str=="madeira" || str=="pedra" || str=="ferro" || str=="ouro" || str=="diamante")
        material=str;
    else
    {   cout<<"esse material nao produz picaretas,usando madeira"<<endl;
        material="madeira";
    }
    initDurabilidade();//depois usara um metodo statico para definir o valor
    cout<<"voce ganhou uma picareta de "<<material<<endl;
}
```

Inicializar valores

```
void Picareta::initDurabilidade()
{
      if (material=="ouro")
      { durabilidade= 32;
        forca=12;
      }
      if (material=="madeira")
      { durabilidade= 59;
        forca=4;
      }
      if (material=="pedra")
      {  durabilidade= 131;
         forca=6;
      }
      if (material=="ferro")
      {  durabilidade= 250;
         forca=8;
      }
      if (material=="diamante")
      { durabilidade= 1561;
        forca=10;
      }
}
```

Configura a durabilidade e eficiência das ferramentas

A nível de curiosidade a de ouro é a mais eficiente porem menos durável

MAIN

```cpp
#include <iostream>
#include <string>
#include "Picareta.h"
#include "Bloco.h"
#include "Mapa.h"
int main()
{
    string str;
    char op;
    cout<<"insira uma cadeia de blocos (ouro=0,pedra=p,ferro=f,diamante=d,inicio=i)"<<endl;
    cin>>str;
    Mapa  mundo(str);
    cout<<"insira uma material para a picareta (pedra,diamante,ouro,ferro,madeira)"<<endl;
    cin>>str;
    Picareta ferramenta(str);
    ferramenta.mostrarDurabilidade();
    while (op!='x')
    {
    system("cls");
    mundo.refresh();
    cout<<"=escolha uma acao="<<endl;
    cout<<"d-atacar direito"<<endl;
    cout<<"a-atacar esquerdo"<<endl;
    cout<<"x-sair"<<endl;
    cin>>op;
    if (op=='d')
    ferramenta.atacar(mundo.bloco(+1));
    else if (op=='a')
        ferramenta.atacar(mundo.bloco(-1));
    system("pause");
    }
    cout<<"tchau!!!"<<endl;
    return 0;
```

2 métodos simples
mostrasDurabilidade e refresh

durabilidade

```
int Picareta::mostrarDurabilidade()
{
    cout<<"restam "<<durabilidade<<" usos"<<endl;
    return durabilidade;
}
```

refresh

```
void Mapa::refresh()
{int i;
 for (i=0;i<tamanho;i++)
 {    if (i!=playerX)grid[i].imprimir();
      else cout<<"()";
 }
 cout<<endl;
}
```

Esse metodo desenha os blocos ou o jogador em suas posicoes

MAIN

```cpp
#include <iostream>
#include <string>
#include "Picareta.h"
#include "Bloco.h"
#include "Mapa.h"
int main()
{
    string str;
    char op;
    cout<<"insira uma cadeia de blocos (ouro=0,pedra=p,ferro=f,diamante=d,inicio=i)"<<endl;
    cin>>str;
    Mapa  mundo(str);
    cout<<"insira uma material para a picareta (pedra,diamante,ouro,ferro,madeira)"<<endl;
    cin>>str;
    Picareta ferramenta(str);
    ferramenta.mostrarDurabilidade();
    while (op!='x')
    {
    system("cls");
    mundo.refresh();
    cout<<"=escolha uma acao="<<endl;
    cout<<"d-atacar direito"<<endl;
    cout<<"a-atacar esquerdo"<<endl;
    cout<<"x-sair"<<endl;
    cin>>op;
    if (op=='d')
    ferramenta.atacar(mundo.bloco(+1));
    else if (op=='a')
        ferramenta.atacar(mundo.bloco(-1));
    system("pause");
    }
    cout<<"tchau!!!"<<endl;
    return 0;
```

Enfim o loop principal
Aqui o usuário escolhe um direção para ir ou finalizar o programa

A ferramenta ataca um bloco(que ela não conhece)
Por isso “pergunta” para o mundo o que existe a direita ou esquerda

Atacar bloco

```
Bloco& Mapa::bloco(int n)
{

      if (playerX+n>=tamanho || playerX+n<0)
      {
       cout<<"fora do mapa"<<endl;
       return grid[playerX];
      }
      else
      {
      playerX+=n;
      return (grid[playerX]);
      }

}
```

Se a posição escolhida não estiver mapeada retorna o bloco na posição atual, caso contrario
move o player para essa posição e retorna o bloco na nova posição

```
//INTERACAO COM OUTROS OBJETOS
bool Picareta::atacar( Bloco & target )
{
    if (durabilidade==0)
    {  cout<<"sua picareta esta quebrada";
       return false;
    }
    if (target.damage(forca))
       cout<<"bloco destruido"<<endl;
    return true;
}
```

Então de posse dessa informação
Prossegue com o metodo atacar se a picareta não estiver quebrada
Causa dano ao bloco

damage

```
bool Bloco::damage(int forca)
{
      if (resistencia!=0)
      {
         cout<<"minerando..."<<endl;
         Sleep(resistencia*1000/forca);
         cout<<(resistencia*1000/forca)/1000<<"segundos para destruir"<<endl;
         resistencia=0;
         return true;
      }
      else
      {
           cout<<"espaco vazio"<<endl;
           return false;
       }

}
```

Se o bloco não estiver destruído minera o bloco
Isso leva tempo dependendo do bloco
Caso o bloco já esteja destruído simplesmente termina i metodo

```
insira uma cadeia de blocos (ouro=0,pedra=p,ferro=f,diamante=d,inicio=i)
oooiiiooofffppppddd
insira uma material para a picareta (pedra,diamante,ouro,ferro,madeira)
diamante
```

```
[][][]____()[][][][][][][][][][][][][]
=escolha uma acao=
d-atacar direito
a-atacar esquerdo
x-sair
```

Um exemplo das entradas e seus resultados

```
[][][]__()__[][][][][][][][][][][][][]
=escolha uma acao=
d-atacar direito
a-atacar esquerdo
x-sair
a
espaco vazio
Pressione qualquer tecla para continuar. . .
```

```
[][][]_____()[][][][][][][][][][][][][]
=escolha uma acao=
d-atacar direito
a-atacar esquerdo
x-sair
d
minerando...
2segundos para destruir
bloco destruido
Pressione qualquer tecla para continuar. . .
```