



Documentação - Rand()venture

Ciência da Computação

Estrutura de Dados 2017

Professor Roberto Ferrari

Aplicação de Lista

Alunos:

Alisson Nunes Vieira Amâncio

Gabriel De Souza Alves

Matheus Bortoleto Da Silva

Rafael Sales Pavarina

Universidade Federal de São Carlos

São Carlos, 2017

Sumário

1. Descrição do Projeto	02
2. Regras do Jogo	02
3. Instalação	02
3.1. Linux	02
3.2. Windows	02
4. Aplicação de Lista Encadeada	02
5. Mecânicas do Jogo	04
5.1. Inimigos	04
5.2. Geração de mapas	04
5. Autores	04

1. Descrição do Projeto

Este projeto foi feito para a matéria de Estruturas de Dados, buscando aplicar os conceitos de implementação de uma lista encadeada. Para isso, implementamos um jogo de temática Roguelike.

2. Regras do Jogo

- Você caiu em um labirinto mágico, com monstros que desejam comer sua cara. Seu objetivo é derrotar todos os monstros, e seguir vivo, destruindo todos os seus inimigos.
- Junto com você, surgem quatro poções, que podem ser usadas por andar. Use com sabedoria. A vermelha recupera sua saúde, a laranja aumenta seu poder de ataque, a cinza aumenta sua armadura, para aguentar maiores danos, e a verde te deixará muito rápido.
- Para se mover utilize WSAD, e para atirar use as setas direcionais.

3. Instalação

3.1. Linux

1. Baixe os arquivos para Linux o link a seguir o repositório do projeto no Github <http://www.github.com/explodingnuggets//Roguelike/releases/latest>;
2. Coloque os arquivos na pasta que desejar;
3. Antes de executar, utilize o comando `export LD_LIBRARY_PATH="../SFML/lib"` na pasta de instalação, ou então, instale a biblioteca SFML.
4. Execute o arquivo `randventure.out`.

3.2. Windows

1. Baixe os arquivos para Windows o link a seguir o repositório do projeto no Github <http://www.github.com/explodingnuggets//Roguelike/releases/latest>
2. Coloque os arquivos na pasta que desejar;
3. Execute o arquivo `randventure.exe`.

4. Aplicação de Lista Encadeada

Os conceitos de lista encadeada foram aplicadas na seleção da classe personagem, sendo que criaram-se os seguintes métodos de percorrimento para a lista de seleção:

```

void ButtonList::toNext() {
    m_actualIndex++;
    if(m_actualIndex >= m_buttonSprites.getSize())
        m_actualIndex = 0;
    m_outlineSprite.setPosition(m_buttonSprites[m_actualIndex]-
>getPosition());
}

void ButtonList::toPrevious() {
    m_actualIndex--;
    if(m_actualIndex < 0)
        m_actualIndex = m_buttonSprites.getSize()-1;
    m_outlineSprite.setPosition(m_buttonSprites[m_actualIndex]-
>getPosition());
}

```

Foi implementado o operador de acesso [], permitindo o acesso mais simples dos elementos da lista encadeada, para isso, utiliza-se o seguinte método:

```

template<class T>
T LinkedList<T>::operator[](int index) {
    this->toFirst();
    while (index > 0 && !this->toNext())
        index--;
    return this->current->value;
}

```

Esse código reseta o ponteiro da lista para o início, e então, vai iterando até chegar no índice desejado.

5. Mecânicas de Jogo

5.1. Inimigos

Os inimigos utilizam o algoritmo A* para determinar o melhor caminho até o herói. Esse algoritmo utiliza o grid criado pelo mapa, e tenta traçar o melhor caminho a partir da posição atual do inimigo.

Para um maior aprofundamento, utilize os links a seguir:

- <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>
- <http://www.redblobgames.com/pathfinding/a-star/introduction.html>
- <http://mat.uab.cat/~alseda/MasterOpt/AStar-Algorithm.pdf>

5.2. Geração dos mapas

Para a geração do mapa aleatório, foi utilizado o algoritmo "Recursive Backtracker", que gera um labirinto delimitado pelos extremos do mapa, e logo em seguida, são criadas salas vazias nesse labirinto.

Alguns materiais do Recursive Backtracker:

- <http://weblog.jamisbuck.org/2010/12/27/maze-generation-recursive-backtracking>
- <https://scratch.mit.edu/projects/17358777/>

6. Autores

- Alisson Nunes - Interface e parte gráfica.
- Gabriel Alves - Implementação da estrutura dos dados e controles.
- Matheus Bortoleto - Parte gráfica e controles.
- Rafael Sales - Documentação.