

DOCUMENTAÇÃO

1. O QUE É O SNAKEWORD?

SnakeWorld é uma variação do clássico jogo Snake, popular durante décadas. Esta aplicação foi desenvolvida afins de utilizarmos o conhecimento adquirido em aula na displina de Estrutura de Dados e com o tempo foi vista como um forte instrumento didático, especialmente para quem está sendo alfabetizado.

2. NÍVEIS DE DIFICULDADE

Nível 1: palavras de 4 letras

Nível 2: palavras de 5 letras

Nível 3: palavras de 6 letras

3. COMO JOGAR

A cobra começa com o tamanho fixo de quadrados dependendo da quantidade de letras de cada nível, a palavra é informada e o jogador deve procurar a partir das opções cada letra da palavra, na ordem correta. A cada letra acertada a cobra diminuirá um quadrado na frente e ganhará 10*n pontos, caso a letra esteja errada não será eliminado o quadrado, perderá 5*n pontos e as letras capturadas se transformaram em quadrados e uma nova palavra será sorteada. Caso acerte toda a palavra ganha 100*n pontos procuramos as letras da palavra e eliminamos da cobra.

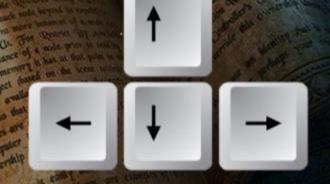
4. COMO PERDER O JOGO

Assim como em uma jogo da cobrinha comum, no SnakeWord o jogador perde quando toca nas extremidades ou toca no seu próprio corpo.

DOCUMENTAÇÃO

5. TECLAS QUE PODEM SER UTILIZADAS:

Seta direcionais: Movimentação da cobra pelo cenário.



Barra de espaço: pausar o jogo.

- Mouse: Interação no menu.

Space

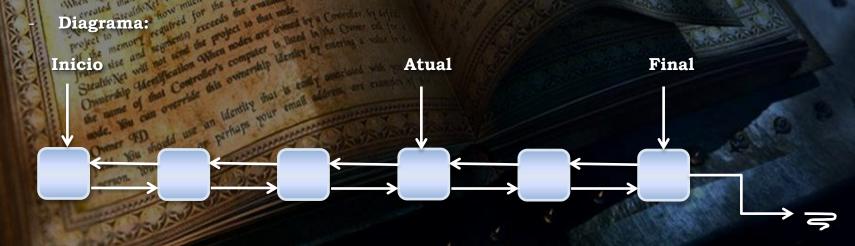
IMPLEMENTAÇÃO

1. COMO FOI IMPLEMENTADO?

Foi implementado na linguagem de programação C++ com auxilio da biblioteca gráfica Allegro 5.

2. ESTRUTURA DE DADOS

Utilizamos a estrutura Lista Cadastral, onde inserimos sem nenhuma regra, mas a retirada acontece a partir do valor dos elementos, ou seja, encontra-se o elemento que deseja retirar em qualquer parte da lista e depois é removido. É importante destacar também que trata-se de uma Lista duplamente encadeada com elementos repetidos.



IMPLEMENTAÇÃO

DIAGRAMA DE CLASSES

Node

- esq: Node

3.

- dir : Node
- info : char
- poslnicialX ; int
- posFinalX : int
- postnicialY ; int - postinalY : int
- + Node() : void
- -Node():void
- + setinfo(info : char) : void
- + getinfo() : char
- + setPosInicialX(posInicialX:int); void
- + getPoshicialX(): int
- + setPosFinalX(posFinalX; int); void
- + getPosFinalXII int.
- + setPoslnicialY[poslnicialY : int) : void
- + getPoshicialY(): int
- + setPosFinalY(posFinalY:int): void
- + getPosFinalYII int
- + setEsq(esq: Node): void
- + getEsg() : Node
- + setDir(dir : Node) : Node
- + getDir(); Node
- + Node(postnicialX: int, postnicialY: int, posFinalX: int, posFinalY: int, info: char): void
- + operator=(node : Node) : Node
- + operator==(node : Node) : boolean

Lista

- Inicio : Node - Atual : Node
- Atual : Node - Final : Node
- + Lista() void
- ~ Lista() ; void + sethicio(lnicio ; Node) ; void
- + gettricio(): Node
- + setAtual(Atual: Node): void
- + getAtual(): Node
- + setFinal(Final: Node): void
- + getFinal(): Node
- + CriaNode(PAux : Node) : Node
- + Vazia() : bool
- + Inserir(PAux : Node) : bool
- + Retirar(PAux Node) bool
- + PegaOPrimeiro(PAux: Node): bool
- + PegaOProximo(PAux: Node): bool
- + PegaOAnterior(PAux : Node) : bool
- + PegaOUltimo(PAux_Node); bool
- + PegaOPrimeiro(PAux: Node): bool
- + DestroiLista() bool
- + EstaNaLista(PAux : Node) : bool

SnakeWord

- cobra il ista
- palayra sorteada; String
- nivel : int
- portuacao int
- + SnakeWord(): void
- ~ SnakeWord(); void
- + setNivel(nivel : int) : void
- + getNivel(): int
- + quantLestras() : int
- + setPalavra sorteada(palavra : String) : void
- + getPalavra sorteada(): String
- + setPontuacao(Pontuacao : int) ; void
- + getPontuacao(): int
- + iniciarAllegro(): int
- + mostrarPalayra(): void
- + LetraPalavra/int pos : int/) : char
- Losa danajintpos. iiig. siid
- + IniciarJogo() : void
- + Pause[] int
- + EscolherNivel(): void
- + InstruccesJogo(): void
- + SairJogo(inti: int): void
- + menu(): void
- + SorteiaPalavra(): void
- + SorteiaLetra(): char
- + SorteiaElementos(char c : int) : Node
- + CriaCobrinha[]: void
- + DesenhaCobrinha() bool
- Described bringing . Box
- + colisaoCobra() : bool
- + colisaoCantos(Aux Node) bool
- + ColisaoNode(ElementoCerto: Lista): int
- + AtualizarPontuacao(tipo : int) : void
- + AtualizarCobrinha/Elemento : Listal : void
- + EscreveCaminho(letra: char[], c: char); void
- + FinalizarAllegro(): void

CRÉDITOS

1. TUTORIAIS ALLEGRO 5

Disponíveis em:

http://www.rafaeltoledo.net/tutoriais-allegro-5/

http://www.geocities.ws/weissengeist/allegro.html

https://www.allegro.cc/manual/5/

2. SOFTWARES E BIBLIOTECAS

Softwares e Bibliotecas Utilizados: Code::Blocks 13.12, Microsoft Office Word, Microsoft Skydrive, Microsoft Team Foudantion Server, Microsoft Codeplex, Astah community, Allegro Gaming Library v5.

3. DESENVOLVEDORES

Breno Silveira

E-mail: brufsc@hotmail.com

Douglas Barbino

E-mail: douglas-448@hotmail.com

Denis Cappelini

E-mail: denis_cappelini@hotmail.com

Gabriela Mattos

E-mail: gaby.vanessa@gmail.com

Colaboração no design gráfico: Claudio Silva

E-mail: claudiocfls@hotmail.com

TESTES ESCREENS



Tela do menu.



Tela do jogo. (Nível 1)



Tela de instruções.



Tela de pontuação.