

PROGRAMAÇÃO OO EM C++: JOGO “BRASIL INDÍGENA”

Trabalho de Técnicas de Programação
Prof.: Dr. Jean M. Simão

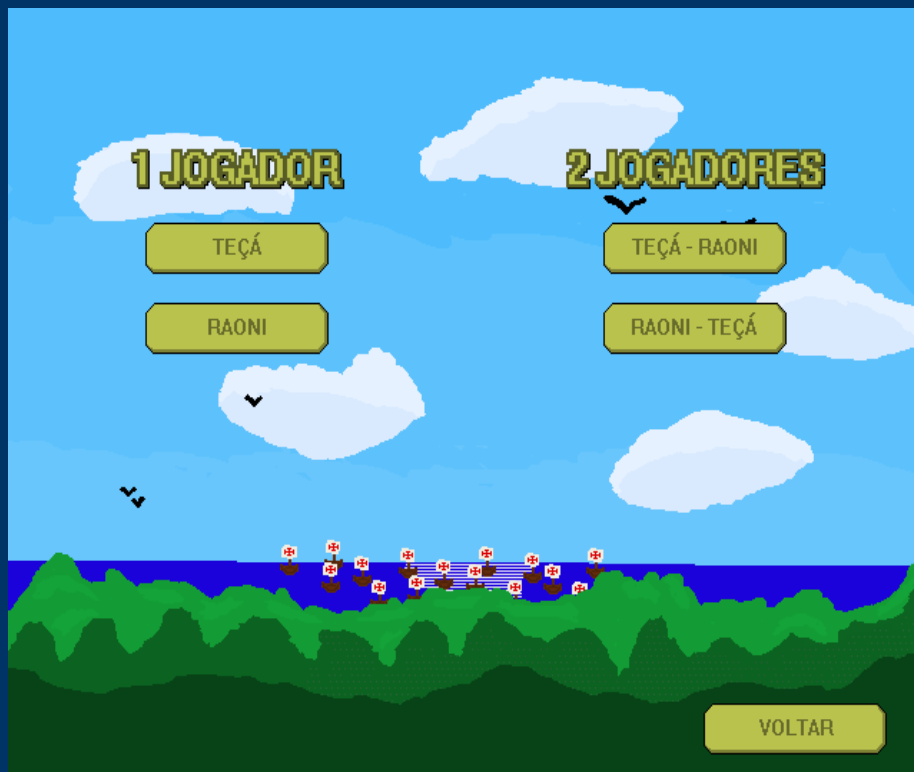


Autores:
Waine Barbosa de Oliveira Júnior
João Felipe Sarggin Machado

Menu do Jogo



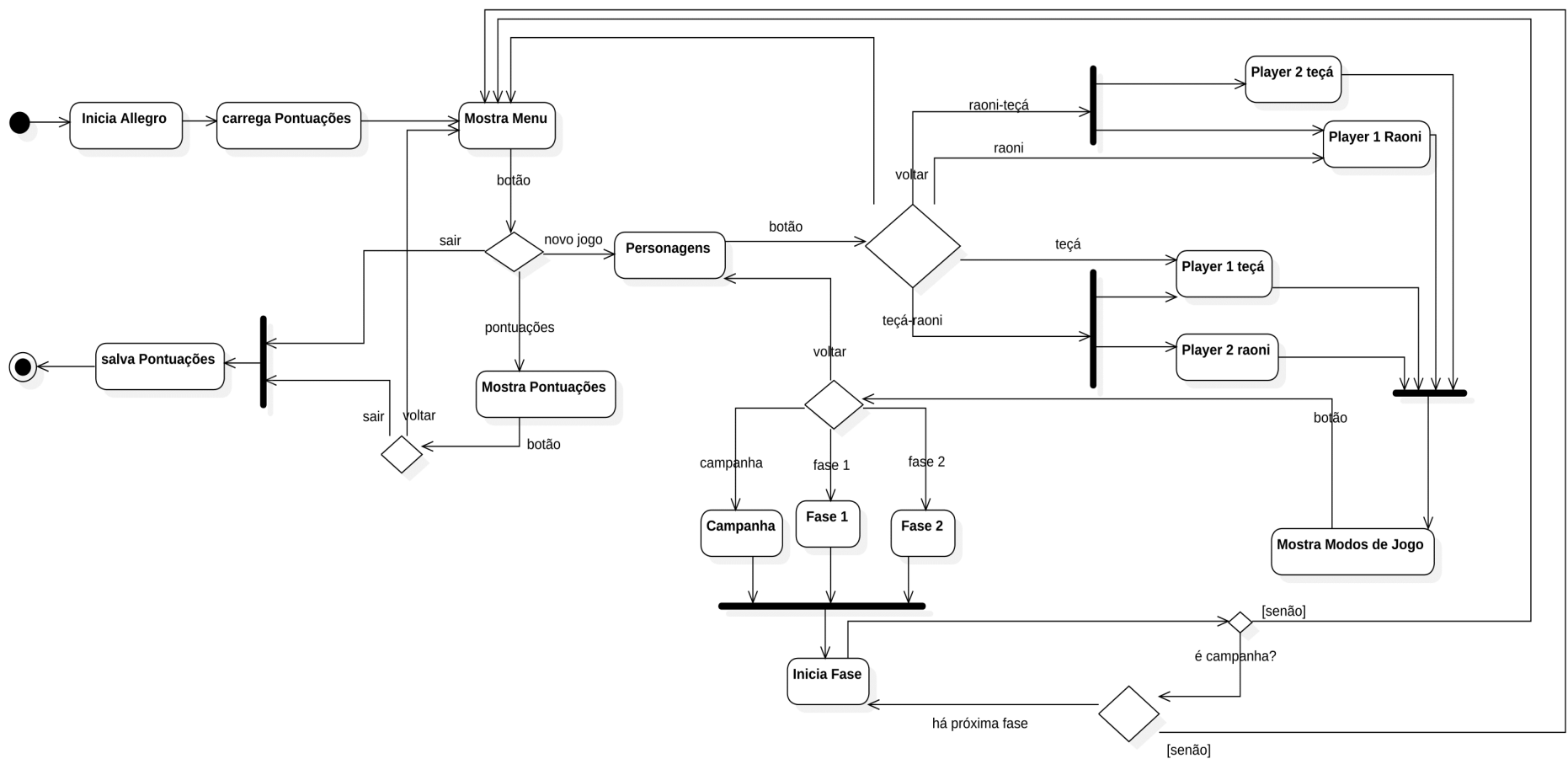
Menu de Personagens e de Fases



Exemplo de Pontuações



Diagrama de Atividades do Jogo



Protagonistas



Tela de Derrota



Tabela de Pré-requisitos e Suas Situações

N.	Requisitos Funcionais	Situação
1	Apresentar menu de opções aos usuários do Jogo	Requisito previsto inicialmente e realizado
2	Permitir um ou dois jogadores aos usuários do Jogo	Requisito previsto inicialmente e realizado
3	Disponibilizar ao menos duas fases que podem ser jogadas sequencialmente ou selecionadas.	Requisito previsto inicialmente e realizado.
4	Ter três tipos distintos de inimigos.	Requisito previsto inicialmente e realizado
5	Ter a cada fase ao menos dois tipos de inimigos com número aleatório de instâncias, podendo ser várias instâncias.	Requisito previsto inicialmente e realizado
6	Ter inimigo "Chefe" na última fase	Requisito previsto inicialmente e não realizado
7	Ter três tipos de obstáculos.	Requisito previsto inicialmente e realizado
8	Ter em cada fase entre um e dez tipos de obstáculos com número aleatório de obstáculos.	Requisito previsto inicialmente e realizado
9	Ter representação gráfica de instâncias.	Requisito previsto inicialmente e realizado
10	Ter em cada fase um cenário de jogo com obstáculos.	Requisito previsto inicialmente e realizado
11	Gerenciar colisões entre jogador e inimigos.	Requisito previsto inicialmente e realizado
12	Permitir cadastrar/salvar dados do usuário, manter pontuação durante jogo, salvar pontuação e gerar lista de pontuação (<i>ranking</i>).	Requisito previsto inicialmente e parcialmente realizado
13	Permitir Pausar o Jogo	Requisito previsto inicialmente e realizado
14	Permitir Salvar Jogada.	Requisito previsto inicialmente e não realizado

[illegible]

Tabela de Conceitos Utilizados

N.	Conceitos	Uso	Onde
1	Elementares:		
	- Classes, objetos,	Sim	Todos .h e .cpp
	- Atributos (privados), variáveis e constantes	Sim	Todos .h e .cpp
	- Métodos (com e sem retorno).	Sim	Todos .h e .cpp
	- Métodos (com retorno <i>const</i> e parâmetro <i>const</i>).	Sim	Todos .h e .cpp
	- Construtores (sem/com parâmetros) e destrutores	Sim	Todos .h e .cpp
	- Classe Principal.	Sim	Jogo.h e .cpp
	- Divisão em .h e .cpp.	Sim	No projeto.
2	Relações de:		
	- Associação	Sim	Mapa e Entidade, dentre outras
	- Agregação via associação	Não	
	- Agregação propriamente dita.	Sim	Jogo.h
	- Herança elementar.	Sim	Fase1.h, Fase2.h e Botao.h
	- Herança em diversos níveis.	Sim	Todas classes que herdam de Entidade.h
	- Herança múltipla.	Não	
3	Ponteiros, generalizações e exceções		
	- Operador <i>this</i>	Sim	Classes Animação, ListaAnimação, Jogador, Personagem
	- Alocação de memória (<i>new</i> & <i>delete</i>)	Sim	Classes Fase, Personagem, gerenciaAnim, gerenciaPontuacoes, Jogador e Mosqueteiro
	- Gabaritos/ <i>Templates</i> criada/adaptados pelos autores (e.g. Listas Encadeadas via <i>Templates</i>)	Sim	Lista.h
	- Uso de Tratamento de Exceções	Não	
4	Sobrecarga de:		
	- Construtoras e Métodos.	Sim	Entidade.h, protoEntidade.h, Personagem.h, Inimigo.h, Arma.h, Projeto.h, Obstaculo.h, Jogador.h e Animacao.h
	- Operadores (2 tipos de operadores pelo menos)	Sim	Jogador.h e Personagem.h
	Persistência de Objetos		
	- Texto via Arquivos de Fluxo	Sim	gerenciaPontuacoes.h
	- Binário	Não	
5	Virtualidade:		
	- Métodos Virtuais.	Sim	Entidade.h, ProtoEntidade.h, Fase.h, Plataforma.h
	- Polimorfismo	Sim	Entidade.h
	- Métodos Virtuais Puros / Classes Abstratas	Sim	Entidade.h, ProtoEntidade.h, Personagem.h, Inimigo.h e Fase.h
	- Coesão e Desacoplamento	Sim	Todo projeto
6	Engenharia de Software		
	- Levantamento de Requisitos Textualmente e Tabelado (Ou por meio equivalente como Diagrama de Requisitos da SysML)	Sim	Tabela 1
	- Diagrama de Classes em <i>UML</i> .	Sim	Figura 10
	- Diagrama de Atividades em <i>UML</i> .	Sim	Figura 11
	- Outros diagramas em <i>UML</i> , Diag. de Estados, Diag. de Sequência, Diag. de Pacotes etc.	Não	
	- E/ou Levantamento de Casos de Uso e sua expressão por meio de Diagrama de Casos de Uso em <i>UML</i> .		
	- E/ou outros diagramas estabelecidos, como Diag. de Fluxo de Dados (DFD) ou Diagrama em SysML (Diag. de Requisitos, de Blocos etc).		
7	Biblioteca Gráfica		

7	Biblioteca Gráfica		
	- Funcionalidades Elementares.	Sim	Jogo.h, Fase.h, derivados de Entidade.h, Botao.h e Animacao.h
	- Funcionalidades Avançadas como:	Não	
	<ul style="list-style-type: none"> tratamento de colisões duplo <i>buffer</i> <i>especificar aqui outras</i> 		
	Obs.: especificar quais funcionalidades.		
8	Interdisciplinaridades por meio da utilização de Conceitos de Matemática, Física etc		
	- Ensino Médio (especificar quais Conceitos aqui)	Sim	Cinética na movimentação dos personagens (Física), colonização da América (História)
	- Ensino Superior (especificar quais Conceitos aqui)	Sim	Coordenadas relativas e movimentação no plano X, Y (Geometria Analítica)
	Organizadores e Estáticos		
	Espaço de Nomes (<i>Namespace</i>) criada pelos autores.	Sim	gerenciaPontuacoes.h e ListaFases.h
	E/ou Classes aninhadas.	Sim	Jogo.h, Fase.h e gerenciaAnim.h
	Atributos estáticos e chamadas estáticas de métodos.	Sim	
9	Standard Template Library (STL) e String OO		
	<i>Vector</i> da STL (p/ objetos ou ponteiros de objetos de classes definidos pelos autores)	Sim	vector da STL na Lista.h e gerenciaPontuacoes.h
	e/ou <i>List</i> da STL (p/ objetos ou ponteiros de objetos de classes definidos pelos autores).		
	e/ou Pilhas, Filas, Bifilas, Filas de Prioridade, Conjuntos, Multi-Conjuntos, Mapas ou Multi-Mapas*.		
	A classe Pré-definida <i>String</i> ou equivalente.	Sim	Classe gerenciaPontuacoes
	*Obs.: Listar apenas os utilizados		
10	Uso de Conceito Avançado no tocante a Orientação a Objetos.		
	Os Padrões de Projeto: GOF	Sim	GOF: Prototype e Flyweight (gerenciaAnim.h)
	Os Programação orientada a eventos e visual: Objetos gráficos como formulários, botões etc (Listar apenas os utilizados)		
	Os Programação concorrente: Threads (Linhas de Execução) no âmbito da Orientação a Objetos, utilizando Posix, C-Run-Time ou Win32API ou afins (com ou sem uso de <i>Mutex</i> , <i>Semáforos</i> , ou <i>Troca de mensagens</i>).		
	Os API de Comunicação em Rede: Cliente Servidor.		Eventos: Fase.h, Jogo.h

Porcentagem de conceitos utilizados:
84,16%

