

INNERSPEAKER

Alessandra Fernandes Lacerda
Enzo Trevisan Topanotti

Técnicas de Programação

Referências do tema

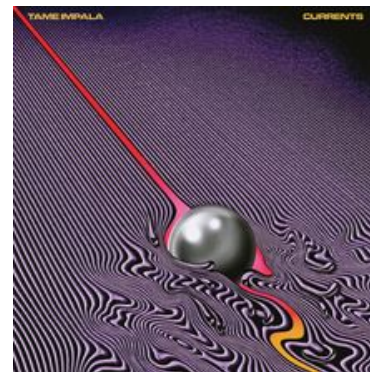
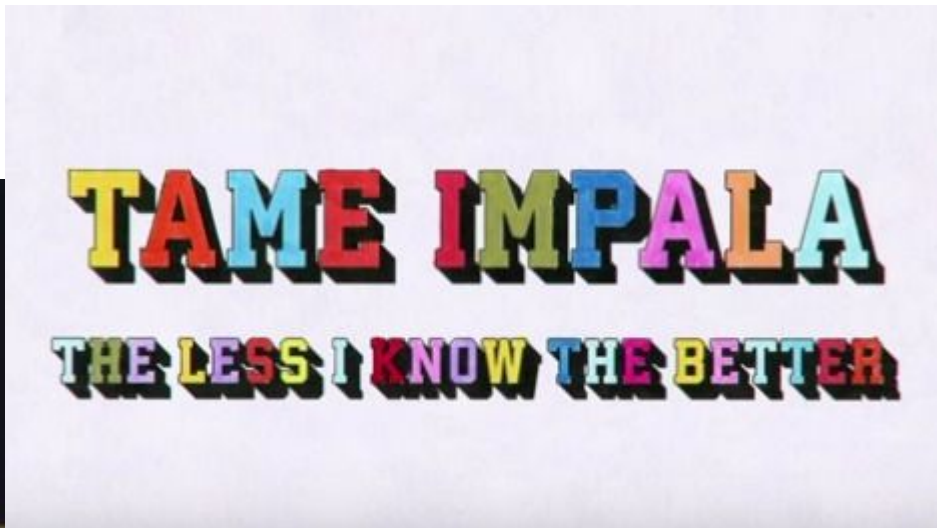


Tabela de Requisitos

Nº.	Requisitos Funcionais	Situação	Implementação
1	Apresentar menu de opções aos usuários do Jogo.	Requisito previsto inicialmente e realizado.	Requisito cumprido via classe Estado Menu e seu respectivo objeto.
2	Permitir um ou dois jogadores aos usuários do Jogo, sendo que no último caso seria para que os dois joguem de maneira concomitante.	Requisito previsto inicialmente e realizado.	Requisito cumprido via classe Jogador.
3	Disponibilizar ao menos duas fases que podem ser jogadas sequencialmente ou selecionadas.	Requisito previsto inicialmente e realizado.	Requisito cumprido via classe Jogo e classe MenuDeFases.

100%

100%

100%

Tabela de Requisitos

4	Ter três tipos distintos de inimigos (o que pode incluir 'Chefão', vide abaixo).	Requisito previsto inicialmente e realizado.	Requisito cumprido, tendo em vista que na hierarquia da classe Personagens há quatro tipos de inimigos, sendo um o 'Chefão'.	100%
5	Ter a cada fase ao menos dois tipos de inimigos com número aleatório de instâncias, podendo ser várias instâncias e sendo pelo menos 5 instâncias por tipo.	Requisito previsto inicialmente e realizado.	Requisito cumprido, via função construtora da classe Fase, herdeira de EstadoFase, com o auxílio da função auxiliar rand, presente na classe mãe.	100%
6	Ter inimigo "Chefão" na última fase.	Requisito previsto inicialmente e realizado	Requisito cumprido via classe Trevor.	100%

Tabela de Requisitos

7	Ter três tipos de obstáculos.	Requisito previsto inicialmente e realizado	Requisito cumprido, tendo em vista que na hierarquia da classe Obstáculo há três tipos de obstáculos.	100%
8	Ter em cada fase ao menos dois tipos de obstáculos com número aleatório de instâncias (i.e., objetos) sendo pelo menos 5 instâncias por tipo.	Requisito previsto inicialmente e realizado	Requisito cumprido, via função construtora da classe Fase, herdeira de EstadoFase, com o auxílio da função auxiliar rand, presente na classe mãe.	100%
9	Ter representação gráfica de cada instância.	Requisito previsto inicialmente e realizado.	Requisito cumprido via biblioteca gráfica SFML	100%

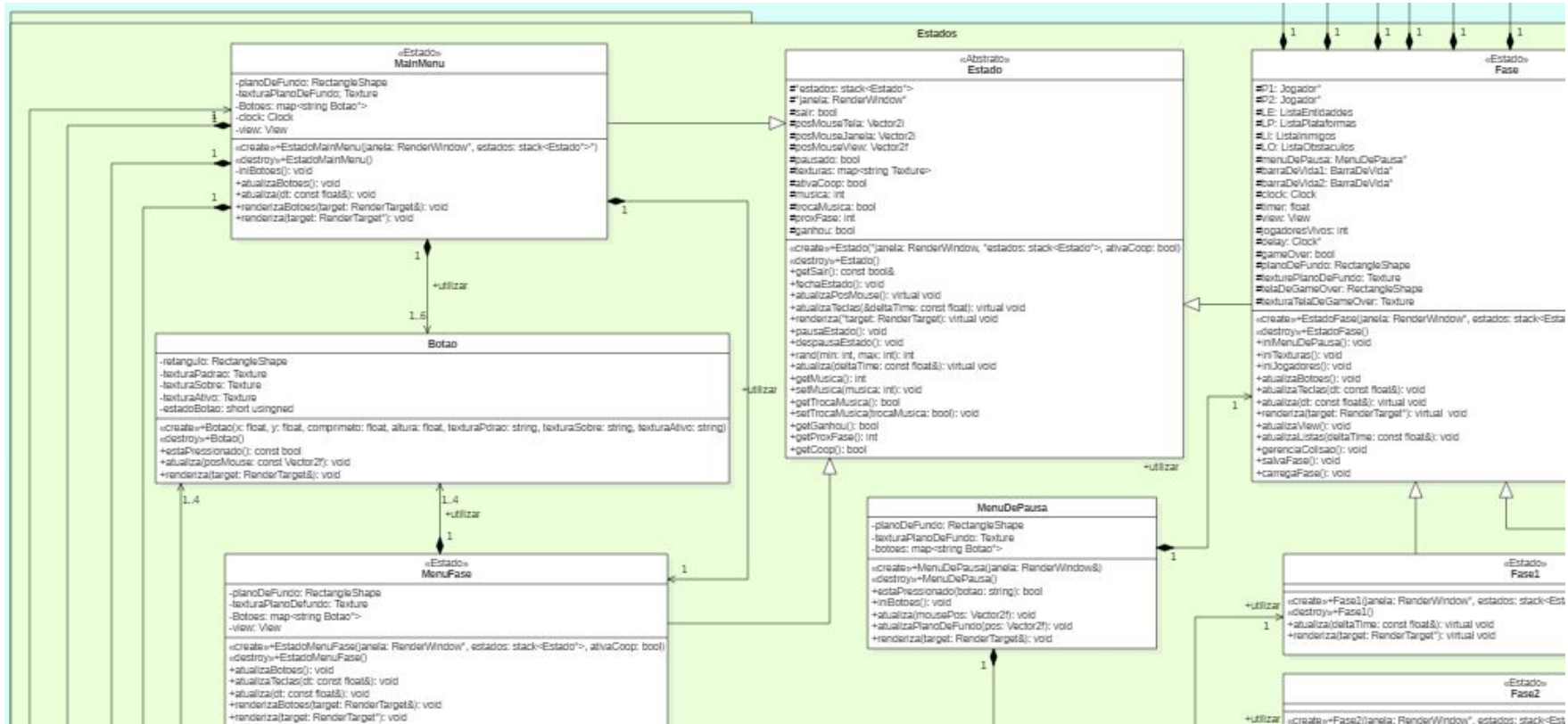
Tabela de Requisitos

10	Ter em cada fase um cenário de jogo com os obstáculos.	Requisito previsto inicialmente e realizado.	Requisito cumprido via classes Fase1, Fase2 e Fase3 herdeira de EstadoFase.	100%
11	Gerenciar colisões entre jogador e inimigos.	Requisito previsto inicialmente e realizado.	Requisito cumprido via função gerenciaColisao pertencente ao EstadoFase.	100%
12	Gerenciar colisões entre jogador e obstáculos.	Requisito previsto inicialmente e realizado.	Requisito cumprido via função gerenciaColisao pertencente ao EstadoFase.	100%

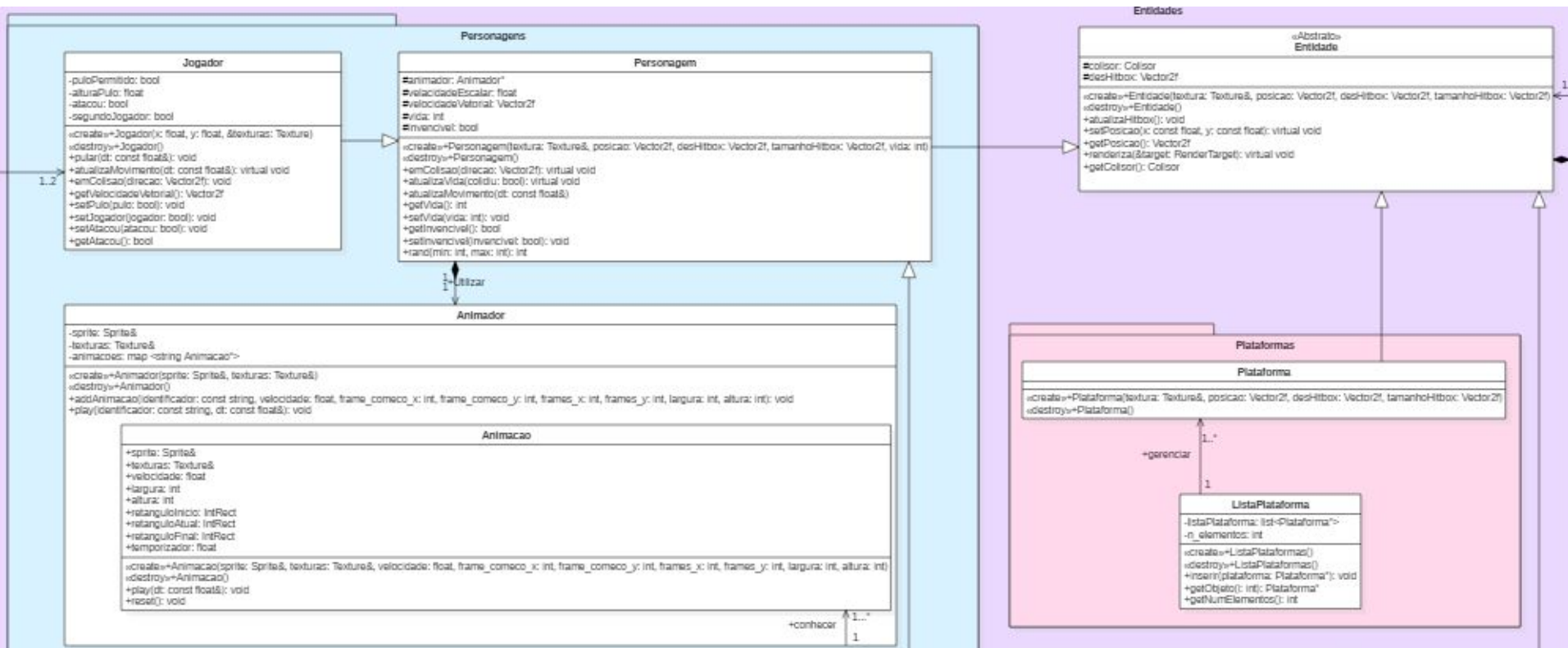
Tabela de Requisitos

12	Gerenciar colisões entre jogador e obstáculos.	Requisito previsto inicialmente e realizado.	Requisito cumprido via função <code>gerenciaColisao</code> pertencente ao <code>EstadoFase</code> .	100%
13	Permitir cadastrar/salvar dados do usuário, manter pontuação durante jogo, salvar pontuação e gerar lista de pontuação (<i>ranking</i>).	Requisito previsto inicialmente e realizado.	Requisito cumprido na função <code>atualizar()</code> da classe <code>Jogo</code> .	100%
14	Permitir Pausar o Jogo.	Requisito previsto inicialmente e realizado.	Requisito cumprido via classe <code>MenuDePausa</code> , o qual possui instância na classe <code>EstadoFase</code>	100%
15	Permitir Salvar Jogada.	Requisito previsto inicialmente e não realizado.		0%

UML - Estados



UML



Hitbox



Tabela de Conceitos

- Classes, objetos. &
- Atributos (privados), variáveis e constantes. &
- Métodos (com e sem retorno).

Sim

10%

- Métodos (com retorno *const* e parâmetro *const*). &
- Construtores (sem/com parâmetros) e destrutores

Sim

- Classe Principal.

Sim

- Divisão em .h e .cpp.

Sim

Tabela de Conceitos

- Associação direcional. & - Associação bidirecional.	Sim	7,5%
- Agregação via associação. & - Agregação propriamente dita.	Sim	
- Herança elementar. & - Herança em diversos níveis.	Sim	
- Herança múltipla.	Não	

Tabela de Conceitos

- Operador <i>this</i> .	Sim
- Alocação de memória (<i>new</i> & <i>delete</i>).	Sim
- Gabaritos/ <i>Templates</i> criada/adaptados pelos autores (e.g. Listas Encadeadas via <i>Templates</i>).	Sim
- Uso de Tratamento de Exceções (<i>try catch</i>).	Não

7,5%

Tabela de Conceitos

Sobrecarga de:

- Construtoras e Métodos.	Sim	
- Operadores (2 tipos de operadores pelo menos).	Não	
Persistência de Objetos (via arquivo de texto ou binário)		
- Persistência de Objetos.	Sim	
- Persistência de Relacionamento de Objetos.	Não	

5%

Tabela de Conceitos

Virtualidade:	
- Métodos Virtuais.	Sim
- Polimorfismo	Sim
- Métodos Virtuais Puros / Classes Abstratas	Sim
- Coesão e Desacoplamento	Sim

10%

Tabela de Conceitos

- Espaço de Nomes (<i>Namespace</i>) criada pelos autores.	Não
- Classes aninhadas (<i>Nested</i>) criada pelos autores.	Sim
- Atributos estáticos e métodos estáticos.	Parcial- mente
- Uso extensivo de constante (<i>const</i>) parâmetro, retorno, método...	Sim

8%

Tabela de Conceitos

5%

- A classe Pré-definida <i>String</i> ou equivalente. & - <i>Vector</i> e/ou <i>List</i> da <i>STL</i> (p/ objetos ou ponteiros de objetos de classes definidos pelos autores)	Sim
- Pilha, Fila, Bifila, Fila de Prioridade, Conjunto, Multi-Conjunto, Mapa OU Multi-Mapa.	Sim
Programação concorrente	
- <i>Threads</i> (Linhas de Execução) no âmbito da Orientação a Objetos, utilizando Posix, C-Run-Time OU Win32API ou afins.	Não
- <i>Threads</i> (Linhas de Execução) no âmbito da Orientação a Objetos com uso de Mutex, Semáforos, OU Troca de mensagens.	Não

Tabela de Conceitos

- Funcionalidades Elementares. &
- Funcionalidades Avançadas como:
 - tratamento de colisões
 - duplo *buffer*

Sim

10%

-
- Programação orientada a evento em algum ambiente gráfico.

Sim

OU

- *RAD – Rapid Application Development* (Objetos gráficos como formulários, botões etc).

Tabela de Conceitos

- Compreensão, melhoria e rastreabilidade de cumprimento de requisitos. &	Sim	10%
- Diagrama de Classes em <i>UML</i> .	Sim	
- Uso efetivo (quicã) intensivo de padrões de projeto (particularmente GOF).	Sim	
- Testes a luz da Tabela de Requisitos e do Diagrama de Classes.	Sim	

Tabela de Conceitos

<ul style="list-style-type: none">- Controle de versão de modelos e códigos automatizado (via SVN e/ou afins) OU manual (via cópias manuais). &- Uso de alguma forma de cópia de segurança (backup).	Sim	5%
<ul style="list-style-type: none">- Reuniões com o professor para acompanhamento do andamento do projeto.	Sim	



Dúvidas?

