

Esse documento corresponde à análise do Diagrama de Classes da Sprint 2. Aponta correções a serem realizadas e possíveis erros. **Observações(OBS)** devem ser analisadas pelo membro responsável pelo artefato!

Autor(es): Aroldo Augusto Barbosa Simões

Data(última modificação): 06/11/2023

Documento de Análise do Diagrama de Classes

Itens a serem verificados

- Sintaxe e semântica:
 - Notação e linguagem de acordo com a UML;
 - Formalidade, linguagem adequada e erros de português.
- Diagrama de Classes:
 - Conformidade dos relacionamentos entre componentes;
 - Nomeação de classes, atributos e métodos;
 - Associações entre classes.
- Outros:
 - Correlação entre Diagrama de Classes e Diagrama de Casos de Uso.
 - Compatível com o padrão ECS(Entity Component System, em português: Sistema de Componente e Entidade).

Correções e Verificações Gerais

Sistemas

Não há correções e/ou verificações a serem feitas.

Entidades

- ☒ **EntidadeQuest**
 - ☒ **Verificar:** ~~aconselho trocar o nome do atributo “gerenciadorTarefa” por “gerenciadorTarefas”.~~
- ☒ **EntidadeMochila**
 - ☒ **Verificar:** ~~no relacionamento entre “EntidadeMochila” e “EntidadeItem” verificar multiplicidade da primeira classe.~~
 - Aparentemente o relacionamento está correto.
- ☒ **EntidadeItem**
 - ☒ **Verificar:** ~~necessidade de um “ComponenteItemGrafico”.~~
 - ☒ **Verificar:** ~~forma com que um colisor e instâncias de objetos são criados no mapa, de forma semelhante ao personagem principal.~~
- ☒ **EntidadeTarefaViagem**
 - ☒ **Verificar:** ~~o método “verificaEstadoConclusao” não deveria retornar um *boolean* ou um ID de estado?~~
 - O método foi verificado e não são necessárias modificações.

Componentes

- ☒ **ComponenteMetricasDaSessao**
 - ☒ **Verificar:** ~~método “incrementarDistanciaTotalPercorrida” e seus parâmetros, visto que esse cálculo deve ser feito, provavelmente, com distância entre pontos cartesianos.~~
 - O método foi verificado e em *sprints* futuras, provavelmente, será melhorado.
- ☒ **ComponenteGerenciadorTarefas**
 - ☒ **Verificar:** ~~métodos “verificaConclusaoDeTarefasViagem” e “verificaConclusaoDeTarefasColeta” não deveriam retornar um *boolean*?~~
 - Os métodos foram verificados e em *sprints* futuras, provavelmente, serão ajustados.

Introdução

- ☒ ~~1-Ortografia;~~
 - ☒ ~~Erro: Página em branco antes do tópico "Introdução".~~
- ☒ ~~2-Texto explicativo.~~

OBS: Bom texto!

OBS: O padrão apresentado no “Template Doc Básico” não foi seguido à risca, no próximo documento por favor seguir o padrão para “Título 1”, “Título 2”, “Subtítulo” e “Texto normal”.

CS00

- ☒ **1-Notação no padrão UML;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
- ☒ **2-Linguagem formal e sem erros de português;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
 - ☒ Descrição do Caso de Uso.
 - ☒ Documentação(-.pdf).
 - ☒ **Erro:** substituir título da seção por “CS00 – Realizar Autenticação”.
 - ☒ **Sugestão:** na subseção “EntidadeAluno” substituir “Com o objetivo de...” por “Visando...”.
 - ☒ **Erro:** na subseção “EntidadeAluno” no atributo “personagem” substitua “e portanto...” por “e, portanto...”
 - ☒ **Sugestão:** na subseção “EntidadeQuest” substituir “de acordo com os...” por “conforme os...” (duas ocorrências).
 - ☒ **Erro:** alinhe o título das subseções de acordo com o padrão, à direita da página (página 12).
 - ☒ **Erro:** na subseção “ComponenteMetricasDaSecao” substitua “descricao” por “descrição” no atributo “descricaoQuest”.
 - ☒ **Erro:** alinhe o título das subseções de acordo com o padrão, à direita da página (página 13).
 - ☒ **Erro:** na subseção “ComponenteGerenciadorTarefas” substituir “à uma” por “a uma” (duas ocorrências).
 - ☒ **Sugestão:** na subseção “SistemaControladorDeAcesso” substituir “de acordo com os...” por “conforme os...”.
 - ☒ **Erro:** na subseção “SistemaControladorDeQuests” adicione uma vírgula após “...concluídas do banco de dados, uma vez...”.
- ☒ **3-Classes;**
 - ☒ As classes representam as entidades no EGS.
 - ☒ Cada classe possui apenas dados (componentes) e não contém lógica.
 - ☒ As classes não possuem métodos que alterem o estado do jogo diretamente.
- ☒ **4-Componentes;**

- ☒ ~~Cada componente é uma estrutura de dados pura, contendo apenas atributos.~~
- ☒ ~~Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.~~
- ☒ **5-Herança;**
 - ☒ ~~Não há hierarquia de herança para os componentes.~~
 - ☒ ~~Herança é usada de forma mínima e apenas para as classes de sistemas.~~
- ☒ **6-Relacionamentos;**
 - ☒ ~~As entidades (classes) são associadas a seus componentes.~~
 - ☒ ~~Os sistemas operam em grupos de entidades com determinados componentes.~~
- ☒ **7-Métodos;**
 - ☒ ~~Os métodos nas classes são apenas para acessar ou manipular os componentes.~~
- ☒ **8-Operações do Sistema;**
 - ☒ ~~As operações do jogo são realizadas pelos sistemas, não pelas classes.~~
- ☒ **9-Renderização;**
 - ☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~
- ☒ **10-Conformidade com requisitos e regras de negócio.**

CS01

- ☒ **1-Notação no padrão UML;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
- ☒ **2-Linguagem formal e sem erros de português;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
 - ☒ Descrição do Caso de Uso:
 - ☒ **Erro:** coloque cada seção no início de uma página, substitua o título da seção por "CS01 - Movimentar Personagem" e mova-o para a próxima página (página 18).
 - ☒ **Erro:** na subseção "EntidadePersonagemPrincipal" substituir "e portanto..." por "e, portanto..."
 - ☒ **Sugestão:** na subseção "SistemaLogicaGeral" substituir "de acordo com as..." por "conforme as..."
- ☒ **3-Classes;**
 - ☒ As classes representam as entidades no EGS.
 - ☒ Cada classe possui apenas dados (componentes) e não contém lógica.
 - ☒ As classes não possuem métodos que alterem o estado do jogo diretamente.
- ☒ **4-Componentes;**
 - ☒ Cada componente é uma estrutura de dados pura, contendo apenas atributos.
 - ☒ Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.
- ☒ **5-Herança;**
 - ☒ Não há hierarquia de herança para os componentes.
 - ☒ Herança é usada de forma mínima e apenas para as classes de sistemas.
- ☒ **6-Relacionamentos;**
 - ☒ As entidades (classes) são associadas a seus componentes.
 - ☒ Os sistemas operam em grupos de entidades com determinados componentes.
- ☒ **7-Métodos;**
 - ☒ Os métodos nas classes são apenas para acessar ou manipular os componentes.
- ☒ **8-Operações do Sistema;**
 - ☒ As operações do jogo são realizadas pelos sistemas, não pelas classes.

☒ **9-Renderização;**

☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~

☒ **~~10-Conformidade com requisitos e regras de negócio.~~**

CS02

- ☒ **1-Notação no padrão UML;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
- ☒ **2-Linguagem formal e sem erros de português;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
 - ☒ Descrição do Caso de Uso.
 - ☒ **Erro:** coloque cada seção no início de uma página, substitua o título da seção por "CS02 - Interagir com o Mundo"
 - ☒ **Sugestão:** na subseção "ComponenteDadosTarefaColeta" substituir "de forma a..." por "para..."
 - ☒ **Erro:** adicione algum texto na seção "Sistemas relacionados" ou a exclua.
- ☒ **3-Classes(entidades);**
 - ☒ As classes representam as entidades no ECS.
 - ☒ Cada classe possui apenas dados (componentes) e não contém lógica.
 - ☒ As classes não possuem métodos que alterem o estado do jogo diretamente.
- ☒ **4-Componentes;**
 - ☒ Cada componente é uma estrutura de dados pura, contendo apenas atributos.
 - ☒ Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.
- ☒ **5-Herança;**
 - ☒ Não há hierarquia de herança para os componentes.
 - ☒ Herança é usada de forma mínima e apenas para as classes de sistemas.
- ☒ **6-Relacionamentos;**
 - ☒ As entidades (classes) são associadas a seus componentes.
 - ☒ Os sistemas operam em grupos de entidades com determinados componentes.
- ☒ **7-Métodos;**
 - ☒ Os métodos nas classes são apenas para acessar ou manipular os componentes.
- ☒ **8-Operações do Sistema(sistema);**
 - ☒ As operações do jogo são realizadas pelos sistemas, não pelas classes.
- ☒ **9-Renderização;**

- ☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~
- ☒ ~~10-Conformidade com requisitos e regras de negócio.~~

CS03

- ☒ **1-Notação no padrão UML;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
- ☒ **2-Linguagem formal e sem erros de português;**
 - ☒ Nomeação de classes.
 - ☒ Nomeação de atributos.
 - ☒ Nomeação de métodos.
 - ☒ Descrição do Caso de Uso:
 - ☒ **Erro:** coloque cada seção no início de uma página, substitua o título da seção por "CS03 - Visualizar Inventário".
 - ☒ **Erro:** na subseção "ComponenteMochilaGrafica" substituir "à bibliotecas..." por "às bibliotecas...".
 - ☒ **Erro:** adicione algum texto na seção "Sistemas relacionados" ou a exclua.
- ☒ **3-Classes(entidades);**
 - ☒ As classes representam as entidades no ECS.
 - ☒ Cada classe possui apenas dados (componentes) e não contém lógica.
 - ☒ As classes não possuem métodos que alterem o estado do jogo diretamente.
- ☒ **4-Componentes;**
 - ☒ Cada componente é uma estrutura de dados pura, contendo apenas atributos.
 - ☒ Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.
- ☒ **5-Herança;**
 - ☒ Não há hierarquia de herança para os componentes.
 - ☒ Herança é usada de forma mínima e apenas para as classes de sistemas.
- ☒ **6-Relacionamentos;**
 - ☒ As entidades (classes) são associadas a seus componentes.
 - ☒ Os sistemas operam em grupos de entidades com determinados componentes.
- ☒ **7-Métodos;**
 - ☒ Os métodos nas classes são apenas para acessar ou manipular os componentes.
- ☒ **8-Operações do Sistema(sistema);**
 - ☐ As operações do jogo são realizadas pelos sistemas, não pelas classes.
- ☒ **9-Renderização;**

- ☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~
- ☒ ~~10-Conformidade com requisitos e regras de negócio.~~

Referências

OBS: Adicione essa seção no documento e inclua as referências e documentos relacionados.

- ☒ ~~Padrão seguido.~~
- ☒ ~~Linguagem formal.~~
- ☒ ~~Descrição atualizada.~~

Referências

- [1] LanguageTool - Para correção de textos e verificação ortográfica. Pode ser acessado em: <<https://languagetool.org/pt-BR>>. Último acesso em: 06/11/2023.
- [2] Diagrama de Classes - Sprint 2- Versão do *commit* "[Diagrama de Classes - Sprint 2](#)". Pode ser acessado no repositório de produto da Equipe 5, link acima. Último acesso em: 06/11/2023.
- [3] ChatGPT. Pesquisa de Checklist para correção de Diagrama de Classes de projeto que utiliza o ECS. Referências: "Game Programming Patterns" por Robert Nystrom; "Entity Systems are the future of MMOG development" por Adam Martin; "Understanding Component-Entity-Systems" por Mike McShaffry. Pode ser acessado em: <<https://chat.openai.com>>. Último acesso em: 04/11/2023.