

Plano de Especificação Formal

1. Introdução

Este documento apresenta o plano de especificação formal do software de jogo interativo baseado em fases, com temática de reciclagem. O objetivo do jogo é incentivar práticas sustentáveis através de uma experiência educativa e divertida, em que o jogador coleta itens de lixo e realiza o descarte correto em centros de coleta e reciclagem, avançando entre fases progressivas.

Para garantir clareza, consistência e ausência de ambiguidades na definição dos requisitos, será utilizada a notação formal Z, representando de forma rigorosa os requisitos funcionais e não-funcionais do sistema.

2. Objetivos da Especificação

A especificação formal visa representar formalmente os requisitos funcionais (RF01–RF09) e não-funcionais (RNF01–RNF03), com o intuito de evitar ambiguidades na definição das regras de negócio e para permitir a verificação da consistência do sistema por meio da notação Z. Ademais, ela apoiará o processo de testes de software, fornecendo uma base formal para a criação de casos de teste, assim como servirá como referência para evolução do projeto, garantindo que futuras modificações respeitem as restrições e invariantes estabelecidos.

3. Escopo

O software consiste em um jogo baseado em fases, no qual o jogador deve coletar lixo e realizar o descarte correto em cada etapa para conseguir progredir. A partir disso, foi projetado para a mecânica envolver a progressão entre fases, cada uma com metas específicas a serem cumpridas para que o jogador avance. Como característica adicional, destaca-se a presença de um “boss” na fase 4, que obriga o reinício da fase em caso de contato com o personagem do usuário.

Dessa forma, pensado como uma forma de estímulo para o usuário manter a jogabilidade por um certo período, será possível realizar o registro do progresso do jogador, armazenando a última fase finalizada e permitindo a retomada a partir do menu “Carregar jogo”.

Assim, a especificação formal proposta dará ênfase nas operações principais relacionadas ao cumprimento dos objetivos das fases, ao processo de login e ao controle do progresso do jogador.

4. Notação Utilizada

Como o objetivo é formalizar as necessidades do software a ser desenvolvido será utilizada a Notação Z, uma linguagem formal baseada em lógica de predicados e teoria de conjuntos. Dessa forma, essa abordagem foi escolhida pois é baseada em modelagem de estados e operações, sendo adequado para representar as mudanças e transições de fases no jogo, com suporte para definição de invariantes, pré-condições e pós-condições.

5. Estrutura da Especificação Formal

Com a determinação da linguagem formal de especificação a sua organização será baseada na construção de conjuntos básicos como formas de definições (Usuário, Fase, Objetivos, Sacola). Além disso, com base na escolha da especificação será possível representar as mudanças e alternâncias de estados como: representação do progresso das fases do usuário, estado de login, fases a serem realizadas, estado das sacolas de coleta para possibilitar a resolução dos objetivos.

Dessa forma, com a determinação das invariantes para possibilitar a aplicação do padrão de regras será possível especificar com precisão as operações principais do usuário como: coleta de lixo, avanços de fase, reinício em casos de falha (fase final) e até nas finalizações das fases.

6. Operações

As operações principais previstas na especificação são baseados no ciclo de possíveis ações realizadas pelos usuários, sendo as etapas:

- Login: realizar a entrada do usuário no sistema por meio de nickname.
- Avançar de fase: avançar quando o jogador cumpre a meta da fase atual.
- Reiniciar fase (fase 4): reiniciar automaticamente quando o jogador falha contra o boss.
- Finalizar fase: concluir uma fase intermediária ou final.
- Salvar progresso: registrar a última fase alcançada, permitindo a retomada da fase.

7. Requisitos a serem especificados

Requisitos Funcionais (RF):

RF01: Usuário realizará entrada por nickname no sistema.

RF02: Usuário acessará às opções do menu inicial.

RF03: Usuário inicializará um novo jogo.

RF04: Usuário realizará a coleta de lixos.

RF05: Usuário avançará para o próximo objetivo.

RF06: Usuário usará o centro de reciclagem para realizar os objetivos finais de cada fase.

RF07: Usuário escolherá a sua próxima ação após finalizar cada fase intermediária do jogo.

RF08: Usuário reiniciará a fase ao encostar no boss.

RF09: Usuário finalizará as fases do jogo.

Requisitos Não-Funcionais (RNF):

RNF01: O sistema deve ter uma boa usabilidade, ser simples de usar.

RNF02: O sistema permite que seja possível realizar o suporte e adições.

RNF03: O sistema deverá possibilitar que suas funcionalidades ocorram como o esperado tendo em vista o pouco custo computacional de operação.

8. Metodologia

A abordagem utilizada envolve a identificação e a explicação de cada requisito funcional do sistema de forma clara e compreensível. Em seguida, esses requisitos são transformados em esquemas formais por meio da notação Z. Define-se o estado global do sistema e os invariantes que devem ser preservados em todas as operações, os quais servem de fundamento para a formalização. Em esquemas Z, as operações que alteram o estado do sistema são representadas, tendo em vista sempre a conformidade com os invariantes definidos.

Em seguida, é feita a análise de consistência entre os esquemas formais para garantir que os invariantes sejam mantidos e que as operações definidas não apresentem contradições. Concluindo, cada operação formalizada está relacionada a casos de teste originados dos requisitos, assegurando que a futura implementação possa ser verificada de acordo com a especificação formal.

9. Critérios de Aceitação

A especificação formal será considerada finalizada quando todos os requisitos funcionais (RF01–RF09) estiverem adequadamente formalizados e os requisitos não funcionais (RNF01–RNF03) devidamente incluídos na descrição geral do sistema. Ademais, é essencial que os esquemas e modelos não contenham contradições internas, a fim de garantir a consistência do documento. Adicionalmente, é necessário que a especificação sirva como uma referência confiável para a criação de casos de teste de software, garantindo, dessa forma, a completude e a qualidade necessárias para o avanço das fases de desenvolvimento e verificação do sistema.