

Versão	Autores	Data	Atividade
1.0	Jean Leopoldo, Jonathan Kuntz	02/04/2018	Levantamento das especificações de requisitos.
1.1	Jean Leopoldo, Jonathan Kuntz	12/04/2018	Alterações na formatação para melhora de legibilidade.
1.2	Jean Leopoldo, Jonathan Kuntz	16/04/2018	Melhora na descrição dos requisitos funcionais.
1.3	Jean Leopoldo, Jonathan Kuntz	16/04/2018	Melhoria nas regras do jogo.
1.4	Jean Leopoldo, Jonathan Kuntz	24/04/2018	Recuperar esboços GUI.
2.0	Jonathan Kuntz	04/07/2018	Alterações para condizer com a realidade do jogo implementado.

1 – Introdução

1.1 – Objetivo

- Desenvolver um jogo por turnos em que haverá dois jogadores, um contra o outro.
- Título do jogo/nome da aplicação: Ice Badge

1.2 – Referência

- Fire Emblem: https://pt.wikipedia.org/wiki/Fire_Emblem

1.3 – Regras do jogo

O nome do jogo é **Ice Badge**.

1.3.1 – Objetivos

- Chegar com um personagem até a área da base inimiga para tomá-la.
- Matar todos os inimigos

1.3.2 – Campo

- Será uma matriz com tamanho 16x16 (256 tiles).
- Terá restrições de acordo com o terreno: Água, Árvores, Montanha, Pedra.

1.3.3 – Personagens

Os personagens possuem características com valores padrão, que aumentam de acordo com o clã de escolha do jogador.

Os personagens possíveis são:

- Swordsman
- Archer
- Clerig
- Bard

1.3.3.1 – Características dos personagens (atributos)

- **Capacidade de movimento:** cada personagem possui um valor padrão de capacidade de movimento em quantidade de casas. Durante a partida o personagem só poderá caminhar no máximo essa quantidade de casas no mapa.
- **Valor de vida:** cada personagem possui um valor de vida padrão. Durante a partida, quando o valor da vida do personagem se esgota, ele é eliminado.
- **Valor de dano de ataque:** cada personagem possui um valor de dano de ataque padrão. Durante a partida, quando o valor da vida do personagem se esgota, ele é eliminado.
- **Valor de área de atuação:** cada personagem possui um valor de área de ataque. Durante a partida, quando a ação 'Atacar' esta área deve ser mostrada e ela representa a distância máxima possível de ataque do personagem selecionado. Apenas personagens dentro desta área poderão sofrer a ação, caso selecionados.

1.3.3.2 – Tabela de atributos dos personagens

Atributos / Classes	Swordsman	Archer	Clerig	Bard
Movimento	5	2	5	6
Vida	40	30	10	20
Dano	25	15	8	10
Área ataque	5	8	X	2

1.3.4 – Bases

1.3.4.1 – Base Principal (Inicial)

- Cada jogador possui apenas uma base principal.
- A posição das bases serão fixas, sendo que, obrigatoriamente, as bases ficarão em lados opostos do tabuleiro.
- A posição inicial dos personagens ao serem criados será perto da área da base.
- A base principal é composta de apenas uma (1) posição.

1.3.5 – Início do Jogo

- Cada usuário deverá montar seu time de acordo com suas preferências.
- O usuário poderá iniciar o jogo com 6 personagens de sua escolha, antes do início da partida.

1.3.6 – Durante o jogo (turno)

- O turno de um jogador acaba quando o mesmo passa a vez ou em que acabe seus movimentos.
- Durante o turno, o jogador não pode atacar duas vezes nem movimentar-se duas vezes.

1.3.7 – Fim do jogo

O jogo finaliza quando:

- A base principal do oponente é tomada
- Um dos jogadores desiste
- Todos os personagens do oponente são eliminados

2 – Visão Geral

2.1 – Arquitetura do Programa

- Cliente-servidor
- Distribuído.

2.2 – Premissas

- Programa desenvolvido em Java.
- Interface gráfica bidimensional (2D).
- Servidor NetGamesNRT.
- Modelagem feita no Visual Paradigm.

3 – Requisitos de Software

3.1 – Requisitos funcionais

- **Menu inicial:** o menu inicial da aplicação possui botões para realizar as operações iniciais.
 - **Conectar:** o menu deve haver um botão “Conectar” para que o usuário consiga conectar-se no servidor através do *NetGames*¹ ao clicá-lo.
 - **Desconectar:** o menu deve haver um botão “Desconectar” para que o usuário consiga conectar-se no servidor através do *NetGames*² ao clicá-lo.
 - **Iniciar:** o menu deve haver um botão “Iniciar” ou “Start Game” para que o usuário consiga iniciar uma partida multiplayer, levando à próxima tela da aplicação.
 - **Sair:** o menu deve haver um botão ‘Sair’ para que o usuário consiga sair da aplicação.
- **Menu pré-partida:** após ter clicado o botão de ‘Iniciar’, o usuário vai para uma tela pré-partida, onde pode realizar escolhas estratégicas antes do início da partida.
 - **Seleção de personagens iniciais:** o jogador começa com uma quantidade de 6 personagens, nesta tela ele pode selecionar seus personagens.
- **Durante a partida:**
 - **Seleção de personagem:** o usuário deve poder selecionar um personagem de seu time, ao clicar sobre ele, para que sejam exibidas as possibilidades de ações disponíveis para aquele personagem.
 - **Ação de personagem “Movimentar”:** personagem selecionado pode se movimentar no mapa respeitando suas limitações de movimento e as do mapa. O mesmo personagem não pode se movimentar mais que uma vez na mesma rodada.
 - **Ação de personagem “Atacar”:** personagem selecionado pode atacar um personagem inimigo ao selecioná-lo, desde que este esteja dentro de sua área de ataque. A quantidade de vida deduzida do personagem atacado será equivalente à força de ataque do personagem atacante. Caso a vida do personagem atacado chegue às 0 ou menos após o ataque, ele será eliminado da partida.
 - **Passar o turno:** durante sua vez, o jogador deve ter acesso a um botão que passa o turno para o jogador inimigo ao clicá-lo.
 - **Fim de jogo:** o jogo pode terminar a qualquer momento que um dos eventos seguintes ocorram.

¹ *NetGames: Framework criado por Leonardo de Souza Brasil:*
R<https://www.inf.ufsc.br/~ricardo.silva/INE5417e5608/NetGamesNRT.htm>

² *NetGames: Framework criado por Leonardo de Souza Brasil:*
R<https://www.inf.ufsc.br/~ricardo.silva/INE5417e5608/NetGamesNRT.htm>

- **Tomar base (ganhar):** quando algum personagem entra na área da base do inimigo, o jogo termina e o jogador possuidor deste personagem ganha a partida.
- **Eliminar personagens do oponente (ganhar):** quando todos os personagens do jogador oponente forem eliminados, o jogador que os eliminou ganhará o jogo, finalizando a partida.
- **Sair do jogo:** durante todo o jogo, qualquer um dos jogadores deve ter acesso a um botão que saia do jogo, finalizando a aplicação após ao clicá-lo.

3.2 – Requisitos não-funcionais

- **Interface gráfica:** a aplicação deverá ter uma interface gráfica para que fique intuitiva a interação do usuário com a aplicação. O menu inicial do programa consistirá em um botão para que o usuário inicie o jogo ou saia do mesmo.
- **Especificação baseada em UML:** a especificação do projeto será realizada utilizando o UML v2.
- **Suporte a conexão utilizando o *NetGames*:** a aplicação deve permitir a função multiplayer (player vs player) utilizando o *NetGames* para realizar a conexão.

4 – GUI



