****

**UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC**

**CENTRO TECNOLÓGICO – CTC**

**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE**

**Franco Camelo Aguzzi,**

**Lohan Rodrigues Silva,**

**Ricardo Romero.**

**ESPECIFICAÇÃO DE REQUISITOS - ELEMENT PO**

**FLORIANÓPOLIS, SC**

**2021**

| **Versão** | **Autores** | **Data** | **Ação** |
| --- | --- | --- | --- |
| 0.1 | Franco Camelo Aguzzi  Lohan Rodrigues Silva  Ricardo Romero | 12/07/2021 | Especificação inicial |
| 0.2 | Franco Camelo Aguzzi  Lohan Rodrigues Silva  Ricardo Romero | 31/08/2021 | Ajustes para novo formato do jogo. |
| 1.0 | Franco Camelo Aguzzi  Lohan Rodrigues Silva | 26/09/2021 | Versão final. |

1. **INTRODUÇÃO**
   1. OBJETIVO DO DOCUMENTO

Este documento busca especificar o sistema Element Po, clarificando como o software deve ser desenvolvido e testado. Também é um material de consulta para desenvolvedores e financiadores do sistema, sempre buscando mantê-lo no escopo definido.

* 1. PÚBLICO ALVO

Este documento destina-se aos desenvolvedores do sistema e aos *stakeholders*.

1. **DESCRIÇÃO GERAL DO SISTEMA**
   1. VISÃO GERAL

Element Po é um jogo desenvolvido pelos graduandos Franco Camelo Aguzzi, Lohan Rodrigues Silva e Ricardo Romero para a disciplina INE5608 (Análise e Projeto de Sistemas), do curso em Sistemas de Informação da Universidade Federal de Santa Catarina.

O jogo é composto por um baralho com 20 cartas iniciais, sendo que cada uma destas cartas possui um elemento (fogo, ar, água ou terra) e um nível (1, 2, 3, 4 ou 5). Element Po é um jogo para ser jogado entre dois jogadores que consiste em ganhar em forma de batalhas as posições de um campo da velha, isso é, completar alguma das posições horizontais, verticais ou diagonais com suas cartas/vitórias.

Ao início de uma nova partida cinco cartas são distribuídas para cada participante, sendo o jogador cadastrado como “Jogador 1” aquele que tem o direito de iniciar a partida. Após iniciar a partida, esse jogador tem que escolher uma da suas cartas da mão e posicionar ela no campo da velha. “Jogador 2” deve fazer o mesmo.

Caso o “Jogador 2” tenha escolhido a mesma posição que o “Jogador 1”, feita a comparação entre estas, que se dá da seguinte forma:

Caso o elemento de uma das cartas se sobressaia ao elemento da outra, o nível daquela é multiplicado por dois.

Elemento Água se sobressai sobre elemento Fogo;

Elemento Fogo se sobressai sobre elemento Ar;

Elemento Ar se sobressai sobre elemento Terra;

Elemento Terra se sobressai sobre elemento Água;

Após multiplicação, ou não, dos níveis das cartas é feita a comparação destes níveis, que se dá de forma numérica, ou seja, o maior número vencerá.

Tendo um vencedor da rodada, a posição do campo é assumida pelo J1 (Jogador 1) ou pelo J2 (jogador 2), sendo em seguida feita a verificação se foi preenchido completamente alguma das posições do campo da velha.

Caso não tenha duas cartas na mesma posição, a partida continua com os jogadores escolhendo cartas e posições até que exista essa comparação ou “batalha”.

Caso as primeiras 5 cartas dos dois jogadores tenham acabado, e não foi completado o jogo da velha, novas cinco cartas são distribuídas a cada um dos jogadores com intuito de completar a partida. Caso acabem as cartas, o baralho é redistribuído a partir de todas as cartas do jogo.

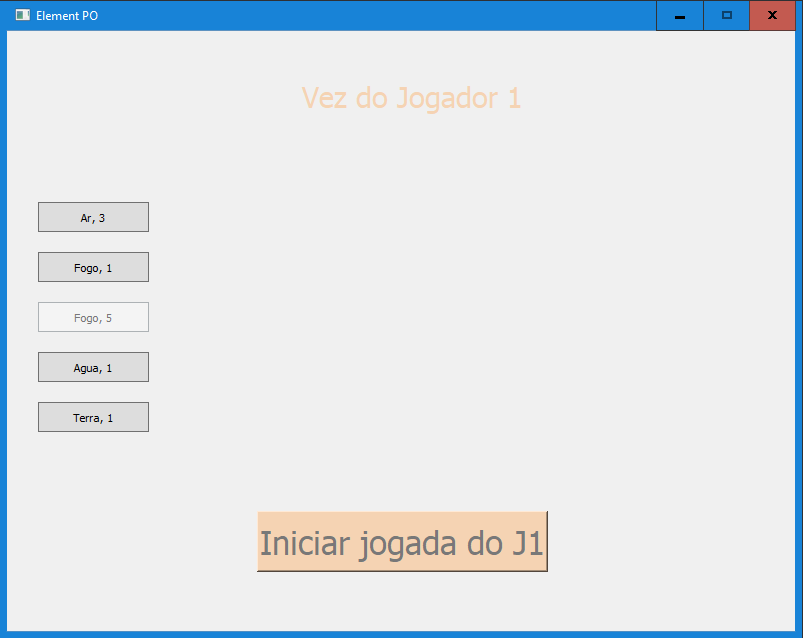
O objetivo neste jogo é ganhar as batalhas com intuito de completar o campo da velha.

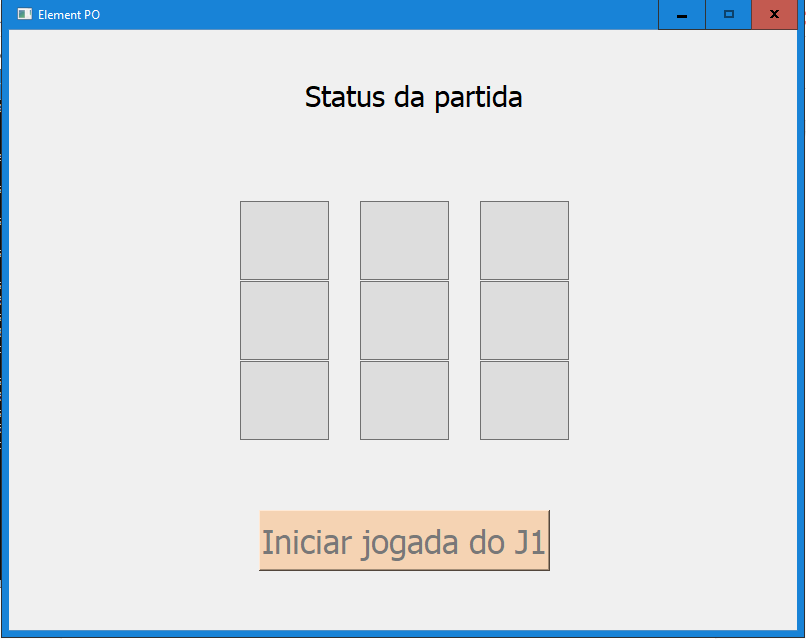
* 1. ARQUITETURA

Programa orientado a objetos e multiusuário (dois jogadores).

* 1. PREMISSAS DE DESENVOLVIMENTO
     + A implementação deverá ser na linguagem Python.
       1. Deve ser compatível com o versão [python-3.9.2](https://www.python.org/downloads/release/python-392/)
     + Deverá ser entregue a modelagem do software em UML 2 produzida com a ferramenta Visual Paradigm (*Community Edition*).
     + O programa deve apresentar uma interface gráfica bidimensional.
  2. INTERFACE GRÁFICA

Encontra-se abaixo um modelo da interface gráfica a ser desenvolvida. Destaca-se, entretanto, que são mostrados apenas elementos referentes ao estado de uma partida já iniciada. A interface será atualizada conforme o momento atual da interação do(s) usuário(s) com o programa.

****

****

1. **REQUISITOS DE SOFTWARE**
   1. REQUISITOS FUNCIONAIS
      * **RF1 - Iniciar partida:** O programa deve permitir ao usuário clicar em um botão na tela principal para iniciar uma partida de Element Po.
      * **RF2 - Escolher cartas para repartir:** No início do jogo, e no início da rodada #5 devem ser escolhidas de forma aleatória 10 cartas do baralho. Quando uma carta é selecionada para repartição, ela é retirada do baralho.
      * **RF3 - Repartir cartas:** Após escolher cartas para repartir, o programa deve repartir 5 delas para cada jogador.
      * **RF4 - Iniciar jogada:** O programa deve permitir ao usuário clicar em um botão na tela principal para iniciar a jogada. Ao iniciar a jogada, ele poderá escolher uma das cartas para jogar.
      * **RF5 - Selecionar carta:** O programa deve permitir ao usuário clicar num botão para selecionar uma carta que será posicionada no campo da velha. Após selecionar a carta, ela se torna indisponível para nova utilização.
      * **RF6 - Selecionar campo:** O programa deve permitir ao usuário clicar em um botão que referencia as posições do jogo da velha o qual a carta selecionada no RF5 será posicionada.
      * **RF7 - Calcular resultado de batalha:** O programa deve comparar a força e elemento das cartas escolhidas por cada jogador, e definir o vencedor no campo selecionado para batalha.  
        Quando uma das cartas é de um elemento vantajoso, seu nível de força é multiplicado por dois, seguindo a regra: Água > Fogo, Fogo > Ar, Ar > Terra e Terra > Água.  
        Após finalizar o cálculo, será necessário aumentar um turno.
      * **RF8 - Verificação de vitória:** Todos os turnos é necessário validar se algum dos jogadores completou uma das 3 posições horizontais, verticais ou diagonais do campo. Caso algum dos jogadores tenha essa condição, deve ser anunciado esse jogador como vitorioso.
   2. **REQUISITOS NÃO FUNCIONAIS**
      * **Especificações de projeto:** O programa deverá ser escrito em *Python*, deverá ser compilado de forma a ser compatível com a versão python-3.9.2 e deve seguir uma modelagem UML 2 conformante com a metodologia vista em aula.
      * **Interface gráfica:** O programa deve possuir uma interface gráfica que represente o estado compartilhado do jogo. Esta interface será implementada em *Python* com a biblioteca *PyQT*.
      * **Características da linguagem:** A linguagem do jogo é intuitiva e de fácil manutenção, *Python* é uma linguagem de programação imperativa, interpretada, de alto nível e com tipagem forte e dinâmica. Ela é considerada uma linguagem multi-paradigma, pois aceita diferentes formas de programação. A saber: programação orientada a objetos, procedural e funcional.