



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

Caroline Martins Alves
Henrique Meireles Costa Gomes

Análise de Requisitos: *Kein*

Florianópolis
2020

Projeto Jogo Kein
Especificação de Requisitos de Software

REQ_PR_jogoKeinOnline_2020set16

Versão: 2.0

Data: 26/10/2020

Versão	Autores	Data	Ação
1.0	Caroline Martins Alves e Henrique Meireles Costa Gomes	24/09/2020	Análise dos requisitos
2.0	Caroline Martins Alves e Henrique Meireles Costa Gomes	22/10/2020	Análise dos requisitos
3.0	Caroline Martins Alves e Henrique Meireles Costa Gomes	15/12/2020	Análise dos requisitos

SUMÁRIO

INTRODUÇÃO	3
OBJETIVO DO DESENVOLVIMENTO	3
DEFINIÇÕES	3
VISÃO GERAL DO SISTEMA	5
ARQUITETURA DE SOFTWARE	5
REQUISITOS DA APLICAÇÃO	6
REQUISITOS FUNCIONAIS	6
REQUISITOS NÃO-FUNCIONAIS	7

1. INTRODUÇÃO

1.1. OBJETIVO DO DESENVOLVIMENTO

O software a ser desenvolvido neste trabalho trata-se de um jogo multiplayer, que será implementado em Java com o apoio da ferramenta NetGames, para que as jogadas dos participantes sejam realizadas em turnos. O nome dado ao jogo é Kein e sua dinâmica requer basicamente encaixar peças.

A partir do desenvolvimento desse software, espera-se criar um programa capaz de simular uma experiência virtual do jogo original chamado Peg Game, The Tricky 6. Além disso, procura-se proporcionar diversão e entretenimento para todas as pessoas que gostam de jogos rápidos e cheios de reviravoltas.

1.2. DEFINIÇÕES

A estrutura do jogo Kein consiste, essencialmente, em uma caixa de madeira que seja internamente oca, pequenos palitos cilíndricos também de madeira (que lembram uma cavilha) e um dado normal. A caixa tem uma tampa que possui 6 furos numerados, sendo que o furo de número 6 fica localizado no centro da tampa e tem tamanho um pouco maior do que os demais, o restante dos furos são do tamanho dos palitos para que seja possível encaixá-los e mantê-los nesses buracos. A quantidade mínima de jogadores é 2 e a máxima 6. No início de cada partida, os participantes recebem 4 palitos, independente da quantidade de participantes.

Figura 1: Jogo original - Peg Game



Fonte: https://gokitoy.com/media/shop/shop/56799_s.jpg

Passos e regras para jogar: O jogo é iniciado quando houver dois participantes conectados e um destes acionar a determinação de início de partida. A partir disso, cada um dos jogadores recebe 5 palitos e então quem deu início na partida - portanto, primeiro jogador - lança o dado e confere o número tirado. Na sequência, esse jogador deve encaixar seu palito no buraco respectivo ao número do dado e dar a vez ao outro participante. Porém, se o buraco já estiver preenchido, o jogador deve coletar a peça do furo para si e passar a vez ao outro jogador. Caso o número da face do dado seja 6, o jogador deve depositar a peça dentro da caixa, uma vez que esse furo possui tamanho maior do que os outros.

As regras acima aplicam-se para os cenários normais. Entretanto, existe uma regra para uma situação especial. No caso do número obtido no dado ser igual ao valor obtido na rodada passada, o jogador irá transferir um de seus palitos ao adversário e também descartar outro palito (na posição 6). Deste modo, nessa situação o jogador poderá eliminar até 2 palitos.

Vale lembrar que, inicialmente, a proposta era um jogo com mais de dois players, além de outras regras adicionais que envolviam interação do usuário com a interface (além de apenas lançar o dado), porém, devido ao tempo hábil, não foi possível cumprir com todas essas exigências e assim para que a entrega ocorresse de modo consistente - estando modelagem, implementação e requisitos alinhados - essas condições foram removidas.

2. VISÃO GERAL DO SISTEMA

2.1. ARQUITETURA DE SOFTWARE

A arquitetura utilizada será cliente-servidor distribuído.

2.2. PREMISSAS DE DESENVOLVIMENTO

- O programa deve utilizar Netgames como suporte para a execução distribuída;
- O projeto será construído com Visual Paradigm;
- Deve ser possível executar o programa em qualquer dispositivo que tenha suporte a máquina virtual Java.

3. REQUISITOS DA APLICAÇÃO

3.1. REQUISITOS FUNCIONAIS

What?	Why?	Who?	When?	Where?	How?
Lançar o dado	Para o programa saber em qual buraco deve colocar a peça	O jogador	Sempre que for sua vez de jogar	Na interface	Clicando no botão que retornará um valor de 1 a 6. De acordo com o número obtido, a peça será encaixada, mas somente se o furo estiver livre, se estiver ocupado, a peça do buraco vai para o jogador que lançou o dado. E no caso de ser o número 6 a peça sempre será depositada dentro da caixa.
Conectar	Para que os jogadores possam conectar-se ao servidor do Netgames	O jogador	Antes do início de cada partida	Na interface	Clicando no menu inicial e selecionando a opção "Conectar"
Desconectar	Para que os jogadores possam desconectar-se do servidor Netgames	O jogador	Após terminar o jogo	Na interface	Clicando no menu inicial e selecionando a opção "Desconectar"
Iniciar partida	Para que o jogador possa dar início a uma nova partida	O jogador	Após conectar-se ao servidor do Netgames	Na interface	Clicando no botão de "Iniciar partida"

Receber determinação de início	Para que haja o definição de que a partida foi iniciada	O servidor Netgame s	Após um dos jogadores conectados clicar em "Iniciar partida"	No servidor Netgames	Setando a variável de início de partida como true
Receber jogada	Para que após um dos jogadores realizar seu movimento , este seja recebido pelo outro jogador	O servidor Netgame s	Após um dos jogadores realizar um lance	No servidor Netgames	Guardando o movimento realizado pelo jogador e enviando-o aos demais

3.2. REQUISITOS NÃO-FUNCIONAIS

- Ser implementado em Java;
- Permitir jogadas em turnos;
- A interface deve apresentar uma espécie de caixa com furos numerados de 1 a 6.