

Documentação de Testes

Trabalho Prático Grupo 15

Fábio Mendes, 8170157

José Baltar, 8170212

Rodrigo Coelho, 8170282

Índice

1. Introdução	2
1.1. Propósito do documento	2
2. Acerca dos testes	3
3. Tabelas de testes.....	4

1. Introdução

1.1. Propósito do documento

Este documento tem como propósito a demonstração e organização de diversos testes executados durante o período de desenvolvimento do projeto. Neste documento, descrevem-se os testes realizados, indicando o problema que foi detectado e a correção efetuada de forma a permitir que o projeto fosse levado a bom porto sem bugs de maior que não permitissem a, em larga medida, correta execução da aplicação ou de alguma das suas componentes.

Neste documento só os testes de considerada importância se encontram presentes.

2. Acerca dos testes

Os testes podem ser divididos em diversas categorias, as quais veremos mais para a frente. De facto, alguns testes foram “Ad-hoc” descritos pelo grupo, uma vez que não se encontravam em nenhuma definição formalizada.

Isto sucede porque, dada à complexidade da arquitetura e da sua testagem, certos problemas e funções foram testadas em ambiente simulado, já com as diversas componentes integradas. Além disso, foi desenvolvida em específico uma página Web que comunicava com o Servidor de WS, isto é, o GameServer, cujo propósito era poder testar os diversos eventos. No entanto, devido à quantidade de verificações lógicas necessárias, a maior parte dos testes, sobretudo os de grande relevo, importância e complexidade, foram testados juntamente com o Cliente de Jogo, ou seja, um teste de sistema já integrado.

A API é, notoriamente, mais fácil de testar, uma vez que utilizando a plataforma Swagger, é possível testar os diversos endpoints e o seu funcionamento de uma maneira não muito dissemelhante a um teste unitário, embora seja mais “Ad-hoc”.

A maior parte do *frontend*, isto é, tanto as partes em browser como as partes em Unity, foram testadas em ambiente integrado com as componentes de *Backend*. Notoriamente, o cliente de jogo em Unity foi testado simultaneamente com o servidor de jogo, embora sejam considerados testes diferentes.

3. Tabelas de testes

Tipo de Teste			Descrição
Testes GL	Game	Logic	Testes <i>ad-hoc</i> utilizando uma página de comunicação WebSocket que permite testar a ligação de uma qualquer cliente ao Servidor de jogo.
Testes Game Client GC			Testes <i>ad-hoc</i> utilizando um build do jogo Unity que permite testar a ligação de uma qualquer cliente ao cliente e sua comunicação com o servidor.
Testes Client Browser CB			Testes <i>ad-hoc</i> utilizando diretamente o front-end e a sua ligação ao servidor web
Testes Unitários TU			Testes desenvolvidos utilizando a abordagem de testes unitários para um qualquer componente do sistema.
Testes REST API API			Testes utilizado a plataforma de documentação Swagger (ou semelhantes) que permite a testagem dos diversos endpoints desenvolvidos na REST API
Testes de Integração Ad-Hoc TI			Testes utilizando uma página de comunicação WebSocket e o Cliente Front-end em Angular que permite testar a integração entre as diversas
Testes Externos TE			Testes desenvolvidos utilizando uma aplicação externa que irá correr <i>snippets</i> de código de difícil testagem com a utilização da aplicação completa ou testes unitários
Teste de Sistema TS			Testes que simulam uma execução normal, em ambiente controlado e verificado, das componentes do sistema a trabalharem entre si e em conjunto. Por não ter sido feito um deploy, este é testado em rede privada Hamachi.

ID	GL-01
Tipo de Teste	GL
Descrição	Teste de Ligação de Clientes
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-01.1
Tipo de Teste	GL
Descrição	Teste de Recebimento de Informação
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-02
Tipo de Teste	GL
Descrição	Efetuar Eventos
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-02.1
Tipo de Teste	GL
Descrição	Efetuar Ataques
Resultado	Erros, Crash de servidor
Bugs identificados	Má geração de random para destruição de unidades
Correções efetuadas	Modificado o random de forma a não sair de Bounds, colocado Catch para a exceção

ID	GL-02.2
Tipo de Teste	GL
Descrição	Sincronização dos timers

Resultado	Erros
Bugs identificados	Má sincronização dos temporizadores
Correções efetuadas	Gestão de Temporizadores melhorada

ID	GL-02.3
Tipo de Teste	GL
Descrição	Sincronização dos timers
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-02.4
Tipo de Teste	GL
Descrição	Capitulação de uma facção
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-03.1
Tipo de Teste	GL (Matchmaking)
Descrição	Colocar 4 clientes (todos em lobby rooms diferentes) a iniciar procura por um jogo através de matchmaking e deixá-los a procurar até uma sala ser criada com eles.
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-03.2
Tipo de Teste	GL (Matchmaking)
Descrição	Colocar 4 clientes (dois lobby rooms cada sala com 2 clientes) a iniciar procura por um jogo através de matchmaking e deixá-los a procurar até uma sala ser criada com eles.
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-03.3
Tipo de Teste	GL (Matchmaking)
Descrição	Colocar 1 cliente (1 lobby room) a procurar por um jogo e depois a cancelar a procura. Depois verifica-se se o lobby do jogador foi removido com sucesso da matchmaking Queue.
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	GL-03.4
Tipo de Teste	GL (Matchmaking)
Descrição	Colocar 5 clientes (dois lobby rooms cada sala com 2 clientes e um lobby room com 1 cliente) o lobby que contem 1 jogador e um dos lobbies que contem 2 jogadores iniciam a procura por jogo. Depois disto o terceiro lobby inicia a procura e o lobby com 1 jogador cancela a procura. O resultado esperado é um lobby de jogo que contém os dois lobbies com 2 jogadores cada.
Resultado	Sucesso

Bugs identificados	-
Correções efetuadas	-

ID	TI-01
Tipo de Teste	TI
Descrição	Um jogador sair da Room e voltar a conectar-se
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	API-01.1
Tipo de Teste	API
Descrição	Testes à API Matchdata
Resultado	Erro 500
Bugs identificados	Construtor mal efetuado no modelo de Request
Correções efetuadas	Apagado o Construtor

ID	API-01.2
Tipo de Teste	API
Descrição	Testes à API Matchdata
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	API-02
Tipo de Teste	API
Descrição	Testes à API Convites
Resultado	Sucesso
Bugs identificados	
Correções efetuadas	

ID	API-03
Tipo de Teste	API
Descrição	Testes à API Pedidos de amizade
Resultado	Sucesso
Bugs identificados	
Correções efetuadas	

ID	API-03
Tipo de Teste	API
Descrição	Receber Lista de pedidos de amizade
Resultado	Conflito com front-end
Bugs identificados	A resposta enviada para o front-end era apenas uma lista de strings com os nomes dos amigos em vez de ser um objeto.
Correções efetuadas	Criado modelo na API para enviar lista com instancias desse modelo.

ID	TE-01
Tipo de Teste	TE
Descrição	Testar pedidos POST HTTP por http factory e ligação à API
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	TS-01.1
Tipo de Teste	TS
Descrição	Testar lançamento do jogo
Resultado	Bugs
Bugs identificados	Problema com a forma de lançamento em href
Correções efetuadas	Corrigida forma de lançamento por TypeScript

ID	TS-01.2
Tipo de Teste	TS
Descrição	Testar lançamento do jogo
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	TS-02.1
Tipo de Teste	TS
Descrição	Testar comunicação dos servers em rede local
Resultado	Erros em WSS
Bugs identificados	Não comunicação com GameServer devido a WSS
Correções efetuadas	Corrigida forma de ligação, remover WSS para WS e lançamento de angular com host 0.0.0.0

ID	TS-02.2
Tipo de Teste	TS
Descrição	Testar comunicação dos servers em rede local
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	TS-03.1
Tipo de Teste	TS => Teste geral do jogo 1
Descrição	Série de Testes Gerais ao jogo
Resultado	Crash de servidor
Bugs identificados	Registos errados no Ficheiro JSON Estado bélico com sinais trocados

	Problemas com a manutenção de exércitos após combate Problemas na destruição de exércitos
Correções efetuadas	Corrigidos ficheiros JSON Corrigida ordem das fações Manutenção modificada para novo dos exércitos Corrigido random nos exércitos

ID	TS-03.2
Tipo de Teste	TS => Teste geral do jogo 2
Descrição	Série de testes gerais ao jogo
Resultado	Crash de servidor, erro 500 API
Bugs identificados	Verificações de derrota erradas Capitulação não atualizava contador de fações derrotadas Servidor não atualizava o cliente com a informação do jogo acabada Matchdata para o vencedor não estava a ser criada Ligação ao servidor estava a interagir com um httpClient nulo Servidor tentava interagir com sala apagada no fim do jogo quando o jogar saia
Correções efetuadas	Verificação de derrota corrigida e modificada Capitulação a contar fação derrotada corrigida Matchdata para vencedor criada Servidor envia informação atualizada para o cliente antes de apagar o room httpClient usa o da instância Matchdata para a API Servidor deixa de tentar interagir com uma sala apagada quando o cliente sai do cliente

ID	TS-03.3
Tipo de Teste	TS => Teste Geral de jogo 3
Descrição	Série de testes gerais ao jogo
Resultado	Sucesso
Bugs identificados	-
Correções efetuadas	-

ID	TS-04.4
Tipo de Teste	TS => Teste Geral de jogo 4
Descrição	Série de testes gerais ao jogo (Declaração de guerra através do client Unity)
Resultado	Mau funcionamento do client e por vezes crash do servidor.
Bugs identificados	Menus e opções que deveria poder executar ou não apareciam (declarar guerra e declarar paz) e apareciam opções que causavam crash do servidor por estarem disponíveis quando não deviam (mover exército)
Correções efetuadas	Alterações na logica do unity (alguns comportamentos esperados, e que não tinham sido considerados foram implementados, nomeadamente so permitir mover para uma região inimiga depois de declarar guerra). Quanto às alterações feitas no servidor, elas estão apontadas no TS-03.1