4
Testes Sistêmicos

O objetivo deste capítulo é apresentar os testes realizados para ajudar a identificar erros na implementação do Ginga-NCL em dispositivos portáteis. Foram realizados apenas testes sistêmicos, onde várias funcionalidades são testadas em conjunto. O objetivo foi verificar o comportamento da execução levando-se em conta a conjunção das funcionalidades, ao invés de se avaliar apenas o funcionamento pontual de cada uma delas, como é feito nos testes unitários. A melhor forma de se fazer esse tipo de teste é desenvolver uma aplicação NCL propriamente dita, onde vários recursos são exercitados ao mesmo tempo. Por exemplo, uma simples aplicação NCL pode por a teste a execução do parser e a criação de regiões, descritores, mídias e portas.

Os testes realizados, portanto, consistem na execução de aplicações NCL que, ao final, também servem como prova de conceito da implementação. Para cada uma das aplicações, serão apresentados a sua dinâmica, os exibidores usados e os problemas mais relevantes identificados, juntamente com as suas respectivas soluções. Também serão apresentadas telas ilustrativas de cada uma dessas aplicações quando executadas em um emulador de dispositivo Symbian S60. Além do emulador, os testes também foram executados em dois dispositivos reais, como descrito no Anexo A.

4.1.
Formula 1

A primeira aplicação desenvolvida para a implementação do Ginga-NCL para dispositivos portáteis foi uma sobre corrida de Fórmula 1. Nela, um vídeo de fórmula 1 é exibido em um pouco mais da metade da tela do dispositivo. O resto do espaço é usado para apresentar três opções de interação. A Figura 9 mostra esse cenário.

Figura 9: Aplicação Fórmula1.

As interações podem ser feitas pressionando-se as teclas 1, 2 ou 3 do dispositivo. O uso da tecla 1 resulta na apresentação de informações sobre pilotos de fórmula 1. No caso do uso da tecla 2, informações sobre as escuderias são exibidas. Já o uso da tecla 3 mostra a pista de Interlagos. Em todos os casos, o vídeo passa a ocupar um quarto de tela, quando uma nova opção de interação é exibida, como pode ser visto na Figura 10.

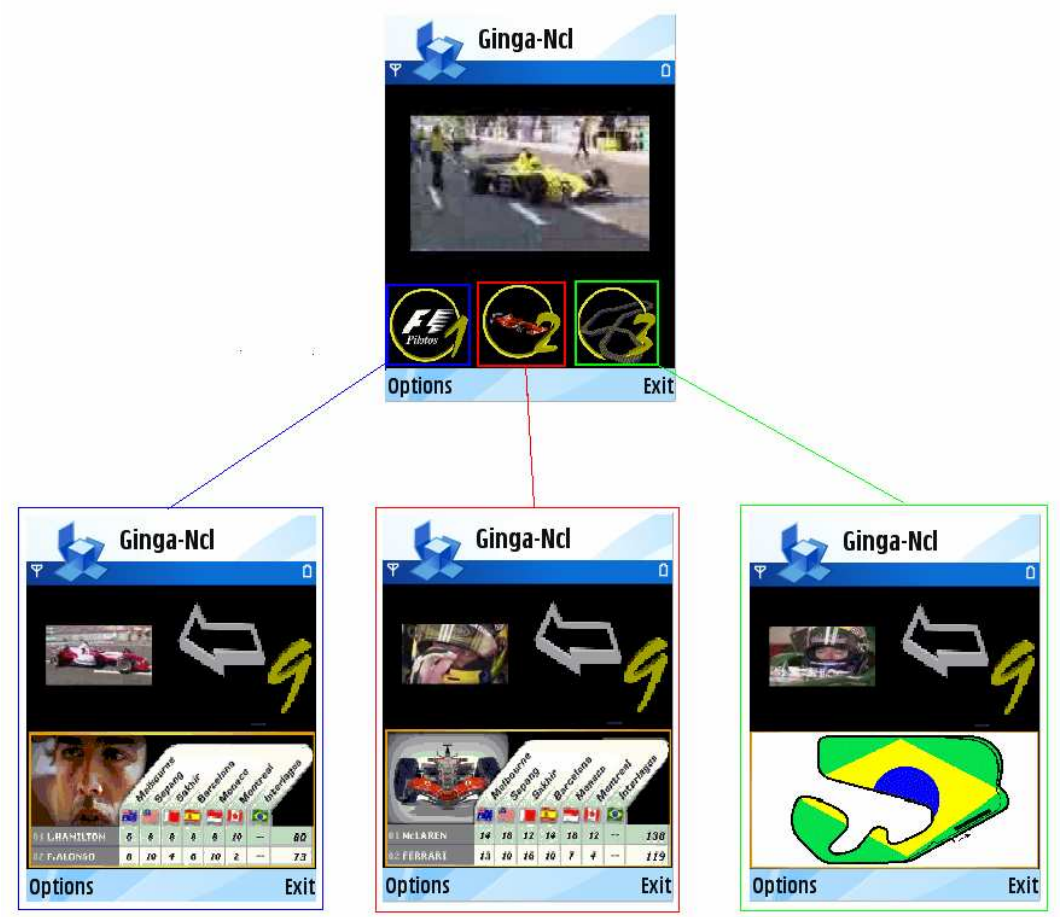

Figura 10: Os três casos de interação da aplicação Fórmula1.

Ao se pressionar a tecla 9, a aplicação volta ao estado inicial.

Os exibidores usados na aplicação Fórmula 1 foram os de vídeo e imagem. A grande maioria dos problemas tidos com esses exibidores foram encontrados e solucionados durante a fase de testes feita com a aplicação em questão. No caso do exibidor de imagem, foi observada a limitação dos tipos de imagens suportados, como descrito no Capítulo 3. Além dessa limitação, o exibidor de imagem não apresentou muitos problemas.

O exibidor de vídeo mostrou-se mais problemático. O maior problema observado ocorreu porque a função de Open de vídeo, no Symbian, é assíncrona. Logo depois de ordenar a abertura de um vídeo, através da chamada à função Open, a máquina de apresentação solicita informações do exibidor, como o tempo de duração e o volume. Nesse momento, pode ser que o vídeo não tenha sido completamente carregado, o que causaria um erro.

A forma mais simples de se resolver esse problema é aguardar até que a função Open termine o seu processamento e, somente então, continuar a execução. Symbian oferece uma API que é capaz de fazer isso, ou seja, ela provê uma forma de se aguardar pelo término de funções assíncronas. Essa API

é, entretanto, complexa, de uso restrito, o que exige um bom conhecimento dos Actives Objects. Apesar da sua complexidade, o uso dessa API solucionou o problema adequadamente.

Como a API do exibidor de vídeo é muito similar à do exibidor de áudio, acredita-se que muitos dos problemas que poderiam ocorrer na implementação do exibidor de áudio foram evitados durante a fase dos testes em questão.

Erros no módulo Conversor também foram encontrados. Os maiores problemas foram aqueles relacionados ao novo procedimento de parser e ao tratamento das tags dependentes. Esses erros foram corrigidos e nenhum outro problema referente aos procedimentos descritos foi encontrado nesse ou em outros testes.

Durante a execução da aplicação Formula1, também foi possível perceber o problema do uso das `threads` no Symbian. As imagens demoravam muito tempo para serem exibidas, mesmo sem a apresentação do vídeo. Foi nesse momento que se iniciou o estudo para a substituição das `threads` pelos Active Objects. Quando a substituição foi feita, o problema foi solucionado e a aplicação passou a ser apresentada eficientemente.

Foi também durante a execução da aplicação Formula1 que ficou evidente a limitação da soma total das mídias que poderiam ser apresentadas ao mesmo tempo em uma aplicação NCL. As três imagens que representam as opções de interação ocupavam, inicialmente, muito espaço de memória. Por causa disso, a soma dos tamanhos dos conteúdos acabava sendo muito grande, e a execução da aplicação no dispositivo era finalizada por falta de recursos.

4.2. Matrix

A segunda aplicação desenvolvida e testada na implementação do Ginga-NCL para dispositivos portáteis foi a Matrix_Mobile. O objetivo principal dessa aplicação foi testar o sincronismo temporal. Ao ser iniciada, a aplicação exibe um trecho do filme Matrix em toda a área disponível do dispositivo, como pode ser visto na Figura 11.

Figura 11: Aplicação Matrix_Mobile.

Em um determinado momento, um dos protagonistas mostra uma pilha. Nesse instante, a aplicação reduz o tamanho do vídeo e apresenta, no espaço restante, uma propaganda de pilha por um determinado período de tempo. Quando esse tempo termina, a propaganda é removida e o vídeo é restaurado ao seu tamanho original. A Figura 12 ilustra esse cenário.

Figura 12: Primeiro caso de sincronismo da aplicação Matrix_Mobile.

Em um outro momento, o mesmo ator fica próximo à câmera, realçando os seus óculos. Nesse instante, o vídeo é reduzido a fim de oferecer espaço para uma opção de interação, que fica disponível por um período fixo tempo. Depois desse tempo, a imagem é removida, o tamanho do vídeo é restaurado e o

usuário não poderá mais realizar a interação. Esse cenário é apresentado na Figura 13.

Figura 13: Segundo caso de sincronismo da aplicação Matrix_Mobile (Sem interação).

Se o usuário optar por fazer a interação, realizada ao se pressionar a tecla 1, o vídeo terá as suas dimensões ainda mais reduzidas e uma propaganda de óculos escuros será apresentada. Poucos segundos depois, a propaganda é removida e o vídeo é restaurado ao seu tamanho original. A Figura 14 ilustra esse cenário.

Figura 14: Segundo caso de sincronismo da aplicação Matrix_Mobile (Com interação).

Os exibidores usados nessa aplicação foram os de imagem e vídeo. Nenhum problema novo foi observado para esses exibidores.

Como já mencionado, o objetivo do teste feito com o uso da aplicação Matrix_Mobile foi verificar se o sincronismo estava funcionando corretamente. Foi nesse momento, portanto, que se observou o problema do tratamento das âncoras temporais, descrito na Seção 3.6. Depois que esse problema foi solucionado, a aplicação Matrix_Mobile passou a funcionar corretamente, e nenhum outro erro relevante foi encontrado.

4.3. Carnaval

A terceira aplicação desenvolvida e testada foi a Carnaval. O seu objetivo foi realizar o uso mais intenso do sincronismo e forçar um pouco mais o uso de recursos. Quando iniciada, Carnaval exibe um vídeo, em tela cheia, de uma escola tocando o seu samba-enredo, como pode ser visto na Figura 15.

Figura 15: Aplicação Carnaval.

Depois de um curto período de tempo, o vídeo tem o seu tamanho reduzido e o espaço restante é preenchido com duas informações: a legenda do samba-enredo sincronizada com o vídeo e duas opções de interação. A Figura 16 ilustra essa situação.

Figura 16: Aplicação Carnaval com legenda e com as duas opções de interação.

A legenda é composta por trinta imagens sincronizadas com o vídeo ao longo do tempo. Uma das interações é ativada pela tecla 1 e a outra pela 2. A primeira interação exibe a bandeira da escola de samba e a segunda exibe uma foto do intérprete, como pode ser visto na Figura 17.

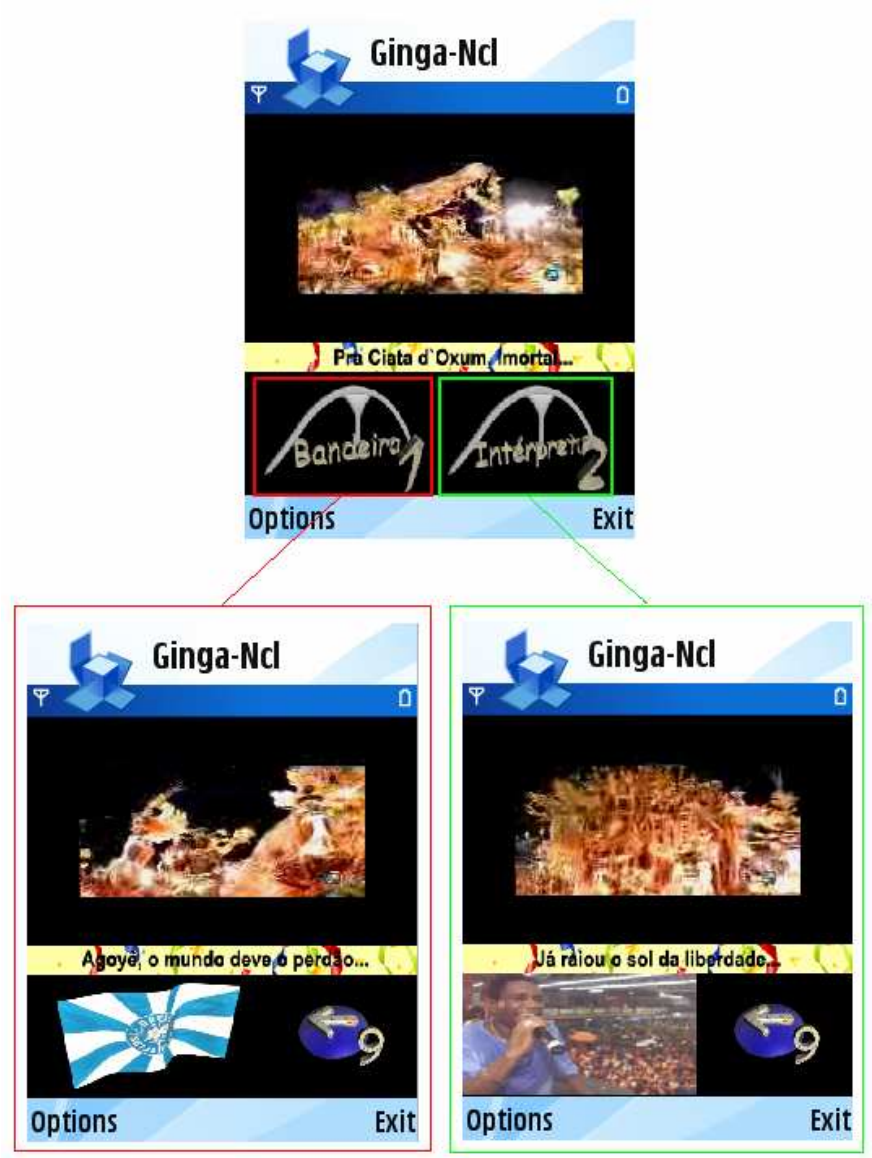

Figura 17: Os dois casos de interação da aplicação Carnaval.

Em ambos os casos, uma nova opção de interação — ativada pela tecla 9 — é oferecida para permitir ao usuário voltar às opções de interação iniciais.

Nenhum problema relevante foi encontrado com a execução da aplicação Carnaval. O sincronismo da legenda funcionou corretamente e de forma eficiente. Não foi observada perda de sincronismo nem mesmo quando as interações eram muito exercitadas. Apesar de ser uma aplicação pesada, o seu funcionamento se deu de forma adequada e eficiente.

4.4. Exibidor HTML com Canal de Retorno

A última aplicação testada não foi baseada em nenhum tema. O seu objetivo foi o de apenas testar a integração com o exibidor HTML, e o funcionamento do recurso de foco e seleção da NCL. Essa aplicação, diferentemente das apresentadas anteriormente, só foi testada em um dos dispositivos disponíveis. Isso porque o outro não possuía uma interface de rede a ser usada com o objetivo de se estabelecer uma conexão com a Internet necessária para o teste do exibidor de HTML. Maiores detalhes sobre os dispositivos são apresentados no Apêndice A desta dissertação.

Ao ser iniciada, a aplicação em questão divide a tela do dispositivo em dois. Na parte superior é exibida uma página HTML e, na parte inferior, quatro imagens de botões, como pode ser observado na Figura 18.

Figura 18: Aplicação 4.

É possível navegar com o foco entre todos os cinco elementos da aplicação. A navegação dentro da página HTML se dá após ela ter sido selecionada. Na aplicação em questão, quando um elemento está focado, uma borda azul é desenha em seu contorno. Já quando o elemento é selecionado, a cor verde é aplicada à borda. A Figura 19 mostra o momento em que a página HTML está focada, depois a sua seleção e, por fim, a navegação dentro dela.

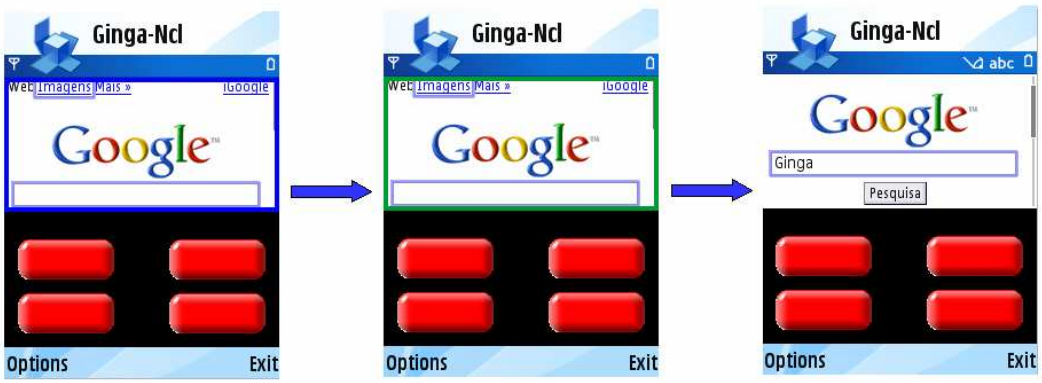

Figura 19: Seleção e uso da página HTML.

Em qualquer momento, a seleção pode ser desfeita, possibilitando novamente a navegação de foco entre os elementos da aplicação.

Cada uma das quatro imagens também podem ser selecionadas. O efeito causado é a exibição momentânea de uma outra imagem que, em seguida, é substituída pela imagem original. Isso permite a criação de efeitos simples de animação. No caso da aplicação em questão, o efeito gerado é o de um botão sendo pressionado, como exemplificado pela Figura 20.

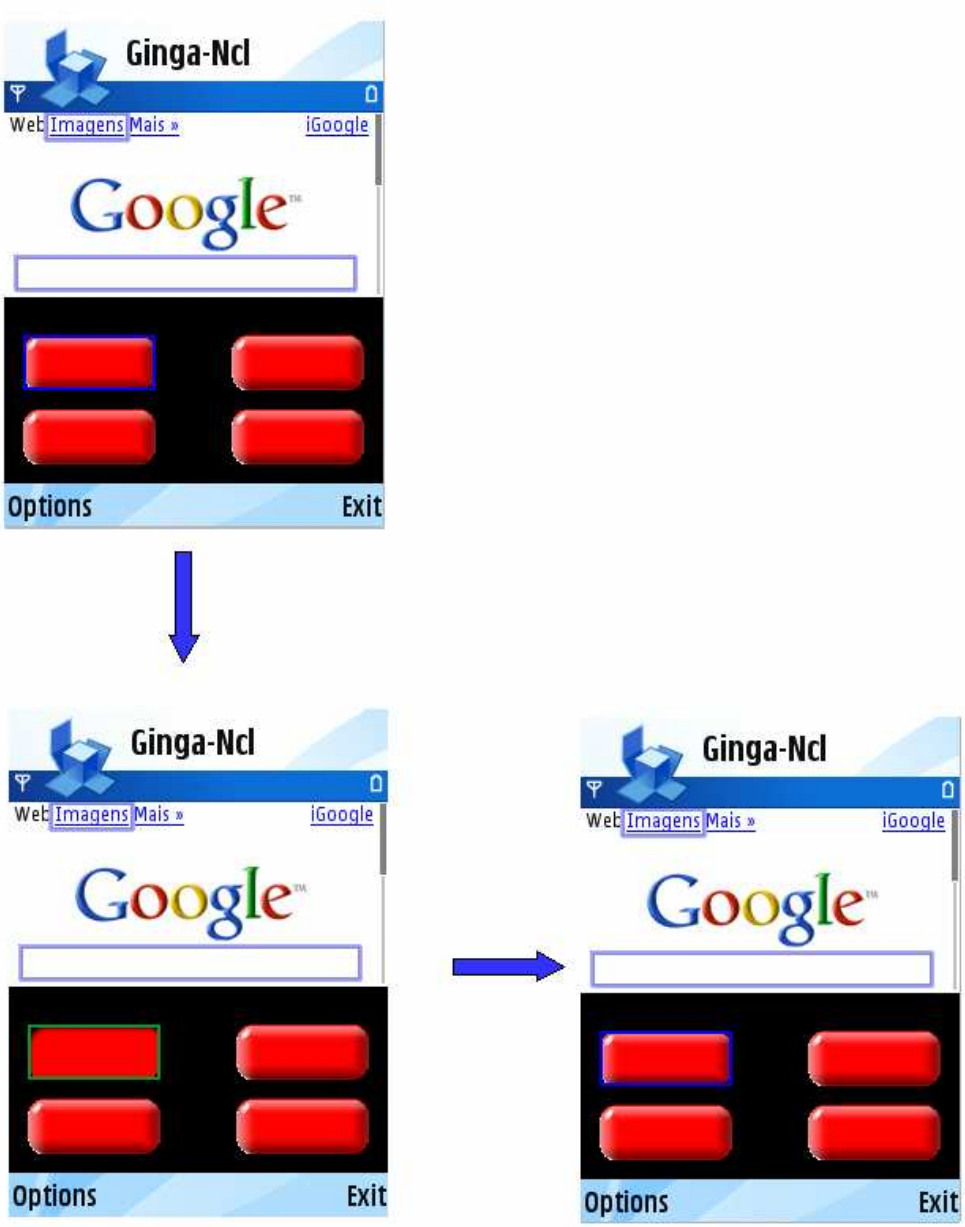

Figura 20: Botão pressionado.

A seleção da imagem que se encontra no canto inferior direito causa o término da aplicação. Já a seleção das outras imagens inicia diferentes conteúdos de áudio.

Os exibidores usados na aplicação foram os de imagem, áudio e HTML. O exibidor de áudio praticamente não ofereceu problemas, assim como o de imagem que já havia sido testado previamente. Já o exibidor de HTML causou muitos problemas. O maior deles, entretanto, é o que fazia com que uma lista de conexões fosse apresentada ao usuário. Isso é feito porque a API que trata das páginas HTML não seleciona uma conexão padrão para estabelecer comunicação com a Internet. Assim, existe a necessidade de se perguntar ao

usuário qual conexão deve ser usada. O problema que ocorreu deveu-se ao fato que, no Symbian, normalmente uma mesma camada gráfica é usada para se desenhar quaisquer elementos na tela. O uso de janelas separadas, onde a exibição de um elemento em uma janela não interfere na outra, é evitado, pois exigiria o uso de mais de um elemento Window. Como já foi explicado na Seção 3.7, o uso de Windows não é recomendado. Dessa forma, a API da página HTML mostra a lista de conexões por cima da aplicação NCL que está sendo apresentada.

A solução mais simples para esse problema foi implementar uma forma de se selecionar uma conexão padrão para a API, evitando, com isso, que a lista fosse apresentada. Uma conexão deve, então, ser inicialmente estabelecida e entregue a API. Essa solução funcionou no emulador, mas não foi suficiente. No dispositivo testado, a API parecia ignorar a conexão estabelecida e voltava a apresentar a lista ao usuário.

Uma segunda solução foi fazer com que a API em questão lançasse a lista de conexões antes da aplicação NCL ser efetivamente iniciada. Assim, a lista é apresentada, o usuário seleciona a opção mais conveniente, a tela é limpa e a aplicação NCL pode, então, ser iniciada sem problemas. Isso resolveu o problema causado pela API que tratava da página HTML. Essa API apresentou, ainda, outros problemas menos relevantes que foram tratados de forma apropriada.

A implementação do exibidor de HTML foi, portanto, como já descrito no Capítulo 3, uma das mais complicadas. O tratamento do recurso de foco e seleção, por outro lado, funcionou corretamente e causou poucos problemas.