

5 Resultados

Neste capítulo são comparados os resultados atingidos em cada um dos protótipos e alguns comentários são feitos sobre a implementação.

Os protótipos propostos foram implementados como descrito no capítulo anterior, e para verificar suas performances foram utilizados os celulares Nokia 6111, Sony Ericsson K750 e Siemens CX 65. Esses telefones possuem 128X160, 176X220 e 132X176 pixels de resolução e 2.0, 1.5 e 1.5 MB de *heap memory*, respectivamente.

A *heap memory* é utilizada para a execução dos aplicativos, e é comparável à memória RAM na arquitetura de computadores pessoais. Normalmente esta memória não está inteiramente disponível para as aplicações.

Todos esses modelos possuem câmera, transmissão de dados via GPRS (transmissão de dados por pacotes) ou EDGE (evolução da rede GPRS) e funções multimídia como reprodução de vídeos e de música em formato MP3.

Apesar de não terem todas as suas configurações de hardware divulgadas ao público em geral, esses três telefones são um bom exemplo da classe de dispositivos-alvo desta dissertação.

5.1. Resolução das Imagens e tamanhos das cenas

Todos os objetos utilizados são representados por *billboards*, utilizando *bitmaps* contendo transparências e com resolução variando entre 50X30 e 200X150.

Todas as imagens usadas estão no formato *Portable Network Graphics* (PNG) e o tamanho final dos pacotes está fortemente ligado ao tamanho destas imagens utilizadas na geração de cada cena.

Por causa de limitações da plataforma J2ME, as imagens utilizadas nas implementações não puderam ser as mesmas.

Todos os testes de *download* utilizaram a rede de dados da operadora de telefonia celular Claro na região de São Conrado, na cidade do Rio de Janeiro .

5.1.1. Implementação utilizando API 2D e desenho direto

Para a implementação utilizando API 2D e desenho direto, sem manipulação dos pixels, a limitação da resolução das imagens teve como gargalo o tamanho ocupado em disco. Durante os testes de download das imagens foi verificado que a plataforma J2ME apresentava erro caso estas ocupassem mais de 270 KB. Porém, o uso de imagens maiores não apresentava erros caso fossem carregadas diretamente junto com o pacote do aplicativo.

Assim, para esta versão do protótipo a maior imagem utilizada foi de 2405 X 140 pixels.

Para esta implementação, foram necessárias, no mínimo, três imagens por cena: duas para o panorama utilizando a técnica de mipmapping para o efeito de zoom e uma para representar os objetos. Devido à alta resolução das imagens de fundo, não foi possível fazer o redimensionamento do panorama no próprio celular, obrigando que existam mais de uma versão dessas imagens de fundo com resoluções diferentes.

Por causa da alta resolução dos panoramas utilizados nos testes, este protótipo apresentou ótima qualidade visual. Apesar de não ocorrerem distorções características do panorama cilíndrico, dando mais realismo ao usuário, tal limitação não é tão perceptível quando utilizados panoramas que representem paisagens distantes da câmera.

O conjunto de imagens utilizado nos testes somou um total de até 415 KB. Em redes de transmissão de dados GPRS/EDGE, que possuem velocidades de transmissão de até 60Kb/s por canal, podendo chegar a 400Kb/s se utilizados oito canais de comunicação, o download de uma cena tem em média duração de menos de um minuto.

Nos testes com um conjunto de imagens que somavam 415 KB, executados no aparelho Nokia 6111, que utiliza tecnologia EDGE, o *download* de uma cena teve duração média de 27,5 segundos, enquanto que no telefone K750i da Sony Ericsson, que utiliza tecnologia de transmissão de dados GPRS que apresenta performance inferior à implementada no aparelho da Nokia, a duração média foi de 59 segundos.

5.1.2. Implementação utilizando Image Warping

Na implementação por warping é necessária a criação de dois *buffers*, um para a leitura da imagem do panorama e outro para a criação da imagem final ao ser desenhada na tela.

Apesar do uso de mais memória devido aos *buffers*, o limite no tamanho das imagens que puderam ser utilizadas foi imposto pelo módulo de *download*. Assim, os resultados quanto ao tamanho dos pacotes e tempo de *download* foram os mesmos quanto da implementação utilizando o desenho direto.

5.1.3. Implementação utilizando API 3D

Neste exemplo o panorama original foi redimensionado para 1024 X 256 pixels, pois pela especificação JSR184, que descreve o *Mobile 3D Graphics* do J2ME, todas as texturas devem ter dimensões que sejam potências de 2. Além disso, cada plataforma possui uma limitação no valor dessas dimensões. Para o emulador presente no *framework* de desenvolvimento *Wireless Toolkit* fornecido pela Sun Microsystems, é de 1024 pixels e para todos os modelos utilizados nos testes, são de apenas 256.

Esta limitação diminui bastante a qualidade do panorama na visualização nos telefones reais. O celular Nokia 6111, apesar de possuir suporte à mipmapping, o controle deste não é disponibilizado para a programação, sendo aplicada automaticamente de acordo com parâmetros fixos característicos da plataforma. Assim essa técnica não traz muitos resultados, já que a maior resolução do panorama possível é de 256 X 256 pixels.

Todas as cenas geradas, mesmo as que possuem as maiores imagens (256 X 256 pixels para o panorama e 150 X 200 pixels para a imagem da estatua do Cristo Redentor), não excederam os 100KB.

Nos testes executados no aparelho Nokia 6111 que possui tecnologia de transmissão EDGE, o tempo médio de *download* das cenas foram de aproximadamente seis segundos,

5.2. Desempenho

Para cada implementação foram gerados alguns testes com imagens de diferentes resoluções, e configurações de cenas diferentes, variando a quantidade e o posicionamento dos objetos.

5.2.1. Implementação utilizando API 2D e desenho direto

Dentre as abordagens propostas a implementação utilizando a API 2D com o desenho direto do panorama foi a que apresentou o melhor desempenho.

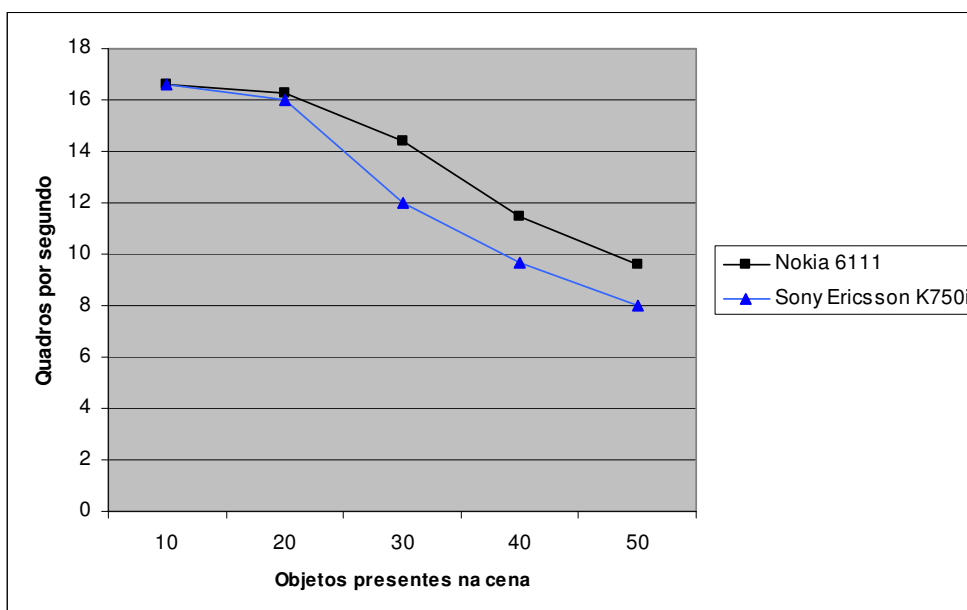


Figura 11 - Resultados para os testes de performance da implementação usando desenho direto da imagem do panorama

Os resultados foram obtidos criando-se cenas em que os objetos são inseridos em uma mesma área da cena, possibilitando que esses sejam desenhados na tela ao mesmo tempo.

Na implementação, os cálculos do posicionamento dos objetos, da área do panorama a ser desenhada e a colisão dos objetos com a “mira”, que possibilita o “clique” do usuário para iniciar uma interação, são todos feitos somente quando o usuário utiliza as teclas direcionais para os lados, provocando a mudança de direção da câmera virtual.

Apesar disso, testes demonstraram que tais cálculos não influem na degradação da performance, e sim o desenho dos objetos na tela. Nestes testes,

mesmo utilizando 50 objetos, mas posicionando-os de maneira espaçada na cena, apresentaram taxa de 16 quadros por segundo no aparelho Nokia 6111.

5.2.2. Implementação utilizando Image Warping

Apesar de utilizar imagens de mesma resolução do protótipo apresentado no item anterior, sua performance foi completamente diferente.

A abordagem é similar a anterior, onde a imagem do panorama é gerada somente com a mudança de direção da câmera virtual, assim os resultados foram os mesmos quando a câmera se encontra parada.

Porém os cálculos necessários para gerar uma imagem do panorama não possuem um bom desempenho nos celulares. Implementando o mesmo algoritmo apresentado em (Matos,98), a taxa de geração de quadros, quando a câmera está em movimento, ficou em torno de 3 (três) por segundo.

Como a visualização apresentou defeitos outros testes implementando outras transformações mais simples foram feitas. Um teste em que a transformação do panorama foi substituída por uma rotação simples da imagem de fundo, em uma cena sem nenhum objeto presente, a taxa manteve-se em apenas 8 (oito) quadros por segundo. Nesta implementação, no loop principal, que lê e calcula o posicionamento de cada pixel da imagem, apenas uma multiplicação com ponto flutuante é efetuada.

5.2.3. Implementação utilizando API 3D.

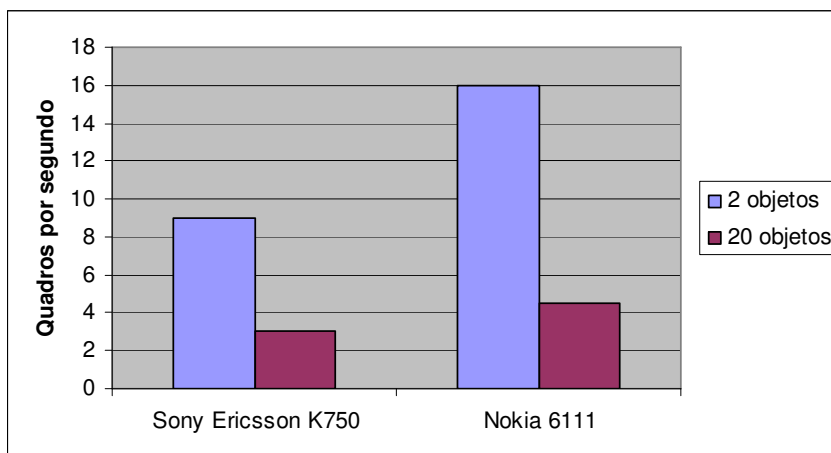


Figura 12 - Resultados de simulações, da implementação utilizando API 3D, de ambientes em que dois e vinte objetos são exibidos ao mesmo tempo

Mesmo apresentando resultados abaixo dos outros dois modelos nos testes apresentados no capítulo 3, o aparelho Nokia 6111 foi o que obteve melhor resultado executando a aplicação, gerando quadros a taxas interativas. Já o Sony Ericsson K750, além de ter apresentado performance mais baixa, ainda apresentou falhas na visualização dos objetos.

Assim como nos casos anteriores, a performance foi aferida para diferentes quantidades de objetos visualizados ao mesmo tempo. Nesta implementação a quantidade total de objetos não interfere nas taxas de reprodução se estiverem espaçados na cena.

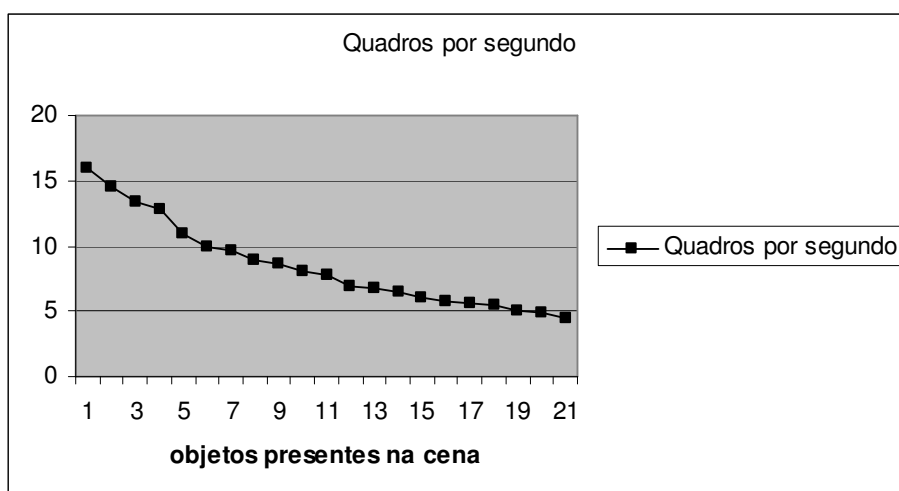


Figura 13 - Desempenho do protótipo implementado com API 3D, aferida no celular Nokia 6111

5.3. Interação com os objetos

O protótipo, seguindo as funcionalidades existentes na aplicação *Quick Time VR*, possibilita a interação do usuário com os objetos presentes em cena.

Este mecanismo foi implementado com a idéia de “clique” em cima do objeto com o qual deseja interagir. Isto é feito através de um marcador em forma de mira posicionado no meio do *display*, que muda para a forma de uma seta, identificando que o usuário pode executar alguma ação associada a este objeto.

Como explicado no capítulo anterior, cada objeto pode pertencer a uma classificação pré-definida, e deste modo apresentar ações que o usuário pode executar.

Criando outras classificações para os objetos é possível a inclusão de outras ações, aumentando a interação do usuário com a cena. Também é possível a inclusão de objetos transparentes, mas que façam parte de alguma das classificações pré-definidas. Assim, pode-se aproveitar algum objeto que faça parte do panorama como um ponto de interação com o usuário.

As imagens e informações do ambiente podem manter-se persistidas no estado corrente da aplicação, ou serem salvas localmente, possibilitando que o usuário retorne a cena após a execução de uma ação.

Como apresentado no capítulo anterior, na atual implementação todos os objetos presentes em uma cena são associados à navegação para uma cena adjacente. Desta forma, a única ação de interação que um usuário pode executar em relação a um objeto é a navegação.

5.4. Processo de autoria

Com a popularidade do *Quick Time VR* e avanços nas técnicas de stitching, como as apresentadas por Szeliski (1997), surgiram várias ferramentas capazes de gerar fotografias panorâmicas em 360º sem a utilização de câmeras especiais como a *Globuscope* ou a *Panoscan*. Um exemplo dessas ferramentas é o conjunto de bibliotecas *Panorama Tools* ou o próprio *Quick Time VR Authoring Studio*.

Apesar de não terem sido criadas ferramentas para ajudar na composição dos objetos com a imagem do panorama, gerando a cena final, o processo de autoria se manteve simples.

O arquivo que descreve a cena, apresentado no capítulo anterior, tem um formato texto simples, onde o autor deve indicar o nome das imagens a serem utilizadas como fundo do panorama e para representar os objetos presentes na cena. Nesta primeira versão, todos os objetos são posicionados em um círculo concêntrico ao cilindro utilizado como base do panorama. O autor identifica apenas o ângulo, em relação à posição inicial da câmera, em que tais objetos se encontram.

Além das informações de posicionamento, o autor também deve incluir as informações sobre o comportamento dos objetos presentes. Na atual versão, apenas objetos que são associados à navegação para cenas adjacentes estão contemplados, portanto, a única informação sobre a interação dos objetos que deve ser fornecida pelo autor é o nome do descritor da cena adjacente.