

Xadrez em Realidade Aumentada - Manual

Douglas Bettioli Barreto (NUSP 6920223)

Giancarlo Rigo (NUSP 6910034)

Rafael Reggiani Manzo (NUSP 6797150)

15 de outubro de 2010

Parte I

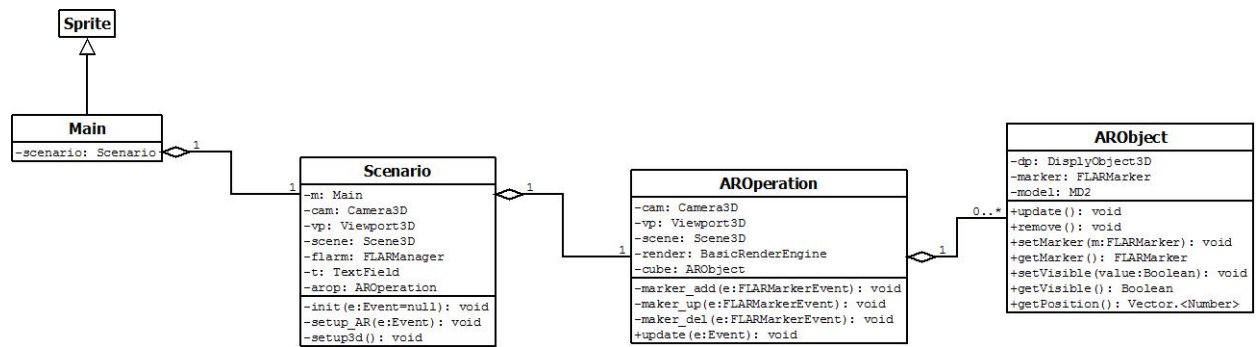
Manual do usuário

Parte II

Manual do Desenvolvedor

Capítulo 1

Diagrama de classes



1

¹Ver observacoes sobre a classe AROperation em 2.3.1

Capítulo 2

Estrutura do código

2.1 Compilação

Este software foi desenvolvido e testado em Linux Ubuntu 10.04 (32 bits). Mas, teoricamente, deve funcionar em qualquer maquina que atenda aos requisitos (2.1.1)

2.1.1 Requisitos

- Adobe Flex Compiler (mxmcl) Version 4.1.0 build 16076

2.1.2 Makefile

Encontrado dentro da pasta src. Dentro dele substitua o caminho para o mxmcl do seu Flex SDK. Existem duas opções de debug:

- Compilar a versão de produção ao: “make”
- Compilar a versão de debug: “make debug”

2.2 Pacote principal

No momento, este pacote possui as classes Main e Scenario. Mas, com o desenvolvimento do projeto, devem ser criadas novas classes.

2.2.1 Classe Main

Esta classe simplesmente instancia um objeto da classe Scenario (2.2.2).

2.2.2 Classe Scenario

Esta classe recebe a instância da classe Main na qual vai adicionar vários “eventlistener” e adicionar elementos como câmera, textos e botões. Seu objetivo é retirar do Main o código que manipula a parte visual da aplicação. Trabalhando com a parte visual, esta classe não poderia deixar de utilizar classes do pacote de realidade aumentada (2.3). Vale ressaltar que os nomes de métodos desta classe estão muito ruins, mas a refatoração disto já foi planejada para a próxima fase.¹

2.3 Pacote AugmentedReality

No momento, este pacote possui as classes ARObject e AROperation. Mas, com o desenvolvimento do projeto, devem ser criadas novas classes.

2.3.1 Classe AROperation

Esta classe concentra as operações do FLARManager, como criar novos objetos de realidade aumentada através da classe ARObject (2.3.2) e atualizar os markers. No momento, esta classe deve ter um atributo para cada modelo 3D que deve ser mostrado. Mas, já esta planejada a refatoração desta classe para que um array dinâmico substitua esta necessidade de vários atributos⁽²⁾.

2.3.2 Classe ARObject

Responsável por carregar o modelo 3D do objeto, guardar seu marker e seu display. Nesta classe são encontrados métodos de get e set, por enquanto públicos³, para quase todos seus atributos. Ela é a classe mais básica que temos hoje no software, sendo muito utilizada pela classe AROperation (2.3.1).

¹Link para a story da refatoração no pivotal: <http://www.pivotaltracker.com/story/show/5588098>

²Link para a story da refatoração no pivotal: <http://www.pivotaltracker.com/story/show/5588098>

³Dependendo do desenvolvimento do projeto estes métodos podem se tornar protected e ser criada uma classe de interface para o pacote

Capítulo 3

Prova de conceito

3.1 Introdução

A prova de conceito consistiu em abordar os pontos chave do projeto no que se refere a realidade aumentada. Nela, foram tratados temas como o reconhecimento de múltiplos marcadores e a capacidade de obter as coordenadas do marcador no espaço.¹

3.2 Marcadores

Na prova de conceito conseguimos com sucesso utilizar 12 marcadores quadrados com 3cm de lado². Demonstrando assim que o FLARManager consegue lidar com muitos marcadores, com tamanho pequeno e muito próximos, como será no nosso jogo.

3.3 Posição do marcador no espaço

É fundamental conseguir identificar a posição de um marcador no espaço, pois sem isso seria impossível montar a matriz de peças no tabuleiro. Esta parte não apresentou problemas uma vez que sua única dificuldade foi consultar a documentação do FLARManager e descobrir que um objeto da classe FLARMarker possui os atributos públicos “x”, “y” e “z”. Com isso foi feito um método na classe ARObject (2.3.2) para pegar estes atributos.

¹A prova de conceito é o branch “12 tags ao mesmo tempo” do repositório

²Uma casa de tabuleiro de xadrez oficial é um quadrado com, no mínimo, 5cm de lado e, no máximo, 6,5cm de lado