

UAWall

#entrega_02 - viabilidade técnica

1. Software necessário

Para a criação de imagem vectorial e tratamento de fotografia, a opção recai sobre o Adobe Illustrator CS5 e o Adobe Photoshop CS5, respectivamente. Neste caso em específico, não há necessidade de fazer um levantamento de outras opções, pois consideramos que ambas as soluções reúnem as funcionalidades necessárias para realizar aquilo que nos propomos a criar e, acima de tudo, são as que mais dominamos.

Para desenvolver a aplicação, surgem diversas dúvidas na escolha do software e respectiva linguagem de programação que se aproprie mais ao tipo de plataforma a desenvolver.

Interessa exemplificar como funcionam e se relacionam as diferentes componentes para clarificar a “matéria”.



Existe uma relação entre os componentes integram todo o sistema que respeitam a hierarquia apresentada, desde o hardware ao software.

Isto implica que o hardware adoptado (sensor de movimento) faça a captura da imagem em 3D e transmita os dados para um computador com um determinado sistema operativo, via USB. Existe então, uma camada chamada middleware que interpreta os dados enviados pelos sensores do dispositivo, compreendendo, segundo um dos algoritmos definidos, os movimentos e interações realizadas pelo utilizador - traduzindo esta informação em inputs, aos quais as frameworks reagem; tudo isto sem a necessidade de nenhum hardware mediador adicional. De seguida, a framework escolhida providencia uma API que permite o desenvolvimento de novas aplicações, neste caso de interação natural, o que descarta a necessidade de um comando físico.

Temos algumas dúvidas relativamente à camada 5 que reside entre duas possibilidades de linguagem a utilizar: HTML/CSS (eventualmente HTML5 e CSS3 juntamente com JavaScript, e outras livrarias.) ou AS3 - Flash. Uma vez que conhecemos as duas linguagens, conseguimos analisar as vantagens e

desvantagens de cada uma. A mais relevante incide sobre a flexibilidade que o flash nos poderá dar relativamente ao desenvolvimento de animações e navegação, o que nos parece mais limitado caso optemos pela outra linguagem. Por outro lado, sentimo-nos mais à vontade com HTML/JavaScript, por termos tido uma maior experiência, e porque as possibilidades que o HTML5/CSS3 oferecem conseguem competir com o Flash - software que tem vindo a cair em desuso.

Mesmo depois desta análise, parece-nos ainda precipitado optar por uma destas linguagens, sem antes termos pensado e encontrado respostas para questões como: grafismo, animação, navegabilidade, etc., que são de alguma forma as condicionantes que nos levarão a optar por um caminho ou por outro. Será de acordo com esta escolha que iremos optar pelas frameworks, middlewares e livrarias que passamos a apresentar juntamente com as vantagens e desvantagens de cada uma.

	VANTAGENS	DESVANTAGENS
MICROSOFT KINECT SDK	Suporta áudio; Suporta múltiplos sensores; Detecção de todo o corpo; Não necessita de uma calibração da pose para iniciar o controlo.	Licença apenas uso não comercial; Consome mais energia do CPU; Não tem um sistema de reconhecimento de gestos; É apenas suportado por Windows 7;
OPENNI PRIMESENSE	Inclui licença para usos comercial; Inclui uma framework para reconhecimento do gesto da mão; Alinha automaticamente a profundidade da - imagem com a sua cor; Consome menos energia do CPU; Suporta múltiplos sensores; É suportado por qualquer tipo de sistema operativo.	Necessita de uma calibração da pose para iniciar o controlo;

	VANTAGENS	DESVANTAGENS
CONCLUSÃO	<p>Depois da análise feita a estas duas frameworks que seleccionámos, a nossa decisão recai sobre o OpenNI.</p> <p>A escolha para a análise da framework Microsoft Kinect SDK surge, apenas por esta ser a única que pode competir ao lado da framework OpenNI devido as suas características. No entanto, foi descartada de imediato por parte do grupo por apenas ser suportada em computadores com sistema operativo Windows7, visto que todos os elementos do grupo possuem o sistema operativo MacOSX.</p> <p>Assim, o OpenNI desenvolvido pela PrimeSense revela-se como a melhor opção para o tipo de aplicação que procuramos desenvolver, por ser a que melhor se adapta às nossas necessidades. Apesar de necessitar de uma calibração da pose para o utilizador iniciar o controlo da plataforma, é uma desvantagem que não se revela como um obstáculos para realização do projeto.</p>	

Posto isto, somos agora capazes de organizar um esquema (embora que não definitivo) que resume as opções para as quais estamos mais inclinados a recorrer, pelos motivos que fomos apresentando.

APLICAÇÃO
OpenNi
NiTE
SISTEMA OPERATIVO
HARDWARE

2. Hardware necessário:

Para criar o ambiente adequado ao funcionamento da aplicação nas melhores condições e fazer com que o utilizador se sinta confortável e motivado durante a sua utilização são necessários alguns aparelhos.

Em primeiro lugar, um sensor de reconhecimento de gesto (representado no esquema anterior na camada 1) e por isso recolhemos as seguintes informações para escolhermos a melhor opção.

Existem neste momento, no mercado, dois principais dispositivos de interação natural com reconhecimento de imagem:

- **Microsoft Kinect** (para Xbox 360 e para Windows);
- **Asus Xtion Pro**

Existem outros dispositivos semelhantes - como é o caso da Nintendo Wii e da PlayStation Move - no entanto, estes não permitem o reconhecimento espacial através de imagem, implicando assim a utilização de um comando físico.

A escolha reside então nos dois equipamentos nomeados em primeiro lugar. Assim, passamos a analisar as vantagens e desvantagens de cada um deles.

O sistema **Kinect** da Microsoft foi o proposto pelo orientador do projeto. Este é um sistema pioneiro que reuniu à sua volta uma grande comunidade de programadores profissionais e/ou entusiastas. Apesar de ser uma tecnologia bastante recente e onde não existem padrões ainda definidos, a utilização de código opensource permitiu que a utilização e desenvolvimento de aplicações para este dispositivo esteja a crescer de uma forma que superou todas as expectativas. Daí que esteja facilmente disponível documentação diversa, tutoriais ou mesmo software para auxiliar o trabalho de desenvolvimento de uma aplicação. Para além disso, é de notar que existe uma “comunidade” de alunos no Deca envolvida no desenvolvimento de projetos ligados a esta área.

O sistema **Asus Xtion Pro** não apresenta, em relação ao sistema Kinect, qualquer vantagem. O preço é superior, a documentação é escassa e não existe uma comunidade “disponível” em volta deste produto.

Após esta análise, é facilmente perceptível que a Kinect da Microsoft seja o sistema a ser escolhido por reunir as condições mais favoráveis numa área onde a informação e o consenso são, ainda, precários.

Necessitamos ainda, de um projetor e de uma “parede”, através dos quais apresentaremos a aplicação. Esta parede, ou em alternativa uma tela, que deverá ser branca e encontrar-se num sítio com as condições necessárias para proporcionar uma experiência ótima ao utilizador. Para isso, deveremos ter em atenção o espaço, a iluminação e interferências naturais do meio. Para além do que já foi referido, acrescenta-se ainda um computador que serve de suporte à plataforma. Embora ainda, não tenhamos a certeza, não será necessário especificar o tipo de Sistema Operativo que este deve possuir, pois tentaremos que aplicação funcione tanto em Windows como em MacOSX.

3. Testes

A tecnologia ligada ao comando por gesto é recente e, para nós, será o primeiro contacto no desenvolvimento de uma aplicação deste tipo. É de referir ainda, que pelo mesmo motivo não se encontram padronizadas formas de interação. Desta forma, o trabalho que iremos realizar terá de se acompanhar por uma série de testes, não só de validação de código e de detecção de erros, mas também, testes de funcionalidade, usabilidade e design. Para tal, contamos com o apoio dos docentes e orientadores, assim como tencionamos criar um “equipa” de testes para a realização dos mesmos.

4. Conclusão

A realização desta entrega exigiu a continuação de uma vasta pesquisa por parte do grupo. Foi necessário avaliar, de forma intensiva, quais as ferramentas a utilizar para o desenvolvimento dos requisitos listados, uma vez que atuamos em “território desconhecido”. Neste sentido, continua ainda, a haver algumas indecisões que apenas serão tomadas no decorrer do projecto, como é normal.

Partindo para a prática, numa primeira fase, será necessário criar o grafismo de toda aplicação, na qual vamos recorrer ao Illustrator e ao Photoshop, sem qualquer dúvida, pois para além de serem ferramentas de elevado potencial, são softwares dominados pelo grupo.

Posto isto, surge então, a grande dúvida: qual o tipo de linguagem a utilizar para o desenvolvimento da aplicação no seu todo. O grupo procurou outros alunos envolvidos em projetos com o mesmo tipo de interação - Kinect - e todos partilham a mesma opinião, sugerindo que se escolha, inicialmente, um “programa” com o qual nos identificamos e que melhor se adeque ao projeto e, só depois de pensado todo o projeto e de este se encontrar consistente, proceder à opção mais acertada. Para já, a dúvida reside entre as opções que apresentamos.

5. Referências

Wired - Kinect;

Frameworks;

Saffer, D. (2008). Designing Gestural Interfaces, O'REILLY

Kean, S., J. Hall, et al. (2011). Meet the Kinect: An Introduction to Programming Natural User Interfaces, Apress.