

hrv_lan – interface gráfica para aquisição de vídeo e frequência cardíaca em uma LAN

1. Introdução/Justificativa

O objetivo deste software é capturar vídeo de webcams de computadores conectados a uma LAN. Paralelamente à captura, o sinal de ECG do usuário também deve ser capturado.

As principais características do software são:

- 1 - Possui janela do operador, onde se controla o início da captura.
- 2 - Permite estabelecer a rotina de captura. tempo em NVNM etc.
- 3 - Possui janelas para os subjects.
- 4 - Deve ser capaz de exibir vídeos pre-gravados, vídeos de câmeras transmitidos via streaming e uma tela vazia com alguma cor.
- 5 - Deve ser capaz de salvar o sinal de ECG.
- 6 - Deve ser capaz consultar as cameras disponiveis no localhost e nos demais hosts da rede. Deve ser possível localizar automaticamente os servidores de câmeras na rede com uma determinada porta aberta.

2. Classes

2.1. Classe LanCamera

A classe LanCamera é uma camada que expõe as funcionalidades das câmeras disponíveis na LAN. Possui os seguintes métodos:

`start_server`

Inicia o servidor que disponibiliza as webcams do localhost para a rede.

`stop_server`

Interrompe o servidor.

`list_servers`

Envia um sinal de broadcast para a rede para localizar todos os servidores ativos.

`list_cams_local`

Lista as câmeras do localhost

`list_cams_lan`

Lista as câmeras da LAN

`handle = stream_start(cam_id, ip_addr)`

Inicia streaming de vídeo da câmera especificada. Retorna um handle.

`stream_show(handle, window)`

Exibe streaming de vídeo com handle especificado em window.

`stream_save(handle, filename)`

Salva streaming de vídeo com handle especificado para arquivo.

2.2. Classe Routine

A classe Routine lê os arquivos que controlam a rotina de captura e exibição de um experimento.

Abaixo está um exemplo de arquivo de rotina de captura.

```
0; message; all; "Welcome!"
0; show; all; #FFFF00
5000; message; all; "Please watch this video"
5000; play; all; "videos/instructions.mp4"
30000; message; all; "No View No Motion"
45000; message; all; "No View Motion"
60000; message; s1; "Please gesticulate"
60000; message; s2; "Spontaneous Imitation"
60000; show; s1; c2
60000; show; s2; c1
90000; message; s2; "Induced Imitation"
120000; message; all; "Please stop. Thank you!"
120000; show; all; #FFFF00
...
```

Abaixo está a lista de comandos suportados. Cada comando recebe dois operandos.

Instrução	Op. 1	Op. 2	Comentário
message	User	String	Show message on screen
show	User	Color/Cam	Show color or camera content on video canvas
play	User	String	Play video and show on video canvas

Os Valores de que User e Cam podem assumir são

```
User = {s1, s2, all}
Cam = {c1, c2}
```

onde s1 corresponde à janela do subject 1, s2 corresponde à janela do subject 2, all corresponde a todas as janelas dos subjects, c1 corresponde à câmera do subject 1 e c2 corresponde à câmera do subject 2.

2.3. Classe WinOp

A classe WinOp cria a janela usada pelo operador do sistema. Expõe as funcionalidades das câmeras da LAN ao usuário, permite definir a rotina de captura, selecionar as câmeras a serem usadas e clientes onde os vídeos serão exibidos.

2.4. Classe WinSubj

A classe WinSubj cria a janela usada pelo subject participante do experimento. É controlada por comandos enviados por WinOp via socket, mesmo que ambos os processos estejam no mesmo host. Exibe o conteúdo de alguma câmera da rede, mostra vídeos e mostra mensagens com instruções.

2. Arquitetura

A Figura 1 mostra as camadas do sistema. A Figura 2 mostra como as janelas são distribuídas entre os hosts e como se comunicam pela rede.

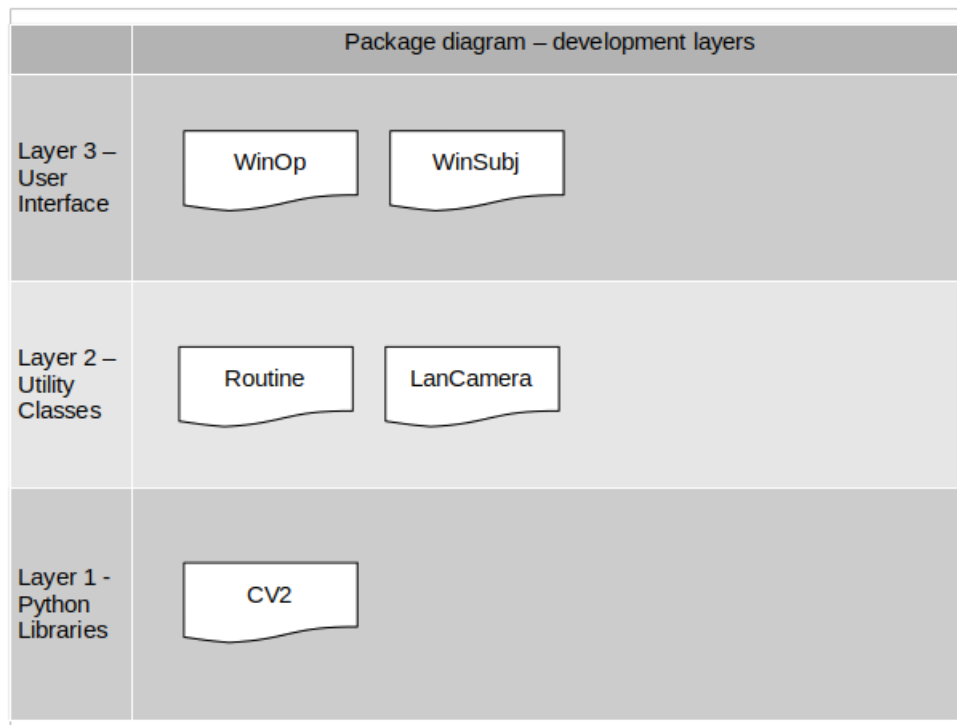


Figura 1 - Package diagram: deployment layers

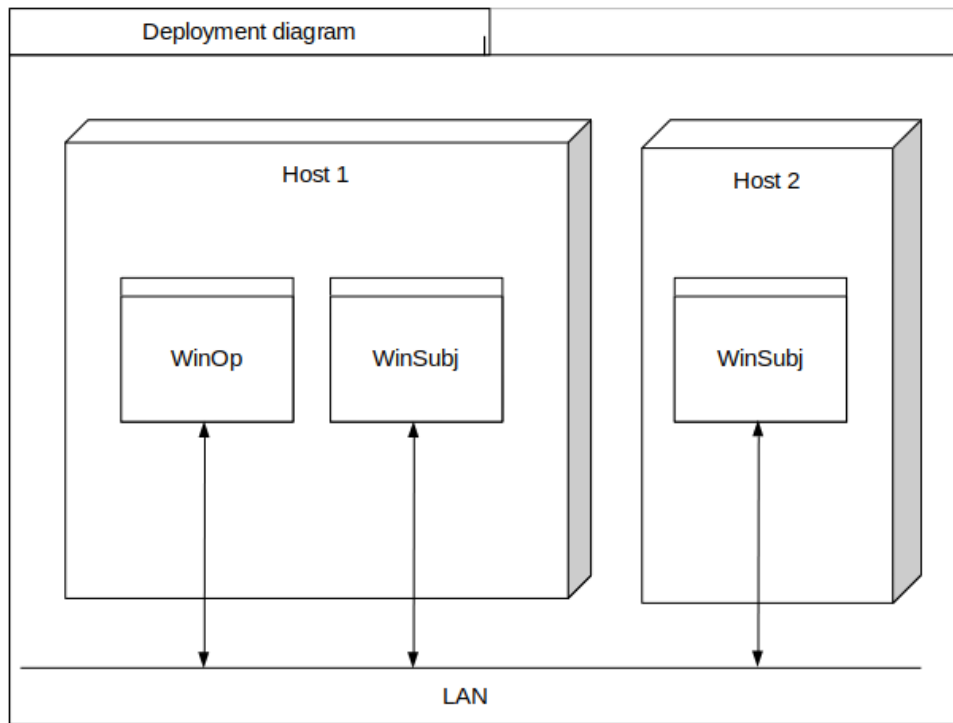


Figura 2 – Deployment diagram