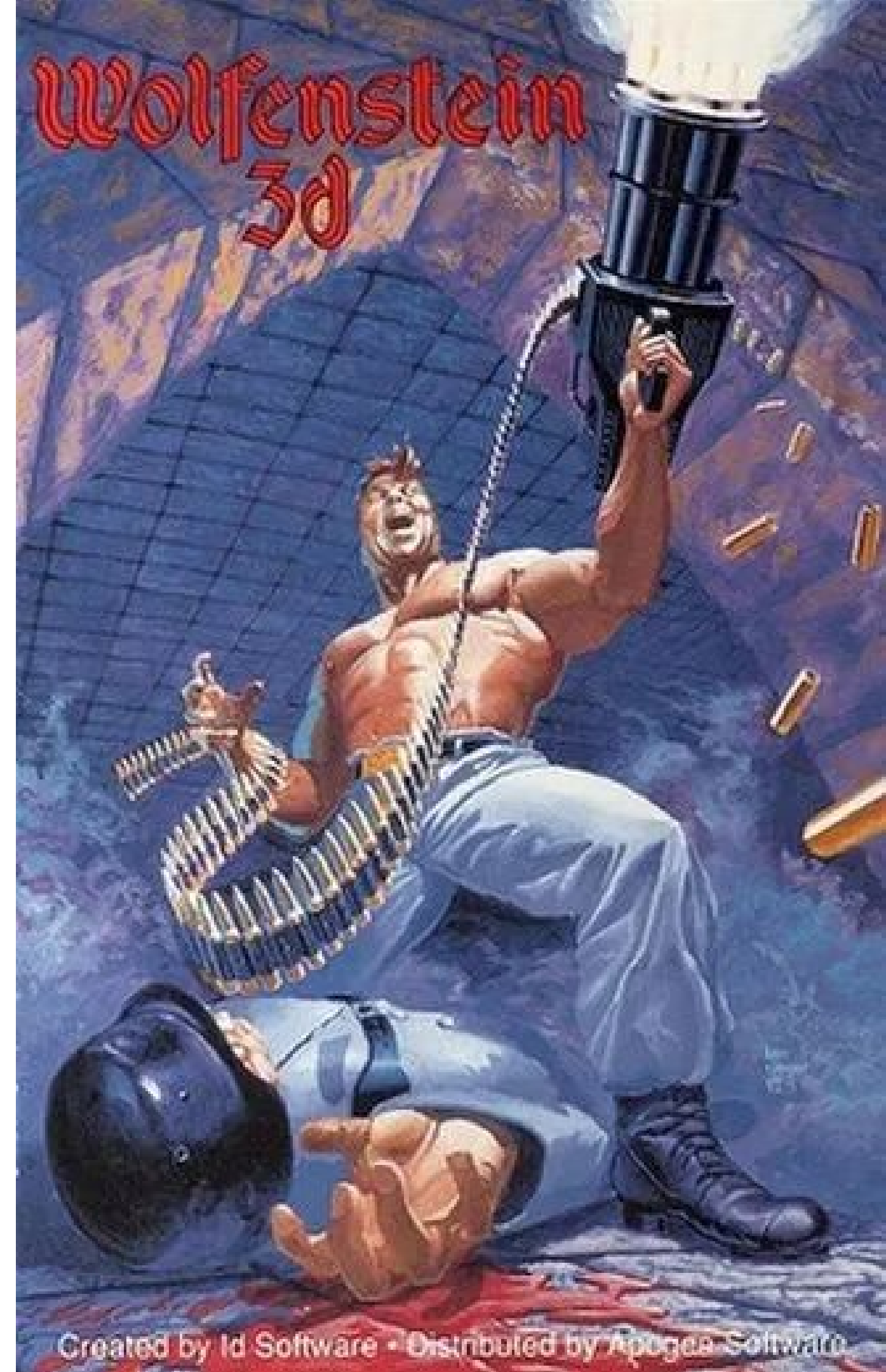


GAME ENGINE BLACK BOOK: WOLFENSTEIN 3D

CAP 4.5: INICIALIZAÇÃO

Erlon Felipe
Guilherme Gasperazzo
Gabriel Araújo
Pablo Barcellos
Lucas Pizzol

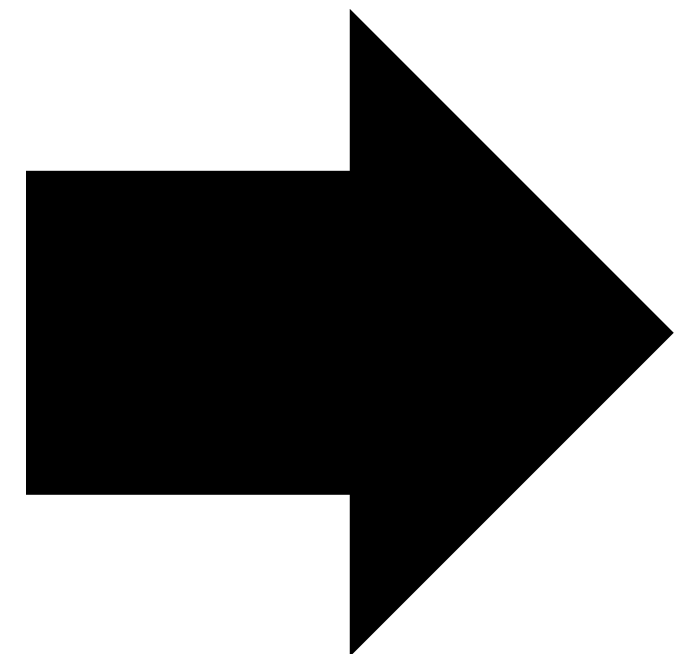


PROBLEMA PRINCIPAL

- Hardware **EXTREMAMENTE** limitado
- Lidar tecnologias com diversas variações através de soluções criativas

Como citado no
Cap.2(Hardware)

Neste contexto a engine do jogo exercia uma série de funções antes da execução efetiva do jogo



4.5.1 - TELA DE INICIALIZAÇÃO (SIGNON)

Desempenhava o papel de um sistema de autodiagnóstico

Função principal: identificava os recursos disponíveis no sistema do usuário e determinar se o jogo poderia ser executado adequadamente. Como:

- mouse
- joystick
- placas de som
- quantidade de memória disponível.



Sendo a limitação da memória RAM um dos pontos principais expostos nessa tela:

MEMÓRIA RAM

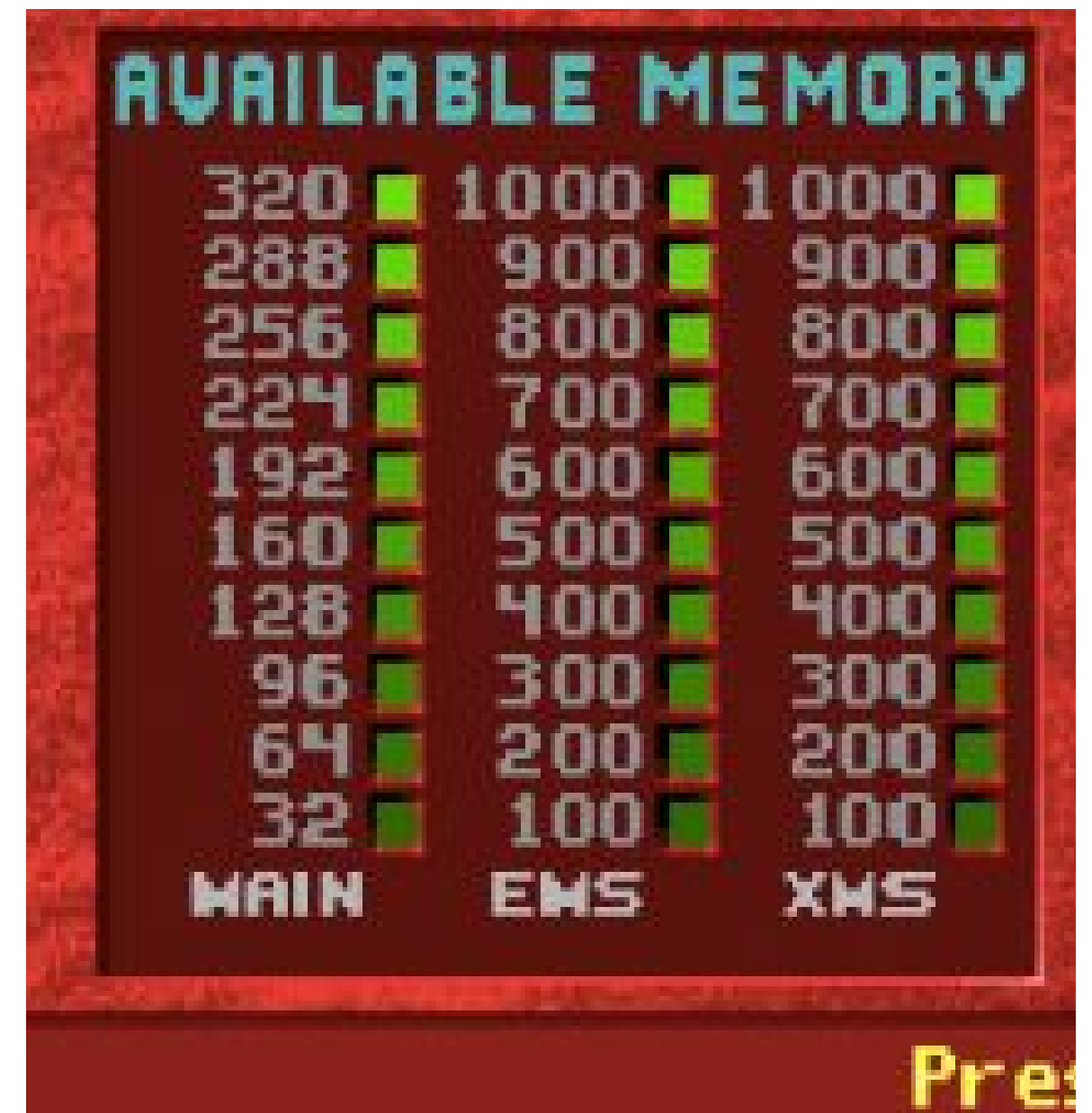
MAIN: representava a quantidade de memória convencional disponível

Sistemas baseados em DOS, os programas tinham acesso a apenas **640 KiB de RAM** convencional

Caso o sistema não dispusesse de memória suficiente para carregar o executável, o usuário recebia a mensagem:

Out of memory

E para o jogo funcionar corretamente, Wolfenstein 3D exigia **no mínimo 320 KiB** de memória convencional livre, metade do total convencional

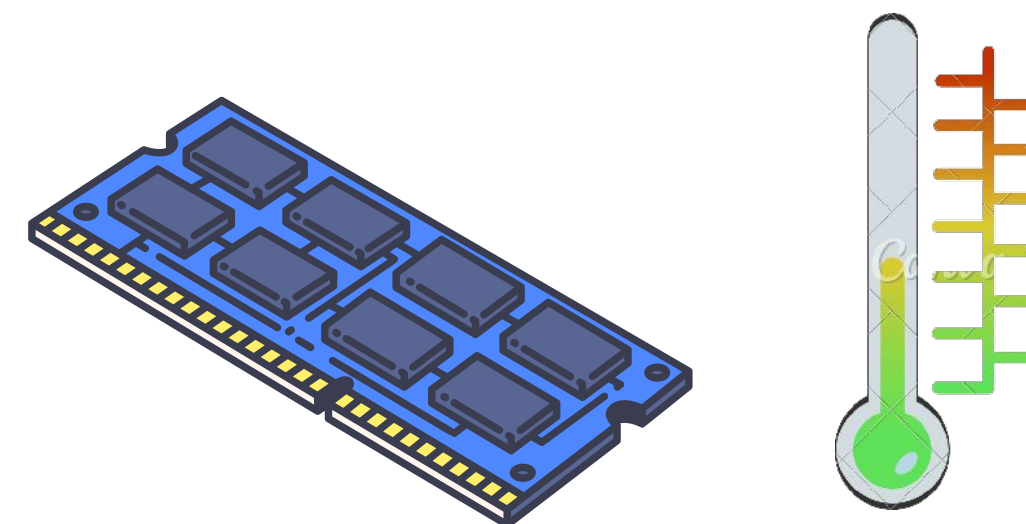




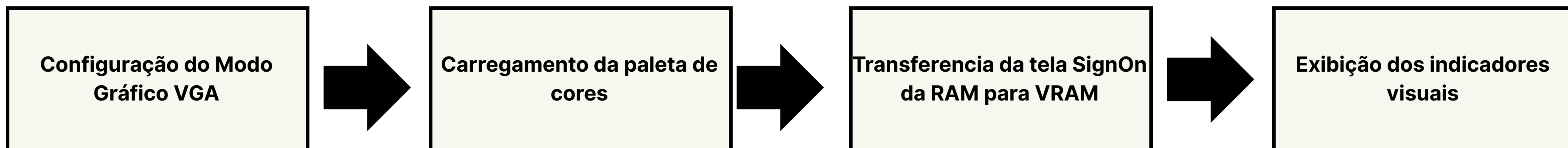
Implementação

Durante a exibição da tela de signon, o motor ainda não dispõe de um sistema de arquivos ativo. O único componente funcional é o gerenciador de memória

- Paleta de cores quanto a imagem da tela são incorporadas diretamente ao executável
- Garantia que os dados estejam carregados na memória assim que o programa é iniciado pelo DOS.



Rotina de inicialização:



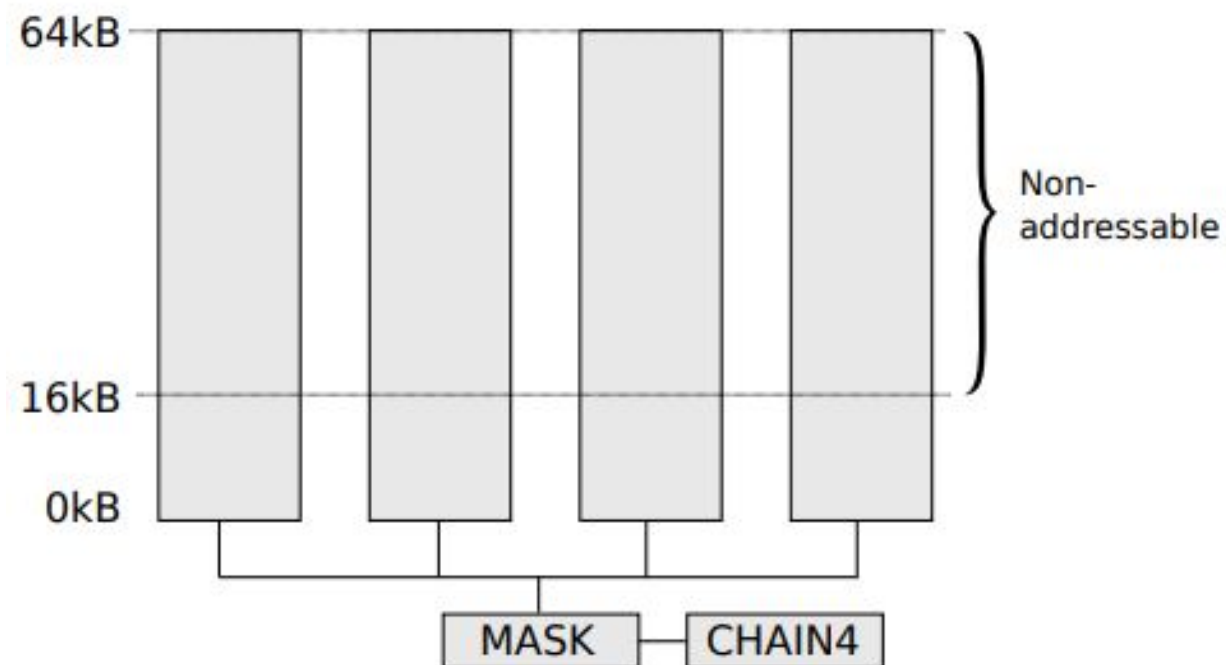
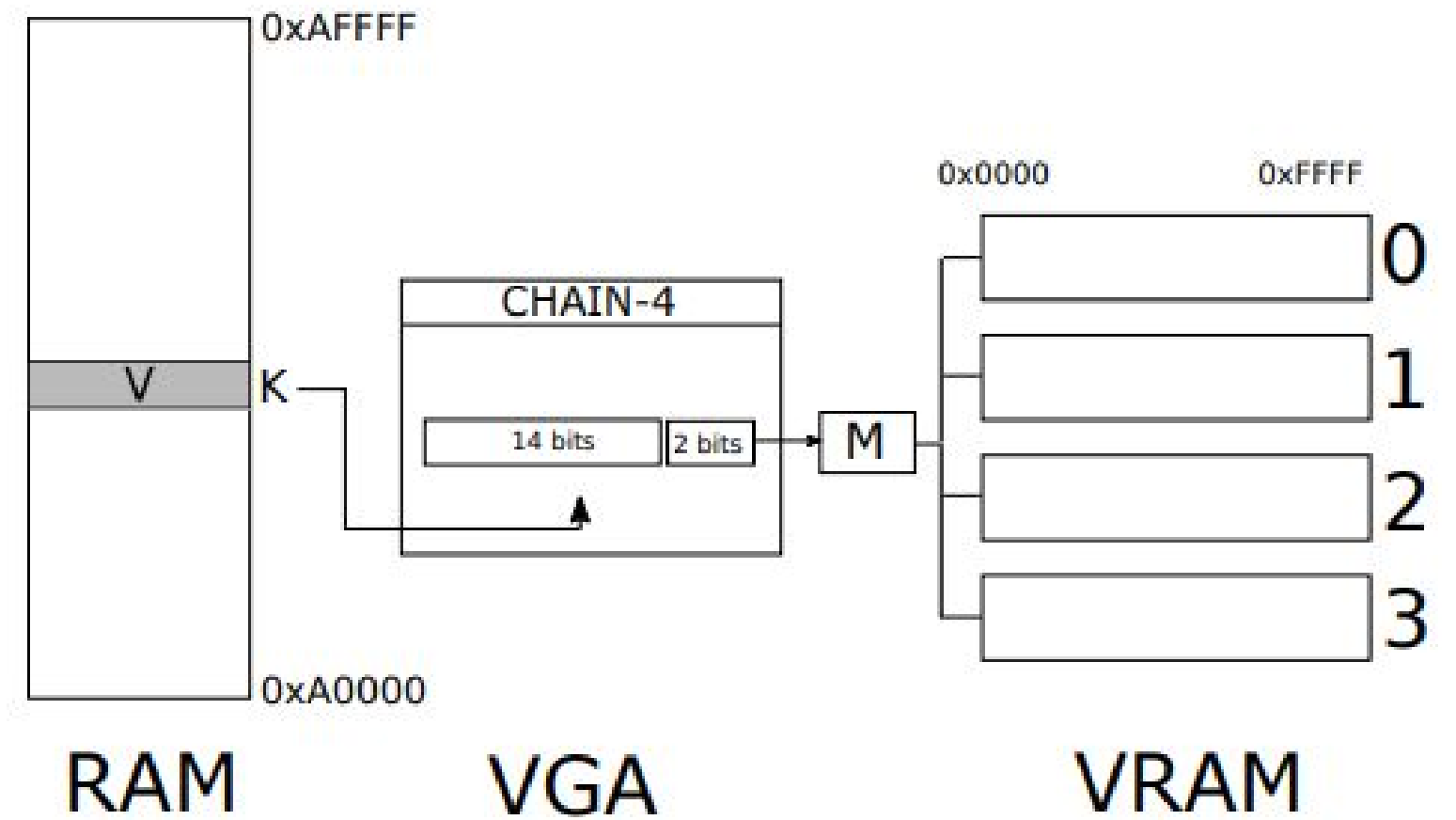
Ao final liberando 64.000 bytes de memória que a imagem ocupava

4.5.2 - RESOLVENDO O VGA

Ausência de Double Buffering causava efeitos visuais indesejados, como *Tearing*. E limitava o desempenho

- Modo padrão de Vídeo (13h): Resolução 320x200, 256 cores. Utilizava o sistema *Chain-4*

Chain-4: Era fácil de programar por utilizar de memória linear, PORÉM desperdiçava cerca de 75% de VRAM e limitava o buffering



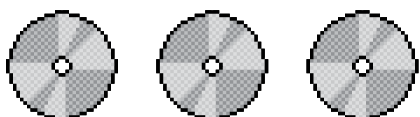
SOLUÇÃO: DESATIVAR O CHAIN-4

Criaram o chamado Mode-Y:

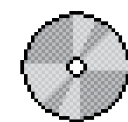
- Mantinha a mesma resolução de 320x200, mas permitia acessar os 256KB completos da VRAM e múltiplos buffers
- Diferente do Mode-X(320x240) que exigia mais processamento, o Mode-Y era mais eficiente e compatível com ferramentas gráficas da época

Logo

- A VRAM passou a ser organizada estrategicamente, garantindo 64Kb para cada framebuffer(até 3)
- Tornou possível a implementação do *triple buffering*, uma técnica que melhora a fluidez e elimina problemas visuais.



Após modificação:



- 64 KB → Framebuffer 1
- 64 KB → Framebuffer 2
- 64 KB → Framebuffer 3
- ~70 KB → Assets (imagens do jogo)



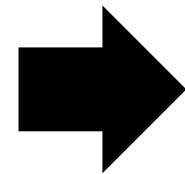
PROBLEMAS SURGIDOS

Performance(velocidade)

Sem o Chain-4, o programador precisava selecionar manualmente o banco(1-4) VRAM pra escrever = LENTIDAO

SOLUÇÃO: Desenhar verticalmente primeiro para minimizar o número de trocas de banco, assim deixando o código duas vezes mais rápido

```
void CleanScreen(int color) {
  for(int y=0 ; y < 200 ; y++) {
    for(int x=0; x < 320 ; x++) {
      selectPlane(x % 4);
      writePixel(x, y, color);
    }
  }
}
```



```
void CleanScreen(int color) {
  for(int x=0; x < 320 ; x++) {
    selectPlane(x % 4);
    for(int y=0 ; y < 200 ; y++) {
      writePixel(x, y, color);
    }
  }
}
```

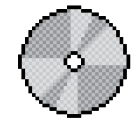
Tearing

Havia "cortes " na imagem pois o monitor desenhava mais rápido que a CPU escrevia.

SOLUÇÃO: Usaram a sincronização vertical (vsync): o computador esperava o monitor terminar de desenhar para só então escrever o novo

Por causa do **Mode Y** permitir uso do Triple Buffering, assim carregava 3 páginas na memória da VGA, onde o motor fica exibindo uma delas, outra pronta, esperando sua vez e a última sendo escrita pela CPU para o próximo frame

PROBLEMAS NO CRT



CRT CONTROLLER

Problema acontecia durante a troca de quadros de imagem na tela, onde duas imagens diferentes e desalinhadas eram apresentadas simultaneamente.

- **Ocorria por**

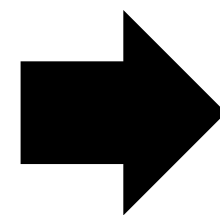
Endereço inicial do CRTC ter 16 bits, enquanto a instrução out era capaz de escrever apenas 8 bits de cada vez.

Por isto a atualização do endereço era dividido em dois passos:

- Gravação do byte superior no registrador
- Gravação do byte inferior no registrador

- **Exemplo**

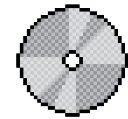
Se página 1 = 0x0000
e página 2 = 0x3E80



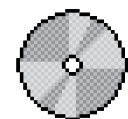
Atualiza o high byte da página de 0x00 para 0x3E, e o low byte de 0x00 para 0x80.



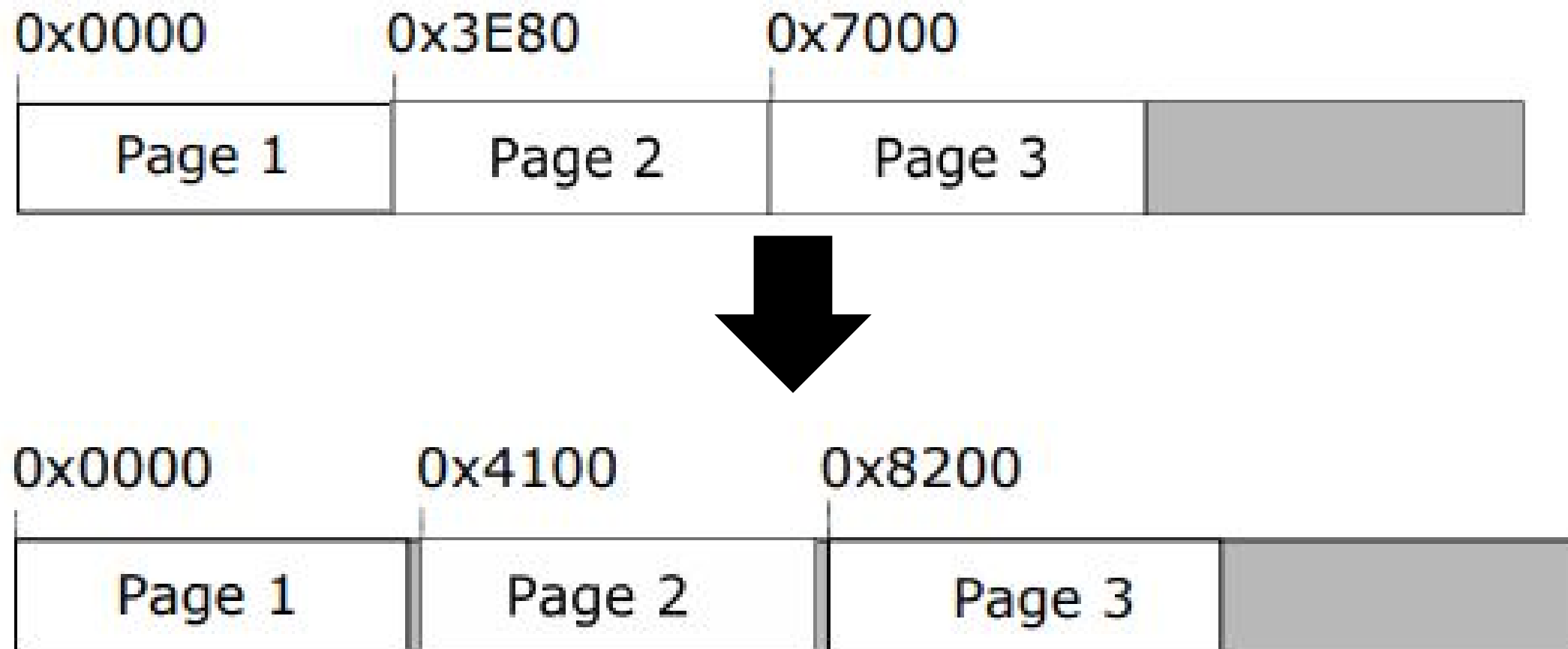
PROBLEMAS NO CRT



PROBLEMA: Pela atualização ser dividida em duas etapas, o endereço lido em algum instante poderia ser uma mistura das duas páginas, como 0x3E00, apresentando uma imagem desalinhada.



SOLUÇÃO: Simular uma resolução de altura 208 e utilizar um Offset de 8 bits para o valor da altura do framebuffer, tornando o endereço inicial múltiplo de 256.



4.5.3- CLASSIFICAÇÃO

Pelo lançamento do jogo no ano de **91**, quando ainda não existia um órgão responsável pela classificaçãoi indicativa dos jogos, como por violência excessiva ou conteúdo impróprio, os Devs desenvolveram a própria classificação ***Profound Carnage(PC-13)*** ao satirizar a utilizada em filmes (**PG-13**)



**OBRIGADO PELA
ATENÇÃO**

