

GCX Manual v0.01b

Copyright (c) 2013, Alysson Ribeiro da Silva & Alexei Manso Correa Machado
under FreeBSD License
All rights reserved

Acknowledgements

AntTweakBar Library was used to help this initial stage of development
Copyright (c) 2005-2013 Philippe Decaudin

Índice

Instalação.....	2
Execução.....	3
Execução pela interface wimp do windows.....	3
Execução pelo prompt de comando.....	3
Argumentos.....	3
Argumentos suportados v0.01b.....	3
Utilização.....	4
Operações básicas.....	7
Depuração.....	8
Terminal de visualização.....	8
Identificador de próxima instrução.....	9
Modo de depuração.....	10
Operações secundárias.....	14
Menu de operações.....	14
Operações secundárias v0.01b.....	14

Instalação

O GCX em sua atual versão não possui instalador, para que o mesmo seja utilizável basta extrair os arquivos oficiais disponibilizados que se encontram compactados em formato ZIP. Para garantir a integridade da execução do simulador é de vital importância que o utilizador verifique o conteúdo do arquivo compactado, que deve possuir 1.050 KB e os seguintes arquivos:

GCX.exe

AntTweakBar.dll

DevIL.dll

glut32.dll

defaultCube.exe

execution.png

nextButton.png

playButton.png

prevButton.png

stopButton.png

Tais arquivos estarão então organizados em quatro pastas, sendo elas a **raiz**, onde estão todos os outros arquivos, a pasta **content**, onde se encontram as pastas **bin** e a pasta **textures**.

Se o processo de extração for bem-sucedido e conter todos os arquivos e pastas citados acima o simulador está pronto para uso.

Nota01: Os arquivos podem ser extraídos em qualquer diretório, fica a critério do utilizador.

Nota02: O programa foi criado com ajuda de algumas ferramentas da Microsoft, caso haja algum problema relacionado a dll's com o que diz respeito a alguma ferramenta da Microsoft, executar solução proposta na Nota03.

Nota03: Caso haja algum problema relacionado a falta de dll's quando o simulador for executado basta fazer o download da mesma e colocar na pasta raiz do programa. Os links para download podem variar, uma pesquisa no site www.google.com.br é aconselhada.

Execução

Execução pela interface wimp do windows

Para executar o simulador basta dar um clique duplo no arquivo **GCX.exe**. Neste caso o arquivo padrão **defaultCube.exe** escrito na linguagem GL será aberto. Para executar um arquivo qualquer basta arrastar o mesmo para cima do executável do simulador.

Execução pelo prompt de comando

Para executar através do *prompt* de comando a sintaxe é:

<exe> <nome do arquivo> <arg(1) ... arg(n)>

Um exemplo de execução seria “C:\GCX\GCX.exe meuCubo.exe”, tal comando executaria o simulador utilizando o arquivo **meuCubo.exe** que é um programa escrito na linguagem GL.

Argumentos

Para executar o programa utilizando argumentos é necessário disponibilizar os mesmos para o simulador de alguma forma, um dos possíveis meios de se fazer isto é executar o simulador através do *prompt* utilizando a sintaxe:

<exe> <nome do arquivo> <arg(1) ... arg(n)>

Um exemplo de execução seria “C:\GCX\GCX.exe quadrado.exe -debug”.

Argumentos suportados v0.01b

-debug → executa o simulador em modo de depuração

IMPORTANTE !!!

Nota04: Todos os arquivos compilados na linguagem GL que o usuário deseje executar no simulador através do *prompt* de comando devem estar presentes na pasta “../content/bin” que se encontra dentro da pasta raiz.

Nota05: Todos os arquivos que o usuário deseje executar no simulador através da interface wimp arrastando os mesmos para cima do executável do simulador devem estar na pasta raiz, ou erros graves podem ocorrer.

Nota06: Caso nenhum arquivo seja especificado um arquivo padrão chamado defaultCube.exe que se encontra na pasta “../content/bin” será utilizado na simulação.

Utilização

Após executar o simulador duas janelas deverão ser abertas, uma delas é o *prompt* de comando, a outra contém a interface primária do simulador, tais janelas podem ser vistas na figura abaixo.

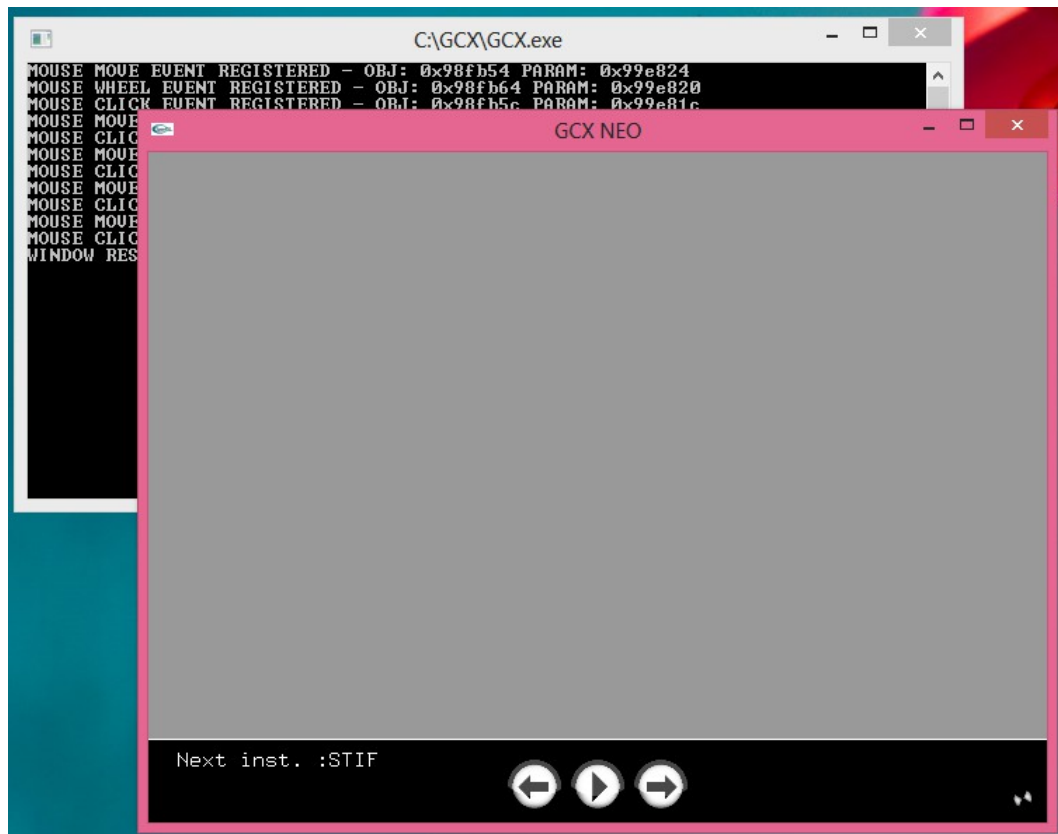


Figura 1 – Interfaces de comunicação do simulador GCX.

Durante o processo de abertura do programa alguns eventos são registrados, tais eventos garantem a execução correta de alguns mecanismos do simulador, são eles:

MOUSE MOVE EVENT

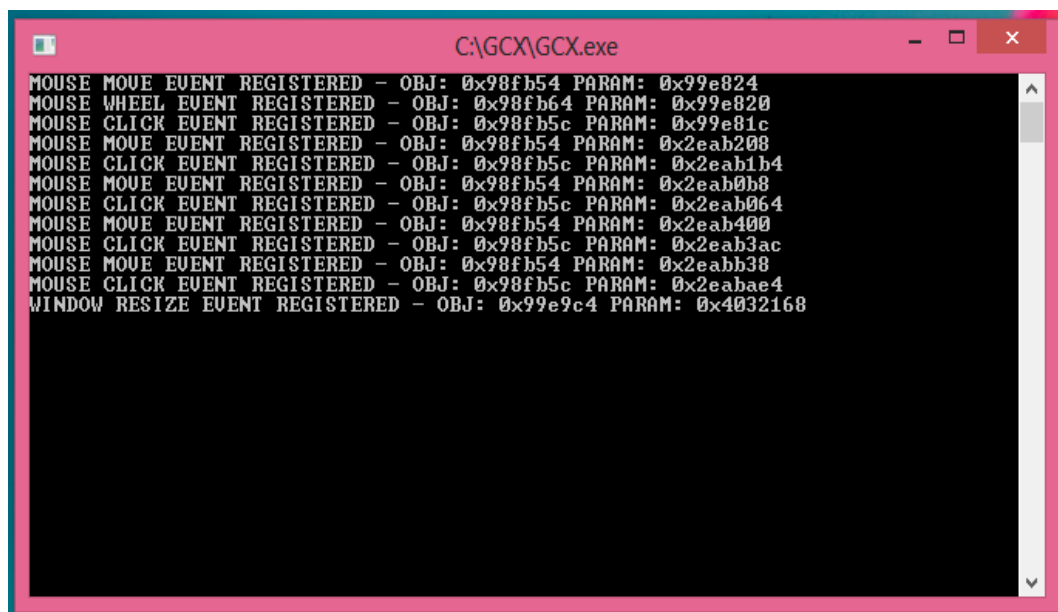
MOUSE WHEEL EVENT

CLICK EVENT

RESIZE EVENT.

É possível visualizar quais eventos foram abertos no *prompt* de comando do simulador, a figura abaixo ilustra quais eventos são obrigatórios e de vital importância para a execução, caso algum deles esteja comprometido a integridade das operações não é garantida. Os valores em hexadecimal presentes representam o endereço de memória onde os objetos do programa foram alocados, a sua posição não é relevante, a menos que seja algum valor absurdo como 0xFFFFFFFF, neste caso um erro grave ocorreu.

Caso qualquer outra mensagem tenha sido mostrada no *prompt* algum erro fatal ocorreu.



```
C:\GCX\GCX.exe
MOUSE MOVE EVENT REGISTERED - OBJ: 0x98fb54 PARAM: 0x99e824
MOUSE WHEEL EVENT REGISTERED - OBJ: 0x98fb64 PARAM: 0x99e820
MOUSE CLICK EVENT REGISTERED - OBJ: 0x98fb5c PARAM: 0x99e81c
MOUSE MOVE EVENT REGISTERED - OBJ: 0x98fb54 PARAM: 0x2eab208
MOUSE CLICK EVENT REGISTERED - OBJ: 0x98fb5c PARAM: 0x2eab1b4
MOUSE MOVE EVENT REGISTERED - OBJ: 0x98fb54 PARAM: 0x2eab0b8
MOUSE CLICK EVENT REGISTERED - OBJ: 0x98fb5c PARAM: 0x2eab064
MOUSE MOVE EVENT REGISTERED - OBJ: 0x98fb54 PARAM: 0x2eab400
MOUSE CLICK EVENT REGISTERED - OBJ: 0x98fb5c PARAM: 0x2eab3ac
MOUSE MOVE EVENT REGISTERED - OBJ: 0x98fb54 PARAM: 0x2eabb38
MOUSE CLICK EVENT REGISTERED - OBJ: 0x98fb5c PARAM: 0x2eabae4
WINDOW RESIZE EVENT REGISTERED - OBJ: 0x99e9c4 PARAM: 0x4032168
```

Figura 2 – ilustração de eventos que devem ser abertos obrigatoriamente junto ao simulador.

O simulador pode trabalhar em modo **normal** e em modo *depuração*, em ambos ele disponibiliza para o usuário quatro operações básicas: **Forward Time Step**, **Backward Time Step**, **Play**, **Pause**. As operações básicas podem ser acessadas através da interface primária do simulador como visto na figura abaixo.

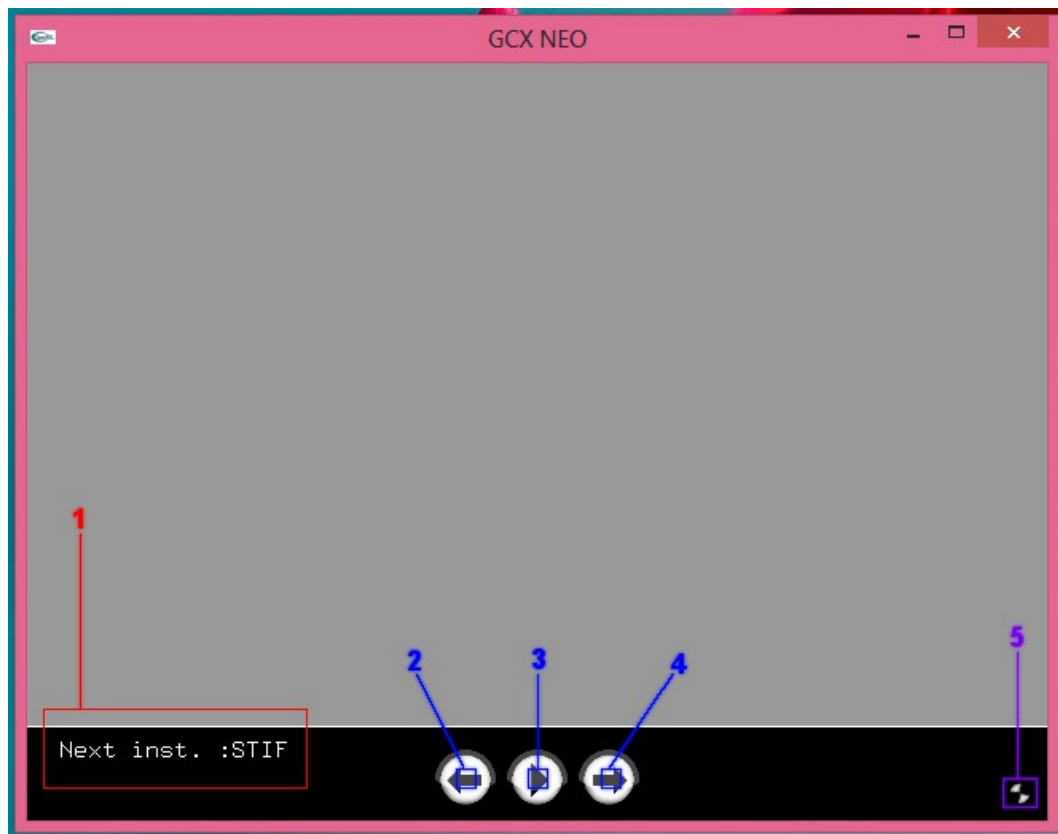


Figura 3 – Identificação de componentes da interface primária do simulador.

Componentes da interface primária:

1 – Identificador de próxima instrução a ser executada pelo simulador

2 – Backward time Step button

3 – Play / Pause button

4 – Forward time Step button

5 – Gui execution flow → permite visualização do fluxo de execução da GUI (velocidade de execução)

Operações básicas

Backward Time step → Permite voltar uma instrução para o passado.

Forward Time step → Permite ir uma instrução para o futuro.

Play/Pause → Permite continuar e pausar a simulação respectivamente.

O simulador pode se encontrar em dois estados, sendo eles, '**pausa**' e '**simulando**'. Seu estado inicial é o estado '**pausa**', para trocar para o estado '**simulando**' basta apertar o botão **Play/Pause**. Ao clicar no botão **Play/Pause** forçando o simulador a ir para o estado '**simulando**' a cena tridimensional então será renderizada conforme descrito no arquivo aberto durante a execução do simulador. Abaixo uma ilustração da execução do arquivo padrão presente na pasta "**.../content/bin**".

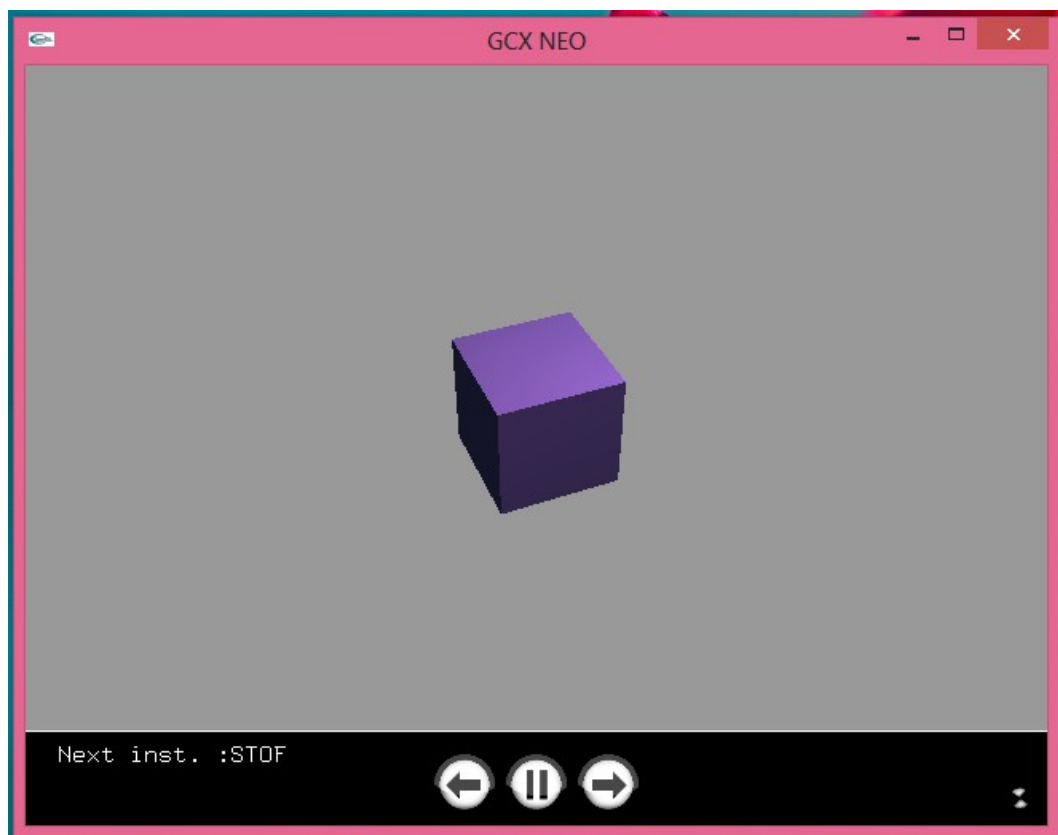


Figura 4 – Ilustração de execução padrão do simulador.

Depuração

Para a depuração do código fonte escrito em GL e compilado para a arquitetura especificada três ferramentas podem ser utilizadas no simulador, o **Terminal de Visualização**, o **Identificador de Próxima Instrução** e o menu de depuração do **Modo de Depuração**.

Terminal de visualização

No terminal de visualização é possível ver os valores dos registradores do simulador para a última (mais recente) instrução executada, sendo eles um total de doze, seis para valores inteiros e seis para valores reais, assim como o valor interno do registrador **PC**. Tais valores são mostrados no terminal de visualização apenas quando uma operação do tipo **Backward Time Step** ou **Forward Time Step** é executada. Abaixo uma ilustração do conteúdo dos registradores para a última (mais recente) instrução executada.

A screenshot of a terminal window titled 'C:\GCX\GCX.exe'. The terminal displays the following text:

```
Bf - 0.100000
Cf - 0.000000
Df - 0.000000
Ef - 0.000000
Ff - 0.000000
-----
Last instruction: LOD
Last PC: 420
Last int registers values:
A - 1
B - 499
C - 0
D - 0
E - 0
F - 0
Last float registers values:
Af - 4.990000
Bf - 0.100000
Cf - 0.000000
Df - 0.000000
Ef - 0.000000
Ff - 0.000000
-----
```

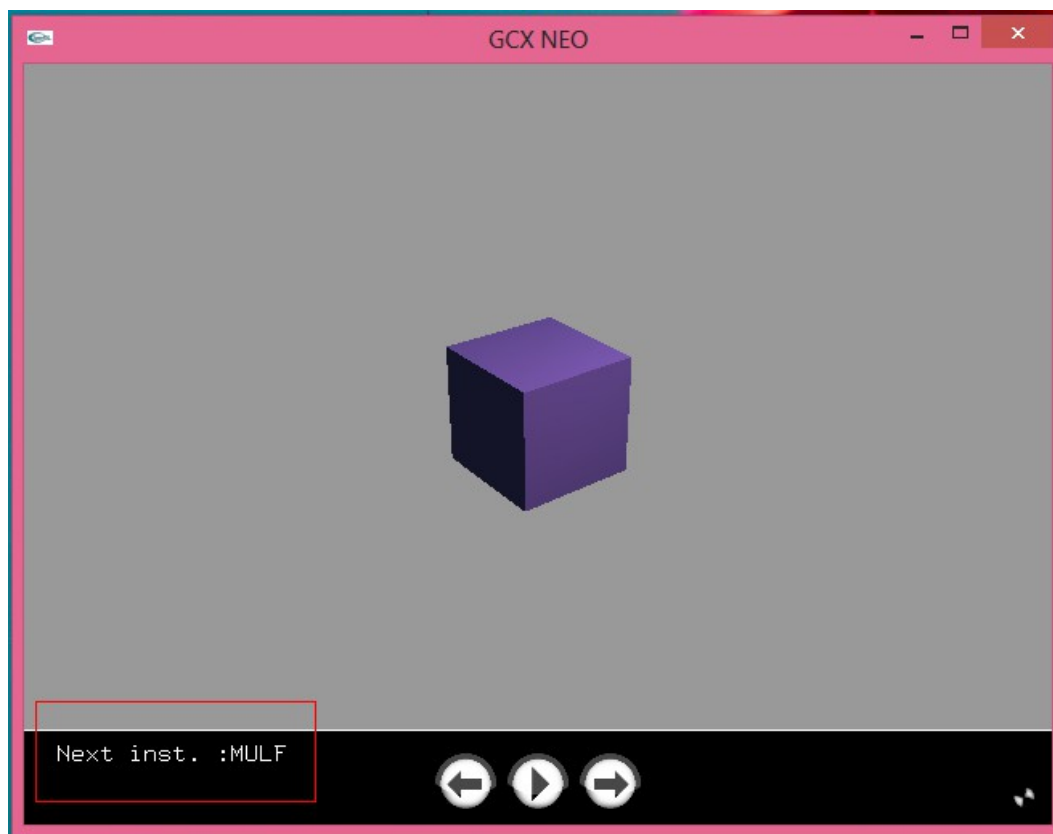
Figura 5 – Visualização do conteúdo dos registradores no terminal do simulador para a instrução mais recente após uma operação de Time Step ser executada.

IMPORTANTE!!!

Nota07: É importante que para executar uma operação Time Step o simulador esteja em estado 'pausa' para que os valores para a última (mais recente) instrução executada que serão mostrados no terminal sejam válidos.

Identificador de próxima instrução

Para visualizar simplesmente qual a próxima instrução que será executada pelo simulador basta olhar o **Identificador de Próxima Instrução**, lembrando que o valor presente neste **Identificador** não é o mesmo apresentado no **Terminal de Visualização**. Abaixo uma ilustração de próxima instrução a ser executada para um determinado momento em uma simulação com o arquivo padrão.



*Figura 6 – Visualização de próxima instrução a ser executada pelo simulador, neste caso **MULF**.*

IMPORTANTE !!!

Nota08: A instrução mostrada pelo Identificador de próxima instrução não é a mesma instrução visualizada no Terminal de Visualização, sendo que o terminal mostra a última (mais recente) instrução que foi executada, e o Identificador mostra qual a próxima instrução a ser executada.

Modo de depuração

Em modo de depuração é possível modificar os valores internos dos registradores em tempo de execução, assim como é possível visualizar mais precisamente o que está ocorrendo dentro do simulador com relação ao valor interno dos registradores em tempo de execução. A interface padrão para o modo de depuração é mostrada na figura abaixo.

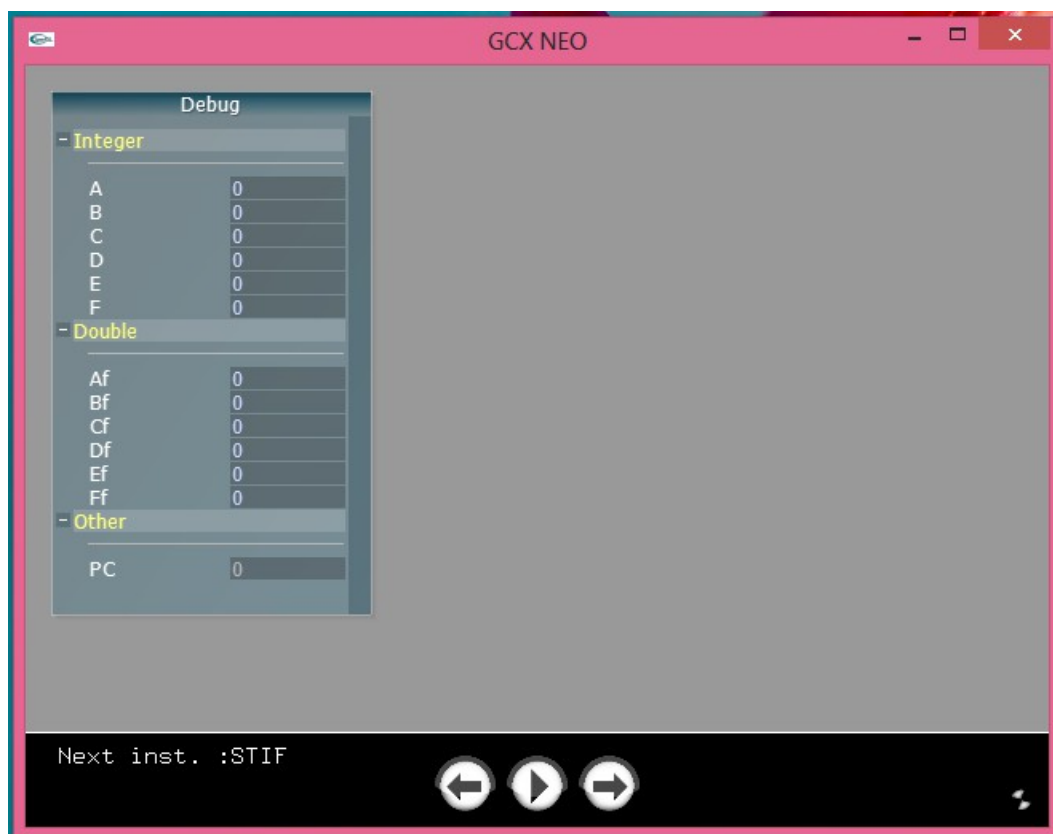


Figura 7 – Interface padrão do simulador em modo de depuração.

IMPORTANTE !!!

Nota09: Os valores mostrados na janela interna são atualizados apenas quando o simulador está no estado 'simulando'.

Nota10: O valor interno do registrador PC não pode ser modificado por questões relacionadas com a integridade de execução do programa.

Através da janela interna '**Debug**' no simulador é possível modificar o valor interno dos registradores utilizando os botões controladores de cada um deles, os mesmos são visíveis apenas ao passar o cursor do mouse sobre a barra representativa onde o valor do registrador é mostrado, como pode ser visto na figura abaixo.

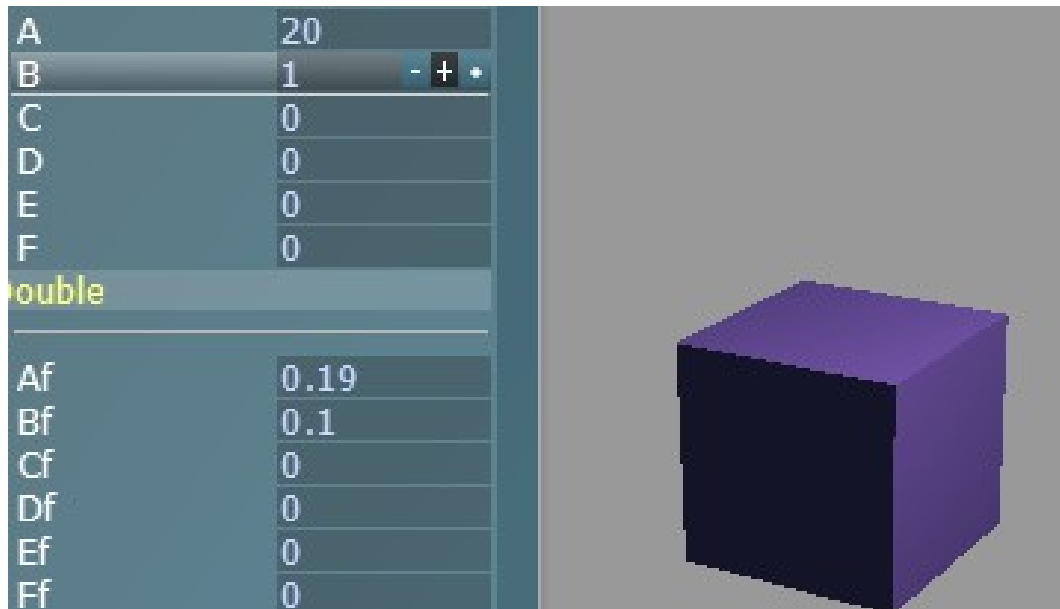


Figura 8 – Botões para ajuste de valor na barra representativa.

A janela interna de '**Debug**' pode ser movida dentro do espaço da janela do simulador, porém não pode ser redimensionada e nem fechada uma vez que o simulador tenha sido aberto em modo de depuração.

IMPORTANTE!!!

Nota11: A renderização é afetada apenas quando o simulador está em estado 'simulando', caso o valor interno de um registrador seja modificado e o simulador esteja no estado 'pausa' seu efeito poderá ser visto quando o estado for modificado para 'simulando'.

O botão representando o símbolo '-' decrementa UMA unidade o valor.

O botão representando o símbolo '+' incrementa UMA unidade o valor.

O terceiro botão representa um círculo trigonométrico, ao segurar o botão esquerdo do mouse sobre o mesmo o círculo poderá ser visualizado, se o botão esquerdo do mouse for mantido pressionado e o cursor do mouse for movido de forma a rodear o círculo apresentado no sentido anti-horário o valor é incrementado, se o cursor do mouse for movido de forma a rodear o círculo no sentido horário o valor é decrementado, a quantidade incrementada e decrementada depende do tipo de variável representada pela barra representativa. Abaixo uma ilustração da utilização do mesmo.

Os valores também podem ser modificados dando-se um clique diretamente sobre a barra e digitando o valor desejado.

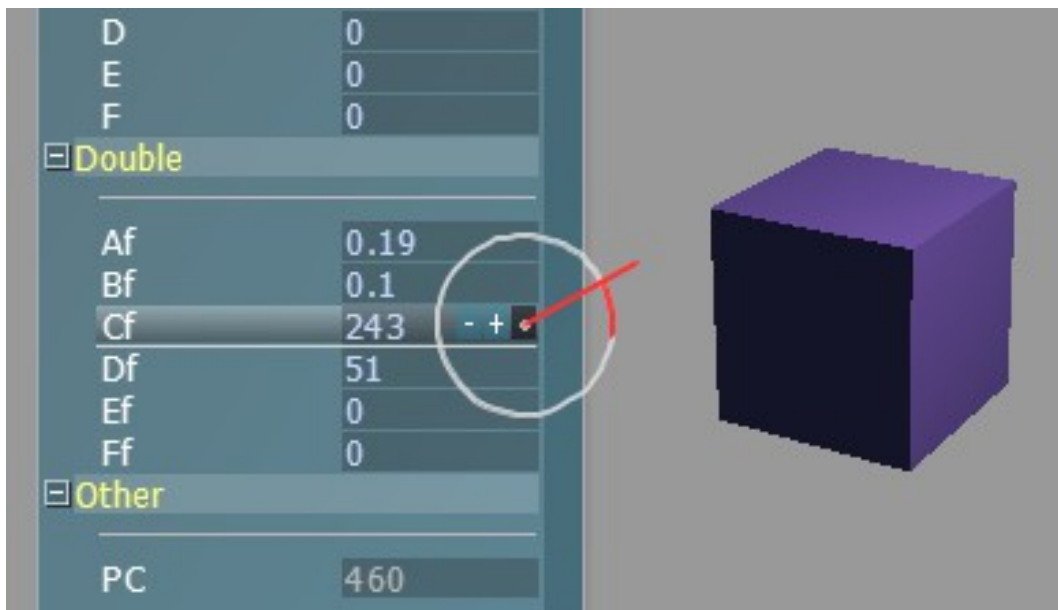


Figura 10 – Utilização do modificar trigonométrico (RotoSlider).

A janela de '**Debug**' pode ser minimizada e sua fonte interna pode ter seu tamanho modificado entre pequeno, grande e médio. Os botões para minimização e mudança de tamanho de fonte são indicados nas figuras abaixo respectivamente.



Figura 11 – Identificação de botão para mudança de tamanho de fonte da janela interna de depuração.

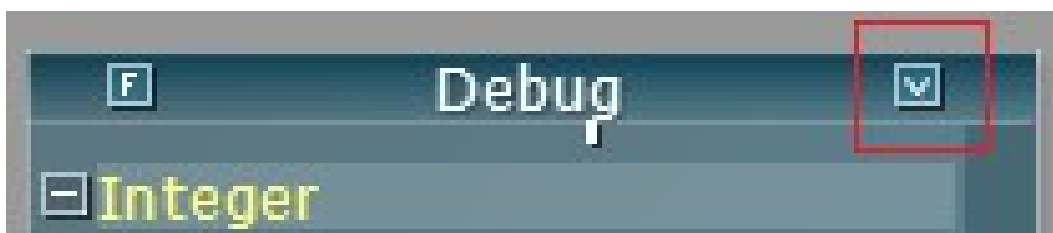


Figura 12 – Identificação de botão para minimização da janela interna de depuração.

Após minimizada, a janela de '**Debug**' pode ser novamente maximizada, um botão indicando que a janela foi minimizada se encontrará no canto inferior esquerdo da janela do simulador logo abaixo do **Indicador de Próxima Instrução** como visto na figura abaixo.

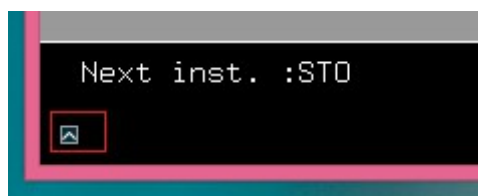


Figura 13 – Identificação de botão para maximização da janela de depuração do simulador.

Operações secundárias

Menu de operações

O menu de operações disponibiliza para o usuário as operações secundárias. Pode ser acessado clicando-se com o botão direito do mouse sobre a janela do simulador, o mesmo pode ser visto na figura abaixo.

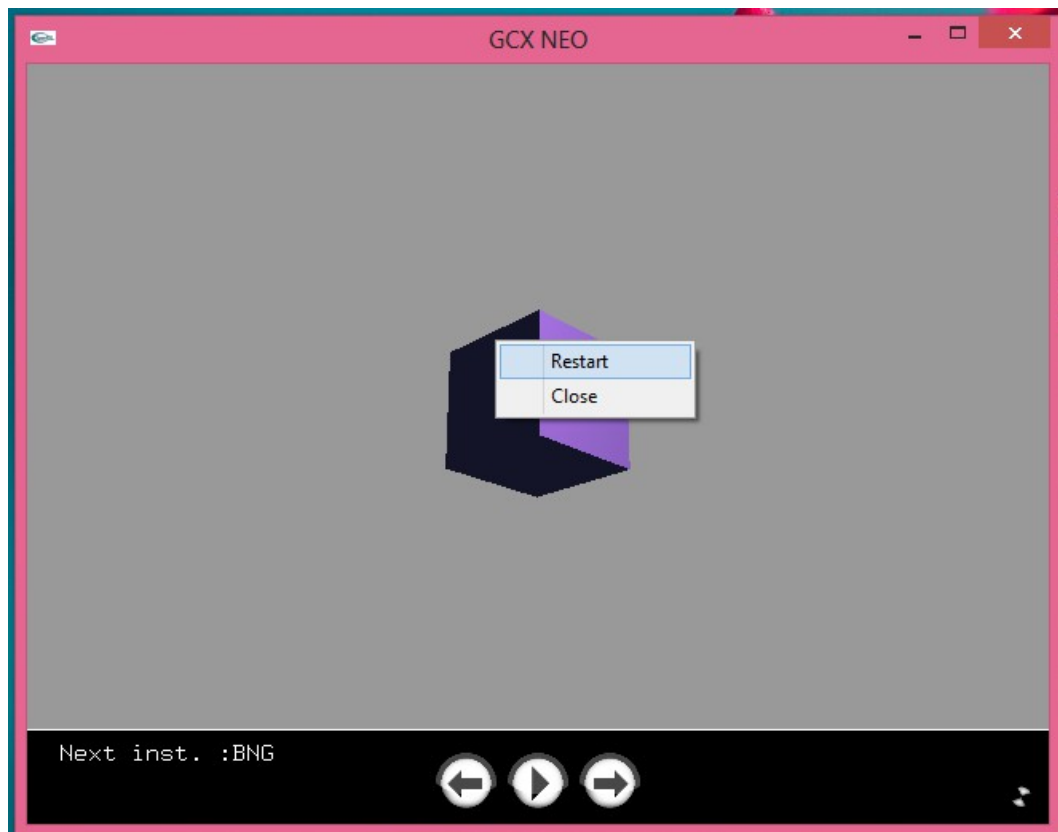


Figura 14 – Menu de operações, agrupa todas as operações secundárias do simulador.

Operações secundárias v0.01b

Restart → Reinicializa a simulação zerando o valor de todos os registradores, incluindo o PC.

Close → Fecha o programa.

IMPORTANTE!!! NOTAS FINAIS!!!

Nota12: Toda a interação entre o usuário e os elementos da interface são feitos através do mouse e teclado.

Nota13: Toda interação relacionada ao acionamento da funcionalidade de cada botão da interface é feito utilizando-se o botão esquerdo do mouse.

Nota14: Caso ocorra algum problema relacionado ao não carregamento das texturas da interface no geral verificar notas anteriores.

Nota15: Nem todas as placas de vídeo suportam OpenGL, logo em alguns computadores podem haver problemas de compatibilidade, neste caso não há o que fazer... ;)