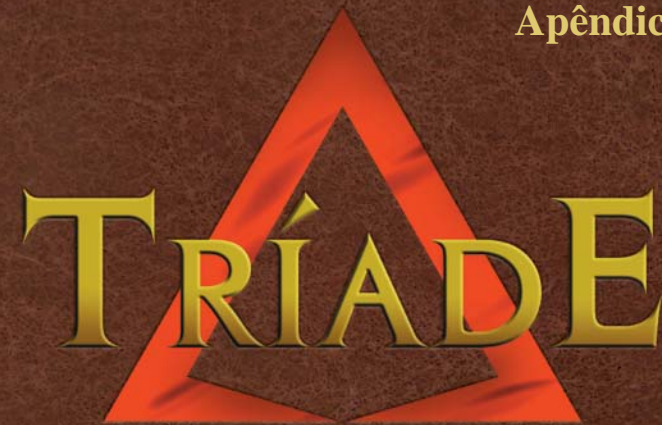


GRUPO DE DESENVOLVIMENTO (GD)
DE PROGRAMAÇÃO - MEMORIAL TÉCNICO

Apêndice 2



LIBERDADE, IGUALDADE E FRATERNIDADE



LIBERDADE, IGUALDADE E FRATERNIDADE

Arivan Bastos

Financiamento:



FINANCIADORA DE ESTUDOS E PROJETOS
MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

Apoio:



UNEB



Fundação de Amparo
à Pesquisa do Estado da Bahia



Conselho Nacional de Desenvolvimento
Científico e Tecnológico



GOVERNO DA
Bahia
TERRA DE TODOS NÓS
Secretaria de Ciência,
Tecnologia e Inovação

Financiamento e Apoio

O Tríade – Igualdade, Liberdade e Fraternidade é um jogo desenvolvido pelo Grupo de Pesquisa Comunidades Virtuais, vinculado ao Mestrado de Educação e Contemporaneidade, do Departamento de Educação do Campus I da UNEB, financiado pelo MCT/FINEP/MEC, com apoio da Universidade do Estado da Bahia (UNEB), bolsas da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Contou com uma equipe formada por profissionais e alunos das áreas de informática, design, arte, história, música e pedagogia que, juntos, aceitaram o desafio de desenvolver um jogo eletrônico voltado para educação com a mesma lógica e qualidade dos jogos comerciais.

O Tríade é um jogo de Aventura com traços de Role Playing Game (RPG), no modo single player, que tem o objetivo de envolver alunos e professores de História no universo da Revolução Francesa.

O jogador é desafiado a resolver quests que estão relacionadas com os acontecimentos que marcaram esse período histórico de forma lúdica, tendo a possibilidade de escolher dois caminhos, durante a segunda e última fase do jogo, ampliando, assim, os níveis de interatividade.

O desafio de desenvolver este jogo teve início em março de 2006, quando começamos a escrever o projeto para atender a demanda do edital do FINEP, mas foi em setembro do mesmo ano, que o grupo, recebeu a notícia de que estava entre os treze selecionados, dentre quase 200 projetos em todo o Brasil. Na Bahia, foram selecionados três projetos (SENAI, UEFS/UFBA e UNEB). As ações ligadas ao desenvolvimento, arte e design foram intensificadas a partir de abril de 2007. Assim, o jogo levou trinta e três meses de desenvolvimento a partir da elaboração do seu projeto.

GD DE PROGRAMAÇÃO MEMORIAL TÉCNICO

TRÍADE

LIBERDADE, IGUALDADE E
FRATERNIDADE

Grupo de Pesquisa
COMUNIDADES VIRTUAIS

Financiamento:



Apoio:



Equipe:**Coordenadora**

Lynn Alves

Game Design

Abelmon de Oliveira Bastos

Alexandre Macêdo Santos

Arivan Bastos

Danilo Assis Santos de Alencar

Felipe Assis Santos de Alencar

Felix de Souza Neto

Gildeon Sena

Isa Beatriz Neves

Leonardo Pessoa

Lynn Alves

Marcus Ventura Araújo

Raphael Montenegro

Roteiro

Camila Santana

Gildeon Sena

Gilmara Silva Santos

Isa Beatriz da Cruz Neves

Jodeilson Mafra Martins

Juliana Moura

Lynn Alves

Historiadores

Gilmara Silva Santos

Isa Beatriz da Cruz Neves

Jodeilson Mafra Martins

Artes

Abelmon de Oliveira Bastos

Alexandre Macêdo Santos

Anderson Alves

Arivan Bastos

Danilo Assis Santos de Alencar

Daniel Araújo

Felipe Assis Santos de Alencar

Felix de Souza Neto

Leandro Marcondes

Marcus Ventura Araújo

Mauro Lira

Ozias Lopes

Raphael Montenegro

Shalon Cerqueira

Tiago Pessoa

Tuíris Azevedo

Programação

Arivan Bastos

Leonardo Pessoa

Nilo Dantas

Trilha Sonora

Aldemar Jr.

Bolsista Pibic-junior

Joseane Santos

Bolsista Pibic

Alessandra Silva

Tatiana Paes

Vanessa Rios

Consultoria

Eduardo Borges (História – UNEB)

Esteban Clua (Game Design – UFF)

Revisora Linguística

Obdália Ferraz (Letras – UNEB)

Agradecimento especial in memoriam

Durval Lordelo (UNEB)

SUMÁRIO

INTRODUÇÃO.....	6
MOTORES DE DESENVOLVIMENTO DE JOGOS.....	6
O TORQUE GAME ENGINE.....	8
PREPARAÇÃO DO AMBIENTE	9
AMBIENTE WINDOWS.....	9
UTILIZANDO O MICROSOFT VISUAL C++ 6.0 PARA COMPILAR A ENGINE AMBIENTE LINUX.....	9
INSTRUÇÕES PARA COMPILAÇÃO EM LINUX.....	10
UTILIZANDO O TORQUEDEV /CODEWEAVER 1.2 PARA EDIÇÃO DE SCRIPTS	10
A ENGINE.....	15
PACKAGES.....	17
PRINCIPAIS CLASSES.....	17
FUNÇÕES ÚTEIS.....	17
RESOURCES UTILIZADOS.....	19
RPGDIALOG.....	20
IMMERSIVEAI.....	20
CLIMABLE OBJECTS.....	21
MELEE COMBAT.....	21
INVENTORY SYSTEM.....	22
DICAS.....	22
UTILIZANDO O FORMATO THEORA (.OGG).....	22
UTILIZANDO SOM (.OGG).....	23

INTRODUÇÃO

Esse documento apresenta a experiência do grupo de pesquisa Comunidades Virtuais (em particular da equipe de programação) no desenvolvimento do projeto Triade: “Mediando o processo ensino aprendizagem da História”, utilizando o motor Torque Game Engine (TGE). O objetivo deste documento é compartilhar o aprendizado adquirido ao longo do projeto, bem como apresentar as principais dificuldades encontradas e como elas foram solucionadas (quando foram).

MOTORES DE DESENVOLVIMENTO DE JOGOS

Um motor de jogos, ou engine, é o software responsável por oferecer ao desenvolvedor os recursos necessários para que ele, a partir de sua idéia, crie um jogo eletrônico. Da mesma forma que quando desejamos criar uma apresentação em slides utilizamos o Microsoft PowerPoint, ou quando desejamos criar uma planta 3D utilizamos uma ferramenta CAD, quando desejamos criar um jogo, utilizamos um motor de jogos.

Dessa forma, um motor de jogo deve envolver diversos módulos, fornecendo ao desenvolvedor todos recursos necessários para criar seu jogo, tais como:

- Reprodução de arquivos multimídia, como vídeos e sons;
- Criação/Edição de interfaces, como telas de save/load, de créditos, mapas, etc;
- Criação/Edição de modelos 2D/3D, animação, e sincronização com os controles do jogo;
- Criação/Edição de cenários;
- Criação/adição de comportamentos a personagens não controláveis (módulo de inteligência artificial);
- Utilização de dispositivos de entrada, tais como joysticks, óculos 3D, etc;

- Utilização de dispositivos de comunicação em rede, para permitir disputa de partidas multijogador;
- Integração com outras plataformas de software, para permitir, por exemplo, sistemas de ranking em Websites.

O TORQUE GAME ENGINE

O Torque Game Engine (TGE), é um motor de baixo custo direcionado para desenvolvimento de jogos de pequeno e médio porte e para entidades educacionais.

O TGE apresenta uma comunidade ativa e relativamente grande, e oferece a maioria dos recursos necessários para desenvolvimento de jogos para projetos educacionais.

Pontos positivos

- Baixo custo
- Acesso a código-fonte
- Multiplataforma (Windows, MAC OS, Linux)
- Comunidade ativa e relativamente grande
- Oferece a maioria dos recursos necessários para desenvolvimento de jogos em diversos estilos

Pontos negativos

- Problemas de desempenho
- BUGs não corrigidos
- Não permite criação de jogos Open-Source

PREPARAÇÃO DO AMBIENTE

Para desenvolver jogos no Torque Game Engine é necessário fazer alterações na SDK da Engine e em seus scripts. Para o primeiro, é necessário utilizar algum compilador C++. Para o segundo, algum editor de texto simples.

A implementação de recursos mais complexos normalmente exige alterações na Engine em si (no SDK), e não apenas em seus scripts. Alterações mais simples e configuração de parâmetros podem ser feitas apenas nos scripts.

Durante o desenvolvimento do projeto utilizamos o Microsoft Visual C++ 6.0 (sem services packs) para alterar a engine. Para edição de scripts, utilizamos uma IDE específica para o Torque Script, o TorqueDev Codeweaver 1.2, que oferece recursos interessantes, como Code Completion e busca no projeto. Ambos os programas encontram-se no SVN, no caminho Triade\Programas\.

AMBIENTE WINDOWS

UTILIZANDO O MICROSOFT VISUAL C++ 6.0 PARA COMPILAR A ENGINE

A Engine Torque já vem com um workspace para Visual C++ 6.0, o que torna simples o processo de compilação da Engine: basta abrir, realizar as alterações e mandar compilar. Um workspace é um conjunto de projetos que se relacionam. No caso da Torque, o workspace trás além do projeto da Engine em si, projetos de bibliotecas auxiliares que são utilizadas pela engine.

Atenção: Antes de compilar o motor no Visual C++ 6.0, deve-se executar o arquivo "VC6 .cc compiling.reg", presente na pasta \$TORQUE/

TGE_1_5_2/vc6. Tal arquivo registrará apropriadamente as extensões “.cc” para serem consideradas durante o processo de compilação.

AMBIENTE LINUX

A compilação/execução do Torque em ambiente Linux é mais complexa, pois requer a instalação de diversas bibliotecas e utilitários. Foi utilizada a distribuição Debian 4.0, (Kernell 2.6.1-6-668 / KDE 3.5.5) para testes, tendo em vista que é a distribuição utilizada nas escolas públicas no estado da Bahia.

INSTRUÇÕES PARA COMPILAÇÃO EM LINUX

Antes de realizar a compilação propriamente dita, certifique-se que:

O compilador GCC está instalado e no PATH: Abra um terminal, digite GCC e tecla ENTER. Se você ver uma mensagem como “comand not found” significa que o GCC não está adequadamente instalado.

Instalação do GGC/G++: No DEBIAN a instalação de pacotes pode ser feita utilizando-se o utilitário apt-get. Use “apt-get update” para atualizar a lista de pacotes, depois “apt-get install gcc” para instalar o GCC. É necessário estar logado como root e ter uma conexão de Internet ativa. Após isso tecla gcc: a mensagem “gcc: no input files” deve ser exibida, indicando que o GCC está instalado.

Após instalar o GCC, use “apt-get install g++” para instalar o g++.

A biblioteca freetype está instalada: Acesse <http://www.freetype.org/>, baixe e descompacte a biblioteca, e siga as instruções de instalação em \$DIRETORIO_DESCOMPACTACAO\$/freetype-X.X.X/docs/install.unix.

O NASM está instalado: Digite nasm no console. Caso a mensagem “comando nao encontrado” seja exibida, você precisa instalá-lo. Para tal, você precisar adicionar um servidor que contenha o NASM na lista de servidores de pacote do DEBIAN. Edite o arquivo /etc/apt/sources.list e adicione a seguinte linha ao fim do arquivo:

```
deb http://ftp.br.debian.org/debian etch main
```

Após isso execute:

```
apt-get update
apt-get install nasm
```

Digite então nasm no console. Se a mensagem “nasm: error: no input file specified” for exibida, o nasm foi instalado.

A biblioteca x11 está instalada:

Use:

```
apt-get install libx11-dev
apt-get install libxft-dev
```

para instalar as dependências necessárias dessa biblioteca.

A biblioteca SDL está instalada:

Use:

```
apt-get install libsdl1.2-dev
```

A biblioteca OpenAL está instalada:

Use

```
apt-get install libopenal-dev
```

ATENÇÃO:

Caso algum dos pacotes listados nos passos anteriores não esteja disponível, você pode procura-lo em <http://www.debian.org/distrib/packages> para certificar-se que não houve mudança em seu nome.

O comando apt-get é da distribuição Debian. Da mesma forma, os nomes de bibliotecas utilizados como parâmetro deste comando, são convenções adotadas pelo servidor de pacotes do Debian. Caso você deseje compilar o torque em outra distribuição linux, verifique se o ambiente fornece algum comando de instalação de pacotes como o apt-get. Ambientes linux frequentemente trabalham com pacotes .RPM, formato no qual também é possível encontrar todas as bibliotecas aqui mencionadas.

Tendo todas as bibliotecas necessárias, os passos para compilação do motor são os seguintes:

Acesse a pasta do torque e execute make:

```
cd ~/projects/torque
make -f mk/configure.mk OS=LINUX COMPILER=GCC4
BUILD=DEBUG
```

Esse procedimento irá demorar um pouco. Diversos warnings serão exibidos, o que é normal.

ATENÇÃO:

Substitua GCC3 pela versão do seu GCC. No momento da escrita deste documento, a ultima versão do gcc era o GCC4.

Executando o jogo em ambiente Linux.

1. Baixar Torque Game Engine Linux no site da Garagegames.
2. Logar como superusuário.
3. Descompactar o Torque em alguma pasta.
4. Acessar essa pasta através do terminal e executar o jogo:


```
cd /$PASTA_TORQUE/example
./torqueDemo.bin
```

ATENÇÃO:

Acessar o jogo através do ambiente gráfico, clicando no ícone torque-Demo.bin pode causar um erro inicialização (main.cs não encontrado).

Desempenho do Torque em ambiente Linux.

Para obter um desempenho aceitável, é necessário instalar os drivers da placa de video em ambiente Linux. O site a seguir, apresenta informações sobre a instalação de drivers NVidia na distribuição DEBIAN.

<http://wiki.debian.org/NvidiaGraphicsDrivers>

Comandos úteis

“uname -r” para verificar a versão do seu kernell.

“more /etc/debian_version” para verificar a versão da sua distribuição debian.

“kde-config --version” para verificar a versão do seu KDE.

Solução de problemas

apt-get : durante a realização desses procedimentos, o apt-get frequentemente acusou um “Mergelist error” durante a atualização de pacotes. Uma rápida pesquisa na internet revela que esse erro pode ser facilmente corrigido adicionando-se a linha “APT::Cache-Limit 16777216123;” ao arquivo /etc/apt/apt.conf.d/00trustedrom.

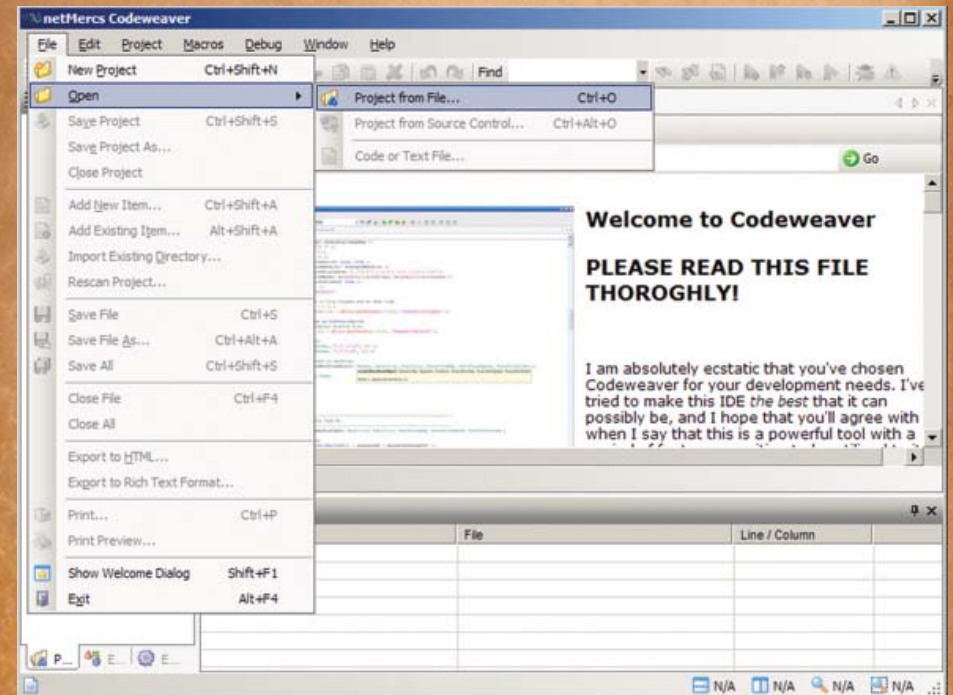
Instalação do Driver NVIDIA: Conforme citado no tópico “Choose a driver version” do site <http://wiki.debian.org/NvidiaGraphicsDrivers>, é importante determinar corretamente qual pacote de drivers deverá ser baixado, de acordo com o dispositivo de vídeo existente. Use o comando “dmesg” para determinar qual versão utilizar. Durante a escrita deste documento foi instalado o Driver para a placa GeForce 4 MX 440, que utilizou os pacotes “-legacy-96.xx”.

Conflito entre pacotes: Utilize `dpkg -r ${NOME_DO_PACOTE}` para remover pacotes que estejam em conflito com novos pacotes instalados.

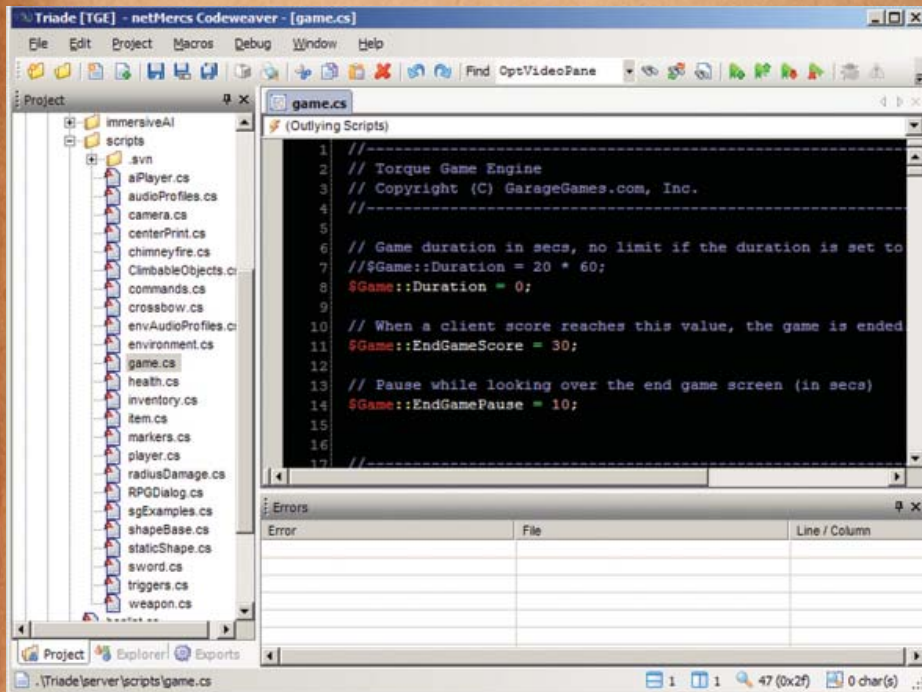
UTILIZANDO O TORQUEDEV / CODEWEAVER 1.2 PARA EDIÇÃO DE SCRIPTS

A edição de scripts pode ser feita

A edição de scripts se resume a abertura do projeto e alteração dos scripts necessários. Não é necessário compilar os scripts, visto que isto é feito quando o jogo é inicializado.



O projeto da Torque encontra-se no caminho \$TORQUE\TGE_1_5_2\example



A ENGINE

PACKAGES

collision

Apresenta as classes que implementam o controle de colisão entre os modelos presentes no jogo.

console

Possui as classes que implementam as funções de console (funções de script) e avaliação/compilação dos scripts.

core

Possui as classes que implementam as funções do núcleo do motor, tais como funções de acesso a arquivos, compactação de dados, transferência, em rede, entre outras.

dgl

Possui as classes que implementam as funções de desenho e manipulação de imagens e fontes.

editor

Possui as classes que implementam as interfaces de edição do motor (editor de mundo, editor de terreno, editor de interface).

game

Possui as classes que implementam os objetos de jogo, tais como physical zones, shapes, players, vehicles, entre outros.

gui

Possui as classes que implementam os elementos de interface do jogo (botões, checkbox, selects, players de vídeo, componentes de texto, entre outros).

interior

Possui as classes que implementam a renderização de arquivos de interior (.DIF).

math

Possui as classes que implementam funções matemáticas utilizadas pelo motor.

platform / platformWin32

Possui as classes que implementam as particularidades ligadas a plataforma onde o motor está rodando (Linux, Windows, Mac OS).

platform / platformWin32

Possui as classes que implementam as particularidades ligadas a plataforma onde o motor está rodando (Linux, Windows, Mac OS).

sceneGraph

Possui as classes que implementam as rotinas de renderização/culling dos elementos do jogo.

sim

Possui as classes que implementam as rotinas de transmissão de objetos através da rede.

terrain

Possui as classes que implementam as rotinas de ligadas a criação/renderização de terrenos (relevo).

audio

Possui as classes que implementam o funcionamento de áudio dentro do jogo.

utility

Classes úteis utilizadas pelo motor.

lightingSystem

Possui as classes que implementam a iluminação do jogo (pontos de luz, cálculo de sombras, pré-lighting).

PRINCIPAIS CLASSES**game\player**

Superclasse que apresenta as principais funções de um jogador no Torque (NPC e jogador humano). Dentre suas principais funções pode-se destacar o controle e definição de animações, realizada pela função `pickActionAnimation()`. Tal função foi bastante alterada durante o desenvolvimento do Tríade, para permitir ao personagem assumir diferentes posições de combate.

game\aiPlayer

Classe que herdeira de `player` e que apresenta as funções básicas de um jogador NPC, como funções de movimentação e de posicionamento.

FUNÇÕES ÚTEIS

O Anexo I possui a listagem de diversas funções que podem ser utilizadas no processo de criação e edição de Scripts.

RESOURCES UTILIZADOS

Resources são artigos disponibilizados na comunidade Torque (www.garagegames.com/index.php?sec=mg&mod=resource&page=home) que demonstram como implementar certas funcionalidades na Engine.

Abaixo estão listados os principais resources utilizados no desenvolvimento do game Triáde.

RESOURCE	LINK
<i>RPGDialog</i>	http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=3531
<p>DESCRIÇÃO</p> <p>Resource que implementa um sistema de diálogos. Ao aproximar-se de um NPC e pressionar uma determinada tecla surge uma tela de diálogos, que inclui um texto com a fala do NPC e opções de perguntas/respostas. Cada NPC tem um arquivo com os textos e as opções de resposta, estruturada de uma maneira própria.</p>	

RESOURCE	LINK
<i>immersiveAI</i>	http://www.garagegames.com/mg/forums/result.thread.php?qt=58585

DESCRIÇÃO

Resource que implementa um sistema de inteligência artificial baseado em Objetivos/Soluções. Foi bastante alterado para trazer um comportamento mais real aos NPC'S, principalmente no que diz respeito ao sistema de combate. Utiliza A* para localização de rotas, gerando um gridmap no terreno. Possui a falha de não realizar localização de rotas em interiores, dificuldade que até o presente momento não foi superada.

RESOURCE	LINK
<i>Climable Objects</i>	http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=11502
<p>DESCRIÇÃO</p> <p>Resource que implementa um sistema de escalagem. O personagem, ao aproximar-se de um objeto escalável, automaticamente assume a posição de escalagem e passa a movimentar-se na vertical.</p>	

RESOURCE	LINK
<i>Melee Combat</i>	http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=5377
<p>DESCRIÇÃO</p> <p>Resource que implementa um sistema de combate com armas de mão (espadas, machados, lanças, etc).</p>	

<p>RESOURCE</p> <p>Inventory System</p>	<p>LINK</p> <p>http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=1955</p>
<p>DESCRIÇÃO</p> <p>Resource que implementa um sistema de inventário.</p>	

DICAS

Utilizando o formato Theora (.OGG - codec de vídeo)

O Torque possui algumas limitações quanto a utilização de videos no formato Theora.

Os principais problemas enfrentados utilizando-se videos Theora no Torque são:

- **Qualidade** : Falta de correspondencia de cores e distorção das bordas do filme.
- **Velocidade** : Travamentos durante a reprodução do video.

O problema de qualidade é solucionado utilizando-se a diretriz `-optimize` durante a conversão avi / ogg. Já a velocidade parece estar ligada a uma limitação do Torque em utilizar filmes grandes (maiores que 30mbs). Uma solução para redução do tamanho é reduzir a resolução do filme durante a exportação. Também, pode-se trabalhar reduzindo a qualidade, ou dividindo um filme grande em dois ou mais arquivos.

Lembrando:

- optimize: Corrige problemas de qualidade na conversao AVI / OGG.
- v : Permite a definição da qualidade do video. Quanto maior a qualidade maior o tamanho. Varia de 1 a 10 e o padrão é 5.
- x, -y: Define largura e altura do filme no formato .OGG.

Uma boa base para iniciar os testes de qualidade/desempenho do filme é:

```
Ffmpeg2theora-0.20.exe -v 8 -optimize -x 720 -y 560 NOME_
DO_VIDEO.AVI
```

Utilizando som (.OGG)

Emissores de som funcionam apenas com arquivos de som MONO.

ANEXO I**Comandos do Torque Script
Retirado de www.garagegames.com**

In this section I will explain each Console command and function used in the TGE CVS HEAD version as of 04/04/02. Examples of each command or function will be given, along with its prototype, and its return value.

Before we begin, let's take a few minutes to define some terms, reserved keywords and operators that will be used within this document.

Document Style

Document text	Document text other than function prototypes and examples will be formatted using a 12pt Arial font
Function prototype	Function prototypes will be BOLD formatted using a 12pt Verdana font
Function example	Function examples will be formatted using a 12pt Courier font

Terms used within the document

string	refers to any single or double quoted collection of characters that can include both numbers and text
boolean	refers to a numeric value of either 1 or 0 (1 being TRUE and 0 being FALSE)
numeric	refers to any intrinsic numeric data type (whole number, boolean, decimal number and or a hexadecimal number)
global	variable refers to a variable that is accessible from within any function or script

local variable	refers to a variable that is only accessible within the function or script it is created
variable	refers to any string that begins with a percent sign (%) denoting a local variable or dollar sign (\$) denoting a global variable
tagged string	tagged strings is used for any string constant that is transmitted across a connection. The entire tagged string is sent only once, when referenced a short tag (numeric value) identifying that string is sent instead of the entire string.

Reserved Key Words

break, case, continue, datablock, default, else, false, function, if, for, new, or, package, return, switch, switch\$, true, and while

OperatorsAssignment operators

= Assigns the value of the second operand to the first operand.

Mathematical Operators:

+	(Addition) Adds 2 numbers
-	(subtraction) Subtracts the value of its argument.
*	(Multiplication) Multiplies 2 numbers.
/	(Division) Divides 2 numbers.
%	(Modulus) Computes the integer remainder of dividing 2 numbers.
+=	Adds 2 numbers and assigns the result to the first.
-=	Subtracts 2 numbers and assigns the result to the first.
*=	Multiplies 2 numbers and assigns the result to the first.

- /= Divides 2 numbers and assigns the result to the first.
- %= Computes the modulus of 2 numbers and assigns the result to the first.
- ++ (Increment) Adds one to a variable representing a number (returning either the new or old value of the variable)
- (Decrement) Subtracts one from a variable representing a number (returning either the new or old value of the variable)

Bitwise Operators:

- ~ (Bitwise NOT) Flips the bits of its operand.
- | (Bitwise OR) Returns a one in a bit if bits of either operand is one.
- & (Bitwise AND) Returns a one in each bit position if bits of both operands are ones.
- ^ (Bitwise XOR) Returns a one in a bit position if bits of one but not both operands are one.
- << (Left shift) Shifts its first operand in binary representation the number of bits to the left specified in the second operand, shifting in zeros from the right.
- >> (Sign-propagating right shift) Shifts the first operand in binary representation the number of bits to the right specified in the second operand, discarding bits shifted off.
- |= Performs a bitwise OR and assigns the result to the first operand.
- &= Performs a bitwise AND and assigns the result to the first operand.
- ^= Performs a bitwise XOR and assigns the result to

the first operand.

- <<= Performs a left shift and assigns the result to the first operand.
- >>= Performs a sign-propagating right shift and assigns the result to the first operand.

String operators::

- @ Concatenates one or more values together to form a new value
- NL Concatenates one value together with a new line to form a new value
- TAB Concatenates one value together with a tab to form a new value
- SPC Concatenates one value together with a space to form a new value

Logical Operators:

- ! evaluates the opposite of the value specified
- && requires both values to be true for the result to be true.
- || requires only one value to be true for the result to be true.

Relational Operators:

- == value1 and value2 are equal
- != value1 and value2 are not equal
- < value1 is less than value2
- > value1 is greater than value2
- <= value1 is less than or equal to value2
- >= value1 is greater than or equal to value2

String comparison Operators:

- \$= string1 is equal to string2

!\$= string1 is not equal to string2

The following console commands and functions are listed in order as viewed in the output from ConsoleDoc¹

OpenALInitDriver()

- Used to initialize the sound driver
- Returns a numeric value

```
if(!OpenALInitDriver())
```

```
error(" Failed to initialize driver.");
```

OpenALShutdownDriver()

- Used to Disable the sound driver
- No return value

```
function OpenALShutdown()
```

```
{
    OpenALShutdownDriver();
}
```

OpenALRegisterExtensions()

- This function does not have a function body or any other associated engine or script code

¹ ConsoleDoc was created by Emanuel Zephir (<http://www.garagegames.com/index.php?sec=mg&mod=resource&page=view&qid=2034>)

-No return value

```
function OpenALRegister()
```

```
{
    OpenALRegisterExtensions();
}
```

alGetString(ALenum)

- Used to return the string representing the specified ALenum passed
- Returns a string value

```
function OptAudioUpdate()
```

```
{
    // set the driver text
    %text = "Vendor: " @ alGetString("AL_VENDOR") @
    "\nVersion: " @ alGetString("AL_VERSION") @
    "\nRenderer: " @ alGetString("AL_RENDERER") @
    "\nExtensions: " @ alGetString("AL_EXTENSIONS");
    OptAudioInfo.setText(%text);
}
```

alxCreateSource(profile, {x,y,z} | description, filename, {x,y,z})

- Creates and validates the Audio source (sound file) and readies it to be played
- Returns a numeric

```
function OptAudioUpdateMasterVolume(%volume)
```

```
{
```

```

if (%volume == $pref::Audio::masterVolume)
    return;
alxListenerf(AL_GAIN_LINEAR, %volume);
$pref::Audio::masterVolume = %volume;
if (!alxIsPlaying($AudioTestHandle))
{
    $AudioTestHandle = alxCreateSource("AudioChannel0", expandFilename("~/data/sound/testing.wav"));
    alxPlay($AudioTestHandle);
}
}

```

alxSourcef(handle, AEnum, value)

- Used to set the value of AEnum for the specified handle
- No return type

```
alxSourcef($ClientChatHandle[%sender], "AL_PITCH", %pitch);
```

alxSource3f(handle, AEnum, 'xyz'|x,y,z)

- Used to set the value of AEnum for the specified handle
- No return type

```
alxSourcef($ClientChatHandle[%sender], "AL_PITCH", %x, %y, %z);
```

alxSourcei(handle, AEnum, value)

- Used to set the value of AEnum for the specified handle
- No return type

```
alxSourcei($ClientChatHandle[%sender], "AL_PITCH", %pitch);
```

alxGetSourcef(handle, AEnum)

- Used to get the value of AEnum for the specified Audio Handle
- Returns a numeric value

```
%pitch = alxGetSourcef(($ClientChatHandle[%sender], "AL_PITCH");
```

alxGetSource3f(handle, AEnum)

- Used to get the value of AEnum for the specified Audio Handle
- Returns a numeric value

```
%pitch = alxGetSourcef(($ClientChatHandle[%sender], "AL_PITCH");
```

alxGetSourcei(handle, AEnum)

- Used to get the value of AEnum for the specified Audio Handle
- Returns a numeric value

```
%pitch = alxGetSourcef(($ClientChatHandle[%sender], "AL_PITCH");
```


alxPlay(handle) | alxPlay(profile) | alxPlay(profile, x,y,z)

-Used to begin audio playback on specified (Handle | profile | profile "x y z")

-Returns a numeric value

```
function OptAudioUpdateChannelVolume(%channel, %volume)
{
    if (%channel < 1 || %channel > 8)
        return;
    if (%volume == $pref::Audio::channelVolume[%channel])
        return;
    alxSetChannelVolume(%channel, %volume);
    $pref::Audio::channelVolume[%channel] = %volume;
    if (!alxIsPlaying($AudioTestHandle))
    {
        $AudioTestHandle = alxCreateSource("AudioChannel"@%channel, expandFilename("~/data/sound/testing.wav"));
        alxPlay($AudioTestHandle);
    }
}
```

alxStop(handle)

-Used to stop the audio playback on specified Handle

-No return value

```
if ( $ClientChatHandle[%sender] != 0 )
    alxStop( $ClientChatHandle[%sender] );
```

alxStopAll()

-Used to stop ALL Audio playback on all registered sound channels

-No return value

```
function clientCmdMissionEnd(%seq,%player)
{
    // Recieved when the current mission is ended.
    alxStopAll();
    ...
    ...
}
```

alxIsPlaying(handle)

-Used to determine if a Handle is currently performing audio playback

-Returns a numeric value

```
if (!alxIsPlaying($AudioTestHandle))
{
    $AudioTestHandle = alxCreateSource("AudioChannel0", expandFilename("~/data/sound/testing.wav"));
    alxPlay($AudioTestHandle);
}
```

alxListener(ALenum,value)

-Used to set the ALenum extension supporting linear gain to value

-No return value

```
function OpenALInit()
{
    OpenALShutdownDriver();

    echo("");
    echo("OpenAL Driver Init.");

    echo ($pref::Audio::driver);
    if($pref::Audio::driver $= "OpenAL")
        if(!OpenALInitDriver())
            error(" Failed to initialize driver.");

    echo(" Vendor: " @ alGetString("AL_VENDOR"));
    echo(" Version: " @ alGetString("AL_VERSION"));
    echo(" Renderer: " @ alGetString("AL_RENDERER"));
    echo(" Extensions: " @ alGetString("AL_EXTENSIONS"));

    alxListenerf( AL_GAIN_LINEAR,
    $pref::Audio::masterVolume );
    for (%channel=1; %channel <= 8; %channel++)
        alxSetChannelVolume(%channel, $pref::Audio::channel
        Volume[%channel]);

    echo("");
}
```

alListener3f(ALenum,|'xyz'|x,y,z)

-Used to set the ALenum extension supporting linear gain to value

-No return value

```
function OpenALInit()
{
    OpenALShutdownDriver();

    echo("");
    echo("OpenAL Driver Init.");

    echo ($pref::Audio::driver);
    if($pref::Audio::driver $= "OpenAL")
        if(!OpenALInitDriver())
            error(" Failed to initialize driver.");

    echo(" Vendor: " @ alGetString("AL_VENDOR"));
    echo(" Version: " @ alGetString("AL_VERSION"));
    echo(" Renderer: " @ alGetString("AL_RENDERER"));
    echo(" Extensions: " @ alGetString("AL_EXTENSIONS"));
    alxListener3f( AL_GAIN_LINEAR,
    $pref::Audio::masterVolume );

    for (%channel=1; %channel <= 8; %channel++)
        alxSetChannelVolume(%channel, $pref::Audio::channel
        Volume[%channel]);
```

```
echo("");
}
```

alxGetListenerf(ALenum)

-Used to get the value of the ALenum extension supporting linear gain

-Returns a numeric value

```
%LinearGain = alxGetListenerf(AL_GAIN_LINEAR);
```

alxGetListener3f(ALenum)

-Used to get the value of the ALenum extension supporting linear gain

-Returns a numeric value

```
%LinearGain = alxGetListenerf(AL_GAIN_LINEAR);
```

alxGetListeneri(ALenum)

-Used to get the value of the ALenum extension supporting linear gain

-Returns a numeric value

```
%LinearGain = alxGetListenerf(AL_GAIN_LINEAR);
```

alxGetChannelVolume(channel_id)

-Used to get the current volume of specified channel

-Returns a numeric value

```
$pref::Audio::channelVolume[%channel] =
```

```
alxGetChannelVolume(%channel);
```

alxSetChannelVolume(channel_id)

-Used to set the volume of the specified channel.

-Returns a numeric value

```
alxSetChannelVolume(%channel, %volume);
```

dumpConsoleClasses()

-Used to dump all registered console classes to the console in native C++ syntax

-No return type

```
dumpConsoleClasses();
```

expandFilename(filename)

-Used to determine the actual path of the specified filename

-Returns a string value

```
$AudioTestHandle = alxCreateSource("AudioChannel0",
expandFilename("~/data/sound/testing.wav"));
```

strcmp(one,two)

-Used to compare two strings.

-Returns a numeric value (1 being TRUE and 0 being FALSE)

```
if ( strcmp( %name, %rawName ) == 0 )
    return false;
```

stricmp(one,two)

-Used to compare two strings with case insensitivity.

-Returns a numeric value

< 0 "one" is less than "two".

0 "one" is equal to "two".

> 0 "one" is greater than "two"

```
if ( strcmp( %name, %rawName ) == 0 )
    return false;
```

strlen(str)

-Used to get the length of a string

-Returns a numeric value.

```
%StringLength = strlen(%player.ShapeName);
```

strstr(string,substr)

-used to locate first occurrence of a "substr" within a "string"

-Returns a string value

```
%pos = strstr( %action, "pov");
```

strpos(stringhay,stringneedle[,intoffset])

-Used to locate the first occurrence of "stringneedle" in "stringhay"

-Returns a numeric value

```
if (%hasNextArg && strpos(%nextArg, "-") == -1)
{
    $showShapeList = $showShapeList @ " " @ %nextArg;
    $argUsed[%i+1]++;
    %i++;
}
```

ltrim(str)

-Used to strip the leading white space from "str".

-Returns a string

```
%trimmed = ltrim(%value);
```

rtrim(str)

-Used to strip the trailing white space from "str"

-Returns a string.

```
%trimmed = rtrim(%value);
```

trim(str)

-Used to strip the leading and trailing white space from "str".

-Returns a string

```
%trimmed = trim(%value);
```

stripChars(string,chars)

-Used to remove "chars" from "string"

-Returns a string value

```
%striped = stripChars(%value, "~");
```

strlwr(string)

-Used to convert "str" to its lower case equivalent.

-Returns a string

```
%var = strlwr(%value);
```

strupr(string)

-Used to convert "str" to its upper case equivalent.

-Returns a string

```
%var = strupr(%value);
```

strchr(string,char)

-Used to locate the first occurrence of "char" in "string"

-Returns a string

```
%var = strchr(%value, "~");
```

strreplace(string,from,to)

-Used to replace all occurrences of "from" and replaces with "to".

-Returns a string

```
%var = strreplace(%value, "~", "-");
```

getSubStr(string,start,numChars)

-Used to get the sub string of "string", starting at "start", and continuing to either the end of the string, or "numChars" characters, whichever comes first.

-Returns a string

```
function serverCmdTeamMessageSent(%client, %text)
{
    if(strlen(%text) >= $Pref::Server::MaxChatLen)
        %text = getSubStr(%text, 0,
$Pref::Server::MaxChatLen);
    chatMessageTeam(%client, %client.team, '\c3%1:
%2', %client.name, %text);
}
```

getWord(text,index)

-Used to get the word at specified index within "text".

-Returns a string

```
function TerrainEditor::offsetBrush(%this, %x, %y)
{
    %curPos = %this.getBrushPos();
    %this.setBrushPos(getWord(%curPos, 0) + %x,
getWord(%curPos, 1) + %y);
}
```

getWords(text,index [,entindex])

-Used to get the word(s) specified at index [,entindex] within

“text”.

-Returns a string

```
%pos = getWords(%obj.getTransform(), 0, 2);
```

setWord(text,index,replace)

-Used to replace the word at index with “replace” within

“text”.

-Returns a string

```
function WorldEditor::dropCameraToSelection(%this)
```

```
{
```

```
    if(%this.getSelectionSize() == 0)
```

```
        return;
```

```
    %pos = %this.getSelectionCentroid();
```

```
    %cam = LocalClientConnection.camera.
```

```
getTransform();
```

```
    // set the pnt
```

```
    %cam = setWord(%cam, 0, getWord(%pos, 0));
```

```
    %cam = setWord(%cam, 1, getWord(%pos, 1));
```

```
    %cam = setWord(%cam, 2, getWord(%pos, 2));
```

```
    LocalClientConnection.camera.setTransform(%cam);
```

```
}
```

removeWord(text,index)

-Used to remove the word at index from “text”.

-Returns a string

```
%cam = removeWord(%cam, 0, getWord(%pos, 0));
```

getWordCount(text)

-Used to return the number of words within “text”.

-Returns a numeric value

```
if ( %pos != -1 )
```

```
{
```

```
    %wordCount = getWordCount( %action );
```

```
    %mods = %wordCount > 1 ? getWords( %action, 0, %wordCount - 2 ) @ “ “ : “”;
```

```
    %object = getWord( %action, %wordCount - 1 );
```

```
    switch$ ( %object )
```

```
{
```

```
    case “upov”: %object = “POV1 up”;
```

```
    case “dpov”: %object = “POV1 down”;
```

```
    case “lpov”: %object = “POV1 left”;
```

```
    case “rpov”: %object = “POV1 right”;
```

```
    case “upov2”: %object = “POV2 up”;
```

```
    case “dpov2”: %object = “POV2 down”;
```

```
    case “lpov2”: %object = “POV2 left”;
```

```
    case “rpov2”: %object = “POV2 right”;
```

```
    default: %object = “??”;
```

```
}
```

```
return( %mods @ %object );
```

```
}
```

```

else
    error( "Unsupported Joystick input object passed to
getDisplayMapName!" );
}

```

getField(text,index)

-Used to get the field at "index" within "text"
-Returns a string

```

function SM_StartMission()
{
    %id = SM_missionList.getSelectedId();
    %mission = getField(SM_missionList.
getRowTextById(%id), 1);

    if ($pref::HostMultiPlayer)
        %serverType = "MultiPlayer";
    else
        %serverType = "SinglePlayer";

    createServer(%serverType, %mission);
    localConnect($pref::Player::Name);
}

```

getFields(text,index [,entindex])

-Used to return the field(s) specified at index [,entindex]
within "text"
-Returns a string

```

%fields = getFields((SM_missionList.getRowTextById(%id),
1, 3 ));

```

setField(text,index,replace)

-Used to replace field at "index" with "replace".
-Returns a string

```

function PlayerListGui::updateScore(%this,%clientId,%score)
{
    %text = PlayerListGuiList.
getRowTextById(%clientId);
    %text = setField(%text,1,%score);
    PlayerListGuiList.setRowById(%clientId, %text);
    PlayerListGuiList.sortNumerical(1);
}

```

removeField(text,index)

-Used to remove a field at "index" from "text".
-Returns a string

```

function removeFromServerGuidList( %guid )
{
    %count = getFieldCount( $Server::GuidList );
    for ( %i = 0; %i < %count; %i++ )
    {
        if ( getField( $Server::GuidList, %i ) ==
%guid )

```

```

    {
        $Server::GuidList = removeField(
$Server::GuidList, %i );
        return;
    }
}

```

// Huh, didn't find it.

```
}

```

getFieldCount(text)

- Used to return the number of fields within "text".
- Returns a numeric value

```
%count = getFieldCount($Server::GuidList);

```

getRecord(text,index)

- Used to return the record at "index" within "text".
- Returns a string

```
function Texture::saveMaterial()

```

```
{

```

```
    %id = $selectedMaterial;
    if (%id == -1)
        return;

```

```
Texture::SaveOperation();

```

```
%data = Texture_Material.getRowTextById(%id);
%newData = getRecord(%data,0);

```

```
%rowCount = Texture_Operation.rowCount();
for (%row=0; %row<%rowCount; %row++)

```

```
    %newdata = %newdata @ "\n" @ Texture_
Operation.getRowText(%row);

```

```
Texture_Material.setRowById(%id, %newdata);
Texture::save();

```

```
}

```

getRecords(text,index[,endIndex])

- Used to return the record(s) at "index [endIndex]" within "text".
- Returns a string.

```
%data = Texture_Material.getRowTextById(%id);
%records = getRecords(%data,1,3);

```

setRecord(text,index,replace)

- Used to replace record at "index" with "replace" within "text".
- Returns a string.

```
%data = Texture_Material.getRowTextById(%id);
%replaced = setRecord(%data,3,"foo");

```

removeRecord(text,index)

- Used to remove record at "index" within "text".
- Returns a string

```
%data = Texture_Material.getRowTextById(%id);
%replaced = removeRecord(%data,3);
```

getRecordCount(text)

- Used to return the number of records within "text".
- Returns numeric value

```
%data = Texture_material.getRowTextById(%id);
Texture_operation.clear();
%recordCount = getRecordCount(%data);
```

firstWord(text)

- Used to return the first word within "text".
- Returns a string.

```
// a target in range was found so select it
if (%scanTarg)
{
    %targetObject = firstWord(%scanTarg);
    %client.setSelectedObj(%targetObject);
}
```

restWords(text)

- Used to return the remaining words in "text".
- Returns a string

```
function Heightfield::showTab(%id)
{
    Heightfield::hideTab();
    %data = restWords(Heightfield_operation.
getRowTextById(%id));
    %tab = getField(%data,1);
    echo("Tab data: " @ %data @ " tab: " @ %tab);
    %tab.setVisible(true);
}
```

detag(textTagString)

- Used to detag a string, that is to return the string value for the tagged string
- Returns a string

```
function isNameUnique(%name)
{
    %count = ClientGroup.getCount();
    for ( %i = 0; %i < %count; %i++ )
    {
        %test = ClientGroup.getObject( %i );
        %rawName = stripChars( detag(
getTaggedString( %test.name ) ), "\cp\co\c6\
c7\c8\c9" );
        if ( strcmp( %name, %rawName ) == 0
)
            return false;
    }
}
```

```
return true;
```

```
}
```

getTag(textTagString)

- Used to return the tag for the specified "textTagString)
- Returns a string

```
%tag = getTag(%variable);
```

echo(text [, ...])

- Used to print "text" to the console
- No return type

```
echo( "This will be printed to the console");
```

warn(text [, ...])

- Used to print text to the console
- No return type

```
warn("Warning, this will be printed to the console);
```

error(text [, ...])

- Used to print text to the console
- No return type

```
error("Error Will Rogers, Error!");
```

expandEscape(text)

- Used to escape characters in "text" (\r becomes \\r)
- Returns a string

```
%expanded = expandEscape(%variable);
```

collapseEscape(text)

- Used to remove escaped characters in "text" (\\r becomes 0xd)
- Returns a string

```
%collapsed = collapseEscape(%variable);
```

quit()

- Used to quit the game
- No return type

```
quit();
```

call(funcName [,args ...])

- Used to execute the "funcName" with supplied "args"
- Returns a string

```
function clientCmdServerMessage(%msgType, %msgString,
%a1, %a2, %a3, %a4, %a5, %a6, %a7, %a8, %a9, %a10)
{
    %tag = getWord(%msgType, 0);
    for(%i = 0; (%func = $MSGCB["", %i]) != "");
    %i++)
        call(%func, %msgType, %msgString,
%a1, %a2, %a3, %a4, %a5, %a6, %a7, %a8, %a9, %a10);

    if(%tag != "")
        for(%i = 0; (%func = $MSGCB[%tag, %i]) !=
""; %i++)
            call(%func, %msgType, %msgString,
%a1, %a2, %a3, %a4, %a5, %a6, %a7, %a8, %a9, %a10);
}
```

compile(fileName)

- used to compile "fileName"

-Returns a numeric

```
if( compile("/fps/client/scripts/script.cs") )
    echo("Compile success");
```

exec(fileName [, nocalls [,journalScript]])

-Used to compile (if needed) and execute the specified "filename" or "journalScript"
-Returns a numeric value

```
if( exec("/fps/client/scripts/script.cs"))
    echo("exec success");
```

export(searchString [, fileName [,append]])

-Used to "export" (save) the values of "searchString" to "fileName"
-No return value

```
echo("Exporting server prefs");
export("$Pref::Server:*", "./server/prefs.cs", False);
```

deleteVariables(wildCard)

-Used to delete global variables specified by "wildcard"
-no return type

```
deleteVariables(*);
```

trace(bool)

-Used to turn trace on or off
-No return type
trace(1);

debug()

-Used to start debug mode

-No return type

```
debug();
```

findFirstFile (string pattern)

-Used to return the first file in the directory system matching the given pattern
-Returns a string

```
findFirstFile("/fps", "*.cs");
```

findNextFile (string pattern)

-Used to return the next file matching a search begun in findFirstFile
-Returns a string

```
findNextFile("/fps", "*.cs");
```

getFileCount (string pattern)

-Used to return the number of files in the directory tree that match the given pattern
-Returns a numeric

```
getFileCount("/fps/client/scripts", "*.cs");
```

getFileCRC(filename)

-Used to get the CRC of specified file name
-Returns a numeric

```
getFileCRC("/fps/client/scripts/script/cs");
```

isFile(filename)

-Used to determine if file exists
-Returns a numeric (0 = FALSE, 1 = TRUE)

```
isFile("/fps/client/scripts/script.cs");
```

isWritableFileName(filename)

-Used to determine if file specified by filename can be written to

-Returns a numeric (0 = FALSE, 1 = TRUE)

```
isWritableFileName("/fps/client/scripts/script.cs");
```

fileExt(filename)

-Used to return the extension of file specified by filename

-Returns a string

```
fileExt("script.cs"); // will return ".cs" as a string
```

fileBase(filename)

-Used to return the base name of the file specified by filename

-Returns a string

```
fileBase("/fps/client/scripts/script.cs"); // will return "fps/client/scripts/script" as a string
```

fileName(filename)

-Used to return the name of the file specified by filename

-Returns a string

```
fileName("scripts.cs"); // will return "scripts" as a string
```

filePath(filename)

-Used to return the path of the file specified by filename

-Returns a string

```
filePath("/fps/client/scripts/script.cs"); // Will return "fps/client/
```

```
scripts/" as a string
```

nextToken(str,token,delim)

-Used to set "token" to next string delimited by delim

-Returns a string

```
nextToken("A,E,I,O,U", vowel, ",");
```

//First pass would be equal to A, the next iteration would set would to E, then I then O and finally U.

setLogMode(mode)

-Used to set the current log mode (0 = Turn logging off, 1 =

Loggin on, 2 = Close current logfile and open a new one)

-No return value

```
SetLogMode(1);
```

setEchoFileLoads(bool)

-Used to turn the echoing of File Loads to the console on or off (0 = OFF, 1 = ON)

-No return type

```
setEchoFileLoads(1);
```

backtrace()

-Used to enable script back tracing (echoing of script call stack to console)

-No return type

```
backtrace();
```

isPackage(packageName)

-Used to determine if packageName is a registered package

-Returns numeric (0 = FALSE, 1 = TRUE)

```
isPackage(Show);
```

activatePackage(packageName)

- Used to activate usage of package specified by packageName
- No return type

```
activatePackage(Show);
```

deactivatePackage(packageName)

- Used to disable usage of package specified by packageName
- No return type

```
deactivePackage(Show);
```

nameToID(object)

- Used to return the ID number of the specified object
- Returns a numeric

```
nameToId(%player);
```

isObject(object)

- Used to determine if object exists
- Returns a numeric (0 = FALSE, 1 = TRUE)

```
isObject(%player);
```

cancel(eventId)

- Used to cancel the specified event
- No return type

```
// if centerprint already visible, reset text and time.
if ($centerPrintActive) {
    if (centerPrintDlg.removePrint !$= "")
```

```
cancel(centerPrintDlg.removePrint);
}
```

isEventPending(%scheduleID)

- Used to determine if there are any pending events for specified schedule

- Returns a numeric (0 = FALSE, 1 = TRUE)

```
$Game::Schedule = schedule($Game::EndGamePause * 1000,
0, "onCyclePauseEnd");
if( isEventPending($Game::Schedule) )
    echo("got a pending event);
```

schedule(time, command, <arg1...argN>)

- Used to schedule an event
- Returns the newly created scheduleID

```
$Game::Schedule = schedule($Game::EndGamePause * 1000,
0, "onCyclePauseEnd");
```

deleteDataBlocks()

- Used to remove all registered dataBlocks from the game
- No return type

```
deleteDataBlocks();
```

telnetSetParameters(port, consolePass, listenPass)

- Used to setup and accept telnet request with the specified options
- No return type

```
telnetSetParameters(4123, "Garage", "Games");
```

dbgSetParameters (port, pass)

-Used to setup and accept debug connects with the specified options

-No return type

```
dbgSetParameters(1130, "morbid");
```

dnetSetLogging(bool)

-Used to turn on network packet logging to the console

-No return type

```
dnetSetLogging(1);
```

setNPatch(bool, int)

-Used to enable Npatching (quadratic Interpolation) at specified level

-No return value

```
setNPatch(1, 1);
```

toggleNPatch()

-Used to toggle the state of Npatch

-No return type

```
toggleNPatch();
```

increaseNPatch()

-Used to increase the level of Npatch by one

-No return type

```
increaseNPatch();
```

decreaseNPatch()

-Used to decrease the level of Npatch by one

-No return type

```
decreaseNPatch();
```

setFSAA

See setNPatch()

IncreaseFSAA

See increaseNPatch()

decreaseFSAA

See decreaseNPatch()

setOpenGLMipReduction(0 – 5)

-Used to control Shape Texture Detail (Higher number represents increase texture detail)

-No return value

```
setOpenGLMipReduction(2);
```

setOpenGLSkyMipReduction(0 – 5)

-Used to Control texture detail for the skybox and clouds (Higher number represents increase texture detail)

-No return value

```
setOpenGLMipReduction(2);
```

setOpenGLInteriorMipReduction(0 – 5)

-Used to Control texture detail for buildings

-No return value

```
setOpenGLInteriorMipReduction(2);
```

setOpenGLTextureCompressionHint (GL_DONT_CARE|GL_FASTEST|GL_NICEST)

- Used to Control how textures are compressed
- No return type

```
setOpenGLTextureCompressionHint(GL_NICEST);
```

setOpenGLAnisotropy(0 – 1)

- Used to set the level of Anisotropy (Advanced texture rendering other than trilinear or bilinear)
- No return type

```
setOpenGLAnisotropy(0);
```

clearTextureHolds()

- Used to free/release any textures that are not being used
- Returns a numeric (size of the TextureObject)

```
// Dump anything we're not using
clearTextureHolds();
```

addMaterialMapping(materialName, Sound, Color)

- Used to add Sound and Puff color to specified material
- No return type

```
addMaterialMapping( "sand" , "sound: 0" , "color: 0.46 0.36
0.26 0.4 0.0" );
```

aiConnect(val 0..N)

- Used to Construct and register a new AI connection. No control object is set
- Returns a numeric (newly created object ID)

```
aiConnect(1);
```

aiAddPlayer ('playerName'[, 'AIClassType'])

- Used to add a bot to the game
- Returns the bots objected

```
aiAddPlayer("BotBoy");
```

setPowerAudioProfiles (powerUp, powerDown)

- Used to powerUp/Down audio files for powerups
- No return value

calcExplosionCoverage(NEED FUNCTION PROTOTYPE!!)

- Used to calculate how much exposure the current object has to the explosive force. The object types listed are objects that will block an explosion. If the object is totally blocked, then no damage is applied.
- Returns numeric (0 = no exposure 1 = exposure)

```
%coverage = calcExplosionCoverage(%position,
%targetObject,
$TypeMasks::InteriorObjectType |
$TypeMasks::TerrainObjectType |
$TypeMasks::ForceFieldObjectType |
$TypeMasks::VehicleObjectType);
if (%coverage == 0)
continue;
```

gotoWebPage(address)

- Used to open clients default browser with specified address
- No return type

```
gotoWebPage("http://www.garagegames.com);
```

deactivateDirectInput()

-Used to disable polling of direct input devices (keyboard, mouse, joystick etc)

-No return type

```
deactivateDirectInput();
```

activateDirectInput()

-Used to activate polling of direct input devices (keyboard, mouse, joystick etc)

-No return type

```
activateDirectInput();
```

strToPlayerName(string);

-Used to limit string to 16 characters and strip leading spaces, underscores an certain characters (“,”, “.”, “\” and “”) from string to be used for player name

-Returns a string

```
%name = stripTrailingSpaces( strToPlayerName( %name ) );
```

stripTrailingSpaces(string)

-Used to strip trailing spaces and underscores from string to be used for player name

-Returns string

```
%name = stripTrailingSpaces( strToPlayerName( %name ) );
```

setDefaultFov(defaultFOV)

-Used to set the default FOV (Field of View)

-No return value

```
setDefaultFov( $pref::Player::defaultFov );
```

setZoomSpeed(speed)

-Used to set the zoom speed

-No return type

```
setZoomSpeed( $pref::Player::zoomSpeed );
```

setFov(FOV)

-Used to set the active FOV

-No return type

```
setFov( $Pref::player::CurrentFOV );
```

screenShot(filename)

-Used to take a screen shot and save it to file specified by filename

-No return type

```
screenShot(“MyScreen”);
```

panoramaScreenShot(filename)

-Used to take a Panoramic Screen shot and save it to file specified by filename

-No return type

```
panoramaScreenShot(“WideEyedScreenShot”);
```

purgeResources()

-Used to purge all game resources

-No return type

```
purgeResources();
```

lightScene(<completeCallback>, <forceAlways,forceWritable>)

-Used to light the mission, “completeCallback” will be called

upon completion of mission lighting
-No return type

```
if (lightScene("sceneLightingComplete", ""))
{
    error("Lighting mission....");
    schedule(1, 0, "updateLightingProgress");
    onMissionDownloadPhase3(%missionName);
    $lightingMission = true;
}
```

flushTextureCache()

-used to delete Textures in the cache
-No return type
flushTextureCache();

dumpTextureStats()

-Used to dump information about each texture to the console
The following information is displayed
type, refCount, holding (Yes or no), textureSpace, texFileName
-No return type

```
dumpTextureStats();
```

dumpResourceStats()

-Used to print information regarding loaded resources
The following information is displayed
path, resource_name, lockCount);
-No return type

```
dumpResourceStats();
```

getControlObjectAltitude()

-Used to get the altitude of a player control object
-Returns a numeric

```
%player.getControlObjectAltitude();
```

getControlObjectSpeed()

-Used to get the Speed of a player control object
-Returns a numeric

```
%player.getControlObjectSpeed();
```

containerFindFirst(type, point, x, y, z)

-Appears to find objects of type in a box
-Returns a numeric

containerFindNext()

-Used to return the next object in a box
-Returns a numeric

snapToggle()

-Used to enable snap to grid functionality
-No return type

```
snapToggle();
```

getVersionNumber()

-Used to get the current hard coded engine version number
-Returns a numeric

```
echo("Version Number: " @ getVersionNumber() );
```

getVersionString()

- Used to get the current hard coded engine version number
- Returns a string

```
echo("Version Number: " @ getVersionNumber() );
```

getCompileTimeString()

- Used to return the compile time and date
- Returns a string

```
function aboutDlg::onWake(%this)
{
    %text="<just:center><font:Arial Bold:20>Torque Game
    Engine Test Application\n( v1.1.1 )\n"@
    "<font:Arial:12>"@ getCompileTimeString() @", "@
    getBuildString() @"Build\n\n"@
    "<font:Arial:16>Copyright (c) 2001 <a:www.
    garagegames.com>GarageGames.Com</a>\n"@
    "Portions Copyright (c) 2001 by Sierra Online, Inc.\n\n"@
    "<bitmap:fps/client/ui/gglogo150.png>";
    aboutText.setText(%text);
}
```

getBuildString()

- Used to get the BUILD type (Release or Debug)
- Returns a string

```
function aboutDlg::onWake(%this)
{
    %text="<just:center><font:Arial Bold:20>Torque Game
    Engine Test Application\n( v1.1.1 )\n"@
    "<font:Arial:12>"@ getCompileTimeString() @", "@
    getBuildString() @"Build\n\n"@
    "<font:Arial:16>Copyright (c) 2001 <a:www.
```

```
garagegames.com>GarageGames.Com</a>\n"@
"Portions Copyright (c) 2001 by Sierra Online, Inc.\n\n"@
"<bitmap:fps/client/ui/gglogo150.png>";
    aboutText.setText(%text);
}
```

getSimTime()

- Used to get the Game time
- Returns a numeric

```
function timeMetricsCallback()
{
    return fpsMetricsCallback() @
    " Time -- " @
    " Sim Time: " @ getSimTime() @
    " Mod: " @ getSimTime() % 32;
}
```

getRealTime()

- Used to get the real time (in milliseconds)
- Returns a numeric

```
echo("Time in milliseconds: " @ getRealTime() );
```

setNetPort(port)

- Used to set the network port
- Returns a numeric (0 = FAILED 1 = SUCCEEDED)

```
function portInit(%port)
{
    %failCount = 0;
    while(%failCount < 10 && !setNetPort(%port)) {
        echo("Port init failed on port " @ %port @ "
        trying next port.");
```

```

    %port++; %failCount++;
  }
}

```

lockMouse(isLocked)

- Used to toggle the mouse state
- No return type

```

function cursorOff()
{
    if ( $cursorControlled )
        lockMouse(true);
        Canvas.cursorOff();
}

```

rebuildModPaths()

- Used to rebuild the default MOD Paths (currently set to "common")
- No return type

```
rebuildModPaths();
```

setModPaths(path)

- Used to set mod path which dictates which directories will be visible to the scripts and the resource engine
- No return type

```

// Set the mod path which dictates which directories will be visible
// to the scripts and the resource engine.
$modPath = pushback($userMods, $baseMods, ";");
setModPaths($modPath);

```

getModPaths()

- Used to return the current mod path
- Returns a string

```
$modPath = getModPaths();
```

createCanvas(WindowTitle)

- Used to create a canvas
- Returns a bool (0 = FAIL 1 = SUCCEED)

```

if (!createCanvas(%windowName)) {
    quit();
    return;
}

```

saveJournal(jname)

- Used to save a journal to file specified by jname
- No return type

```

//-----
case "-jSave":
    $argUsed[$i]++;
    if ($hasNextArg)
    {
        echo("Saving event log to journal: " @ $nextArg);
        saveJournal($nextArg);
        $argUsed[$i+1]++;
        $i++;
    }
    else
        error("Error: Missing Command Line argument. Usage: -jSave <journal_name>");

```

playJournal(jname,[break])

- Used to play back saved journal specified by jname

-No return type

```
//-----
case "-jPlay":
    $argUsed[$i]++;
    if ($hasNextArg)
    {
        playJournal($nextArg,false);
        $argUsed[$i+1]++;
        $i++;
    }
else
    error("Error: Missing Command Line argument. Usage: -jPlay
    <journal_name>");
```

addTaggedString (string)

-Used to "tag" a string and add it to the NetStringTable
-Returns a string

```
// Tag the name with the "smurf" color:
$client.nameBase = %name;
$client.name = addTaggedString("\cp\c8" @ %name @ "\co");
```

removeTaggedString(tag)

-Used to remove a "tag" string from NetStringTable
-No return type

```
removeTaggedString(%client.name);
```

getTaggedString(tag)

-Used to get the string associated to the "tag"
-Returns a string

```
%name = getTaggedString( %client.name );
```

buildTaggedString(fmtTag, <arg1, ...arg9>)

-Used to build a "tag" string with specified fmtTag
-Returns a string

```
%MyTagedString = builtTaggedString(%client.name, %string);
```

commandToServer(func, <arg1,...argn>)

-Used to issued command specified by func with args on the server

-No return type

```
function SADSetPassword(%password)
```

```
{
    commandToServer('SADSetPassword', %password);
}
```

commandToClient(client, func, <arg1,...argn>)

-Used to issue func on client with args
-No return type

```
function chatMessageClient( %client, %sender, %voiceTag,
    %voicePitch, %msgString, %a1, %a2, %a3, %a4, %a5, %a6,
    %a7, %a8, %a9, %a10 )
```

```
{
    //see if the client has muted the sender
    if ( !%client.muted[%sender] )
        commandToClient( %client, 'ChatMessage',
            %sender, %voiceTag, %voicePitch, %msgString,
            %a1, %a2, %a3, %a4, %a5, %a6, %a7, %a8,
            %a9, %a10 );
}
```

allowConnections(bool)

- Used to enable/disable connections to the game server
- No return type

```
// Make sure the network port is set to the correct pref.
portInit($Pref::Server::Port);
allowConnections(true);
```

connect(addr)

- Used to connect to specified address
- No return type

```
connect(ip:192.168.0.1:4123);
```

localConnect()

- Used to create a local connection to a server
- No return type

```
//-----
function SM_StartMission()
{
    %id = SM_missionList.getSelectedId();
    %mission = getField(SM_missionList.getRowTextById(%id), 1);

    if ($pref::HostMultiPlayer)
        %serverType = "MultiPlayer";
    else
        %serverType = "SinglePlayer";

    createServer(%serverType, %mission);
    localConnect($pref::Player::Name);
}
```

startRecord(filename)

- Used to record a demo
- No return type

```
startRecord(MyNiftyDemo);
```

stopRecord()

- Used to stop recording a previously started demo
- No return type

```
stopRecord();
```

playDemo(filename)

- Used to play back a recorded demo
- No return type

```
playDemo(MyNiftyDemo);
```

isDemoRecording()

- Used to determine if a demo is currently being recorded
- Returns a numeric (0 = FALSE 1 = TRUE)

```
if( isDemoRecording())
    echo("Can only record one demo at a time");
```

msg(id,msg)

- Used to send a message (event?) to specified object ID
- No return type

```
msg(%someObject, somemessage);
```

queryMasterServer (port,flags,gametyoe,missiontype,minplayers,maxplayers,maxbots,regionmask,maxping,maxcpu,filterflags)

- Used to query a master server looking for specified information

-No return type

```
function JoinServerGui::query(%this)
{
    queryMasterServer(
        28000, // lanPort for local queries
        0, // Query flags
        $Client::GameTypeQuery, // gameTypes
        $Client::MissionTypeQuery, // missionType
        0, // minPlayers
        100, // maxPlayers
        0, // maxBots
        2, // regionMask
        0, // maxPing
        100, // minCPU
        0 // filterFlags
    );
}
```

cancelServerQuery()

-Used to cancel the current query, if there is anything left on the ping list, it's dropped.
-No return type

```
function JoinServerGui::cancel(%this)
{
    cancelServerQuery();
}
```

stopServerQuery()

-Used to Cancel the current query, anything left on the ping list is moved to the finished list as "done".
-No return type

```
function JoinServerGui::stop(%this)
{
```

```
    stopServerQuery();
}
```

startHeartbeat()

-Used to update server to Master Server
-No return type

```
if ($pref::Net::DisplayOnMaster != "Never" )
    schedule(0,0,startHeartbeat);
```

stopHeartbeat()

-Used to stop update to Master Server
-No return type

```
function destroyServer()
{
```

```
    $Server::ServerType = "";
    $missionRunning = false;
    allowConnections(false);
    stopHeartbeat();
```

```
    ...
    ...
}
```

getServerCount()

-Used to return the number of servers from the Master Server
-Return numeric

```
//-----
```

```
function JoinServerGui::update(%this)
{
    // Copy the servers into the server list.
    JS_queryStatus.setVisible(false);
    JS_serverList.clear();
    %sc = getServerCount();
    ...
    ...
}
```

setServerInfo(index)

-Used to update the Master Server with our server information
 -Return type numeric (0 = FAILED 1 = SUCCEED)

```
//-----
```

```
function JoinServerGui::join(%this)
```

```
{
    cancelServerQuery();
    %id = JS_serverList.getSelectedId();

    // The server info index is stored in the row along with
    // rest of displayed info.
    %index = getField(JS_serverList.
getRowTextById(%id),6);
    if (setServerInfo(%index)) {
        connect($ServerInfo::Address,$Client::Password,
$pref::Player::Name);
    }
}
```

setShadowDetailLevel(0..1)

-Used to set the level of detail for shadows

No return type

```
// Copy saved script prefs into C++ code.
setShadowDetailLevel( $pref::shadows );
setDefaultFov( $pref::Player::defaultFov );
setZoomSpeed( $pref::Player::zoomSpeed );
```

showShapeLoad (shapeName,faceCamera)

-Used with the “show” mod to load and test DTS files
 -No return type

```
command = “getLoadFilename(“*.*.dts”, showShapeLoad);”;
```

showSequenceLoad(sequenceFile,[sequenceName])

-Used to load a sequence from sequence file for a running
 ”show”
 -No return type

```
command = “getLoadFilename(“*.*.dsq”,
showSequenceLoad);”;
```

showTurnLeft(amt)

-Used to rotate dts associated with running “show” to the left
 -no return type

```
showMoveMap.bind(keyboard, z, showTurnLeft);
```

showTurnRight(amt)

-Used to rotate the dts associated with the running “show” to the
 right
 -No return type

```
showMoveMap.bind(keyboard, x, showTurnRight);
```

showUpdateThreadControl()

-No return type

showUpdateThreadControl();

showSelectSequence()

-No return type

showSelectSequence();

showPlay([threadNum])

-No return type

command = "showPlay(threadList.getValue());";

showStop([threadNum])

-No return type

command = "showStop(threadList.getValue());";

showSetScale(threadNum,Scale)

-No return type

new GuiTextEditCtrl(showScale)

```
{
    profile = "GuiTextEditProfile";
    position = "80 20";
    extent = "50 20";
    altCommand = "showSetScale(threadList.
    getValue(),showScale.getValue()); Canvas.
    popDialog(TSShowEditScale);";
}
```

showSetPos(theadNum, pos)

-No return type

showSetPos(threadList.getValue(),showScale.getValue());

showNewThread()

-No return type

command = "showNewThread();";

showDeleteThread(threadNum)

-Used to delete thread specified by threadNum

-No return type

command = "showDeleteThread(threadList.getValue());";

showToggleRoot()

-No return type

command = "showToggleRoot();";

showToggleStick()

-No return type

command = "showToggleStick();";

showSetCamera(bool)

-Used to put camera in a free fly or orbital mode (FALSE = free fly TRUE = orbital)

-No return type

command = "showSetCamera(true); showSetKeyboard(false);";

showSetKeyboard(bool)

-Used to enable keyboard control of shape (FALSE = no control
TRUE = control)

-No return type

```
command = "showSetCamera(true); showSetKeyboard(false);";
```

showSetLightDirection()

-Used to set the light direction based on camera position

-No return type

```
new GuiButtonCtrl ()
```

```
{
  profile = "GuiButtonProfile";
  position = "40 330";
  extent = "60 20";
  text = "Set Direction";
  command = "showSetLightDirection();";
};
```

showSetDetailSlider()

-Used to set the level of detail based on the slider or auto detail

-No return type

```
function showToggleDetail()
```

```
{
  if ($showAutoDetail)
  {
    showDetailText.setValue("Slider Sets Detail
Level");
```

```
  showSetDetailSlider();
  $showAutoDetail = false;
}
```

else

```
{
```

```
    showDetailText.setValue("Auto Detail Using
Distance");
```

```
    $showAutoDetail = true;
```

```
}
```

```
}
```

StripMLControlChars(string)

-Used to strip ML control characters from string

-Returns a string

```
function PlayerListGui::update
```

```
(%this,%clientId,%name,%isSuperAdmin,%isAdmin,%isAI,%score)
```

```
{
```

```
// Build the row to display. The name can have ML control tags,
// including color and font. Since we're not using and
// ML control here, we need to strip them off.
```

```
    %tag = %isSuperAdmin? "[Super]":
```

```
    (%isAdmin? "[Admin]":
```

```
    (%isAI? "[Bot]":
```

```
    "");
```

```
    %text = StripMLControlChars(%name) SPC %tag TAB
```

```
%score;
```

```
...
```

```
...
```

```
}
```

setInteriorRenderMode(modeNum)

-Used to set the detail render level for interiors

-No return type

```

function cycleDebugRenderMode(%val)
{
    if (!%val)
    return;
    if ($MFDebugRenderMode == 0)
    {
        // Outline mode, including fonts so no stats
        $MFDebugRenderMode = 1;
        GLEnableOutline(true);
    }
    else if ($MFDebugRenderMode == 1)
    {
        // Interior debug mode
        $MFDebugRenderMode = 2;
        GLEnableOutline(false);
        setInteriorRenderMode(7);
        showInterior();
    }
    else if ($MFDebugRenderMode == 2)
    {
        // Back to normal
        $MFDebugRenderMode = 0;
        setInteriorRenderMode(0);
        GLEnableOutline(false);
        show();
    }
}

```

setInteriorFocusedDebug()

-Used to enable Debug mode for interior focused objects
 passing a value to this function enables debugging, no
 value passed disables debugging
 -No return type

```
setInteriorFocusedDebug();
```

isPointInside(point)

-Used to determine if point is “inside” another object
 -Return numeric (0 = FALSE, 1 = TRUE)

```
isPointInside(“143 34 567”);
```

VectorAdd(vect1,vect2)

-Used to add two vectors together
 -Returns a string

```
%muzzleVelocity = VectorAdd(
    VectorScale(%muzzleVector, %projectile.muzzleVelocity),
    VectorScale(%objectVelocity, %projectile.velInheritFactor));
```

VectorSub(vect1,vect2)

-Used to subtract vect2 from vect1
 -Returns a string

```
// Apply the impulse
if (%impulse) {
    %impulseVec = VectorSub(%targetObject.getWorldBoxCenter(),
    %position);
    %impulseVec = VectorNormalize(%impulseVec);
    %impulseVec = VectorScale(%impulseVec, %impulse *
    %distScale);
    %targetObject.applyImpulse(%position, %impulseVec);
}
```

```
VectorScale(vect,scalar)
```

-Used to scale the vector by scalar

-Returns a string

```
// Apply the impulse
if (%impulse) {
    %impulseVec = VectorSub(%targetObject.getWorldBoxCenter(),
%position);
    %impulseVec = VectorNormalize(%impulseVec);
    %impulseVec = VectorScale(%impulseVec, %impulse *
%distScale);
    %targetObject.applyImpulse(%position, %impulseVec);
}
```

VectorNormalize(vec)

-Used to "Normalize" the vec

-Returns a string

```
// Apply the impulse
if (%impulse) {
    %impulseVec = VectorSub(%targetObject.getWorldBoxCenter(),
%position);
    %impulseVec = VectorNormalize(%impulseVec);
    %impulseVec = VectorScale(%impulseVec, %impulse *
%distScale);
    %targetObject.applyImpulse(%position, %impulseVec);
}
```

VectorDot(vec1,vec2)

-Used to return the DotProduct of vect1 and vec2

-Returns a string

```
// Add a vertical component to give the object a better arc
%verticalForce = %throwForce / 2;
%dot = vectorDot("0 0 1",%eye);
```

VectorCross(vec1,vec2)

-Used to get the CrossProduct of vec1 and vec2

-Returns a string

```
%vecCross = vectorCross("x y z","x y z");
```

VectorDist(vect1,vect2)

-Used to get the distance between vect1 and vect2

-Returns a string

```
%disatance = vectorDist(vector1,vector2);
```

VectorLen(vec)

-Used to get the length of the vector

-Returns a string

```
%length = vectorLen(vector);
```

VectorOrthoBasis(AngAxisF)

-Used to get the OrthoNormal for AngAxisF

-Returns a string

```
%ortho = vectorOrthoBasis("x y z angle");
```

MatrixCreate(Pos,Rot)

-Used to create a Matrix from supplied values

-Returns a string

```
%mat = matrixCreate("x y z", "x y z angle");
```

MatrixMultiply(Left,Right)

-Used to multiple two matrices

-Returns a string

```
%newMatrix = matrixMultiply(matrix1,matrix2);
```

MatrixMulVector(transform, vector)

-Used to multiple a matrix by a vector

-Returns a string

```
%daMatrix = matrixMulVector(matrix,vector);
```

MatrixMulPoint(transfor,point)

-Used to multiple a matrix by a point

-Returns a string

```
%daMatrix = matrixMulPoint(matrix,point);
```

getBoxCenter(Box)

-Used to get the center of a box

-Returns a string

```
// Set the object's position and initial velocity
%pos = getBoxCenter(%this.getWorldBox());
```

setRandomSeed([seed])

-Used to set the required "seed" for the MOD

-No return type

```
function initCommon()
```

```
{
```

```
    // All mods need the random seed set
    setRandomSeed();
```

// Very basic functions used by everyone

```
exec("./client/canvas.cs");
```

```
exec("./client/audio.cs");
```

```
}
```

getRandomSeed()

-Used to get the "seed" for the MOD

-Numeric return type

```
%seed = getRandomSeed();
```

getRandom([[max]][[min,max]])

-Used to get a random number

-Numeric return

```
%random = getRandom(34,176);
```

MatrixCreateFromEuler ("x y z")

-Used to create a matrix from given arguments

-Returns a string

```
%daMatrix = MatrixCreateFromEuler("x y z");
```

mSolveQuadratic(a,b,c)

-Used to solve for Quadratic

-Returns a string

```
%quad = mSolveQuadratic(a,b,c);
```

mSolveCubic(a,b,c,d)

-Used to solve for a Cube

-Returns a string

```
%cube = mSolveCubic(a,b,c,d);
```

mSolveQuartic(a,b,c,d,e)

- Used to solve for a Quartic
- Returns a string

```
%quartic = mSolveQuartic(a,b,c,d,e);
```

mFloor(float)

- Used to return the largest integral value not greater than "float"
- Returns a numeric

```
%pageLines = mFloor(%chatScrollHeight / %textHeight);
if (%pageLines <= 0)
    %pageLines = 1;
```

mCeil(float)

- Used to return the smallest integral value not less than "float"
- Returns a numeric

```
%min = mCeil(%chatScrollHeight / %textHeight);
```

mFloatLength(float, numDecimals)

- Used to return float with a "numDecimals" padding
- Returns a numeric

```
%newFloat = mFloatLength((7/3),5);
```

mAbs(float)

- Used to return absolute value of float
- returns a string

```
%abs = mAbs(76.3);
```

mSqrt(float)

- Used to return the square root (sqrt) of float
- Returns a numeric

```
%sqrt = mSqrt(69);
```

mPow(floatA,floatB)

- Used to get the value of floatA raised to the power of floatB
- Returns a numeric

```
%pow = mPow(2,4);
```

mLog(float)

- Used to return the natural logarithm of
- Returns a numeric

```
%log = mLog(7654.98);
```

mSin(float)

- Used to return the "sine" of float measured in radians
- Returns a numeric

```
%sin = mSin(65);
```

mCos(float)

- Used to return the "cosine" of float measured in radians
- Returns a numeric

```
%cos = mCos(69);
```

mTan(float)

- Used to return the “tangent” of float
- Returns a numeric

```
%tan = mTan(87.6);
```

mAsin(float)

- Used to return the “arc sine” of float
- returns a numeric

```
%asin = mAsin(-3,8);
```

mAcos(float)

- Used to return the “arc cosine” of float
- Returns a numeric

```
%acos = mAcos(-8,3);
```

mAtan(float)

- Used to return the “arc tangent” of float
- Returns a numeric

```
%atan = mAtan(-10,3);
```

mRadToDeg(float)

- Used to convert radiant to degrees
- Returns a numeric

```
%r2d = mRadToDeg(5);
```

mDegToRad(float)

- used to convert degrees to radians
- Returns a numeric

```
%c2r = mDegToRad(171);
```

ValidateMemory()

- Used to validate memory space for the game
- No return type

```
ValidateMemory();
```

FreeMemoryDump()

- Used to print out some useful statistics regarding free memory
- No return type

```
FreeMemoryDump();
```

dumpMemSnapshot(filename)

- Used to dump memory statistics to file
- No return type

```
dumpMemSnapshot(memdump.txt);
```

redbookOpen(<device>)²

- Used to open a redbook device
- Returns a numeric (0 = FAILED other = SUCCESS)

```
redbookOpen();
```

redbookClose()

- Used to close a redbook device

2 I am a bit stumped on the following, don't some searching of the code;

But it makes absolutely no sense to me. Does “redbook” refer to the “OpenGL” RedBook

-Returns a numeric (0 = FAILED 1 = SUCCES)

```
redbookClose();
```

redbookPlay(track)

-Used to play a track on a redbook device
-Returns a numeric (0 = FAILED 1 = SUCCES)

```
redbookPlay(2);
```

redbookStop()

-Used to stop playing
-Returns a numeric (0 = FAILED 1 = SUCCES)

```
redbookStop();
```

redbookGetTrackCount()

-Used to return the number of redbook tracks
-Returns a numeric

```
%tracks = redbookGetTrackCount();
```

redbookGetVolume()

-Used to get the current volume level of a redbook device
-Returns a numeric

```
%volume = redbookGetVolume();
```

redbookSetVolume(volume)

-Used to set the volume level of a redbook device
-Returns a numeric (0 = FAILED 1 = SUCCES)

```
redbookSetVolume(%volume);
```

redbookGetDeviceCount()

-Used to return the number of redbook devices
-Returns a numeric

```
%count = redbookGetDeviceCount();
```

redbookGetDeviceName(idx)

-Used to get the device name of specified redbook device index
-Returns a string

```
echo("Red Book (whats that?) device name :") @  
redbookGetDeviceName(1);
```

redbookGetLastError()

-Used to get the last know error from a redbook device
-Returns a string

```
echo("RedBook (whats that?) last known error :") @  
redbookGetLastError() );
```

videoSetGammaCorrection(gamma)

-Used to set the Gamma correct
-No return type

```
videoSetGammaCorrection($pref::OpenGL::gammaCorrection);
```

setDisplayDevice(deviceName{, width{, height{, bpp{, fullScreen}}}})

-Used to setup the display device with specified parameters
-Returns a numeric (0 = FAILED 1 = SUCCESS)

```
function optionsDlg::applyGraphics( %this )
```

```

    {
        %newDriver = OptGraphicsDriverMenu.getText();
        %newRes = OptGraphicsResolutionMenu.getText();
        %newBpp = OptGraphicsBPPMenu.getText();
        %newFullScreen = OptGraphicsFullscreenToggle.
getValue();

        if ( %newDriver !$= $pref::Video::displayDevice )
        {
            setDisplayDevice( %newDriver, firstWord(
                %newRes ), getWord( %newRes, 1 ), %newBpp,
                %newFullScreen );
            //OptionsDlg::deviceDependent( %this );
        }
        else
            setScreenMode( firstWord( %newRes ), getWord(
                %newRes, 1 ), %newBpp, %newFullScreen );
    }

```

setScreenMode(width, height, bpp, fullScreen)

-Used to setup the screen with specified parameters
 -Returns a numeric (0 = FAILED 1 = SUCCESS)

```

setScreenMode( firstWord( %newRes ), getWord( %newRes, 1
), %newBpp, %newFullScreen );

```

toggleFullScreen()

-Used to switch between fullscreen mode and windowed mode
 -Returns a numeric (0 = FAILED 1 = SUCCESS)

```

toggleFullScreen();

```

isFullScreen()

-Used to determine if screenmode is set to fullscreen or not
 -Returns a numeric (0 = NO 1 = YES)

```

if(isFullScreen())
    echo("We be full screened!");

```

switchBitDepth()

-Used to switch between 16 and 32 bpp (for fullscreen mode only)
 -Returns a numeric (0 = FAILED 1 = SUCCESS)

```

if(!switchBitDepth())
    echo("Unable to switch screen bpp");

```

prevResolution()

-Used to decrease screen resolution to the next lowest resolution
 -Returns a numeric (0 = FAILED 1 = SUCCESS)

```

if(!prevResolution ())
    echo("Unable to switch screen resolution");

```

nextResolution()

-Used to increase screen resolution to the next highest resolution
 -Returns a numeric (0 = FAILED 1 = SUCCESS)

```

if(!nextResolution ())
    echo("Unable to switch screen resolution");

```

getResolution()

-Used to get the current screen resolution
 -Returns a string


```
%res = getResolution();
```

setResolution(width, height, bpp)

-Used to set the screen resolution to specified parameters
-Returns a numeric (0 = FAILED 1 = SUCCESS)

```
if(!setResolution(640,480,32);)
    echo("Unable to set resolution to 640x480x32");
```

setRes width, height, bpp)

-See setResolution

getDisplayDeviceList()

-Used to get the device name for each display device
-Returns a string

```
echo("Display Device(s) : " @ getDisplayDeviceList() );
```

getResolutionList(devicename)

-Used to get all possible resolution combinations for specified device
-Returns a string

```
echo("Possible resolutions : " @ getResolutionList(%device) );
```

getVideoDriverInfo()

-Used to get device driver info
-Returns a string

```
echo("Device driver info : " @ getVideroDriverInfo() );
```

isDeviceFullScreenOnly(devicename)

-Used to determin if device is capable of fullscreen only
-Reuturns a numeric (0 = NO 1 = YES)

```
if(isDeviceFullScreenOnly(%devicename) )
    echo("You are limited to fullscreen mode only!");
```

setVerticalSync(<bool>)

-Used to enable/disable the use of Vertical Syn
-Reuturns a numeric (0 = FALSE 1 = TRUE)

```
if(!setVerticalSync(1) )
```

```
echo("Unable to enable Vertical Sync for your video device");
```

profilerMarkerEnable(markerName, true/false)³

-Used to enable/disable profiling for markerName -No return type

```
profilerMarkerEnable(mark,true);
```

profilerEnable(true/false)

-Used to enable/disable profiling
-No return type

```
profilerEnable(false);
```

profilerDump()

-Used to dump NetStringTable statistics to the console
-No return type

```
profilerDump();
```

³ Profiler commands are only available when the engine code is compiled with the ENABLE_PROFILER set

profilerDumpToFile(filename)

-Used to dump NetStringTable statistics to file specified by
fileName

-No return type

```
profileDumpToFile(dump.txt);
```

enableWinConsole(bool)

-Used to display the console window

-No return type

```
case "-console":
    enableWinConsole(true);
    $argUsed[$i]++;
```

isJoystickDetected()

-Used to determine if a joystick is present

-No return type

```
if(!isJoystickDetected())
```

```
echo("No Joystick was detected");
```

getJoystickAxes(instance)

-Used to get the current Axes of the joystick pointed to by
instance

-Returns a string

```
%joyAxes = getJoystickAxes( 3 );
```

enableMouse()

-Used to enable DirectInput polling of the mouse device

-Returns a numeric (0 = FAILED 1 = SUCCESS)

```
enableMouse();
```

disableMouse()

-Used to disable DirectInput polling of the mouse device

-No return type

```
disableMouse();
```

permDisableMouse()⁴

-Used to permatnatly disable DirectInput polling of the mouse
device

-No return type

```
permDisableMouse();
```

echoInputState()

-Used to display (to the console) the current state of DirectInput,
mouse, keyboard and joystick

-No return type

```
echoInputState();
```

toggleInputState()⁵

-Used to toggle DirectInput state between enable and disable
will also print the new Input state (same as echoInputState) to
the console

-No return type

4 Currently this function is only used in the Mac port of the engine

5 Currently this function is only used in the Mac port of the engine

```
toggleInputState();
```

MathInit(detect|C|FPU|MMX|3DNOW|SSE|...)

- Used to enable the use of certain Math extensions based on the users CPU type.
- No return type

```
MathInit(detect);
```

AddCardProfile(vendor,renderer,safeMode,lockArray,subImage,fo gTexture,noEnvColor,clipHigh,deleteContext,texCompress,interior Lock,skipFirstFog,only16,noArraysAlpha,proFile)⁶

- Used to store certain aspects of a card for later usage
- No return type

addOSCardProfile(vendor,renderer,allowOpenGL,allowD3D,prefer OpenGL)⁷

- Used to store certain aspects of a card for later usage
- No return type

getDesktopResolution()

- Used to get the current desktop resolution
- Returns a string

6 Looks to be one of those “whack” jobs done when GG had to rip out the T2 proprietary information. The code executes a script (CardProfiles.cs) that does not exist in the CVS tree.

7 Looks to be one of those “whack” jobs done when GG had to rip out the T2 proprietary information. The code executes a script (CardProfiles.cs) that does not exist in the CVS tree.

```
%res = getDesktopResolution();
```

activateKeyboard()

- Used to enable DirectInput polling of the keyboard
- Returns a numeric (0 = FAILED 1 = SUCCESS)

```
if(activateKeyboard())
```

```
echo(“Keyboard has been activated”);
```

deactivateKeyboard()

- Used to disable DirectInput polling of the keyboard
- No return type

```
deactivateKeyboard();
```

GLEnableLogging(bool)⁸

- Used to enable/disable OpenGL logging to gl_log.txt
- No return type

```
GLEnableLoggin(true);
```

GLEnableOutline(bool)

- Used to draw outlines around OpenGL elements (wire mesh?)
- No return type

```
GLEnableOutline(true);
```

GLEnableMetrics(bool)

- Used to draw a Metrics around each OpenGL entity?
- No return type

8 Only accesable in the Win32 port of TGE

GLEnableMetrics(1);

inputLog(string)

-Used to enable/disable logging of DirectInput events to log file specified by string

-No return type

inputLog(DI.log);

launchDedicatedServer(missionType, map, botCount)

-Used to launch a dedicated game server with specified

arguments

-No return type

launchDedicatedServer(mymission,damap,0);

isKoreanBuild()

-Silly Korean registry key checker

-Returns a string

```
if(isKoreanBuild())
    echo("Silly Korean Build!");
```

debug_testx86unixmutex()

-Used to see if the Unix system TGE is running one can

create a mutex⁹

-No return type

debug_testx86unixmutex();

debug_debugbreak()

-Used in windows to causes the program to display a dialog box as if it had crashed. Under Unix we force a segfault.

-No return type

debug_debugbreak();

resetLighting()

-Used to reset the current lighting (looks to be OGL only)

-No return type

resetLighting();

getMaxFrameAllocation()

-Used to return the Max Frame Allocation unit

-Returns a numer

%maxFrameAlloc = getMaxFrameAllocation();

9 A mutex enforces a policy of mutual exclusion. Only one thread at a time may hold a particular mutex. Threads trying to lock a held mutex will block until the mutex is unlocked.

Mutexes are strictly bracketing and may not be recursively locked. That is to say, mutexes should be exited in the opposite order they were entered, and cannot be reentered before exiting.

dumpNetStringTable()

- Used to dump NetStringTable statistics to the console
- No return type

```
dumpNetStringTable();
```

InitContainerRadiusSearch ('x y z', radius, mask)

- Used to search for objects of type "mask" within a radius around the specified coordinates given
- No return type

```
InitContainerRadiusSearch ("0 450 76", %somerad,
%somemask)
```

ContainerSearchNext()

- Used to return the next object in a container search
- Returns a numeric

```
while(ContainerSearchNext() != -1 )
{
    ...
    ...
}
```

ContainerSearchCurrDist()

- Used to return the Container Search Distance
- Returns a numeric

```
%dist = ContainerSearchCurrDist();
```

ContainerSearchCurrRadiusDist()

- Used to return the Container Search Radius Distance
- Returns a numeric

```
%rad = ContainerSearchCurrRadiusDist();
```

ContainerRayCast ('x y z', 'x y z', mask, [exempt object])

- Used to find objects of type "mask" between the two cords supplied

- Returns a string

```
// Search for objects within the range that fit the masks above
// If we are in first person mode, we make sure player is not
// selectable by setting fourth parameter (exempt
// from collisions) when calling ContainerRayCast
%player = %client.player;
if ($firstPerson)
{
    %scanTarg = ContainerRayCast (%cameraPoint,
    %rangeEnd, %searchMasks, %player);
}
else //3rd person - player is selectable in this case
{
    %scanTarg = ContainerRayCast (%cameraPoint,
    %rangeEnd, %searchMasks);
}
```

ContainerBoxEmpty(Mask, Loc, Rad [,yRad, zRad])

- Used to see if any objects of given types are present in box of given extent
- Returns a numeric (0 = No objects found 1 = Objects found)

pathOnMissionLoadDone()

- Used to construct MOD paths
- No return type

```
pathOnMissionLoadDone();
```

makeTestTerrain(filename)

- Used to make a test terrain file
- No return type

getTerrainHeight(Point2I pos)

- Used to get the terrain height at the specified position
- Return numeric

```
%TerHeight = getTerrainHeight(%pos);
```

Versões do Memorial

1ª Versão	13 de setembro de 2007	Iniciada a escrita do memorial técnico.
2ª Versão	03 de março de 2008	Adicionadas informações sobre uso de videos em formato Theora.
3ª Versão	02 de dezembro de 2008	Adicionado capítulo de introdução.
4ª Versão	22 de janeiro de 2009	Adicionada instruções para compilação em Linux / Estrutura de pacotes.