
Mugen: Crie aqui seus jogos de luta

Confira neste tutorial todos os passos para criar personagens no Mugen



Quando gostamos de algo, é natural que procuremos aprofundar nossos conhecimentos em tal assunto. E, enquanto não conseguirmos reunir o máximo de informações a respeito, não nos satisfazemos. Esse comportamento também acontece na área de jogos. Nos últimos tempos, tem sido comum as pessoas passarem do nível de jogador para o de desenvolvedor, já que, após ter esmiuçado todos os detalhes e comandos do jogo, o caminho natural é obter conhecimentos sobre sua lógica e estrutura. E para facilitar o desenvolvimento de games caseiros, atualmente há diversas ferramentas disponíveis na Internet. Uma delas é o Mugen, que só requer uma leitura atenta de sua documentação, permitindo que qualquer pessoa com um mínimo de conhecimento em informática se aventure na criação de seus personagens e cenários. Se você se enquadra nesse perfil, o que está esperando para fazer as suas primeiras criações com esse programa?



Básico do Mugen

Antes de colocar a mão na massa, conheça os conceitos básicos que regem o funcionamento do Mugen:

Estrutura de diretórios do Mugen

CHARS

Neste diretório estão localizados os arquivos dos lutadores.

DATA

Pasta que reúne arquivos de configuração do Mugen.

DOCS

Documentação e tutoriais do Mugen.

GRAPHICS

Estão localizados nesta pasta as imagens e animações utilizadas no Mugen, como tela de seleção de lutadores, tela de entrada, tela de opções, entre outros.

FONT

Pasta com fontes usadas no jogo.

SOUND

Pasta onde serão salvos arquivos de som do Mugen.

STAGES

Os cenários deverão ser salvos exatamente nesta pasta.



Personagem (pasta CHARS)

Arquivos necessários para a criação dos personagens

personagem.DEF

Arquivo com as configurações que definem o personagem. Pode ser aberto e editado em qualquer editor de textos, inclusive o Bloco de Notas. Informações como nome, versão e os arquivos utilizados pelo lutador são definidos no arquivo DEF.

personagem.SFF

Coletânea de imagens do personagem. Neste arquivo, estão reunidas todas as imagens do personagem.

personagem.CNS

Arquivo que determinará o comportamento do personagem, ou seja, neste arquivo, serão definidos todos os movimentos. Cada personagem pode ter vários arquivos com a extensão CNS.

personagem.CMD

Arquivo que reúne comandos do personagem. Você pode relacionar uma animação/imagem a um determinado comando.

personagem.ACT

Se refere à paleta de cores. Neste arquivo, são especificadas as cores do seu personagem ao escolher botões diferentes. Um personagem pode ter até 12 paletas.

personagem.SND

Arquivo que reúne os efeitos de áudio e som do personagem lutador. Os arquivos estão compactados em apenas um arquivo.

personagem.AIR

Arquivo responsável pelas caixas de colisão e de ataque do personagem.



Cenário (Pasta Stages)

Para compor um cenário no Mugen, você precisará modificar apenas três arquivos: cenário.DEF (arquivo com toda a programação do cenário, coordenadas, entre outras especificações; cenário.SFF (arquivo com as imagens e animações compactadas do cenário) e audio.xxx (arquivo opcional, pode estar no formato MP3 ou MID).

➤➤ Atalhos do Mugen

Para facilitar a vida do jogador, os desenvolvedores do Mugen criaram algumas teclas de atalho que permitem mais interatividade entre o jogador e o Mugen. Confira:

F1 - Zera a energia do jogador 2

F2 - Configura os níveis de energia dos jogadores 1 e 2 para o valor um

F3 - Configura os dois jogadores com energia máxima

F4 - Reinicia a partida

F5 - Finaliza a luta por um tempo

F12 - Captura imagem da tela

CTRL+F1 - Zera a energia do jogador 1

CTRL+F2 - Reduz a energia do jogador 1 ao valor 1

SHIFT+F2 - Reduz a energia do jogador 2 ao valor 2

CRTL+A - este comando serve para congelar movimentos dos personagens

CRTL+C - Visualize as caixas de colisão e de ataque dos lutadores

CTRL+D - Visualize o debug

CRTL+L - Faz desaparecer as barras de energia e super

CRTL+S - Ao ativar essas teclas, você aumenta a velocidade do jogo na capacidade máxima de seu computador.

Para voltar à velocidade normal, basta pressionar novamente as teclas CTRL+S

CTRL+V - Habilita o v-Sync

CRTL+ALT+# valor de 1-4
- opção habilita/desabilita o jogador selecionado

Pause - Interrompe o jogo

Barra de Espaço -

Repõe a energia dos jogadores

Esc - Finaliza o jogo

Instalação do Mugen

Para ter o Mugen instalado, basta copiar a pasta Mugen 01.01 (\Tutoriais\Mugen\Ferramentas\Mugen 01.01) do CD-ROM para o qualquer pasta de seu computador. Nessa pasta, estão reunidas todas as ferramentas, programas e documentos necessários para elaboração de novos personagens e cenários.

Agora que você já está familiarizado com os conceitos básicos do Mugen, podemos passar para a próxima etapa da criação de jogos: a elaboração do personagem.

Para esta tarefa, serão necessários um editor de imagens como o Photoshop ou PaintShop Pro, as ferramentas do Mugen sprmaker e AirEditor e um editor de textos simples que pode ser até mesmo o Bloco de Notas.

Para um maior aproveitamento deste tutorial, é necessário um mínimo de conhecimento sobre editores de texto, edição de imagens, conceitos de jogabilidade e linhas de comando, caso contrário, você terá dificuldade de seguir o que preparamos para você. Neste caso, é indicado a leitura de livros para iniciantes ou web sites. Para aqueles que tem noção de lógica de programação, o tutorial é bastante fácil, já que os arquivos de configuração estão comentados. Podemos começar?

Criando um personagem

Para criar os seus próprios lutadores, vá à pasta chars/ e crie um novo diretório para o seu personagem. O nosso se chamará Fighter (escolha o nome que desejar para ele). Coloque o mesmo nome do personagem na pasta a ser criada dentro do diretório chars (chars\fighter).

Feito isso, vamos configurar os arquivos-base do personagem. São eles: fighter.air, fighter.cmd, fighter.cns, fighter.def, fighter.sff, fighter.snd e fighter.act. Uma alternativa é criá-los do zero, mas isso demandaria um maior nível de detalhamento, fugindo do propósito inicial deste tutorial que é explicar os principais conceitos da criação de jogos com o Mugen de uma maneira simples e descomplicada. Para facilitar a explicação, ilustraremos a criação do personagem com os arquivos de exemplo do Goku Super Sayajin 2 (que estão localizados no diretório \Tutoriais\Mugen\Personagem do CD-ROM que acompanha a revista). Copie os arquivos do personagem Goku Super Sayajin 2 para o diretório correspondente ao personagem (/chars/fighter).

Os arquivos a ser copiados e renomeados são:

```
\\Tutoriais\\Mugen\\Personagem\\goku-ssj2.air ->> \\chars\\fighter\\fighter.air  
\\Tutoriais\\Mugen\\Personagem\\goku-ssj2.cns ->> \\chars\\fighter\\fighter.cns  
\\Tutoriais\\Mugen\\Personagem\\goku-ssj2.cmd ->> \\chars\\fighter\\fighter.cmd  
\\Tutoriais\\Mugen\\Personagem\\Goku-SSJ2.def ->> \\chars\\fighter\\fighter.def
```

Esses quatro arquivos são os arquivos que reúnem as configurações do personagem. Eles podem ser abertos e editados em qualquer editor de texto como o EDIT do MS-DOS ou o Bloco de Notas do Windows.

Vamos começar com o arquivo fighter.def. Este arquivo é o de configuração mais básico do personagem a ser criado. Abra-o com o Bloco de Notas e modifique os campos para os valores apropriados, conforme pode ser visto no exemplo:

[Info]

name = "Fighter"

;Nome do personagem

displayname = "Fighter"

;Nome do personagem a ser mostrado na tela

versiondate = 15,01,2004

;Data da versão do personagem

mugenversion = 04,14,2001

;Versão do Mugen na qual o personagem funciona

author = "Nomedoautor"

;Nome do autor do personagem, neste caso, você pode colocar o seu nome

pal.defaults = 1

;Paletas padrão em ordem de preferência

[Files]

cmd = fighter.cmd

;Conjunto de comandos

cns = fighter.cns ;Constantes

st = fighter.cns ;Estados

stcommon = common1.cns

;Estados comuns

sprite = fighter.sff ;Sprites

anim = fighter.air;Animações

sound = fighter.snd ;Som (deixar em branco se não houver)

pal1 = fighter.act

;Paletas (pode haver até 12)

[Arcade]

;intro.storyboard =

;Deixar em branco por enquanto

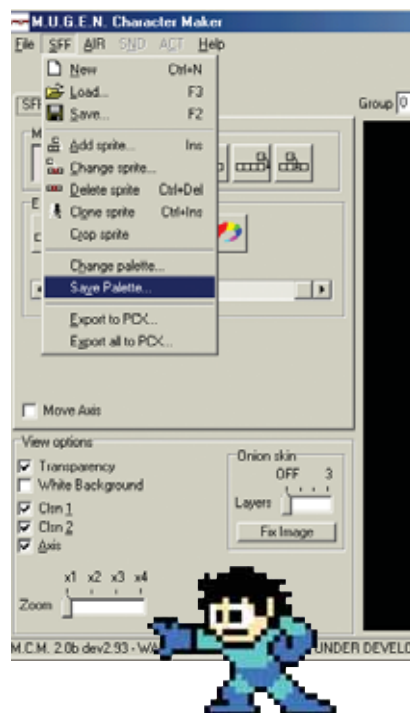
;ending.storyboard =

;Deixar em branco por enquanto

Depois de salvar o arquivo fighter.def, vamos às paletas de cores. Elas são elementos muito importantes no jogo, pois a partir delas é que as cores podem ser exibidas corretamente. No Mugen, o número máximo de cores que você pode trabalhar é 256 cores. Você pode criar a sua paleta de cores por meio de um editor de imagens como

o Photoshop ou o Paint Shop Pro. Outra forma de gerar paletas é por meio de um arquivo SFF. Para essa tarefa, execute o programa Mugen Character Maker (localizado na pasta Ferramentas do CD-ROM), abra um arquivo SFF através do menu SFF, Load. Neste caso, usamos o modelo do personagem Goku Super Sayajin. Com o personagem aberto, vá ao menu SFF e escolha Save Palette para salvar uma nova paleta.

Gerando o arquivo de sprites



Agora que já temos a paleta de cores, o próximo passo será gerar o arquivo com extensão .sff, que reúne todas as imagens do personagem. Aqui vamos usar imagens de personagens prontos. Com o programa Mugen Character Maker, exporte os arquivos de imagem PCX de um personagem já criado e salve as imagens na pasta /work/fighter. As figuras que utilizaremos são as fighter0.pcx - fighter6.pcx. Para montar os movimentos do seu

personagem, você terá que atribuir alguns valores como número do grupo e da imagem e os valores dos eixos X e Y para cada sprite.

O número do grupo e da imagem para cada arquivo PCX será utilizado para que o arquivo .air (de animação) consiga identificar cada quadro da imagem. Dessa forma, esse par de números estará diretamente associado a um determinado sprite. Por exemplo, ao se referir ao arquivo fighter0.pcx, você usará o par 0,0. O grupo 0, em geral, é utilizado para posições em que o personagem esteja de pé. Para saber quais são os números recomendados para outras animações, confira alguns grupos de movimento na tabela abaixo:

Grupo Descrição

0 Parado

5 Virando

10 Abaixando

11 Abaixado

12 Levantando-se

20 Andando pra frente

21 Andando pra trás

40 Pular (no chão)

41 Pular para cima)

42 Pular pra frente (indo pra cima)

43 Pular pra trás (indo pra cima)

44 Pular descendo

45 Pular pra frente (descendo)

46 Pular pra trás (descendo)

47 Pouso (no chão)

100 Correr pra frente

105 Correr pra trás

110 Esquivar-se para cima

115 Esquivar-se para baixo

120 Início de defesa (parado)

121 Início de defesa (abaixado)

122 Início de defesa (voando)

130 Defendendo-se (parado)

131 Defendendo-se (abaixado)

132 Defendendo-se (voando)

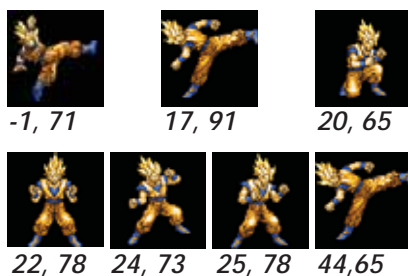
140 Parar de defender (parado)

141 Parar de defender (abaixado)

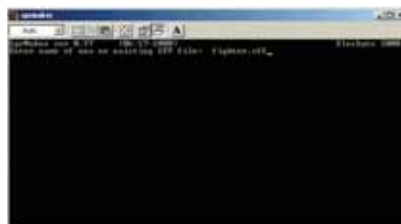
142 Parar de defender (voando)
150 Defendendo-se de um golpe sem movimento
151 Defendendo-se de um golpe com o personagem abaixado
152 Defendendo-se de um golpe com o personagem voando
200-299 - Ataques em pé
300-399 - Mais ataques com o personagem em pé e correndo
400-499 - Ataques com o personagem agachado
500-599 - Mais ataques com o personagem agachado
600-699 - Ataques aéreos
700-799 - Mais ataques aéreos
1000-2999 - Ataques especiais
3000-4999 - Ataques mais poderosos
5000-5999 - Faixa de números reservados. Para saber quais são, verifique em *air.txt*

Além do grupo e do número da imagem, dentro de cada sprite, você terá que atribuir os eixos X e Y. Para frames em que o personagem esteja em pé, ele geralmente está na parte inferior central do sprite. No caso de sprites de pulo, o eixo está onde os pés do personagem estariam se ele estivesse em pé. Para verificar os eixos de cada personagem, execute o Mugen e pressione as teclas CTRL+C enquanto joga. Isso mostrará as caixas de colisão e eixos do sprite. Veja abaixo alguns exemplos de posição de eixos:

É muito importante o posicionamento dos eixos, se eles estiverem desalinhados, você vai perceber seus sprites



tremendo. Se cometer mais erros de posicionamento, seus sprites podem terminar aparecendo em lugares completamente distantes uns dos outros. O próximo passo é copiar a ferramenta *sprmaker.exe* (localizada no caminho Tutoriais\Mugen\Ferramen-



tas\Tools 98 do CD-ROM) para o mesmo diretório das imagens PCX (*work\fighter*). Feito isso, clique duas vezes sobre o *sprmaker.exe*. Uma janela do prompt do MS-DOS será aberta. Seguindo as instruções da tela, digite o nome do arquivo SFF a ser criado. Neste caso, digitamos *fighter.sff* e depois teclamos Enter. Feito isso, o próximo passo será indicar o nome do arquivo PCX. Como nomeamos esses arquivos de imagem como *fighter0.pcx* a *fighter6.pcx*, é só digitar o primeiro

Com o arquivo SFF pronto, qual é o próximo passo?



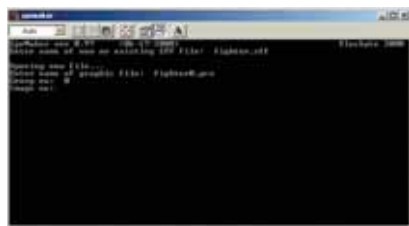
Agora que você já tem os sprites do personagem em pé no arquivo *fighter.sff*, vamos adicionar movimento ao personagem. Para fazer a animação, será necessário criar uma ação para isso. Para essa tarefa, abra o arquivo *fighter.air* em um editor de texto para começarmos a adicionar uma ação para o personagem. Como pegamos o arquivo .air do Goku Super Sayajin 2 como modelo, você notará alguns comandos prontos neste arquivo. Ele se parecerá com a estrutura de comandos abaixo:

```
[Begin Action 0] --> O valor da ação é 0, pois se trata de uma Clsn2Default:
7      --> animação com o personagem em pé
Clsn2[0] = -18,-56, 8,-36
Clsn2[1] = 24,-51, 8,-38
Clsn2[2] = -2,-56, 8,-63
Clsn2[3] = -9,-36, 6,-26
Clsn2[4] = 16,-12,-16,-26
Clsn2[5] = 23, 0, 8,-12
Clsn2[6] = -23,-12,-14, 2
0,0, 0,0, 21  --> Os valores desta linha significa: número de
0,1, 0,0, 21  --> grupo, número de imagem, posição em X, posição
0,2, 0,0, 21  --> em Y, gameticks*
0,3, 0,0, 7
0,4, 0,0, 7
0,5, 0,0, 7
0,6, 0,0, 7
0,7, 0,0, 7
```

* 1 game-tick equivale a 1/60 segundos, então 60 ticks é igual a 1 segundo

Salve o arquivo e agora você está pronto para ver como está o seu personagem.

arquivo da série e depois teclar Enter novamente. Depois, surgirá a linha de comando "Grupo no", digite 0 e tecla Enter. O valor 0 se refere à posição do personagem que nesses sprites está em pé.



Depois de definir o grupo de imagem, iremos definir o número dela. Para seguir um padrão, atribua um número relacionado à imagem. Em nosso caso, como o arquivo de imagem é o fighter0.pcx, atribuímos 0 como número da imagem. Na linha de comando "Image no", digite 0 e tecla Enter.

Agora, iremos atribuir os eixos X e Y. Como em um sistema cartesiano você pode atribuir valores numéricos positivos e negativos. Sendo que quanto maior o valor do eixo X, mais a esquerda a imagem aparecerá e para o eixo Y, quanto maior o valor de Y, mais para baixo a imagem será posicionada. Esse script deverá ser repetido para todas as imagens. Assim que terminar de digitar os atributos de cada sprite, tecla Enter para finalizar o programa e concluir a geração do arquivo SFF. Ao fim desse processo, copie o arquivo fighter.sff criado para a pasta /chars/fighter.

Checando as ações do personagem

Verifique se todos os arquivos estão nos diretórios corretos, incluindo paletas e arquivos .cns, .cmd, .air, .sff e .def. Se todos os arquivos estiverem OK, carregue o personagem no Mugen através do prompt de comando do MS-DOS. Digite a linha de comando abaixo e tecla Enter.

Mugen fighter fighter

O Mugen será carregado no modo Versus. Neste modo você poderá verificar se a posição e as ações do personagem estão corretas. Caso você perceba que um sprite está em um posição incorreta, altere o eixo da imagem correspondente. Depois de fazer os ajustes necessários em todos os sprites, o seu personagem não deverá parecer deslizando ou tremendo durante a animação. Outra situação que geralmente acontece é o número incorreto dos sprites no arquivo .air. Verifique cuidadosamente o arquivo .air para ver se os números correspondem àqueles usados para a construção de seu .sff. Corrigido todos os erros encontrados, você pode prosseguir a construção de ações, só que agora fazendo o personagem andar ou pular.

Caixas de colisão

Uma vez que você tenha montado uma animação, poderá aperfeiçoar ainda mais os movimentos e ações para o seu personagem. Para isso, será necessário o conhecimento sobre as caixas de colisão. Saiba abaixo explicações sobre esse recurso no Mugen.

Tipos de caixas de colisão

O que são caixas de colisão?

É a área do personagem onde ele será atingido. Dessa forma, quando o oponente acertar essa região, este somará pontos e fará com que o seu personagem perca pontos. Para visualizar as caixas de colisão, pressione as teclas CTRL+C no Mugen e você visualizará caixas azuis nos personagens.

CLSN1 - é o local onde o personagem vai acertar o oponente em animações de ataque. São mostradas caixas em vermelho pelo Mugen.

CLSN2 - é a área onde o personagem pode ser atingido. São mostradas caixas na cor azul pelo Mugen.

Trabalhando com o AirEditor



Para definir as caixas de colisão para os seus personagens, será necessário o programa AirEditor. Copie o arquivo airedit.exe para o mesmo diretório do mugen.exe e o airedit.cfg para o diretório data. Execute o arquivo, vá ao menu File, Open e indique a localização do arquivo .sff e o .air correspondentes ao seu personagem. Você o verá na tela. Para definir uma caixa de colisão, é simples: basta desenhar uma caixa com o mouse sobre qualquer parte do personagem. Esta caixa será mostrada na cor verde claro. Começaremos criando uma caixa para o corpo do personagem, assim que terminar o desenho, clique em Save1 para salvá-la. Agora, defina uma segunda caixa para a cabeça e clique em Save2 para salvá-la. Siga esses mesmos passos até que

todo o corpo do personagem tenha sido coberto por caixas de colisão. Note que as caixas salvas são mostradas em uma cor mais escura. Caso queira corrigir uma dessas caixas, basta arrastar uma nova caixa verde, e depois, salve no mesmo espaço ocupado pela caixa que deseja corrigir. Por fim, você deve salvar este conjunto de caixas para o arquivo .air. Para isso, vá ao menu File, Save. Aparecerá as opções Clsn1 e Clsn2, escolha um dos tipos e repita esse processo até que você tenha definido todas as caixas para cada elemento da animação.

Criando um movimento de ataque simples



Depois de aprender a criar animações e sprites para o seu personagem, agora vamos elaborar exatamente o estado dele, ou seja, o seu comportamento. Tudo o que o seu personagem faz está descrito no arquivo .cns. Abra o arquivo fighter.cns em um editor de texto. A primeira linha do

arquivo é o grupo [Data]. Aqui é onde estão reunidas as variáveis do lutador, responsáveis por muitos aspectos básicos do personagem. Para não se perder no posicionamento do personagem, é importante ter em mente que o sistema de coordenadas é positivo em X para a direita e negativo em Y para cima. Todas as distâncias são medidas em pixels e as velocidades em pixel/tick (lembrando que um game tick equivale a 1/60 de um segundo).

[Data]

life = 1000

; Vida que o personagem vai começar o jogo

attack = 100

; A força de ataque do personagem (quanto maior o valor, mais forte serão os ataques)

defence = 100

; a força de defesa do personagem

fall.defence_up = 50

; Porcentagem para a defesa aumentar toda vez que o personagem sofrer um ataque

liedown.time = 60

; Tempo que o lutador leva caindo no chão

airjuggle = 15

sparkno = 2

; Refere-se a faísca para os ataques certos de seu personagem

guard.sparkno = 40

; Função igual a de cima, com a diferença de que esta variável se aplica aos ataques defendidos

KO.echo = 0

; Função para ativar/desativar o eco no grito de seu personagem quando ele morre.

volume = 0

; Variável para aumentar o volume dos sons do seu personagem

IntPersistIndex = 58

FloatPersistIndex = 40

[Size]

xscale = 1

;Parâmetro que define o tamanho horizontal do seu personagem

yscale = 1

;Parâmetro que define o tamanho vertical do seu personagem

ground.back = 15

;Largura do personagem (atrás, no chão)

ground.front = 16

;Largura do personagem (frente, no chão)

air.back = 12

;Largura do personagem (atrás, no ar)

air.front = 12

;Largura do personagem (frente, no ar)

height = 60

;Altura do personagem (para o oponente pular por cima)

attack.dist = 160

;Valor padrão para a distância de ataque (É recomendável não alterar esse valor)

proj.attack.dist = 90

;Valor padrão para a distância de projéteis (É recomendável não alterar esse valor)

proj.doscale = 0

head.pos = -5, -90

;Posição aproximada da cabeça

mid.pos = -5, -60

;Posição aproximada do centro do personagem

shadowoffset = 0

;Número de pixels em que a sombra fica distante do personagem

draw.offset = 0,0

;Função para mudar a posição do personagem. Não mude o valor!

[Velocity]

;Grupo que irá regular a velocidade do personagem

walk.fwd = 2.4

;Andar para frente

walk.back = -2.2

❖ Criação de um ataque simples

Como copiamos os arquivos de configuração do Goku Super Sayajin 2, inclusive o .csn, então o seu personagem já tem todos os estados deste personagem. O que faremos é alterar o soco em pé do GSS para que ele funcione com o seu personagem. Não se preocupe com todos os outros estados por enquanto, você poderá modificá-los mais tarde ou remover aqueles que não for usar.

O soco fraco em pé do Goku possui estado de valor 200, por este motivo trabalharemos com sprites que envolvam imagens deste grupo e uma animação do mesmo grupo. O primeiro passo a ser executado é fazer os sprites e a animação para este ataque. Para este exemplo, será necessário que você adicione três novos sprites e reconstrua o arquivo .sff. No



arquivo fighter.air, crie uma nova ação de animação com o grupo 200.

Como construímos o nosso arquivo fighter.air a partir do arquivo do personagem Goku Super Sayajin 2, localize a ação de animação 200 e substitua a antiga pela nova. As instruções dessa nova ação deve ser igual aos comandos abaixo:

[Begin Action 200]

200,0,0,0,2

200,1,0,0,3

200,2,0,0,5

200,1,0,0,4

200,0,0,0,3

Salve o arquivo .air e execute o AirEditor para criar as caixas de colisão para este ataque. Para

essa tarefa, siga as instruções dadas anteriormente nesse tutorial. Para o quadro de animação em que o personagem golpeará o oponente, não se esqueça de criar uma caixa de ataque (cls1) sobre a caixa de impacto. É um bom recurso criar as caixas de impacto nas áreas das caixas de ataque. Caso isso não seja feito, o seu personagem não poderá trocar golpes com outro lutador que esteja tentando golpeá-lo naquela área. Assim que tiver terminado de salvar suas caixas de colisão no arquivo fighter.air, abra o arquivo fighter.cns com um editor de texto, localize a linha de comando [Statedef 200] e delete todos os comandos desta seção. Pronto, você acabou de excluir o soco fraco em pé do Goku.



Dica: É importante lembrar que as caixas de colisão devem ser criadas para todos os sprites do personagem


```

;Andar para trás
run.fwd = 4.6, 0
;Correr para frente (x, y)(y deve
ser igual a 0)
run.back = -4.5,-3.8
;Correr para trás (x, y)
jump.neu = 0,-8.4
;Velocidade do pulo neutro (x, y)
jump.back = -2.55
;Velocidade do pulo para trás (x, y)
jump.fwd = 2.5
;Velocidade do pulo para frente (x, y)
runjump.back = -2.55,-8.1
;Velocidade do pulo enquanto
corre (opcional)
runjump.fwd = 4,-8.1
airjump.neu = 0,-8.1
airjump.back = -2.55
;Velocidade do pulo no ar aéreo
(opcional)
airjump.fwd = 2.5

```

```

[Movement]
;Variáveis usadas para o mo-
vimento do personagem
airjump.num = 1
;Número de pulos aéreos per-
mitidos (opcional)
airjump.height = 35
;Distância mínima do chão
para que você possa dar o pu-
lar no ar (opcional)
yaccel = .44 ;Aceleração
vertical
stand.friction = .85
;Coeficiente de fricção quan-
do o personagem está em pé
crouch.friction = .82
;Coeficiente de fricção quando
o personagem está abaixado

```

Depois de ter feito os ajustes nas variáveis, vamos dar prosseguimento na criação de um ataque simples.

Criando um State

Como deletamos o estado 200 de nosso personagem, criaremos um do zero. Para começar, abra o arquivo fighter.cns no Bloco de Notas ou qualquer editor de texto de sua preferência e copie o código abaixo para o arquivo fighter.cns:

```

[Statedef 200]
;Número corresponde a animação
definida no arquivo .air e ao grupo
de imagem do personagem
type = S
;Tipo de golpe. S significa stand
ou seja "em pé"
movetype = A
;O golpe é um ataque
physics = S
;Opção padrão para golpes em pé
juggle = 1
;Número de vezes que o persona-
gem pode ser atingido no ar
velset = 0,0
;Velocidade inicial é 0 tanto para
X quanto para Y
ctrl = 0
;Durante o golpe o jogador não
terá controle sobre o personagem
anim = 200
;Animação do golpe, no arquivo
Air ela corresponde ao número
200.
poweradd = 40
;Ao finalizar o golpe será adiciona-
do 40 unidades à barra de poder
sprpriority = 2

```

;Prioridade. Neste caso, ele fica à frente de sprites com prioridades menores

```

[State 200, 1]
;Estado que toca sons
type = PlaySnd
trigger1 = AnimElem = 2
;O som será executado no se-
gundo elemento da animação
value = 1,3
;Será executado o som 1,3 do
snd do personagem

```

```

[State 200, 2]
;Estado de HitDef
type = HitDef
trigger1 = AnimElem = 3
;Será executado o elemento 3
da animação
attr = S, NA
;Tipo de ataque. S (em pé) e NA
(ataque normal)
damage = 40
;Subtrai 40 unidades da barra de
energia do oponente
animtype = hard
guardflag = M
;A defesa é Média (M)
hitflag = MAF
hitsound = s0,3
sparkno = s730
guard.sparkno = s740
priority = 3, Hit
;Prioridade. Ele ganha de golpes
inferiores aos golpes de valor 3
pausetime = 6,8
sparkxy = 0,-76
;Local em que serão mostra-
dos os sparks

```



```

ground.type = High
;Tipo de pancada
ground.slidetime = 5
ground.hittime = 11
;Tempo em que o personagem
ficará com o golpe
ground.Velocity = -7.5
;A velocidade do golpe
air.Velocity = -2,-3
;A velocidade do golpe no ar

```

```

[State 200, 3]
type = ChangeState
trigger1 = AnimTime = 0
;Mudará o estado quando a ani-
mação terminar
value = 0
;Mudará para o statedef 0
ctrl = 1
;O jogador novamente terá con-
trole sobre o personagem

```

Agora que já criamos a parte do arquivo CNS, salve-o e abra o arquivo fighter.cmd:

Na segunda parte do CMD, quando começar as linhas de comando statedef -1, insira um estado de changestate para quando for pressionado o botão dedesejado, o personagem desferir o golpe.

```

[Statedef -1]
[State -1]
type = ChangeState
;Muda o statedef
value = 200
; para o statedef 200 (soco)
triggerall = command = "z"
;Caso o jogador pressione a tecla z
triggerall = command != "holddown"

```

```

;e não estiver pressionando a seta
direcional para baixo
trigger1 = statetype = S
;quando o state for do tipo stand (em pé)
trigger1 = ctrl = 1
; o jogador novamente retomará o
controle do personagem
Ao terminar de copiar essas linhas

```

de código, o soco deve estar funcionando em seu personagem. Confira abaixo o resultado:



Conforme você pôde notar, neste tutorial procuramos explicar de forma simples e descomplicada a idéia principal da construção de personagens no Mugen. Não tivemos a pretensão de oferecer um curso completo sobre criação de jogos com essa ferramenta, já que as alternativas de criação são infindáveis. Caso tenha gostado do que criou, fique sabendo que isso é apenas o básico no que se refere à criação de jogos. Conforme você for adquirindo conhecimento sobre os comandos dessa ferramenta de criação, você poderá aperfeiçoar os movimentos de seus personagens além de também poder criar cenários para os seus jogos. Para isso, só um conselho, faça uma boa leitura da documentação do Mugen e em breve você estará apto para testar e criar novos golpes, personagens e cenários.

