

Erweiterung eines Spiels mit Hilfe der Skriptsprache Tcl

Jan Hendrik Dithmar

26.01.2006

Motivation

- Mehr Spielspaß durch höhere Flexibilität
- Spiel muss nicht bei jeder Änderung neu gebaut werden
- Add-Ons können leicht erstellt werden

Beispiel

Charakterklasse in einem Rollenspiel:

- Hinzufügen von neuen Charakterklassen
- Abändern von Charakterklassen
- Hinzufügen von neuen Aufgaben für Charakterklassen
- Schnittstelle muss im Spiel vorhanden sein

Beispiel – Code (Auszug)

```
#include <tcl.h>
int ChAdd(ClientData cd,
          Tcl_Interp *interp,
          int argc, char *argv[])
{
    if (argc != 2) return TCL_ERROR;

    interp->result = AddCharacter(argv[1]);

    return TCL_OK;
}
```

Beispiel – Code (Auszug)

```
#include <tcl.h>
int Tcl_AppInit(Tcl_Interp *interp)
{
    if (Tcl_Init(interp) == TCL_ERROR)
        return TCL_ERROR;

    Tcl_CreateCommand(interp, "chadd",
        ChAdd,
        (ClientData) NULL,
        (Tcl_cmdDeleteProc *) NULL);

    return TCL_OK;
}
```

Beispiel – Ausgabe

```
$ cc -o gameshell GameAppInit.c -ltcl -lm
```

```
$ ./gameshell
```

```
% chadd Doctor
```

```
Character "Doctor" added. Please edit the  
attributes of the new class.
```

```
% chadd Doctor
```

```
Character "Doctor" already added. Maybe you  
want to change the attributes of the  
class.
```

```
% _
```