

```
// GAME.CPP
#include <stdio.h>
#include <dos.h>
#include <ctype.h>
#include "game.h"

int ende=0;

int STEIN::movetest()
{
    if ((dx==0) && (dy==0))
        return 4; // keine Bewegung

    if (bild->in(x+dx,y+dy))
    {
        if (bild->get(x+dx,y+dy)==-1)
            return 0; // leeres Feld
        else
            return 1; // besetztes Feld
    }
    else
        return 2; // Bildrand
}

void STEIN::move()
{
    int index = bild->get(x,y);
    hide();
    bild->del(x,y);
    x+=dx;
    y+=dy;
    bild->set(x,y,index);
    draw();
}

int SPIELER::play()
{
    //Tastaturabfrage
    //setzt Bewegungswunsch
    //und Abschluß
    if (kbhit())
    {
        char c=getch();
        switch (toupper(c))
        {
            case '7': // Abbruch
                ende=1;
                break;
            case '1':
                game->speed=1000;
                break;
            case '2':
                game->speed=500;
                break;
            case '3':
                game->speed=200;
                break;
            case '4':
                game->speed=100;
                break;
            case '5':
                game->speed=50;
                break;
            case '6':
                game->speed=1;
                break;
            case 'S': // nach links
                dx=-1;
                break;
            case 'D': // nach rechts
                dx=1;
                break;
            case 'X': // Halt
                dx=0;
                break;
            case 'F': // Feind
                dx=0;
                break;
            case 'E': // Feuer
                dx=0;
                if (bild->get(x,y-1)==-1)
                {
                    STEIN *k =
                        new KUGEL(game,bild,x,y-1);
                    game->set(k);
                }
            else
                {

```

```

// Stein als getroffen markieren
game->hit(bild->get(x,y-1));
        }
        break;
    }
}

switch (movetest())
{
    case 0: // OK, Stein kann bewegt werden
        move();
        return 1;
    case 1: // Feindberührung, stehen bleiben
        dx=0;
        break;
    case 2: // Spielfeldrand, stehen bleiben
        dx=0;
        break;
    case 4: // keine Bewegung
        break;
}
return 0;
}

int KUGEL::play()
{
    switch (movetest())
    {
        case 0: // unbelegtes Spielfeld
            move(); // Position verschieben
            return 1;
        case 1: // getroffen
            tot=1;
            game->hit(bild->get(x+dx,y+dy));
            return 1;
        case 2: // Außerhalb des Spielfeldes
            tot=1;
            break;
        case 4: // keine Bewegung
            break;
    }
    return 0;
}

int ZIEL::play()
{
    if (ishit())
    {
        STEIN *s;
        s=create_ziel_random(game, bild);
        game->set(s);
        s=create_ziel_random(game, bild);
        game->set(s);
    }
    return -1;
}

STEIN *create_ziel_random(GAME *g, BILD *b)
{
    // erzeugt ein zufälliges Zeichen
    // im Bereich des Spielfeldes
    // außer in der untersten Bildschirmzeile

    STEIN *t =
        new ZIEL(g, b,(rand()%MAXX),
                rand()%(MAXY-1));
    return t;
}

STEIN *create_spieler(GAME *g, BILD *b)
{
    // erzeugt einen Spieler
    // in der untersten Bildschirmzeile
    STEIN *t = new SPIELER(g,b,41);
    return t;
}

GAME::GAME(BILD *b)
{
    // Spielsteinpositionen initialisieren
    bild=b;
    for (int i=0; i<SPIELSTEINE i++)
        stein[i]=NULL;
    // Spieler und Ziel einsetzen
    STEIN *s;
    s = create_ziel_random(this,bild);
    if (!set (s))
        printf(,"**AUS**");
    s = create_spieler(this,bild);

```

```

if (!set (s))
    printf(,"**AUS**");
    speed = SPEED;
}

GAME::~GAME()
{
    // Speicherplatz freigeben
    for (int i=0; i<SPIELSTEINE i++)
        if (stein[i]!=NULL)
            delete stein[i];
}

int GAME::set(STEIN *s)
{
    // In der Liste der Spielsteine vormerken
    for (int i=0; i<SPIELSTEINE i++)
    {
        if (stein[i]==NULL)
        {
            stein[i]=s;
            stein[i]->draw(i);
            bild->set(s);
            return -1;
        }
    }
    return 0;
}

void GAME::play()
{
    while (!ende)
    {
        for (int i=0; i<SPIELSTEINE i++)
        {
            if (stein[i]!=NULL)
            {
                if (stein[i]->istot())
                {
                    delete stein[i];
                    stein[i]=NULL;
                    continue;
                }
                if (stein[i]->ishit())
                {
                    stein[i]->play();
                    delete stein[i];
                    stein[i]=NULL;
                    continue;
                }
                if (stein[i]->play())
                { // Stein wurde bewegt
                }
                else
                {
                }
            }
        }
        delay(speed);
    }
}

void main(void)
{
    directvideo=1;
    wscroll=0;
    BILD *bild = new BILD;
    GAME *game = new GAME(bild);
    game->play();
    delete game;
    delete bild;
}

```

Mit
Computern
irrt man viel genauer.