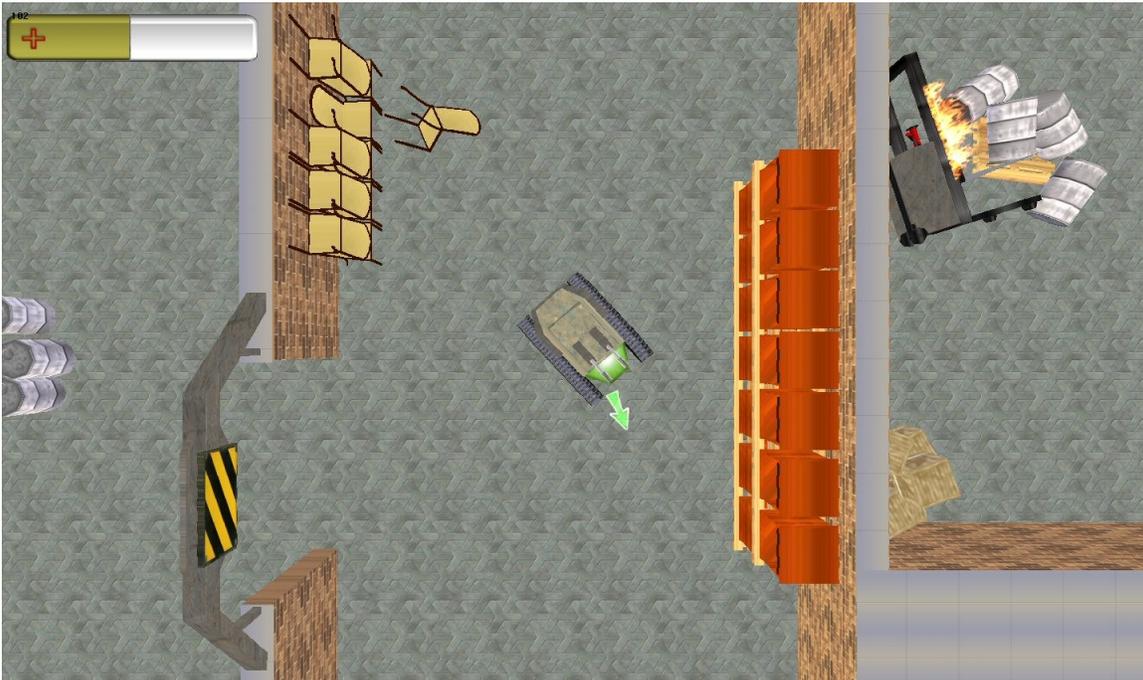


# TREK SOFTWARE

## Tutorial 7 : User Control

This tutorial and the next one are the last big pieces of the puzzle that will help you start making real 3D games.

I've already covered the basics of how to setup a fully 3D world so given enough time and practice you could nearly duplicate a scene like this one from the Trek Software game, Radon: Front Lines (visit [www.TrekSoftware.org](http://www.TrekSoftware.org) if you'd like to see a video of it):



However, one important element missing between making a scene like the above screenshot and turning it into a game is user control. That is one of the main elements that turns your program from simply artwork, into an exciting 3D game. As usual, DarkBASIC makes this very easy to do:

```

autocam off
sync on
sync rate 30

position camera 0,10,0

make object cube 1,3
position object 1,0,0,25

do
  if upkey()=1 then move object 1,1
  if downkey()=1 then move object 1,-1
  if leftkey()=1 then turn object left 1,3
  if rightkey()=1 then turn object right 1,3
  sync
loop

```

The first new command in here is the `upkey()` statement. This returns a 0 if the up arrow key is not being pressed and a 1 if it is. So the first statement says that if the up arrow key is being pressed, then move the object 1 unit forward. Similarly, if the down arrow key is being pressed, I tell it to move the object -1 units forward which is the same as telling it to go 1 unit backward. The `leftkey()` and `rightkey()` statements check the right and left arrow keys and, as you can see, I have turn object left/right commands that take care of turning the box in one direction or another. Go ahead and run the program and have some fun driving around for a bit.

After you play the above program for awhile, you might notice one problem. The box can leave the view of the camera. There are several ways to fix this, depending on the type of game you want. If you want a game where the camera is in a fixed position and simply points at the box wherever it is (which is what I did in Codename Countdown except every now and then I repositioned the camera to a new location) you can do that easily by simply adding this command before the "sync" :

```

point camera object position x(1),object position y(1), object
position z(1)

```

Can you see what it will do? This continually points the camera at the center of the box. Go ahead and add it so you can see how it looks. It may look kind of strange at

this point with no other objects to give us a point of reference but you can probably get an idea for how it works.

On the other hand, suppose you wanted to make a top down racing kind of game. Basically what you would do then is position the camera directly over the object's x and z position while keeping it's y position constant. To give you an example to practice with, here is a mini-racing game where you have to dodge the cones in the road. As of now, there is no collision or scoring – those two things will be mentioned in the next tutorial. Your challenge is to alter the code to make the camera follow the car from a top-down view as described above:

```
autocam off
sync on
sync rate 30

position camera 0,50,20
point camera 0,0,50

make object cube 1,3
position object 1,0,1.5,25

make object plain 2,12,700
position object 2,0,0,350
xrotate object 2,90

ConeZ = 50

for n=3 to 43
  make object cone n,2
  position object n,rnd(10)-5,1,rnd(3)+ConeZ
  ConeZ = ConeZ+15
next n

do
  move object 1,.5
  if leftkey()=1 then turn object left 1,5
  if rightkey()=1 then turn object right 1,5
  sync
loop
```

If you get stuck, feel free to e-mail me at [mail@treksoftware.org](mailto:mail@treksoftware.org)

Written by Trek Software - [www.TrekSoftware.org](http://www.TrekSoftware.org)