

```
1 #include "data/scripts/vars/anim.h"
2 #include "data/scripts/vars/entity.h"
3 #include "data/scripts/com/draw0001.h"
4 #include "data/scripts/vars/frames.h"
5
6 void bind0010(void vEnt, int iIndex, int iX, int iY, int iZ, int iDir, int iFrame, int A){
7
8     /*
9     bind0010
10    Damon Vaughn Caskey
11    12/31/2008
12    Grapple binding. Binds and sets target entity into desired pose, or sets position and releases.
13
14    iIndex:    Entity index. Allows binding of multiple entities.
15    iX, iY, iZ: Location offsets.
16    iDir:      Direction; 0 = no change, 1 = same target, -1 = opposite of target, 2 = right, -2 = left.
17    iFrame:    -1 = Release bind, 0+ Animation frame to set.
18    vEnt:      Anchor entity.
19    int A:     Expansion.
20    */
21
22    void vTarget;
23    float fRatio;
24    int iInv;
25    int iTime;
26
27    if (!vEnt){ vEnt = getlocalvar("self"); } //If no entity passed, use caller.
28
29    fRatio = getentityvar(vEnt, ADSCALER); //Ent's current scale ratio.
30    vTarget = getglobalvar(vEnt + ".bind." + iIndex); //Get entity by bind index.
31
32    if (vTarget) //Make sure there is a legit target.
33    {
34        iTime = openborvariant("elapsed_time"); //Get elapsed time.
35        fRatio = getentityvar(vEnt, ADSCALER); //Get caller's current scale ratio.
36
37        if (iX){ iX = draw0002(fRatio, iX); } //If X bind, apply scaling to fX.
38        if (iY){ iY = draw0002(fRatio, iY); } //If Y bind, apply scaling to fY.
39
40        bindentity(vTarget, vEnt, iX, iZ, iY, iDir, 0); //Execute bind.
41        changeentityproperty(vTarget, "stalltime", iTime + 1000000); //Jack up stalltime to disable default special moves.
42        draw0001(vTarget); //Refresh draw settings.
43
44        if (iFrame < 0) //Frame -1? Then let's release bind.
45        {
46            if (iFrame == -2)
47            {
48                setglobalvar(vEnt + ".bind." + iIndex, NULL()); //Remove record of binding from self.
49
50                setentityvar(vTarget, BOUNDA, NULL()); //Remove record bound to self anchor from target.
51                setentityvar(vTarget, BOUNDI, NULL()); //Remove record bound to self Index from target.
52                setentityvar(vTarget, BOUNDX, NULL()); //Remove record bound to self X from target.
53                setentityvar(vTarget, BOUNDY, NULL()); //Remove record bound to self Y from target.
54                setentityvar(vTarget, BOUNDZ, NULL()); //Remove record bound to self Z from target.
55                setentityvar(vTarget, BOUND, NULL()); //Remove record bound to self direction from target.
56                setentityvar(vTarget, BOUND, NULL()); //Remove record bound to self frame from target.
57            }
58            bindentity(vTarget, NULL()); //Release bind.
59        }
60        else
61        {
62            setglobalvar(vEnt + ".bind." + iIndex, vTarget); //Make record of binding.
63            setentityvar(vTarget, BOUNDA, vEnt); //Make record bound to self anchor on target.
64            setentityvar(vTarget, BOUNDI, iIndex); //Make record bound to self index on target.
65            setentityvar(vTarget, BOUNDX, iX); //Make record bound to self X on target.
66            setentityvar(vTarget, BOUNDY, iY); //Make record bound to self Y on target.
```

```
67     setentityvar(vTarget, BOUNDZ, iZ);           //Make record bound to self Z on target.
68     setentityvar(vTarget, BOUND, iDir);         //Make record bound to self direction on target.
69     setentityvar(vTarget, BOUND, iFrame);       //Make record bound to self frame on target.
70
71     performattack(vTarget, DEFPOSE, 1);         //Set defense pose.
72     changeentityproperty(vTarget, "animpos", iFrame); //Set frame.
73
74 }
75
76 /*
77 Reset engine's lasthit variants. Otherwise, target will assume sweep values when
78 knocked down if last normal attack hit low.
79 */
80 changeopenborvariant("lasthitx", 0);
81 changeopenborvariant("lasthita", 0);
82 changeopenborvariant("lasthitz", 0);
83 changeopenborvariant("lasthitt", 0);
84 }
85 }
86
```