

Cain, Billy

From: Sommers, Andy
Sent: Friday, February 14, 1997 1:36 PM
To: ML WING DESIGN
Cc: ML WING V PROGRAMMERS
Subject: Game Update 2-14-97

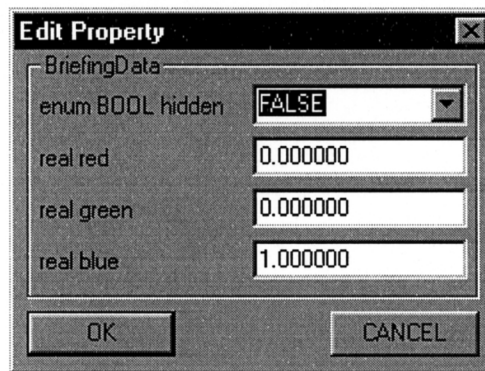
PROGRAMMERS

- 1) add src\game\briefing to your include directory settings
- 2) add src\game\briefobj.cpp to your project
- 3) add src\game\briefanm.cpp to your project

DESIGNERS

The first pass of the briefing system is in. The briefing system uses the "X,Z" view of the MED sector file to place ships for the briefing animation.

Ships and nav points now have briefing data properties :



BriefingData

hidden - whether the object should be visible in the briefing. Defaults to TRUE. If you want an object visible, you have to explicitly make it visible.

red, green, blue - color of the briefing object. rgb values are from 0 to 1 (similar to light sources)

FLO_Main - ships have an additional property known as the briefing program. This unique script is run for that object during the briefing.

BRIEFING PROGRAM COMMANDS

[any of the SF_ commands)

```
void BRIEF_Activate (void);
```

Allow the object to be viewed

```
void BRIEF_Deactivate (void);
```

Disallow the object from bring viewed

```
void BRIEF_Explode (void);
```

Destroy the object (not implemented, will simply deactivate for now)

```
void BRIEF_MoveSeconds ( int x, int y, int seconds );
```

Move the object from it's current position to <x,y> specified and take *seconds* seconds to perform the move

```
void BRIEF_MoveMilliseconds ( int x, int y, int milliseconds );
```

Move the object from it's current position to <x,y> specified and take *milliseconds* milliseconds to perform the move

```
void BRIEF_MoveObjSeconds ( _ObjId o, int seconds );
```

Move the object from it's current position to the location of object *o* and take *seconds* seconds to perform the move

```
void BRIEF_MoveObjMilliseconds ( _ObjId o, int milliseconds );
```

Move the object from it's current position to the location of object *o* and take *milliseconds* milliseconds to perform the move

```
void BRIEF_WaitSeconds ( int seconds );
```

Have the object wait for *seconds* seconds

```
void BRIEF_WaitMilliseconds ( int milliseconds );
```

Have the object wait for *milliseconds* milliseconds

```
void BRIEF_Exit (void);
```

Exit the briefing system

SAMPLE BRIEFING PROGRAM

```
void BriefPlayer ()
{
    // activate the object
    BRIEF_Activate();

    BRIEF_MoveSeconds(4000,0,1); // goto nav 1
    BRIEF_MoveSeconds(0,4000,1); // goto nav 2
    BRIEF_MoveSeconds(-4000,0,1); // goto nav 3
    BRIEF_MoveSeconds(0,-4000,1); // goto nav 4

    // exit the briefing
    BRIEF_Exit();
}
```

NEW MISSION PROGRAM COMMANDS

```
void MS_RunGameflow ( long g );
```

Run gameflow, and have the player start in room *g*.

```
void MS_RunBriefing ();
```

Run the briefing for the first sector listed

```
void MS_RunPADD ();
```

Run the padd (currently not yet supported)

```
void MS_RunSpaceflight ( long s );
```

Run spaceflight for the first sector listed (*s* is now ignored)

SAMPLE MISSION PROGRAM

```
void MissionMain ()
{
    // start the player in room 0
    MS_RunGameflow (0);
}
```

```
// once gameflow finished, run the briefing for this mission
MS_RunBriefing ();
```

```
// run spaceflight for this mission
MS_RunSpaceflight(0)
```

```
}
```

NOTES :

- * Right now, ships are triangles, nav points are circles
- * All nav points are active at the start of the briefing
- * All ships are inactive at the start of the briefing and must be activated with activate object
- * All objects can only perform actions on themselves, for example, you can't have an object "activate" another object
- * There can only be one sector file in the mission now. If there is more than one sector file, only the first sector file is used.
- * Shifted keyboard states appear to be broken. I have moved the cycle nav keys to [N] and [P].
- * sample3.mis contains a sample briefing
- * There is a new command line option -p to be used in conjunction with -m[filename] to run the mission program for a non-series mission.

For example :

```
# run the spaceflight portion of sample (ignoring the mission program)
-msample
```

```
# run the gameflow portion of sample (ignoring the mission program)
-msample & -g
```

```
# run the mission sample (mission program and all!)
-msample & -p
```

```
# run the sample series
-ssample
```

TODO

- * Use splines for movement
- * Take initial orientation into account
- * Camera zooms
- * Better art

INFORM ME IF ANYTHING IS BROKEN.

-ALS