

[Introduction](#)**HAM Tutorial :: Day 10 :: Sprites #3 - Animation #2**[Day 1](#)

GBA Hardware

Today's tutorial is very similar to [Day 8](#), except that we will animate the plane image so that it looks like the propellor is spinning. I started out with the graphics from Day 8 and then manipulated them. You can see what I mean [here](#). For the sake of saving time I decided that the plane will only move up, down, left or right and not in any diagonal direction.

[Day 2](#)

"Hello, World!"

[Day 3](#)

Input

One of the purposes of this tutorial is to give you a basic understanding of timers. Make sure that you understand the *frames* variable and how it is used to animate the propellor. You'll notice it's basically the same as *vbl_count* from [Day 7](#). Also notice how *animcnt* is used - this is very important.

[Day 4](#)Backgrounds -
Bitmapped Modes

To convert the graphic, use the following:

```
gfx2gba -fsrc -m -pbackground.pal -t8 water.bmp
```

[Day 5](#)

Sprites

Again, the sprite will not be in a tile/map format, so type:

```
gfx2gba -D -fsrc -pobject.pal -t8 red_plane_anim.bmp
```

[Day 6](#)Backgrounds -
Tile Modes

Your **gfx** directory should have the following files:

```
background.pal.c  
object.pal.c  
red_plane_anim.raw.c  
water.map.c  
water.raw.c
```

[Day 7](#)Project 1 -
Tetris

On to the code:

[Quiz - Week 1](#)[Day 8](#)Sprites #2 -
Animation

```
// The Main HAM Library
#include <mygba.h>

// Graphics Includes
// gfx2gba -fsrc -m -pbackground.pal -t8 water.bmp
#include "gfx/background.pal.c"
#include "gfx/water.raw.c"
#include "gfx/water.map.c"
// gfx2gba -D -fsrc -pobject.pal -t8 red_plane_anim.bmp
#include "gfx/object.pal.c"
#include "gfx/red_plane_anim.raw.c"

// Defines
#define ANIM_UP      0
#define ANIM_RIGHT  1
#define ANIM_DOWN   2
#define ANIM_LEFT   3

// Global Variables
u32 dir_plane = ANIM_RIGHT; // Direction the plane is facing
u8  plane;                 // Sprite index for the plane
```

[Day 9](#)

Maps

[Day 10](#)Sprites #3 -
Animation #2[Day 11](#)Backgrounds -
Rotation[Day 12](#)Sprites #4 -
Mosaic[Version History](#)

[Poll](#)[Discussions](#)[Graphics FAQ](#)[GFX2GBA Readme](#)[Translations](#)[Downloads](#)[Games](#)[Projects](#)[Credits](#)[Links](#)[Support](#)

```

u32 plane_x      = 110;          // X position of the plane
u32 plane_y      = 50;          // Y position of the plane
u32 animcnt      = 0;           // Current frame of animation
u32 frames       = 0;           // Global frame counter
u32 array_spot   = 0;           // Sprite's current image location

// Function Prototypes
void vbl_func();                // VBL function
void query_buttons();           // Query for user input
void update_plane_gfx();        // Apply plane's new graphic
void update_plane_pos();        // Apply plane's new position

// Function: main()
int main()
{
    // Variables
    map_fragment_info_ptr bg_water; // Background pointer

    // Initialize HAMLlib
    ham_Init();

    // Setup the background mode
    ham_SetBgMode(1);

    // Initialize the palettes
    ham_LoadBGPal((void*)background_Palette,256);
    ham_LoadObjPal((void*)object_Palette,256);

    // Setup the tileset for our image
    ham_bg[0].ti = ham_InitTileSet(
        (void*)water_Tiles,SIZEOF_16BIT(water_Tiles),1,1);

    // Setup the map for our image
    ham_bg[0].mi = ham_InitMapEmptySet(3,0);
    bg_water = ham_InitMapFragment(
        (void*)water_Map,30,20,0,0,30,20,0);
    ham_InsertMapFragment(bg_water,0,0,0);

    // Display the background
    ham_InitBg(0,1,0,0);

    // Setup the array spot
    array_spot = (16384 * dir_plane) + (4096 * animcnt);

    // Setup the plane
    plane = ham_CreateObj(
        (void *)&red_plane_anim_Bitmap[array_spot],
        0,3,OBJ_MODE_NORMAL,1,0,0,0,0,0,plane_x,plane_y);

    // Start the VBL interrupt handler
    ham_StartIntHandler(INT_TYPE_VBL,(void*)vbl_func);

    // Infinite loop to keep the program running
    while(1) {}

```

```

        return EXIT_SUCCESS;
    } // End of main()

// VBL function
void vbl_func()
{
    // Copy plane sprite to hardware
    ham_CopyObjToOAM();

    // Process the following every VBL
    query_buttons(); // Query for user input
    update_plane_gfx(); // Apply plane's new graphic
    update_plane_pos(); // Apply plane's new position

    ++frames; // Increment the frame counter

    return;
} // End of vbl_func()

/*****
// Function: query_buttons()
// Purpose: Query for user input
*****/
void query_buttons()
{
    // UP only
    if(F_CTRLINPUT_UP_PRESSED
        && !F_CTRLINPUT_RIGHT_PRESSED
        && !F_CTRLINPUT_LEFT_PRESSED)
    {
        if (plane_y > 0) --plane_y;
        dir_plane = ANIM_UP;
        return;
    }

    // RIGHT only
    if(F_CTRLINPUT_RIGHT_PRESSED
        && !F_CTRLINPUT_UP_PRESSED
        && !F_CTRLINPUT_DOWN_PRESSED)
    {
        if (plane_x < 176) ++plane_x;
        dir_plane = ANIM_RIGHT;
        return;
    }

    // DOWN only
    if(F_CTRLINPUT_DOWN_PRESSED
        && !F_CTRLINPUT_RIGHT_PRESSED
        && !F_CTRLINPUT_LEFT_PRESSED)
    {
        if (plane_y < 96) ++plane_y;
    }
}

```

```

        dir_plane = ANIM_DOWN;
        return;
    }

    // LEFT only
    if(F_CTRLINPUT_LEFT_PRESSED
        && !F_CTRLINPUT_UP_PRESSED
        && !F_CTRLINPUT_DOWN_PRESSED)
    {
        if (plane_x > 0) --plane_x;
        dir_plane = ANIM_LEFT;
        return;
    }

    return;
} // End of query_buttons()

/*****
// Function: update_plane_gfx()
// Purpose: Apply plane's new graphic
*****/
void update_plane_gfx()
{
    // We'll only update the animation every 5th frame
    if (frames > 5) {
        // Reset the frame counter
        frames = 0;
        // Figure out where to load the image from
        array_spot = (16384 * dir_plane) + (4096 * animcnt);
        // Update it
        ham_UpdateObjGfx(plane,
            (void*)&red_plane_anim_Bitmap[array_spot]);
        // Increment the animation counter
        if (animcnt == 3) {
            animcnt = 0;
        } else {
            animcnt++;
        }
    }

    return;
} // End of update_plane_gfx()

/*****
// Function: update_plane_pos()
// Purpose: Apply plane's new position
*****/
void update_plane_pos()
{
    ham_SetObjX(plane,plane_x);
    ham_SetObjY(plane,plane_y);

    return;
}

```

```
} // End of update_plane_pos()
```

Code Explanation

// Defines ...

This is similar to [Day 8](#). There are fewer directions because it would have taken a while for me to create the extra graphics.

// Global Variables

All of these are basically the same except for *animcnt*. This is used to keep track of the current image (out of 4 possible) for the current plane direction.

// Main function

int main()

The main thing here the following code:

// Setup the array spot

*array_spot = (16384 * dir_plane) + (4096 * animcnt);*

Let me explain the numbers. Remember the plane is a 64x64 pixel graphic and that $64 * 64 = 4096$. Therefore one plane image is an array of 4096 pixels. There are 4 frames per direction (to make the propellor appear to be moving) and $4 * 4096 = 16384$. Huh? Take a look at the following chart.

	Index	Index	Index	Index
Direction	Frame 1	Frame 2	Frame 3	Frame 4
UP	0	4096	8192	12288
RIGHT	16384	20480	24576	28672
DOWN	32768	36864	40960	45056
LEFT	49152	53428	57344	61440

// VBL function

void vbl_func()

This is the same as in previous days. Notice that *frames* is incremented every VBL.

void query_buttons()

This should be pretty easy to understand as well.

void update_plane_gfx()

This is where the animation happens. Every five frames the plane sprite gets updated with the next frame of animation. So, if the plane doesn't change direction, then it takes 20 VBLs to go through one cycle (4 frames * 5 VBLs).

Hmm. I guess that's it for Day 10. Make sure to tinker with the code to see that you really understand it. You may want to try updating the graphic

every 10 or 20 frames (instead of 5) to see what happens (or less to make the propellor spin faster - hint, hint).

Download Code

NOTE: You may need to Right-click and choose Save As.

HAM Version 2.80 And Higher

Download All Files In One Zip: [Day10_Sprites3_Animation2.zip](#)

[View Demo Now](#)

[Discuss Day 10](#)

[<<](#)

[HOME](#)

[>>](#)