



Collision Map Debugging

In 6 easy steps!

Oh no! You're trying to get a collision map working 3 hours before an assignment is due and everything is broken! All is lost!

In times like these, you need to break out your trusty *Collision Map Emergency Bug-Fixing Procedure* (a.k.a. this document!)

Before you talk to a TA about your collision map issue, please go through each of these steps in order! Odds are your issue will be resolved by the end!

If you come to me and ask me about a collision map bug, I'm going to tell you to run through this checklist first for my sanity's sake!

1. Check that the collision checks are actually the problem

Remove all of the collision checks. Does your sprite move as expected (ignoring the collision map?) If not, fix that first!

2. Check your Collision Map in Usenti

Open your collision map in Usenti and check the following:

- Is my palette ordered correctly?
 - Proper palette order is:



Black at the 0th index, then **any number of other colors you want to use (or none if you don't want to make things complicated)**, then white.

- Does my collision map overlay my background properly? In other words, do the top left corners of both of them line up? If not, fix it!
- What is the width and height of my map? Remember this value for later.



3. Check your export settings

GBA Exporter (don't panic!) X

☒ Gfx

bitmap (GBA)

bpp 8

cprs none

trans FF00FF

☐ Map

☒ flat

☒ reduce

☒ pal

☒ flip

☐ sbb

☐ meta-tile pal

☐ affine

cprs none

0 ofs

Meta/Obj

sqr

t

wide

a

ll

1

2

4

8

☐ cstm

nx 1

ny 1

☐ Pal

start 0

num 256

trans 0

Area

☐ custom

☒ as img

☐ as sel

left 0

top 0

width 240

height 160

File

C:\Users\Owner\Desktop\Tiled Demo\Untitled.c

...

type C (*.c)

☒ h file

☐ append

☐ name

☐ riff

Untitled

u8

u16

u32

☐ ext tileset

...

gfx: bitmap (GBA), 8bpp, cprs: none
 area: (0,0)-(240,160) [240, 160]
 obj: -
 map: -
 pal: -
 files: Untitled.h Untitled.c

OK

Cancel

Validate

Make sure you're exporting to the right folder!

Always run the clean task after exporting!

4. Check your MAPWIDTH macro

This needs to match the actual width of your map! If it's 256 wide, MAPWIDTH should be 256. If it's 240 wide, MAPWIDTH needs to be 240.

Always run the clean task after modifying a header file!

5. Check that your sprites are where they should be

Open the Sprite Viewer and check the position of the sprite that isn't following the collision map properly. Are it's col and row correct? If not, your issue likely isn't with your collision map at all! Fix that first, then see if there are other issues!

Is your sprite starting inside of a non-walkable area?



6. Check that you're calling collisionCheck properly

```
int collisionCheck(unsigned char *collisionMap, int mapWidth, int col, int row, int width, int height, int colShift, int rowShift) {
    int smallestSoFar = 256;
    int current;

    // Top Left
    current = collisionMap[OFFSET(col + colShift, row + rowShift, mapWidth)];
    if (current < smallestSoFar) {
        smallestSoFar = current;
    }
    // Top Right
    current = collisionMap[OFFSET(col + width - 1 + colShift, row + rowShift, mapWidth)];
    if (current < smallestSoFar) {
        smallestSoFar = current;
    }
    // Bottom Left
    current = collisionMap[OFFSET(col + colShift, row + height - 1 + rowShift, mapWidth)];
    if (current < smallestSoFar) {
        smallestSoFar = current;
    }
    // Bottom Right
    current = collisionMap[OFFSET(col + width - 1 + colShift, row + height - 1 + rowShift, mapWidth)];
    if (current < smallestSoFar) {
        smallestSoFar = current;
    }

    return smallestSoFar;
}
```

Here are what you should pass in for each of these parameters:

- **collisionMap:** Your collisionMap pointer, which you have cast to a char * near the top of your file.
- **mapWidth:** Your MAPWIDTH macro, which is the actual width of your map.
- **col/row:** The current col/row of the entity you want to check. **Do not add width, rdel, cdel, or anything else to this!** collisionCheck handles all of that, just give it the col and row!
- **width/height:** The width and height of the entity you want to check. **Make sure you've initialized these properly!**
- **colShift/rowShift:** How much you want to move in the col/row direction. **Do not always pass in (cdel, rdel) for these!** If you're moving up, pass in (0, -rdel). If moving right, pass in (cdel, 0)!