Subject: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by devilsclaw on Wed, 25 Jan 2023 16:02:12 GMT
View Forum Message <> Reply to Message

So I found a combination of actions that will cause clipping to return a invalid pointer in RenderPath of BufferPainter.

1) Use Clip and Draw out side of the clip region
2) Attach the object to follow the mouse
3) Repeatedly move the mouse in and out of the window frame at the bottom in random locations up to 30 seconds.

Attached is an example app in code of how to cause it.

Below is the offending code and the output of it getting a corrupted pointer

Y2 is the pointer getting corrupted. Also I am on linux

```
if(clip.GetCount()) {
  printf("DC: Y1 = %i\n", y);
  const ClippingLine& s = clip.Top()[y];
  printf("DC: Y2 = %p\n", &s);
  if(s.IsEmpty()) {
    goto empty;
  }
  printf("DC: Y3 = %i\n", y);
  if(!s.IsFull()) {
    mf.Set(rg, s);
    rf = &mf;
  }
}
```

Results:
DC: Y1 = 525
DC: Y2 = 0x7f82b80571e8
DC: Y1 = 526
DC: Y2 = 0x7f82b80571f0
DC: Y3 = 526
DC: Y1 = 525
DC: Y2 = 0x7f82b80571e8
DC: Y1 = 526
DC: Y2 = 0x7f82b80571f0
DC: Y3 = 526
DC: MouseMove: x 16620 : y 16906 : 00000000
DC: Y1 = 526
DC: Y2 = 0x1070

File Attachments
1) DrawCrash.zip, downloaded 294 times

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by devilsclaw on Wed, 25 Jan 2023 17:13:45 GMT
View Forum Message <> Reply to Message

I also tested it in windows and it just straight crashes the program with out the crash handler like in linux.

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by devilsclaw on Mon, 30 Jan 2023 16:49:38 GMT
View Forum Message <> Reply to Message

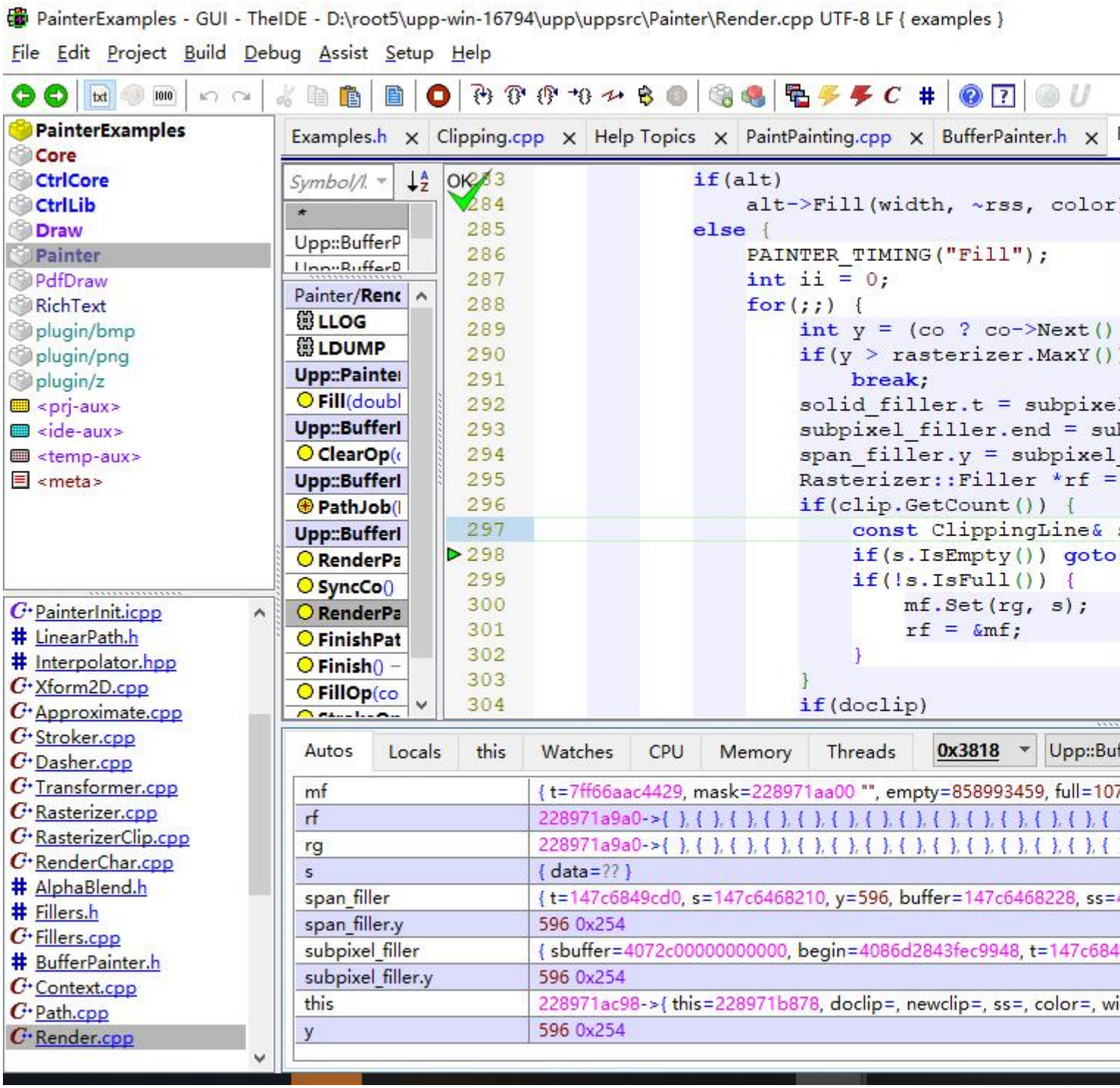
So I am guessing no one has a clue how to fix it or could not confirm the bug ?

I am more of a C programmer and not sure how to handle a situation where the pointer should always be valid because of the object& return type and it not being valid.

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by zouql on Thu, 13 Apr 2023 14:08:36 GMT
View Forum Message <> Reply to Message

hi, all.
I also encountered this exception. If someone has solved this problem, please help.

File Attachments
1) bufferpainter_renderpath.jpg, downloaded 1421 times



Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by devilsclaw on Thu, 13 Apr 2023 14:36:59 GMT
View Forum Message <> Reply to Message

I re-wrote how the mouse handler works by overloading it and it fixed the problem for me

below is sample code of what I did. I made it work more like how the java mouse handling works

```
//Header
enum {
  MOUSE_NONE   = (0 << 0),
  MOUSE_LEFT   = (1 << 1),
  MOUSE_RIGHT  = (1 << 2),
  MOUSE_MIDDLE = (1 << 3),
  MOUSE_DOWN   = (1 << 4),
  MOUSE_UP     = (1 << 5),
  MOUSE_DOUBLE = (1 << 6),
  MOUSE_TRIPLE = (1 << 7),
  MOUSE_DRAG   = (1 << 8),
  MOUSE_MOVE   = (1 << 9),
};

virtual Upp::Image MouseEvent(int event, Upp::Point p, int zdelta, Upp::dword keyflags);
virtual void MouseWheel(Upp::Point p, int zdelta, Upp::dword keyflags);
virtual void MouseClicked(Upp::Point p, Upp::dword keyflags, unsigned int type, int clicks);
virtual void MouseReleased(Upp::Point p, Upp::dword keyflags, unsigned int type);
virtual void MouseDrag(Upp::Point p, Upp::dword keyflags, unsigned int type);
virtual void MouseMove(Upp::Point p, Upp::dword keyflags);
virtual void MouseEnter(Upp::Point p, Upp::dword keyflags);
virtual void MouseLeave();

//CPP
void frm_network::MouseWheel(Upp::Point p, int zdelta, Upp::dword keyflags) {
}

void frm_network::MouseEnter(Upp::Point p, Upp::dword keyflags) {
}

void frm_network::MouseLeave() {
}

void frm_network::MouseDrag(Upp::Point p, Upp::dword keyflags,  unsigned int type) {
}

void frm_network::MouseClicked(Upp::Point pos, Upp::dword keyflags, unsigned int type, int
clicks) {
}

void frm_network::MouseReleased(Upp::Point p, Upp::dword keyflags, unsigned int type) {
}

void frm_network::MouseMove(Upp::Point p, Upp::dword keyflags) {
}
```

```
Upp::Image frm_network::MouseEvent(int event, Upp::Point p, int zdelta, Upp::dword keyflags) {
  int mouse_event = event & ~(LEFT | RIGHT | MIDDLE);
  Upp::dword mouse_flags =  keyflags & (Upp::K_MOUSELEFT | Upp::K_MOUSERIGHT |
Upp::K_MOUSEMIDDLE | Upp::K_MOUSEDOUBLE | Upp::K_MOUSETRIPLE);
  if(mouse_event == REPEAT || mouse_event == CURSORIMAGE) {
    goto exit;
  }

  switch(mouse_event) {
    case MOUSEWHEEL: {
      MouseWheel(p, zdelta, keyflags);
      goto exit;
    }
    case MOUSEENTER: {
      MouseEnter(p, keyflags);
      goto exit;
    }
    case MOUSELEAVE: {
      MouseLeave();
      goto exit;
    }
  }

  if(mouse_event != UP && keyflags == 0) {
    mouse_state = 0;
  }

  if((mouse_state & MOUSE_UP)) {
    mouse_state = 0;
  }

  //Add rejection of other buttons once one is in use
  if((mouse_state & MOUSE_LEFT) != 0 && (keyflags & (Upp::K_MOUSERIGHT |
Upp::K_MOUSEMIDDLE)) != 0) {
    goto exit;
  }
  if((mouse_state & MOUSE_RIGHT) != 0 && (keyflags & (Upp::K_MOUSELEFT |
Upp::K_MOUSEMIDDLE)) != 0) {
    goto exit;
  }
  if((mouse_state & MOUSE_MIDDLE) != 0 && (keyflags & (Upp::K_MOUSERIGHT |
Upp::K_MOUSELEFT)) != 0) {
    goto exit;
  }

  if((keyflags & Upp::K_MOUSELEFT) == Upp::K_MOUSELEFT) {
    mouse_state |= MOUSE_LEFT;
```

```
  } else if((keyflags & Upp::K_MOUSERIGHT) == Upp::K_MOUSERIGHT) {
    mouse_state |= MOUSE_RIGHT;
  } else if((keyflags & Upp::K_MOUSEMIDDLE) == Upp::K_MOUSEMIDDLE) {
    mouse_state |= MOUSE_MIDDLE;
  }

  if((keyflags & Upp::K_MOUSETRIPLE) == Upp::K_MOUSETRIPLE) {
    mouse_state &= ~(MOUSE_DOWN | MOUSE_UP | MOUSE_DOUBLE | MOUSE_MOVE |
MOUSE_DRAG);
    mouse_state |= MOUSE_TRIPLE;
  } else if((keyflags & Upp::K_MOUSEDOUBLE) == Upp::K_MOUSEDOUBLE) {
    mouse_state &= ~(MOUSE_DOWN | MOUSE_UP  | MOUSE_MOVE | MOUSE_DRAG);
    mouse_state |= MOUSE_DOUBLE;
  } else if(mouse_event == DOWN) {
    mouse_state &= ~(MOUSE_UP | MOUSE_MOVE | MOUSE_MOVE | MOUSE_DRAG);
    mouse_state |= MOUSE_DOWN;
  } else if(mouse_event == UP) {
    mouse_state &= ~(MOUSE_DOWN | MOUSE_DOUBLE | MOUSE_TRIPLE | MOUSE_MOVE |
MOUSE_DRAG);
    mouse_state |= MOUSE_UP;
  } else if(mouse_event == DRAG) {
    mouse_state &= ~(MOUSE_DOWN | MOUSE_MOVE | MOUSE_UP);
    mouse_state |= MOUSE_DRAG;
  } else if((mouse_state & MOUSE_DRAG) != MOUSE_DRAG) {
    mouse_state |= MOUSE_MOVE;
  }

  if(mouse_state & MOUSE_DRAG) {
    MouseDrag(p, keyflags, mouse_state & (MOUSE_LEFT | MOUSE_RIGHT |
MOUSE_MIDDLE));
  } else if((mouse_state & MOUSE_DOWN) && (mouse_state & MOUSE_MOVE) == 0) {
    MouseClicked(p, keyflags, mouse_state & (MOUSE_LEFT | MOUSE_RIGHT |
MOUSE_MIDDLE), 1);
  } else if((mouse_state & MOUSE_DOUBLE) && (mouse_state & MOUSE_MOVE) == 0) {
    MouseClicked(p, keyflags, mouse_state & (MOUSE_LEFT | MOUSE_RIGHT |
MOUSE_MIDDLE), 2);
  } else if((mouse_state & MOUSE_TRIPLE) && (mouse_state & MOUSE_MOVE) == 0) {
    MouseClicked(p, keyflags, mouse_state & (MOUSE_LEFT | MOUSE_RIGHT |
MOUSE_MIDDLE), 3);
  } else if(mouse_state & MOUSE_UP) {
    MouseReleased(p, keyflags, mouse_state & (MOUSE_LEFT | MOUSE_RIGHT |
MOUSE_MIDDLE));
  } else if(mouse_state & MOUSE_MOVE) {
    MouseMove(p, keyflags);
  }

exit:
  return Upp::Image::Arrow();
```

}

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by devilsclaw on Thu, 13 Apr 2023 14:39:08 GMT
View Forum Message <> Reply to Message

Yes I know goto statements are frowned on in C++. I come from the linux kernel coding style and embeded programming mostly.

It should be easy enough to remove goto statements.

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by mirek on Thu, 13 Apr 2023 16:16:46 GMT
View Forum Message <> Reply to Message

devilsclaw wrote on Thu, 13 April 2023 16:39Yes I know goto statements are frowned on in C++. I come from the linux kernel coding style and embeded programming mostly.

It should be easy enough to remove goto statements.

goto statements are considered fine in U++, as long as they are the most direct solution to the problem...

However, would it be possible to post complete testcase to save my time? (ideally .zip of whole package)

EDIT: Appologies, I missed it. All is fine now, investigating.

Mirek

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by mirek on Fri, 14 Apr 2023 03:27:46 GMT
View Forum Message <> Reply to Message

Hopefully fixed in master.

Mirek

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by zouql on Sat, 15 Apr 2023 08:16:47 GMT

hi, mirek
The program is working well now, thank you very much.
The clip.Top() may be sometimes empty.

File Attachments

1) render.jpg, downloaded 1375 times

```
284                 alt->Fill(width, ~rss, color);
285             else {
286                 PAINTER_TIMING("Fill");
287                 int ii = 0;
288                 for(;;) {
289                     int y = (co ? co->Next() : ii++) + rasterizer.MinY();
290                     if(y > rasterizer.MaxY())
291                         break;
292                     solid_filler.t = subpixel_filler.t = span_filler.t = (*ip)[y
293                     subpixel_filler.end = subpixel_filler.t + ip->GetWidth();
294                     span_filler.y = subpixel_filler.y = y;
295                     Rasterizer::Filler *rf = rg;
296                     if(clip.GetCount() && clip.Top()) {
297                         const ClippingLine& s = clip.Top()[y];
298                         if(s.IsEmpty()) goto empty;
299                         if(!s.IsFull()) {
300                             mf.Set(rg, s);
301                             rf = &mf;
302                         }
303                     }
304                     if(doclip)
305                         clip_filler.Clear();
306                     rasterizer.Render(y, *rf, j.evenodd);
```

Subject: Re: BufferPainter Clip Crash - Fatal error: Invalid memory access!
Posted by mirek on Sun, 16 Apr 2023 06:02:39 GMT

Correct, that is exactly what I have fixed:

https://github.com/ultimatepp/ultimatepp/commit/a2cefe2b6648 006f308c237649f802dc095a4a6d

---