Subject: Re: BufferPainter::Clear() optimization
Posted by Tom1 on Fri, 15 May 2020 21:13:27 GMT
View Forum Message <> Reply to Message

Hi Mirek,

The game is not over yet, I'm afraid. I did some additional benchmarking with varying buffer lengths to set. It get's more complicated...

```
 RGBA c = Red();

 int bsize=8*1024*1024;
 Buffer<RGBA> b(bsize,(RGBA)Blue());

 String result="\"N\",\"Fill()\",\"memsetd()\",\"MemSet()\"\r\n";
 for(int len=1;len<=bsize;len*=2){
  int maximum=1000000000/len;
  int64 t0=usecs();
  for(int i = 0; i < maximum; i++) Fill(~b, c, len);
  int64 t1=usecs(t0);
  t0=usecs();
  for(int i = 0; i < maximum; i++) memsetd(~b, *(dword*)&(c), len);
  int64 t2=usecs(t0);
  t0=usecs();
  for(int i = 0; i < maximum; i++) MemSet(~b, c, len);
  int64 t3=usecs(t0);
  result.Cat(Format("%d,%f,%f,%f\r\n",len,1000.0*t1/maximum,1000.0*t2/maximum,1000.0*t3/max
imum));
 }

 SaveFile(GetHomeDirFile("Desktop/memset.csv"),result);
```

Now, if you import the resulting memset.csv to your spreadsheet program and create a log-log plot, you will see that the different buffer lengths have a huge impact on the performance of each algorithm. As filling lengths can be quite diverse, I think we need to think about some combination of the different algorithms. Additionally, we need to look at the results on different CPUs. I will keep tinkering on this one for a while here.

(Now I'm running on Core i7 here at home, so this one I can test easily, and also the Core i9 at the office every now and then, as the situation is what it is...)

Best regards,

Tom