Subject: Problem in dasher.cpp and proposed solution
Posted by koldo on Fri, 13 Apr 2012 14:02:11 GMT
View Forum Message <> Reply to Message

Hello Mirek

In dasher.cpp if the line drawn is very long, it enters in an almost endless loop.

```
void Dasher::Line(const Pointf& p)
{
 if(sum == 0) {
  PutLine(p);
  return;
 }
 Pointf v = p - p0;
 double len = Length(v);
 double pos = 0;
 while(pos + rem < len) {
  pos += rem;
  Put(pos / len * v + p0);
  flag = !flag;
  rem = (*pattern)[patterni];
  patterni = (patterni + 1) % pattern->GetCount();
 }
 rem -= len - pos;
 Put(p);
 p0 = p;
}
```

If "len" is very high, the program gets stuck in the while loop.

If you could limit this in any way, by limiting or len or the number of iterations of the while loop, the problem gets solved.

Thank you .

Subject: Re: Problem in dasher.cpp and proposed solution
Posted by mirek on Sat, 14 Apr 2012 09:22:56 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 13 April 2012 10:02Hello Mirek

In dasher.cpp if the line drawn is very long, it enters in an almost endless loop.

```
void Dasher::Line(const Pointf& p)
{
 if(sum == 0) {
```

```
  PutLine(p);
  return;
 }
 Pointf v = p - p0;
 double len = Length(v);
 double pos = 0;
 while(pos + rem < len) {
  pos += rem;
  Put(pos / len * v + p0);
  flag = !flag;
  rem = (*pattern)[patterni];
  patterni = (patterni + 1) % pattern->GetCount();
 }
 rem -= len - pos;
 Put(p);
 p0 = p;
}
```

If "len" is very high, the program gets stuck in the while loop.

If you could limit this in any way, by limiting or len or the number of iterations of the while loop, the problem gets solved.

Thank you .

If I understand it well, this is the same problem as

http://www.ultimatepp.org/redmine/issues/214

correct?

There is no easy solution to this... Perhaps some rough clip estimate before doing stroke?

---

Subject: Re: Problem in dasher.cpp and proposed solution
Posted by koldo on Sat, 14 Apr 2012 11:00:47 GMT
View Forum Message <> Reply to Message

Hello Mirek

Yes. It seems the same problem.

I think in this case a rough solution could be valid.

---