
Subject: Simple way to develop 2D Game
Posted by [Xemuth](#) on Tue, 11 Jun 2019 10:28:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello there,

Me and friends would like to develop a 2D Games.
And we're wondering if Upp include an easy way to develop 2D games with is TopWindowCtrl.

If yes, do you guys have an example more complex than 2DBox examples ?

Thanks in advance

Best regard.

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Tue, 11 Jun 2019 14:02:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Try to take a look at 2D games developed in Adobe Flash. Flash is providing 2d vector graphics. U++ has similar functionality. Take a look at PainterExamples. Flash is using such abstractions as timeline and frames, which you'll need to implement by yourself. The rest is already available. You can start by reading AS3 (Flash programming language) books. They describe such abstractions as Movie and MovieClip, and explain how to create animation.

Hope this helps.

Subject: Re: Simple way to develop 2D Game
Posted by [Xemuth](#) on Tue, 11 Jun 2019 14:26:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Novo,

I will take a look up

Subject: Re: Simple way to develop 2D Game
Posted by [mirek](#) on Tue, 11 Jun 2019 15:33:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

https://store.steampowered.com/app/872700/JetStory_2018/?curator_clanid=4777282&utm_source=SteamDB

is developed in U++

I plan to opensource it soon.

Subject: Re: Simple way to develop 2D Game
Posted by [Xemuth](#) on Tue, 11 Jun 2019 17:02:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Haha Mirek !

when I posted the subject, I was hoping you spoke about Jet-Story and how you did it!

I'm happy you planned to make it opensource
By saying soon how much time you planned ?

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Tue, 11 Jun 2019 18:05:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Tue, 11 June 2019 11:33
https://store.steampowered.com/app/872700/JetStory_2018/?curator_clanid=4777282&utm_source=SteamDB

is developed in U++

I plan to opensource it soon.
I guess plugin/tess2 was added because of this game ...
When are you planning to release a brand-new hardware accelerated version of U++ GUI ?

Subject: Re: Simple way to develop 2D Game
Posted by [mirek](#) on Tue, 11 Jun 2019 21:02:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 11 June 2019 20:05
I guess plugin/tess2 was added because of this game ...
When are you planning to release a brand-new hardware accelerated version of U++ GUI ?

Actually not, but in general I am working on OpenGL accelerated Draw / Painter subset. It is however pretty hard, GPUs are not well suited for this, so the work was postponed. Actually, probably postponed until I find a better alternative to tess2

Mirek

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Wed, 12 Jun 2019 00:20:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Tue, 11 June 2019 17:02 It is however pretty hard, GPUs are not well suited for this, so the work was postponed.
A couple of examples of what can be done.

Check this app. It is a Flash Player with hardware acceleration.
Try to run included examples.
Ctrl-W will show you how tessellation works.
F1 works as well.
It is compiled with MSVS 2008, but, theoretically, should work on any machine ...

ImGui looks interesting as well.

Subject: Re: Simple way to develop 2D Game
Posted by [mirek](#) on Wed, 12 Jun 2019 07:19:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Wed, 12 June 2019 02:20 mirek wrote on Tue, 11 June 2019 17:02 It is however pretty hard, GPUs are not well suited for this, so the work was postponed.
A couple of examples of what can be done.

Check this app. It is a Flash Player with hardware acceleration.

Well, sorry for original poster for being off-topic... Anyway:

It all seems to depend on "this" in above quote. If "this" is defined as "reproduce the Painter polygon semantics with 90% accuracy", then it really is a problem and tessellation is at the heart of it, the other issue being batching / OpenGL state changes.

It simply might be less problematic to render polygon with 100000 vertices in software than to tessellate it, send all vertices to GPU and then render.

Even more trouble: As what I draw is often map polygons, I have even tried the tactics of: tessellate map polygon once, store it in the GPU and then draw from data in GPU. Guess what: This is still slower than software rendering, because by the time OpenGL is finished with state changes to draw the polygon, software renderer is already done with that. So it appears that as long as you are doing `glDrawElements` in equivalent of `Painter::Fill` / `Painter::Stroke`, you have already lost. Means ideally you need to do single `glDrawElements` for the whole rendering, less ideally but still useably single `glDrawElements` per many Fill/Strokes.

That said, I have it in "postponed" state now. For now, I got stuck at "I need faster tessellator than `tess2`, but at the same time I need tessellator that is as accurate as `tess2`". While there are faster tessellators (e.g. <https://github.com/mapbox/earcut.hpp>), they are not as accurate, trivial polygon

examples tend to fail with them. I have tried to implement my own, but the work is still in progress...

Mirek

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Wed, 12 Jun 2019 17:13:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 12 June 2019 03:19

It all seems to depend on "this" in above quote. If "this" is defined as "reproduce the Painter polygon semantics with 90% accuracy", then it really is a problem and tessellation is at the heart of it, the other issue being batching / OpenGL state changes.

"this" is defined as an inspiring example of that 2D graphics on GPU is not just possible, but this was already done ~10 years ago. BTW, tessellation was developed by the author of the Anti-Grain Geometry.

It was a commercial project, but most of the problems you listed were solved.

Everything is possible.

Subject: Re: Simple way to develop 2D Game
Posted by [mirek](#) on Wed, 12 Jun 2019 18:58:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Wed, 12 June 2019 19:13 inspiring example of that 2D graphics on GPU is not just possible

I never argued about that. You can definitely do great things with GPU.

What I argue about is that if you are about to draw very complex polygons with little pixels and draw a lot of them, with external API / semantics similar to Painter, then beating Painter is difficult.

Tessellation was definitely developed many times before. However, the only opensourced code I have really found is

- tess2. Slow.
- earcut. Does not correctly support winding semantics.

If you have any other suggestion...

(BTW, polygon rasterization of Painter is adopted from AGG. I feel no shame about it, because he in turn adopted it from FreeType

Mirek

Subject: Re: Simple way to develop 2D Game
Posted by [Sgifan](#) on Fri, 14 Jun 2019 15:59:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

I know it is not multiplatform, but Direct2D is really powerful and simple to use.

It is as capable as painter, antialiasing quality is perfect.

For instance look at this project to create something like painter using direct2d

<https://github.com/livingcreative/kcanvas>

Look also here: <https://github.com/microsoft/Windows-classic-samples/tree/master/Samples/Win7Samples/multimedia/Direct2D>

to have simple and small examples of using Direct2d in Win32 (not UWP, not WinRT)

I would even go as far as to say it could easily be used by SystemDraw instead of GDI on win32.

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Sat, 15 Jun 2019 16:58:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 12 June 2019 14:58
If you have any other suggestion...

(BTW, polygon rasterization of Painter is adopted from AGG. I feel no shame about it, because he in turn adopted it from FreeType)

Mirek

The only one I'm familiar with is from the author of AGG. Part of the algorithm is patented.
"he in turn" asked an author of FreeType for a permission and got it ...
Tessellator was completely 100% his algorithm and code.

mirek wrote on Wed, 12 June 2019 14:58 What I argue about is that if you are about to draw very complex polygons with little pixels and draw a lot of them, with external API / semantics similar to Painter, then beating Painter is difficult.

Maybe, but "this" app is an example of exactly that.
Most of the GUI code is using semantics of Draw (DrawRect/DrawText/DrawImage).
Painter-related stuff can be rendered into bitmaps in the beginning ...
But even ImGui, which is redrawing everything each frame seems to be able to handle this load.
I'm pretty sure you can do much better than that.
IMHO, with your talent to design very complicated things in a very simple way you should be able to do this easily.
Hardware Abstraction Layer, tessellator, display tree, immutable data structure to pass data between threads ...
It would be interesting to see how simple and elegant will be your design ...

Subject: Re: Simple way to develop 2D Game
Posted by [Xemuth](#) on Sun, 16 Jun 2019 12:42:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Sgfan,

Thanks for your proposition, I will take a look !

Subject: Re: Simple way to develop 2D Game
Posted by [Novo](#) on Mon, 01 Jul 2019 03:22:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

How to Write Your Own C++ Game Engine
This guy was an architect with Ubisoft (one of the biggest game developing companies) ...

Subject: Re: Simple way to develop 2D Game
Posted by [sunitapr](#) on Wed, 11 Sep 2019 05:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

It is helpful thread. Thanks for information. I am also planning to develop 2D game as my project.
I hope this will help me.

Subject: Re: Simple way to develop 2D Game
Posted by [huadong](#) on Wed, 18 Sep 2019 08:20:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

hi Mirek,
where can I get the source code of the jet-story 2018
thanks!

Subject: Re: Simple way to develop 2D Game
Posted by [huadong](#) on Thu, 19 Sep 2019 06:03:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

I find the repository but it's empty
<https://github.com/mirek-fidler/JetStory2018>

Subject: Re: Simple way to develop 2D Game
Posted by [mirek](#) on Sat, 21 Sep 2019 12:11:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

JetStory 2018 now opensourced:

<https://github.com/mirek-fidler/JetStory2018>

Subject: Re: Simple way to develop 2D Game
Posted by [Xemuth](#) on Sat, 21 Sep 2019 14:05:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sat, 21 September 2019 14:11JetStory 2018 now opensourced:

<https://github.com/mirek-fidler/JetStory2018>

Thanks Mirek

Subject: Re: Simple way to develop 2D Game
Posted by [huadong](#) on Sun, 22 Sep 2019 10:25:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Mirek

Subject: Re: Simple way to develop 2D Game
Posted by [Didier](#) on Sun, 22 Sep 2019 12:48:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

I just downloaded and compiled JetStory on my Linux and had a success ... after having to tweak my config.

The compilation of JetStory package uses a lot of RAM (I have only 4Gb) and with no multi-threading (only one hydra thread) it almost uses 4Gb of SWAP to compile (which means it needs at least 8Gb of RAM to compile)

What in the code makes the compiler need so much RAM ? the JetStory package isn't so big even with all the images (37Mb).

The same happens with GCC and CLANG

The game works fine on my (now) old computer

Subject: Re: Simple way to develop 2D Game

Posted by [Novo](#) on Sun, 22 Sep 2019 13:54:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

You can try to compile code with GCC and extra C++ options: -v -ftime-report -fmem-report

You should get detailed info on compiler memory and time usage.

Clang doesn't seem to support -fmem-report.

Subject: Re: Simple way to develop 2D Game

Posted by [mirek](#) on Mon, 23 Sep 2019 07:23:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Sun, 22 September 2019 14:48: What in the code makes the compiler need so much RAM ?

Graphics in .iml files.

Mirek
