Godot tileset sprite sheet

Hello, TLDR: Should I work with Godot import add-ins from Tiled or make a custom Tiled Export Plugin that would make .scene and .tres files (.tmx and .tsx respectively)? This issue was caused by the fact that the existing add-in has performance issues and crashes the editor. For more information about the situation, read below to the end. I use Tiled over the years for small online games mainly with Phaser. For the past 2-3 months I've been playing with and loving Godot 3 so instinctively I went tiled as a tilemap and level editor. There was a problem importing the .tmx and .tsx files in Godot Editor. The existing TMX import to Godot add-in is: . It works, but... It is made some time ago and after importing creates tile sets with one tile - this leads to tile sets with 500+ tiles for the usual 16x16px sprite sheet (704x196 size image), so godot editor crashes during import. This can be avoided by setting the project settings to Godot: message queue -> Max Size Kb-> 102400 (or another high value by default it is 1024), but we all know that this is a solution, not a solution. The existing add-in analyzes .tmx/tsx files and creates scene files and tilemap nodes in single tile mode. At some point, a new Tileset mode was added during the development of the Godot Engine – Atlas mode, which has a much better performance. Currently, if you try to import a tile map through an add-in that has a size greater than 500 x 500 tiles, Godot Editor crashes without error or log message... I think i just ran out of memory. I did some performance tests with Tilemap, produced by Godot Engine with Atlas mode and 2300x1000 tile map, and it won't even curse 60fps. The same experiment with the imported .tmx file and Single Tiles dropped the frame rate to 20fps with a 812x215 tile map (both maps are 16x16px tiles), so surprise - Atlas is faster. The tile editor does a much better job of drawing, layering, and editing maps. Maybe in the future Godot will have all the cool features built in, but at the moment it is not. Creating complex tile maps with multiple layers is complicated. So to solve this problem, I see two paths: I can try editing the Godot import add-in and making changes to import the .tmx file in new Atlas mode. To write an export plugin using the new JS API Tiled 3.2, which will export tilesets to .tres files that Godot uses, and .tmx to .scene files. Both Godot files are text formats. More information on .tscn: So as I have already looked through the Godot plugin code and all xml balance (500+ string code) I am more inclined to make tiled export plugin because it looks simpler and less errors. As I see it, things should be: Export a tile set as .tres with all tiles with their position and collision and a link to the video sprite sheet. .scene export export with tilemap node and tilemap position of each tile and its collision polygon, if such a polygon is available. For each layer shape tiled, a new sub-scene will be created other features can come later one by one when the basics work. Features such as custom features, navigation maps, etc. This topic is announced as being on The Godot Forum/Reddit as well because I really appreciate any recommendations on the 2D mapping and equal building process, and the overall support of both godot and tile communities. 3 Likes hello @Mihail_Ilinov! Thank you for bringing up this topic here. I had no idea that the existing Importer of Godot suffers from such operational problems and glad to hear that you want to do something! Personally, I think it is better to write a Godot exporter than to have an importer's plugin. Whether writing this exporter C++ or JavaScript will depend on what you feel most comfortable with. Either way I would like to be there to help you with the project, and I would be willing to ship a good job of Godot extension along with Tiled. Of course, you may want to take into account the resulting differences in the workflow of the importer and the exporter. How many steps will it take to see the game change after it has been tiled? Anyway, please keep us up to date and let me know if I can help anyone! 1 How hi @bjorn, so it was a while and now I have enough free time to progress with the export plugin task. I have two questions because today: When I console.log (tileset) or tile.log (tileset) write: function (tileset, fileName) { ... } What I see in the console is: Tiled::EditableTileset(0x9dbb40) I tried JSON.stringify but this crashes tiled editor. Also, when I call tileset.image, what is returned is an absolute path ... But as I check the .tsx and .json export path to tileset image is relative and relative path is what I'm looking for too, because Godot uses res:// path, which is the current project root folder, not an absolute path like c:/something/something Except for the two questions above. I managed to export hard encoded .tres and .tscn files that (not surprisingly) works in Godot. And the question about your compared to the import plugin that exists will not make any difference in the steps. After exporting the map or set of tiles, the engine reloads them to fly. If you edit something from Godot Editor and export again... and you overwrite your changes. It's understandable that maybe after all the export work is done I could do a reading function that will analyze and upload .tscn files, but it looks a lot more work and I'm not sure if it really needs to be. Btw if I would get a better debug and code completion if I go c++ way export please tell me. The reason though I use JS more I really want to have an easier to combine And another strange thing that happens. When you export a map or the contents of tiled and even you file valid. I can not upload them to Godot - as the file is not there or not there. I can upload them to Sublime or any other text and everything is fine. Then after I close tiled editor (I think it keeps some link to the file node) I can upload the same files godot without any problems and everything works fine. So the problem is after exporting tiled something to do and must be closed until I can use the exported files Godot. Edit 2: And one more question before I go to bed. I see an example of export scenario cycles trough layer: (var i = 0; i < map.layerCount; ++i) { var layer = map.layerAt(i); if (layer.isTileLayer) { for (y = 0; y < layer.height; ++y) { (x = 0; x < layer.width; ++x) { I try it with an infinite map, and the code above the loops is only positive tiles (with coordinates, e.g. [0,0] [0,1] [1,1] [1,0] and yes one, but all tiles that are [-1,0][0,-1][-1,-1] are not detected and then skipped. So how do I travel on infinite maps? Except that in the example the script exports even empty tiles, so I added the following check: layer.cellAt(x, y).tileId !== -1 So only tiles with reference to the tile set are exported all others are considered empty. Mihail_Ilinov: I have two questions because today: When I console.log (tileset) or tile.log (tileset) write: function(tileset, fileName) { ... } What I see in the console is: Tiled::EditableTileset(0x9dbb40) I tried JSON.stringify but this crashes tiled editor. I'll look at the fact that no accident, but what do you expect to happen in this case? Mihail_Ilinov: Also when I call tileset.image, what is returned is the absolute path ... But as I check the .tsx and .json export path to tileset image is relative and relative path is what I'm looking for too, because Godot uses res:// path, which is the current project root folder, rather than an absolute path like c:/something/something right, the relative paths are determined by an absolute road load and turns into relative paths to save again. We probably need to add a few scripts to the API that helps convert between the two. On the C++ side, we have QDir::relativeFilePath for this. Mihail_Ilinov: Btw if I would get a better debug and code completion if I go c++ way export please tell me. The reason though I use JS more I really want to have an easier to combine environment. If you create C++ you have regular debugging support for your IDE. I personally use Qt Creator, which supports gdb, lldb and CDB debugging, but you should be able to use the Visual Studio IDE as well if you want (see README.md). If C++ is easier for you, then write the C++ add-in all of us (for example, you can pick up existing add-ins). In this case, we should distribute the add-in together with Tiled for binary compatibility reasons (just open the pull query). Mihail_Ilinov: Then after I close editor (I think it keeps some link to the file node) I can upload the same files Godot Godot no problems and everything works well. So, the first thing I wonder is, does your script call file.close() at the end? Mihail_Ilinov: So how do I travel on endless maps? From the script, it seems that there is no effective way to iterate only nonblank segments at this time, but you can repeat the layer content boundary field by using layer.region().boundingRect (this will include content with negative coordinates). Hello, thanks for the quick answers. About JSON. stringify hope to see some JSON with an object and its features or some lines of representation that can indicate me the content. @bjorn: We probably need to add some script API that helps convert between the two. On the C++ side, we have QDir::relativeFilePath for this. I'm waiting for a relative path and now I think I'll try to get a map folder if there's a way and take it off the other absolute path so I can have a relative. Because I don't see any other way to get him to work for Godot. I will try to finish exporting the plugin js, but if C++ offers me more API features well it will not be difficult to choose and be really honest, it will be kinda fun to write some C++ in a fairly controlled environment bjorn: Mihail_Ilinov: Then after I close tiled editor (I think it keeps some link to file node) I can upload the same files Godot without any problems and everything works well. So, the first thing I wonder is, does your script call file.close() at the end? It works now dunno why I thought file.commit also shut it down. It's very nice now cause I don't need to close Tiled after export and Godot detects changes after the scene reboot. So at this point I can export the main map and the main set of tiles, I will try the infinite and maybe until the weekend I will have a beta version that I can share. After that, we could think about converting it to a C++ plugin would be a better way. 2 Likes Mihail_Ilinov: Dunno why I thought file.commit also shut it down. Actually, I think that should, because there is nothing you can do with the file after commit. It's a little silly to require manual closure in this particular case. Also, I just noticed that I didn't call close() to sample papers either. github.com/bjorn/tiled Usually commit () to also close the device unless file recording is disabled. Make sure your device is always closed.... Mihail_Ilinov: I think I'll try to get a map folder if there's a way and take it off the other absolute path so I can have a relative. Yes, you could really do some string manipulation of JavaScript. The asset.fileName property gives you the full absolute file path. bjorn: From the script it seems that there is no effective way to iterate only nonbund pieces at this time, but you can repeat the layer content boundary box by using layer.region().rect (this will include content with negative coordinates). Ok, I just tested it, there is only a small let's correct = = let rect_json = JSON.stringify (correct); console.log (remedy); console.log(rect_json); results: qml: Region(...) qml: {boundingRect: {x:-6,y:-6,width:32,height:20,left:-6,right:25,top:-6,bottom:13}} And the internal property is limited toRect - with x, y and width and height I think it is enough to loop the trough of all tiles. Also, id verification (!== -1) should not impair performance too much. Except that I downloaded and tried to install Qt with Qt Creator, but it hangs on 4% installation I will try again later. Btw I use JetBrains IDE's so I have to check the CLion configuration. Mihail_Ilinov: And the inner property is bordered byRect Right, spell my side, sorry. I edited my post to avoid confusion for new readers. Mihail_Ilinov: Except that I downloaded and tried to install Qt with Qt Creator, but it stuck on a 4% installation I will try again later. Btw I use JetBrains IDE's so I have to check the CLion configuration. Yes, it seems that their servers are currently having problems? I couldn't run an upgrade on my laptop just now either, it was a lot of time and getting stuck as well. It would be nice if you can get CLion to work, but now there is no CMake project file tiled, only qmake and qbs project, and I recommend using Qbs. If CLion can work with Makefiles you can try to generate them using qbs generate -g makefile. I see, I understand. I go with Qt Creator when the installation is complete ... it's 12% after an hour. I made an endless map to work with Godot. And also tried to connect a map, layer or something more complex object that makes the editor crash (something with a console maybe). This makes it kinda difficult to see, for example, what is map.tilesets). Additionally, it doesn't quite get your offer asset.fileName. From which property should I try to get the root folder of the .tmx map? From the map reference? and then I should remove the map.tmx form at the end of the row to get the root folder? Mihail_Ilinov: And also tried to connect a map, layer or something more complex object that makes the editor crash (something with a console maybe). This makes it kinda difficult to see, for example, what is map.tilesets). Try using Object.keys() to test objects and, of course, you can read the script API link, which should be quite detailed. I tried to use JSON.stringify on several objects, add a set of tiles, but I kept getting typeerror: Type error and not crash. If you have a script with which you can recover this crash, open the problem on GitHub. Mihail_Ilinov: From which property should I try to get the root folder of the .tmx map? From the map reference? and then I should remove the map.tmx form at the end of the row to get the root folder? Yes, just find the last/ character and cut the file name. bjorn: I tried to use JSON.stringify in multiple objects, add a set of tiles, but I keep getting TypeError: Type If you have a script with which you can recover this crash, open the problem on GitHub. To check, also just make a new blank map and then export as a custom map format. Here I made an example that you can run and check: github.com/bjorn/tiled Here's a simple example of a script that custom tilemap export: You can check the comments and try different ways to crash ... Except that sometimes some objects, I type the error too. Also what's strange is that even though the map inherits the property map reference I get to write doesn't matter fileName or fileName at all. Both return an empty line in the console.log (map.asset.fileName); console.log (map.fileName); Hello, @bjorn that I could not completely run the code Qt Creator. When I click run it wants an executable and I tried a few things but without success ... I continued to read the Qt documents. Edit: Ok after some trial and error I think I realized my mistake. When installing qt system, I choose MinGW, but only dev tools. Perhaps adding some screenshots to the reading topic/guide will help others get started. Now the debug build is working (although I have some warnings about missing .dll it seems that everything works, and Tiled starts operating after construction and runs in the latest version 1.3.3). Qt Setup: Warnings: I got something more I updated the GitHub problem with a little more information about recording and crashing. Console problem.log or JSON.stringify is a cyclic reference: tileset.tiles[0].tileset.tiles[0].tileset... and so on... More on this: stackoverflow.com javascript, json, google-chrome, google-chrome-extension and problem solving uses this small package: npm super light and fast circular JSON analyzer. Then you can use it as: console.log (--- set of tiles ---); console.log (Flatted.stringify(tileset)); console.log (--- end --- of the tile set); This displays the text mapping of the entire entity in the Editor console. And to my surprise, everything is up and running. So the set of tiles gives everything you need. 2 Likes After a month I'm ready and releasing tiled extensions. It can be downloaded from GitHub: GitHub Tiled plugins used for Tilemaps and Tilesets for export in Godot format - MikeMnD/tiled-to-godot-export Also I made a short video tutorial: versions used in the video are Godot 3.2.1, Tiled 1.3.3 Cheers. 3 Likes These are some amazing work @Mihail_Ilinov! I'm pleased to see if others will pick up this exporter as well and it's a great demonstration of the JavaScript extension system. I put it in the tile extensions repository README. Btw, in his video Tiled ran into a claim failure. It would be great if you know how to reproduce that because it looks like someone has set. It can be a little hard, because it was rather unlinking some tiles ... Thanks for the feedback and quick in GitHub. Yesterday was a bit late and didn't pay much attention to the accident if this happens again I will report what I found. I received the first question today as Leemayn wrote on reddit: 'I messed with him a little bit this morning, and ran into a small problem when exporting tilesets. Godot did not read them properly, and when I looked at the generated tscn file, the region to correct the width and height was set as null or something like that. I tried it with several tile sets. It's 256x256 pixels with 16x16 tiles. Godot will not open the .tres file and an error will occur: The intended float constructor. I manually add width and height using text editor and then Godot opens them without issuing I use Windows 10 and I have the latest stable version of both Tiled and Godot. (I think Tiled 1.3.3 and Godot 3.2.1, but I'm not on my computer at the moment.) And I wrote: Ок. I just downloaded again Tiled 1.3.3 on a different laptop and I can confirm that tileset lacks the following two features: imageWidth and imageHeight. So to solve this problem, we will have to wait until Thorbjørn releases the next version of the patch 1.3.4 - it is more difficult to set qt and create the current version of dev ... but if I would like to wait and put it manually until then @bjorn a quick question is ... Is there a planned release 1.3.4 1 As Mihail_Ilinov: I received the first question today congratulations! Mihail_Ilinov: @bjorn a quick question is... Whether the planned release of 1.3.4 Hmm, in fact Tiled 1.3.4 would only solve a few errors. Tileset.imageWidth and Tileset.imageHeight properties were added again in January, but they will be tiled 1.4. In the meantime, although they are already available for creating snapshots, I would suggest you add a note to README that users should use the latest snapshot. Edit: Eventually I decided to backport these two properties to Tiled 1.3.4, which is now released. 2 Likes Hello! I made this account just to say thank you for this exporter! I've been following your progress since your initial reddit post, you have no idea how useful this will be to my Project 2 Likes likes