“Niet zomaar een project”

vraaggesprek met Gábor Kiss-Vámosi, ontwikkelaar van LittlevGL

De vragen werden gesteld door **Mathias Claußen** en **Jens Nickel** (Elektor)

Tegenwoordig kan geen software worden ontwikkeld zonder open-source bibliotheken en frameworks. Sommige van deze bibliotheken ontstaan wanneer jonge softwareliefhebbers niet-commerciële code ontwikkelen die ze zelf regelmatig nodig hebben. En dan wordt het project omvangrijker en professioneler. Een voorbeeld is de LittlevGL-bibliotheek van Gábor Kiss-Vámosi, die grafische gebruikersinterfaces met aanraakfunctionaliteit zelfs bij 16-bit-µC’s met weinig RAM mogelijk maakt (zie ook elders in dit nummer). Hier vertelt Gábor over de achtergrond, enkele interessante details en de toekomst van zijn populaire bibliotheek.

Elektor: Gábor, waarom zouden we jouw GUI-bibliotheek gebruiken als er al open source-bibliotheken bestaan die gratis kunnen worden gebruikt voor niet-commerciële doeleinden, zoals µGFX?

Gábor: µGFX en LittlevGL hebben een aantal voordelen gemeen, zoals open source en platformonafhankelijkheid, maar LittlevGL kan ook zorgen voor goede animaties, anti-aliasing, transparantie en schaduwen zonder dubbele buffering. Zo is slechts één framebuffer vereist in de displaycontroller of de MCU, en een kleine hoeveelheid grafisch RAM voor LittlevGL. Een ander voordeel is dat LittlevGL ingebouwde ondersteuning heeft voor navigatie en bediening met een toetsenbord of zelfs een encoder.
En tenslotte ondersteunt LittlevGL MicroPython. Je kunt dus Python3-code schrijven om een gebruikersinterface te maken. Daarom kun je de gebruikersinterface in realtime wijzigen zonder de MCU opnieuw op te bouwen en te flashen.

Elektor: Wat is je achtergrond, je beroep? Hoe komt het dat je zoveel tijd in een softwareproject investeert?

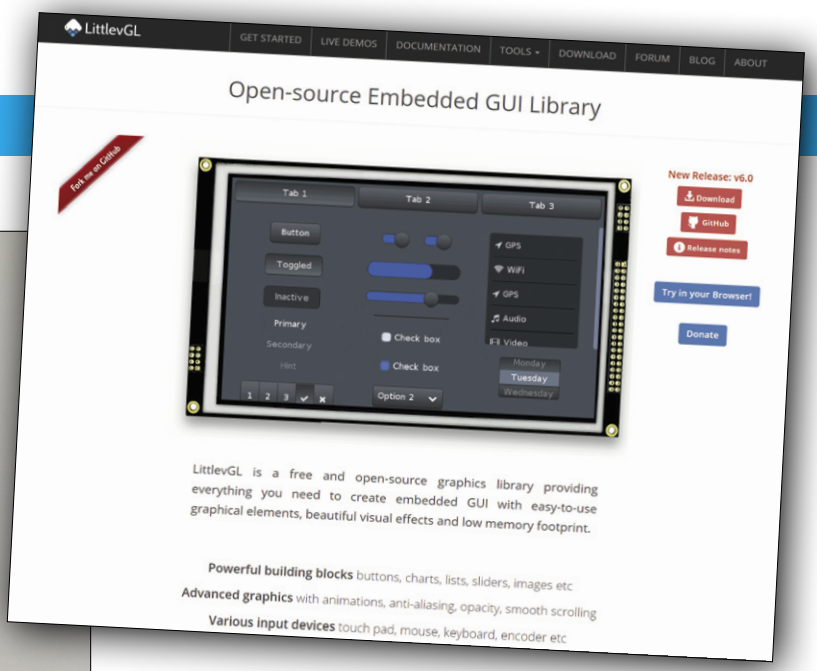
Gábor: Ik ben elektrotechnisch ingenieur en werk nu aan een hardware-beveiligingsproject.
In mijn studietijd hebben één van mijn vrienden en ik in onze vrije tijd veel hobbyprojecten ontwikkeld, zoals versterkers, instrumenten en andere gadgets. We vroegen ons af of het geen gek idee zou zijn om dit spul uit te rusten met een mooi grafisch display in plaats van 7-segment- of 2x16 LC-displays. Ik zei: “Oké, laten we een TFT-scherm kopen. Ik zal proberen er iets op te tekenen. Als ik één pixel kan instellen, moet de rest eenvoudig zijn”. Uiteindelijk duurde het slechts twee uur om een pixel aan te sturen, en daarna heb ik acht jaar besteed aan die ‘eenvoudige rest’!
Eigenlijk mag ik graag nieuwe dingen leren en uitzoeken, dus ik zie het als een hobby. Toen het project openbaar werd, kreeg ik gelukkig veel feedback van andere, meer ervaren ontwikkelaars. Ze stelden voor hoe het beter kon of wat er ontbrak. En ik kan geen nee zeggen tegen een uitdaging. Ik voel een persoonlijke verantwoordelijkheid voor mijn project, omdat mensen het gebruiken en erop vertrouwen.

Elektor: Hoeveel tijd investeer je in het feitelijke schrijven van code? En hoeveel tijd besteed je aan andere werkzaamheden?

Gábor: Het was een paar jaar geleden gemakkelijker, maar nu heb ik een gezin en dus ik moet het juiste evenwicht proberen te vinden. Meestal sta ik vroeg op, als mijn vrouw en kind nog slapen, om e-mails en vragen te beantwoorden en aan nieuwe functies te werken. Als ik ’s middags tijd over heb, probeer ik ook nog iets te doen. Als ik de uren tel, is het naast mijn eigenlijke werk een echte bijbaan. Het is een hele uitdaging (maar niet onmogelijk) om tijd te hebben voor werk, LittlevGL, familie, vrienden en andere vrijetijdsactiviteiten.
Vroeger, toen ik LittlevGL publiceerde, moest ik veel zaken leren die niet met programmeren te maken hebben, zoals marketing, zoekmachine-optimalisatie, web-ontwikkeling, grafisch ontwerpen enzovoort. Ik herinner me dat het maanden duurde om SEO-tips en -trucs te lezen, en ik analyseerde dagenlang websites. Maar het was de moeite waard waard, want als je nu zoekt naar ‘embedded GUI’ zal LittlevGL waarschijnlijk door Google als eerste genoemd worden.

Elektor: Krijgt je op de achtergrond hulp van de community of bedrijven, voor het schrijven van code en andere zaken?

Gábor: Gelukkig zijn er in de loop van de tijd steeds meer mensen bijgekomen die het project op verschillende manieren ondersteunen. Sommigen delen de verbeteringen die ze bedacht hebben, zoals: “Hoi, ik heb een plaatshouder toegevoegd aan het tekstveld-object. Interesse?” Anderen voegen enorme verbeteringen toe; de MicroPython-koppeling is van



Gábor Kiss-Vámosi (29) woont in Boedapest (foto's: Ramóna Erdei).

"amirgon" en de nieuwe font converter is van "puzrin". Anderen helpen bij het beantwoorden van vragen van gebruikers en bij het oplossen van problemen. "Embeddedt" besteedt bijvoorbeeld elke dag uren aan de ondersteuning van gebruikers. Wat ze doen is erg belangrijk en waardevol. LittlevGL zou niet kunnen bestaan zonder hen!

Elektor: Heeft je hulp van de community gekregen voor het porten naar Arduino?
Gábor: Eigenlijk was porten naar Arduino lastig. De structuur van de bibliotheek moest worden gewijzigd om met het build-systeem van Arduino te kunnen werken. Ik heb de eerste wijzigingen zelf doorgevoerd, maar een paar maanden geleden heeft "Pablo2048" het in een nieuw jasje gestoken om een Arduino-pakket uit te brengen; hij helpt ook bij het beantwoorden van vragen die daarmee verband houden. Hij had veel meer ervaring met Arduino dan ik.

Elektor: Gebruikers kunnen geld doneren aan je project. Heb je daar veel steun aan? Of slechts een beetje?
Gábor: De donaties zijn voldoende om de kosten van het project te dekken, zoals een webserver, het kopen van ontwikkelboards enzovoort, maar ik kan er niet van leven.

Elektor: Hoe krijg je nieuwe ideeën voor de bibliotheek? Krijg je feedback van ontwikkelaars?
Gábor: Soms vragen gebruikers expliciet om een bepaalde functie. Daarnaast beschouw ik vakergestelde vragen als iets dat moet worden verbeterd in de bibliotheek of in de documentatie. V6.0 was bijvoorbeeld een conceptuele wijziging in de bibliotheek om voornamelijk veelgestelde vragen en verzoeken op te lossen.
Naast suggesties van gebruikers, probeer ik trends in grafisch ontwerp en embedded UI's te volgen en daar een oplossing voor te bieden.

Ik gebruik de bibliotheek ook zelf. Ik maakte als freelancer verschillende projecten voor bedrijven, maar ik gebruik het nu ook voor mijn professionele werk.

Elektor: Ga je ook naar shows en evenementen? Heb je de kans om persoonlijk met gebruikers van uw bibliotheek te praten?
Gábor: Eerlijk gezegd mis ik het persoonlijke contact met gebruikers. Het zou geweldig zijn om ze persoonlijk te ontmoeten, maar ze zijn verspreid over de hele wereld, wat het niet makkelijk maakt.
Sommigen hebben me al aangeboden om hen te bezoeken. Ze willen ons zelfs hosten als we hun land of stad bezoeken. In feite hebben we onze vakantie van volgend jaar georganiseerd bij een LittlevGL-gebruiker. En ik vind deze kans echt super.
LittlevGL zal op de achtergrond worden veranderd in echt professionele software en niet zomaar een project. Als onderdeel hiervan zijn we van plan deel te nemen aan conferenties, en webinars en seminars te organiseren.

Elektor: Om LittlevGL te gebruiken, zijn 16-, 32- of 64-bit MCU's vereist's nodig. Er is dus geen ondersteuning voor de 8-bit MCU-familie?
Gábor: Theoretisch is er geen probleem met 8-bit MCU's, maar meestal hebben ze zo weinig RAM dat ze LittlevGL niet kunnen uitvoeren. Minimaal 16 kB RAM is nodig om LittlevGL uit te voeren, wat zelfs op 16-bit MCU's krap kan zijn.

Elektor: Kun je ons wat meer vertellen over het technische achtergrond van de bibliotheek? Uit de eerste voorbeelden blijkt dat de UI werkt met schermen/scènes.
Gábor: Het basisidee is om objecten (of widgets) te maken, zoals knoppen, labels, diagrammen, schuifregelaars enzovoort. Je hoedt dan alleen hun eigenschappen zoals grootte, positie, stijlen, waarde of status en call-backs (om de gebruiker tewaarschuwen dat er iets is gebeurd, zoals op een knop geklikt) in te stellen. Zo worden alle tekeningen en andere onderliggende zaken beheerd door LittlevGL en hoeft de gebruiker zich alleen met high-level zaken bezig te houden.
Het object kan in realtime worden aangemaakt en verwijderd. Op deze manier kun je het geheugengebruik optimaliseren door alleen het vereiste object actief te houden. Als je bijvoorbeeld

Webtools voor een softwareproject

Het schrijvan van code is één aspect van een open source-softwareproject, maar je hebt ook documentatie nodig en een manier om de gebruikers te ondersteunen. Dit is wat Gábor Kiss-Vámosi vertelt over de website en andere hulpmiddelen om zijn bibliotheek te ondersteunen:

Als elektrotechnisch ingenieur wist ik niet veel over web-ontwikkeling. Eerst heb ik WordPress geprobeerd, maar toen ik het probeerde aan te passen, kwam ik erachter dat het niet de vrijheid bood die ik nodig had. Later heb ik Bootstrap gevonden en kon ik de website helemaal zelf ontwikkelen. Het was vrij eenvoudig om met Bootstrap werken, zelfs als een beginnende webontwikkelaar, omdat het fraaie elementen heeft en een ingebouwd reactievermogen. Later zijn de bijkomende zaken verplaatst naar apartee subdomeinen en nu werken ze met verschillende engines. De blog wordt gehost op GitHub waar de berichten in markdown worden geschreven, en een engine met de naam Jekyll compileert automatisch een statische website uit de markdown-bestanden. Op deze manier kan iedereen eenvoudig berichten toevoegen aan de blog, er is niet meer nodig dan een pull-request te verzenden.
De documentatie wordt ook gehost op GitHub, maar wordt offline met Sphinx gecompileerd. De documenten kunnen door de gebruikers worden vertaald op een online platform met de naam Zanata. Het forum heeft een Discourse-engine die draait op een DigitalOcean Virtual Server. DigitalOcean heeft een sponsorprogramma voor open source-projecten. Ik heb een aanvraag ingediend en kreeg een jaarlijkse sponsoring die ik kan uitgeven zoals ik wil. Dus begon ik een VPS om het forum te hosten. “seyyah” van GitHub heeft veel geholpen met de VPS-configuratie. Mijn doel was om een gemeenschap op te bouwen waar mensen hun kennis kunnen delen en over hun projecten en ervaringen kunnen praten. Dat is de reden waarom LittlevGL een open blog heeft waar iedereen berichten kan schrijven. En er is een categorie ‘Mijn projecten’ op het forum, waar men interessante creaties kan delen.

een berichtvenster nodig hebt, maak je het gewoon en verwijder je het weer wanneer de gebruiker op de knop “OK” klikt. Het berichtvenster gebruikt alleen geheugen als het zichtbaar is.
De objecten hebben een parent/child-hiërarchie. Je kunt bijvoorbeeld een container (parent) maken en er drie knoppen (children) aan toevoegen. Als je de container verplaatst, bewegen de knoppen mee. Als je de container verwijdert, worden de knoppen ook verwijderd.
Een ander interessant concept van LittlevGL is dat het een beperkt soort objectoriëntatie realiseert. Elk object is afgeleid van het basisobject dat alleen de meest algemene eigenschappen heeft, zoals positie, grootte, parent enzovoort. Dit is vrij gebruikelijk in C++ maar uniek in C.

Elektor: We kunnen diverse prefab-widgets voor de gebruikersinterface gebruiken. Als we iets nodig hebben dat er nog niet is, is er dan een handleiding om deze te maken?
Gábor: Alle widgets functioneren op dezelfde manier. Ze hebben enkele aangepaste gegevens, een functie die de widget tekent en een signaalfunctie die de interne low-leve events afhandelt. Kortom, als je de code van de widget bestudeert, kun je op dezelfde manier je eigen widget maken. Er zijn c- en h-templates om het eenvoudiger te maken om aan de slag te gaan met je eigen widget.

Elektor: Wordt het RAM statisch toegewezen aan de UI-elementen? Of hebben we een soort geheugenpool op een gegeven moment te klein kan worden?
Gábor: LittlevGL heeft een eigen geheugenmanager die dynamisch geheugen toewijst aan de widgets en andere gerelateerde zaken. Hij plaatst gegevens in een array waarvan de grootte kan worden aangepast of die zelfs in het externe geheugen kan worden ondergebracht. In tegenstelling tot de standaard ‘malloc’ en ‘free’ biedt de geheugenmanager van LittlevGL ook monitoring om het gebruik en de fragmentatie van het geheugen te zien.
Dus ja, je kunt zonder geheugen komen te zitten, maar je kunt het geheugengebruik controleren en de geheugenruimte die is gereserveerd voor LittlevGL aanpassen. Als je onvoldoende geheugen hebt, krijg je een foutmelding met een regelnummer; het programma stopt daar. Op die manier kun je eenvoudig achterhalen waar het probleem is opgetreden.

Elektor: Hoe wordt het tekenen beheerd, bestaan er trucs om overlappende meervoudige tekeningen te voorkomen?
Gábor: Een heleboel trucs optimaliseren het tekenen. Op het hoogste tekenniveau houdt LittlevGL bij welke gebieden opnieuw moeten worden getekend (bijvoorbeeld als een knop wordt ingedrukt) en elke paar milliseconden (bijvoorbeeld 30) worden die gebieden opnieuw getekend. Voordat dit gebeurt, zoekt het op de achtergrond welke widget de eerste is die op het opnieuw te tekenen gebied moet komen. Als je de tekst op de knop wijzigt, worden daarom alleen de knop en het label opnieuw getekend, maar de achtergrond onder de knop niet. Maar als je de knop verplaatst, moeten ook de achtergrond, de knop en het label worden getekend. Maar dit alles is eigenlijk triviaal.
Het echt interessante aan LittlevGL is dat het niet rechtstreeks naar het display tekent (zoals veel eenvoudige GUI-bibliotheken doen), noch dubbel buffert (zoals de geavanceerde GUI-biblio-

Ontwikkeling begint in alle vroegte.

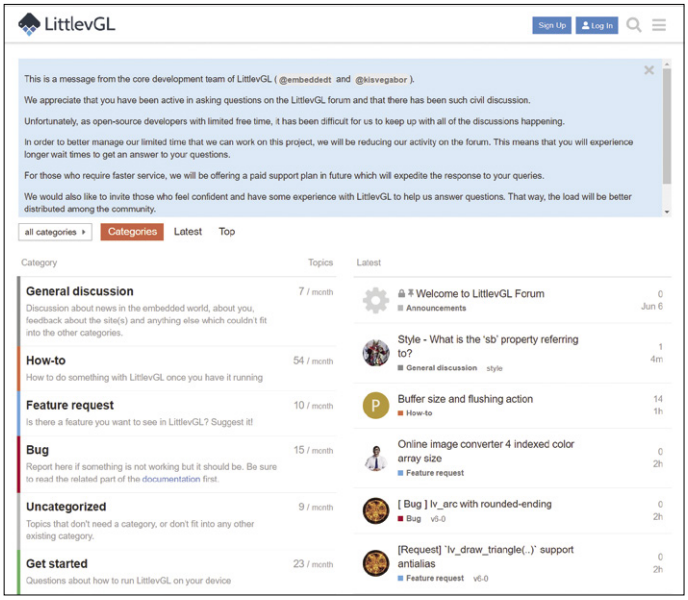
theken), maar met tiles (tegels) werkt. Het gebruikt een kleiner geheugen (typisch 1/10 van het displayformaat) en tekent eerst daar. Wanneer de tekening klaar is, wordt de buffer met de getekende afbeelding naar het display gesstuurd. Omdat het een gebufferde tekening is, treedt er geen flikkering op tijdens het tekenen, waardoor vloeiende animaties mogelijk zijn. In de nieuwste versie van de bibliotheek (die nog dit jaar gepubliceerd zal worden) is de teken-engine compleet herschreven. Dat maakt de engine flexibeler en beter uitbreidbaar, en ook wordt de kwaliteit in het algemeen beter (zoals betere anti-aliasing en schaduwen); maar het belangrijkste doel was het ondersteunen van maskeren. Met het maskeren van de pixels worden ronde hoeken mogelijk of kunnen teksten worden weergegeven met een afbeelding als achtergrond, of een verloopkleur. Maar er zijn ook enkele nieuwe functies toegevoegd, zoals horizontaal verloop, verschoven schaduwen en additieve en subtractieve overvloeimodi.

Elektor: Lettertypen kunnen online op je pagina worden geconverteerd; heb je ook een eenvoudige offline-converter voor lettertypen? En hoe zit het met symbolen?
Gábor: Het lettertype-conversieprogramma is geschreven in Node.js, dus het kan zowel online als offline worden uitgevoerd. We hebben nog geen applicatie voor het converteren van desktop-lettertypen, maar je kunt de converter van GitHub klonen en vanaf de opdrachtregel gebruiken.
Symbolen kunnen ook uit fonts gehaald worden. FontAwesome is een bekend, zeer populair symboollettertype waarvan sommige symbolen standaard in LittlevGL zijn opgenomen. Wanneer je echter je eigen lettertype maakt, kun je meerdere lettertypebestanden kiezen om samen te voegen tot één C-bestand. Bijvoorbeeld, bepaalde sets van Arial en een paar symbolen van FontAwesome.

Elektor: De bibliotheek is niet thread-safe. Zijn er plannen om RTOS-ondersteuning aan je bibliotheek toe te voegen?
Gábor: Klopt, de bibliotheek is niet standaard thread-safe, maar het is niet moeilijk om hem thread-safe te maken. Je hoeft slechts een mutex (mutual exclusion) te gebruiken wanneer je LittlevGL-gerelateerde functies aanroept en deze op het juiste moment vrij te geven.

Elektor: LittlevGL heeft een mooie op Eclipse gebaseerde omgeving voor snelle UI-ontwikkeling zonder het gedoe om de code naar echte hardware te flashen. Komt er in de toekomst een soort GUI Designer die de ontwikkelingsfase nog meer versnelt?
Gábor: Zeker weten. Daar is al verschillende keren om gevraagd. Ik was erg blij toen ik zag dat sommige gebruikers spontaan begonnen zijn met GUI Designers. Er zijn drie onafhankelijke ontwikkelingen die tot nu toe allemaal grote vooruitgang hebben geboekt. Het is mogelijk dat één daarvan de officiële GUI Designer van LittlevGL wordt.

Elektor: LittlevGL draait momenteel onder de MIT-licentie, die een eenvoudige integratie in commerciële producten mogelijk maakt. Uit de laatste online enquête die je hebt gehouden bleek dat er vragen waren over commerciële licentiemogelijkheden. Wat zal er in de toekomst veranderen?



Gebruikersondersteuning is een groot deel van het werk. Op het forum beantwoorden ervaren bibliotheekgebruikers vragen.

Gábor: De populariteit van LittlevGL neemt razendsnel toe. Een zeer groot bedrijf heeft interesse getoond in LittlevGL, maar als individu zijn de onderhandelingen erg moeilijk. De volgende stap is om een bedrijf op te richten dat de belangen van LittlevGL beter kan vertegenwoordigen dan een individu. Naarmate de community groeit, wordt het moeilijker om de ontwikkeling, ondersteuning en marketing als vrijetijdsactiviteiten te blijven doen. Het doel is om een businessmodel voor LittlevGL te vinden dat een salaris kan bieden aan sommige mensen die fulltime aan LittlevGL werken, maar dat ook voor de gebruikers nog acceptabel zal zijn.
Betaalde ondersteuning met snelle bugfixes en toegewijde professionele hulp en een soort licentie wordt ook overwogen. We proberen nog steeds het juiste bedrijfsmodel te vinden.

Elektor: Wat zou je jongeren willen aanraden die interessante open source-softwareprojecten ontwikkelen?
Gábor: Mijn advies is om een plan en visie te hebben wanneer je een open source-project publiceert of start. Gaat het om een kleine tool die niet echt onderhouden hoeft te worden? Of om iets groots dat jarenlang veel tijd zal vergen? Wat is het doel van het project? Waarom wil je het doen? Voor de lol? Voor de roem? Om ervaring op te doen? Of gewoon om anderen te helpen? Je moet deze vragen van tevoren kunnen beantwoorden en dienovereenkomstig beslissingen nemen. Je kunt gefrustreerd raken als dingen je gewoon overkomen en je niet weet hoe je moet beslissen. Zeg gewoon nee als iets niet is zoals je wilt en verwelkom kansen die je in de juiste richting helpen. Zoals het spreekwoord gaat: “Er waait geen goede wind voor een zeeman die niet weet waar de haven is.” ⏮

(190353-04)

Weblink

[1] LittlevGL-startpagina: https://littlevgl.com/