

組み込み機器で3D GUIを効率的に開発できる グラフィックスライブラリ RIANT

RIANT Graphics Library to Improve Efficiency of Development of 3D GUIs on Embedded Devices

深井 祐介 アルマン ジリエ 中西 晃
■ FUKAI Yusuke ■ Armand GIRIER ■ NAKANISHI Akira

携帯電話やデジタルテレビ (TV) など組み込み機器へのGPU (Graphics Processing Unit) 搭載が進み、美しいグラフィックス表示やアニメーション効果を持つGUI (Graphical User Interface) を備えた製品が実現されている。しかしGPUの操作は非常に複雑であり、効率的にGUIを開発することは難しかった。

東芝は、GPUを搭載した組み込み機器向けに、簡単な操作で2D・3D (2次元・3次元) アニメーション効果を付加できるグラフィックスライブラリ “RIANT (ライアント)” を開発した。既存の2D GUI部品をGPU機能で描画することにより、2D GUIツールキットの機能性と移植性を確保しながら3D描画機能と描画速度を向上させ、効率的な3D GUI開発を可能にした。

Embedded devices equipped with graphics processing unit (GPU) chips, such as smartphones, digital televisions, and so on, increasingly feature graphical user interfaces (GUIs) that offer attractive graphics and animation effects. However, it is difficult to efficiently develop such GUIs due to the complex processes involved.

As a solution to this issue, Toshiba has developed the RIANT graphics library for embedded devices to support the development of GPUs, which makes it possible to add two-dimensional (2D) and three-dimensional (3D) animation effects without difficulty. RIANT can improve 3D drawing capabilities and speed by drawing existing 2D widgets through GPU functions, while also maintaining the functionality and portability of 2D GUI toolkits.

1 まえがき

近年、パソコン (PC) だけでなく組み込み機器へもGPU (Graphics Processing Unit) が搭載され、スマートフォンや、タブレットPC、デジタルTVなどで美しいグラフィックス表示やアニメーション効果を持つGUI (Graphical User Interface) が実現されている。

組み込み機器でGPUを操作するAPI (Application Programming Interface) としては、Khronosグループ^(注1)によってOpenGL[®] ES^(注2)が業界標準化され、対応するLSIが数多く製品化されている。従来はハードウェアへの依存性が高かったグラフィックスドライバの操作がこれによって共通化され、容易に導入できるようになった。

しかし、OpenGL ESが提供するAPIはもっとも基本的な機能だけであり、自由度が高い反面、GUIで使用するようなボタンや四角形、画像など各種のオブジェクトから構成される画面を効率的に開発することが難しい。また、オブジェクトの移動、拡大と縮小、及び回転には行列計算が必要になり、更にアニメーション効果を付加する場合にも専門的な知識が必要

になる^{(1), (2)}。

東芝は、このような問題を解決するため、OpenGL ESをベースとしたGPUの機能を使いやすくし、効率的にGUIアプリケーション (以下、GUIと略記) を開発できるグラフィックスライブラリ “RIANT (ライアント)” を開発した。ここでは、RIANTの概要とその効果について述べる。

2 グラフィックスライブラリ RIANT

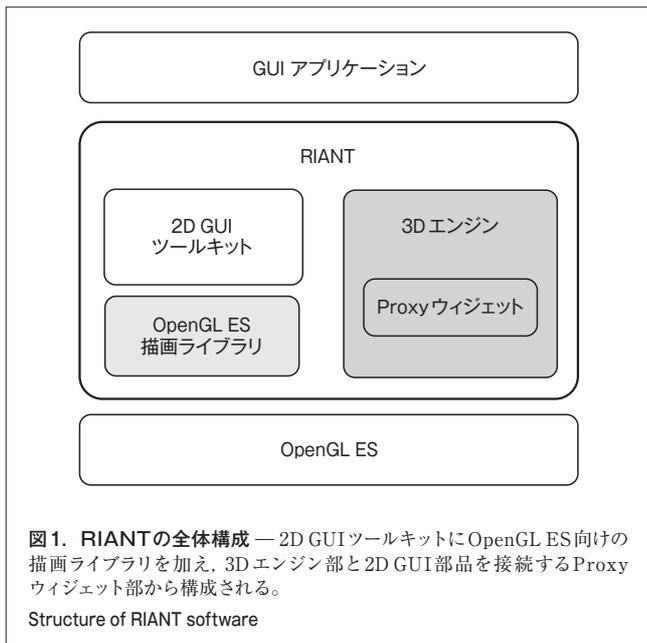
これまでの一般的な組み込み機器のグラフィックスライブラリでは、線や四角形を描く、画像を貼る、文字を描く、などの必要最低限の2Dグラフィックス機能だけが提供されていた。そのため、複雑な図形を描いたり、アニメーション効果を付加する場合には、非常に大きな開発コストがかかっていた。

RIANTの開発では、GPUの機能を使用して美しいグラフィックス表示やアニメーション効果を実現するだけでなく、より効率的にGUIを開発できるようにすることを目指した。そこで、GUIの開発効率を向上させるため、既存の2D GUIツールキットの機能を流用することにした。更に、3Dアニメーション効果を実現する3D APIを提供し、2D GUIツールキットが備えるGUI部品に適用する方式とした。

RIANTの全体構成を図1に示す。また、これらの設計方針をまとめると以下のとおりである。

(注1) 情報技術分野の様々な企業が参加し、オープンなAPIの策定を目指しているコンソーシアム。

(注2) ロイヤリティフリーで、クロスプラットフォームなグラフィックスAPIであり、デスクトップ向けAPIであるOpenGLのサブセット。OpenGL、OpenGL ESは、Silicon Graphics, Inc.あるいはその子会社の米国及びその他の国における商標あるいは登録商標。



- (1) OpenGL ESをベースとしたGPUの機能を使用し、GUIを高機能・高速化
- (2) 既存の2D GUIツールキットの機能を活用し、GUI開発効率を向上
- (3) 使いやすい3D APIを提供し、アニメーション効果を容易に実現

2.1 OpenGL ES描画ライブラリ

2D GUIツールキットは、一般に、プラットフォーム間の差異を吸収するため2D描画インタフェースがブリッジパターンとして実装され、各種プラットフォーム向けの描画ライブラリ及びラップ^(注3)が用意されている。これにより、例えばWindows^(注4)で作成したアプリケーションがそのままLinux^(注5)でも動作するように移植性を確保している。

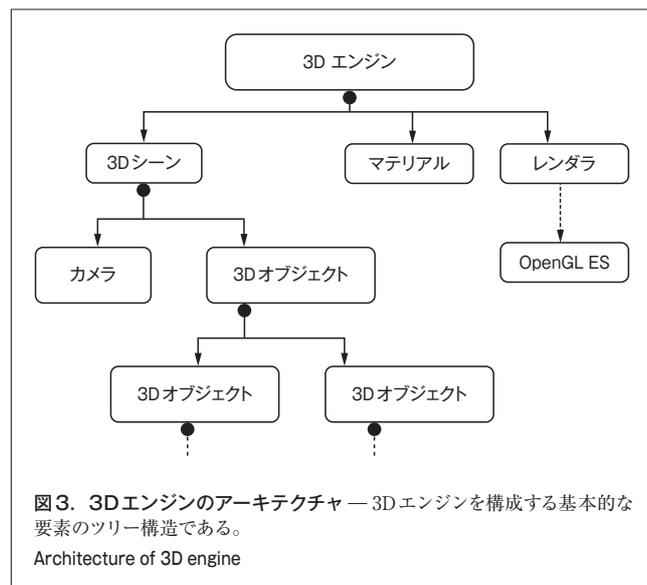
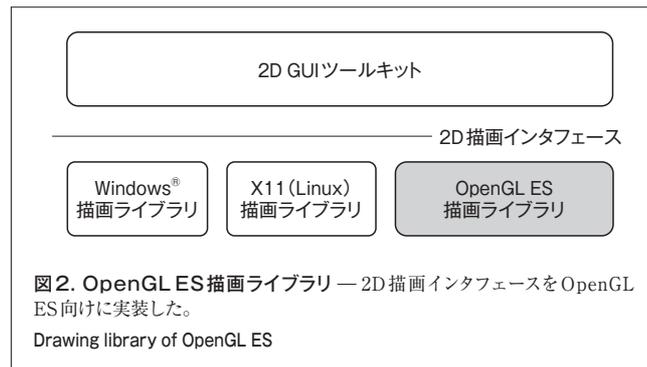
RIANTは、既存の2D GUIツールキットを流用するために、この2D描画インタフェースに従い、OpenGL ES向けの描画ライブラリを用意した(図2)。これにより、2D GUIツールキットが提供するGUI部品をそのままOpenGL ESで描画できるようになった。

現在は当社製品に搭載された2D GUIツールキット向けに実装しているが、2D描画インタフェースを持つ任意の2D GUIツールキットにも大きな変更なしに適用可能である。

2.2 3Dエンジン

RIANTは、OpenGL ESをベースとしたGPU機能を利用するための3Dエンジンを内蔵している。ゲームエンジン⁽³⁾など

(注3) ある関数やデータが提供する機能を、別の形で提供するプログラム。
 (注4) Windowsは、米国Microsoft Corporationの米国及びその他の国における商標又は登録商標。
 (注5) Linuxは、Linus Torvalds氏の米国及びその他の国における登録商標。

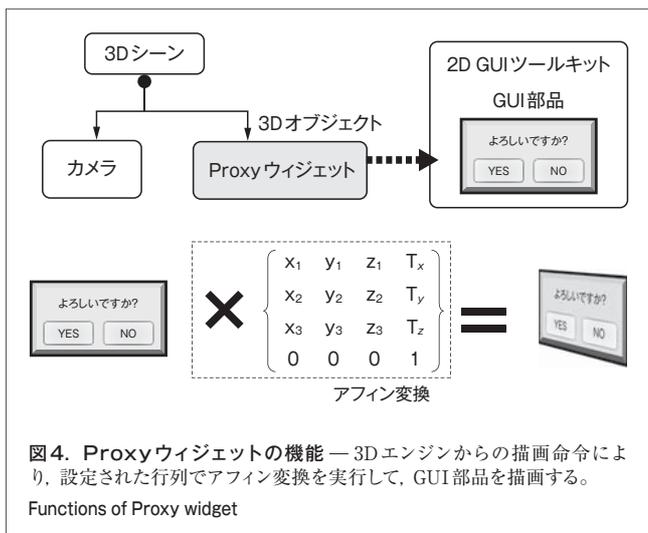


で用いられる一般的な3Dエンジンを構成する基本的な要素を図3に示す。

一般的な3Dエンジンと同様に、RIANT内部では、描画のメインループ関数が定期的呼び出される。例えば、30 fps (フレーム/s) と設定すれば33 ms周期で呼び出される。図3の3Dシーンにツリー構造で構築された3Dオブジェクトは、それぞれが自分自身の3D空間上の頂点データと、色や画像などのマテリアル情報へのポインタを持っている。これらの3Dオブジェクトが、カメラから見た2D平面上に投影されて、レンダラ^(注6)によって描画される。

また、RIANTはこれらを容易に操作するC言語ベースの3D APIを提供する^(注7)。OpenGL ESでは、行列情報やGPUプロセッサへの命令であるシェーダプログラムを直接記述する必要があるが、これらを隠蔽して抽象度の高い、使いやすいAPIとしている。3D APIは、次に示すように五つの主要なカテゴリーに分類される。

(注6) データや、パラメータ、描画ツールなどを処理して、具体的な画素を得るためのソフトウェア。
 (注7) RIANTの3D APIは、Java言語などの他の言語からも呼び出せるように、C言語の関数としている。Javaは、米国Oracle Corporationあるいはその関連会社の登録商標。

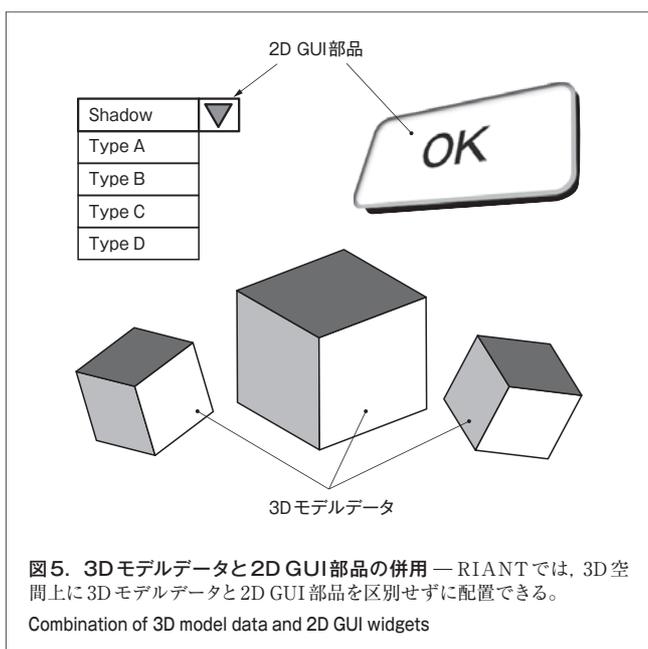


- (1) 3次元空間の位置情報を操作
- (2) 行列情報を生成及び設定
- (3) アニメーション情報を生成及び設定
- (4) 影やぼかしなどの描画効果を生成及び設定
- (5) 共通的に使われるデータ構造などを提供

2.3 Proxyウィジェット

Proxyウィジェットの導入により、既存の2D GUIツールキットのGUI部品を、3Dエンジンの描画対象である3Dオブジェクトとして関連付けることができる。

3Dエンジンから定期的呼び出される描画命令は、Proxyウィジェットを介してGUI部品に転送され、OpenGL ES向けの描画ライブラリによりOpenGL ESの命令に変換されて描画される。また、3D APIにより設定された行列情報はProxy



ウィジェット内に保持されており、描画時にアフィン変換が適用されて、GUI部品が3D空間上に描画される(図4)。

2.4 3Dモデルデータの描画

RIANTは、2D GUIツールキットが提供するGUI部品を描画する機能に加え、3Dモデリングツールで作成した3Dモデルデータをインポートして描画する機能も提供している。

一般に、3Dモデリングツールは各種のフォーマットで出力する機能を搭載しており、OpenGLと同様にKhronosグループが管理しているCOLLADA™(注8)フォーマットであれば、RIANTに入力できる。RIANTが提供する3Dモデルデータ描画機能を使用することにより、前述の2D GUI部品と同様の手順で、3D API操作を行うことができる(図5)。

3 GUI開発環境

組込み機器において開発効率が向上しない原因の一つに、実機が必要であることが挙げられる。プロジェクト開始初期はPC用エミュレータなどを用意してあっても、昨今の組込み機器の開発速度に追いつかず、十分にメンテナンスされない結果、使用されなくなる例も多い。RIANTは、下位描画機能をOpenGL ESのAPIだけに依存しているため、PC上で容易に実行できることが特長の一つである。

RIANTはPC用と組込み機器用のmakeファイル及びそのテンプレートを備えており、コンパイルスイッチでターゲットを切り換えることで、ソースコードを変更することなくそれぞれの実行ファイルを生成できる。

また、キー入力や、アプリケーション起動の開始と終了、メッセージ通知などのRIANTアプリケーション用のインタフェースを用意し、これらをポーティングレイヤとしてGUIやRIANT本体とは隔離している。ターゲット機器を変更する場合は、ポーティングレイヤ部だけを実装すればよく、GUIやRIANT本体のソースコードを変更する必要はない(図6)。

このように、PC版RIANTを使用することにより、実機がない状況でもGUI開発を進めることが可能である。例えば、新しいハードウェアを開発中である、試作機を大量に生産する予算がない、オフショアの開発リソースを利用するなど、従来は実機がないことでGUI開発が停滞していた要因を取り除くことができる。

また、GUIの単体検査をPC上でを行い、実機が完成し次第、結合検査を実機上で開始することができる。GUIの品質が低いとその他の機能の結合検査にも大きな影響を及ぼすため、あらかじめGUIの品質を十分に確保できることは、組込み機器の開発効率を高めるうえで非常に有効である。

(注8) COLLADAは、Khronos Group Inc.の商標。

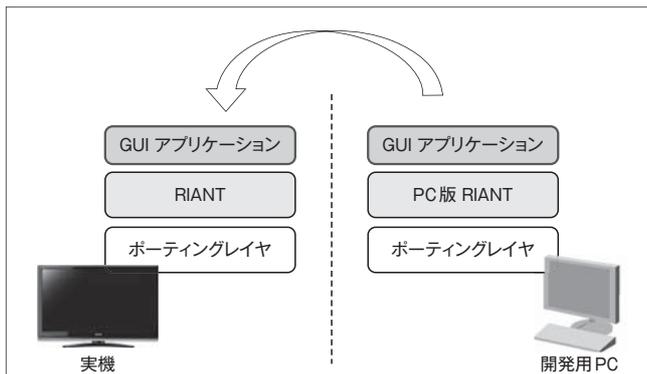


図6. RIANTによるGUI開発環境 — ターゲット機器を変更する場合、ポーティングレイヤとOpenGL ESを用意すれば、GUIやRIANTを変更せずにPCや組込み機器で実行できる。
GUI development environment applying RIANT

4 RIANT導入の効果

2011年に製品化された、当社の地上・BS(放送衛星)・110度CS(通信衛星) デジタルハイビジョン液晶TV レグザ(REGZA) Z2/Z3シリーズでは、GPU機能を備えた“レグザエンジンCEVO™”と呼ばれるシステムLSIを搭載した。RIANTはこれらの製品群に搭載され、GUI描画を高速化して、アニメーション効果を付加することにより、直観的な新感覚のメニュー画面である“レグザメニュー”を実現した(図7)。

レグザメニューの開発では、GUIのプロトタイプ設計から製品向けGUIの実装、検査までの工程で、PC版RIANTを使用した。新しいLSIを備えたハードウェアが存在しない状況でも、PC上で先行して開発することにより、従来よりも大幅に短い期間で製品化することが可能になった。



図7. 液晶TV レグザ Z2/Z3シリーズに搭載したレグザメニュー — GPU機能を活用し、軽快な操作感を実現した。
REGZA Menu display of REGZA Z2/Z3 series

5 あとがき

ここでは、従来の2D GUIツールキットの開発効率を維持しながら、GPU機能により高機能化、高速化することで、3D GUIを効率的に実現するRIANTの概要と、RIANTを液晶TV レグザZ2/Z3シリーズのレグザメニューの開発に適用した例について述べた。

今後は、RIANTを活用したGUIを継続的に開発するとともに、GPUの機能をより引き出せるようにRIANTの機能強化を進めていく。

文献

- (1) Foley, J. D. 他. コンピュータグラフィックス理論と実践. 東京, オーム社, 2001, 1189p.
- (2) アフタブ ムシン 他. OpenGL ES 2.0 プログラミングガイド. 東京, ピアソン桐原, 2009, 480p.
- (3) ジェイソン グレゴリー 他. ゲームエンジン・アーキテクチャ. 東京, ソフトバンク クリエイティブ, 2010, 884p.



深井 祐介 FUKAI Yusuke

デジタルプロダクツ&サービス社 コアテクノロジーセンター エンベディッドシステム技術開発部主務。組込み機器向けGUIのプラットフォーム開発に従事。
Core Technology Center



アルマン ジリエ Armand GIRIER

デジタルプロダクツ&サービス社 コアテクノロジーセンター エンベディッドシステム技術開発部。組込み機器向けGUIのプラットフォーム開発に従事。
Core Technology Center



中西 晃 NAKANISHI Akira

デジタルプロダクツ&サービス社 コアテクノロジーセンター エンベディッドシステム技術開発部主務。組込み機器向けGUIのプラットフォーム開発に従事。
Core Technology Center