

DirectX 勉強会 第4回

内容

- 頂点バッファ
 - サンプルプログラム: [VertexBuffer.cpp](#)
- 入力(DirectInputについて)
 - サンプルプログラム: [Dinput.cpp](#)

頂点バッファ

- 描画以前に用意しておく頂点リストのようなもの
- 頂点が変わらないオブジェクトを描画する場合に有効

頂点バッファの準備

- LPDIRECT3DVERTEXBUFFER8 vb[6]
とすると、頂点バッファが6つできる
- `VertexBuffer`では、`PrepareVB()`という関数を作って、vb[6]に頂点の情報を設定している
- アプリケーションの終了時に、頂点バッファを開放する必要がある。
`VertexBuffer`では、`ReleaseVB()`という関数で行っている

頂点バッファを使った描画

- **SetStreamSource()**
使用する頂点バッファを指定する
 - 第2引数 使用する頂点バッファへのポインタ
 - 第3引数 使用する頂点バッファの大きさ
- **DrawPrimitive()**
描画する
 - 第1引数 ポリゴンの描画法
(D3DPT_TRIANGLEFAN等)
 - 第2引数 3角形ポリゴンの数

入力

- 入力を取得する方法には、いくつか種類がある
- 今回は、DirectInputを利用した方法を説明する
 - マウス、キーボードなどの入力装置を統一的な方法で管理できる
- DirectInputを使う為に
 - プログラムに次の一行を加える
`#include <dinput.h>`
 - プロジェクト->プロパティ->リンカ->入力
->追加の依存ファイル に次を加える
`dinput8.lib`

DirectInputの初期化

- CreateDevice()

デバイスの取得

- 第1引数 入力デバイスの指定
 - GUID_SysKeyboard : キーボード
 - GUID_SysMouse : マウス など

- SetDataFormat()

デバイスからのデータのフォーマットを設定

- 第1引数 データ形式を記述する構造体へのポインタ
 - c_dfDIKeyboard : キーボード用(256バイトの配列)
 - c_dfDIMouse : マウス用 など

入力の取得

- `GetDeviceState()`

デバイスの状態を取得する

- 第1引数 取得するデータサイズ
- 第2引数 データ格納先のポインタ
(キーボードの場合は、
256バイトの配列)

キーボードからの入力

- 256バイトの配列に、0番から256番までのキーボード・デバイス定数の状態が入っている
- キーが押されると、最上位ビットが1になる
- キー入力の判定の仕方

```
diKeyState : キー入力を受け取る256バイトの配列
if (diKeyState[DIK_RETURN] & 0x80) {
    リターンキーが押されている
}
```

キーボード・デバイス定数

DIK_SPACE	スペースキー
DIK_A~DIK_Z	AからZまで
DIK_0~DIK_9	メインキーボードの1から9まで
DIK_RETURN	リターンキー

- 上記以外にもある

<参考>

http://www.geocities.jp/toru_website/di/list1.html

サンプルプログラム

Dinput.cpp

- 入力

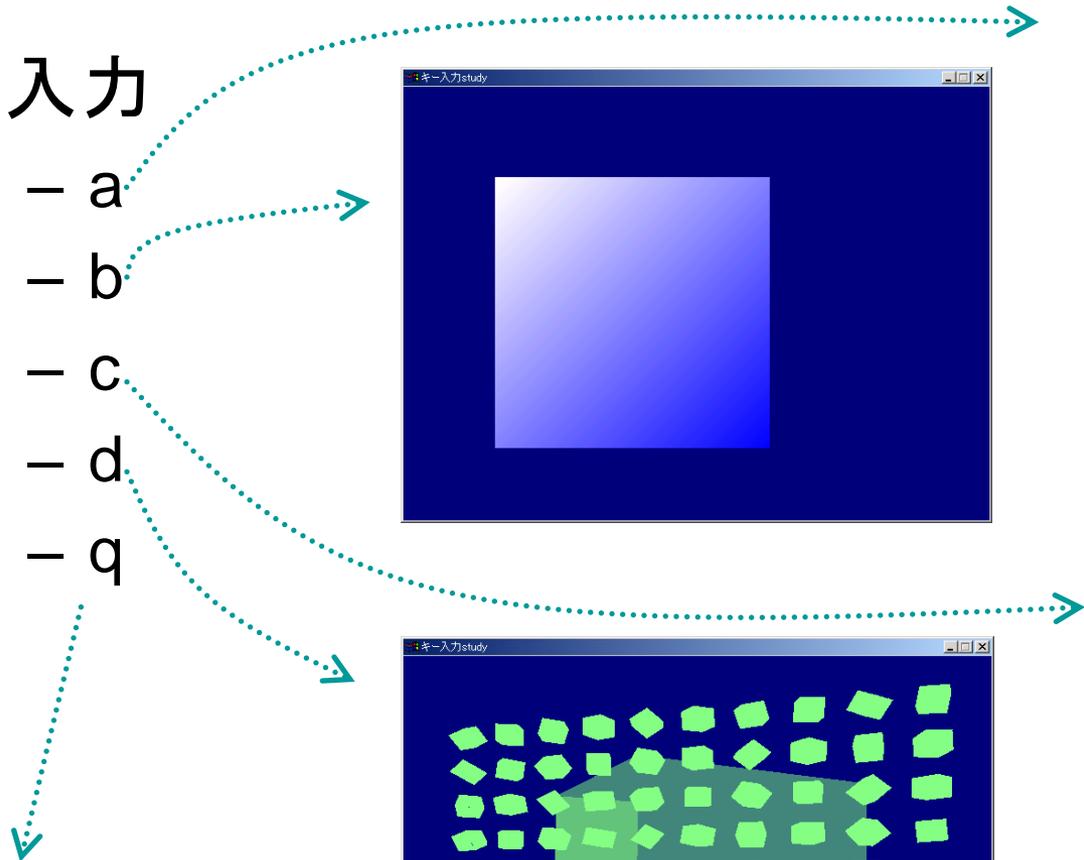
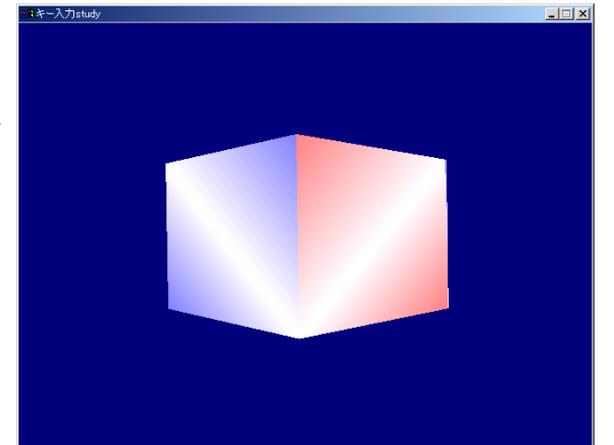
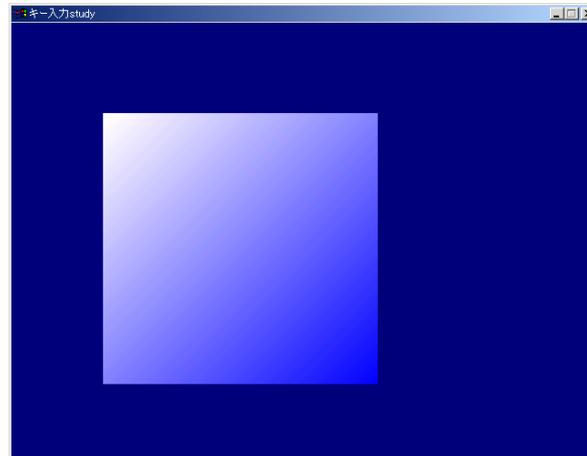
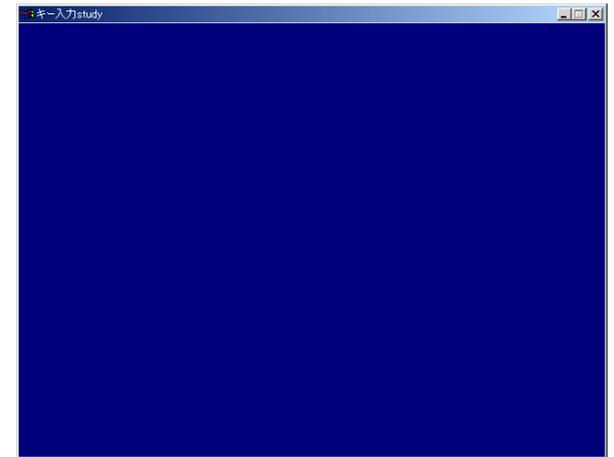
- a

- b

- c

- d

- q



参考

- 土井淳 大澤文孝 成田拓郎 著
DirectX8 実践プログラミング
工学社 2003
- 登大遊 著
DirectX8.0 3Dアクションゲーム・プログラミング
工学社 2002
- http://www.geocities.jp/toru_website/index.html
ToruのDirectXプログラミング講座