

# ソフトウェア設計及び実験

## 第5回レポート

6119019056 山口力也

2019/05/21 日提出

### 1 原理

#### 1.1 SDL

SDL とは (Simple DirectMedia Layer) の略であり, マルチメディアを簡単かつ直接に扱うためのライブラリである. 特徴としては

- グラフィックス, サウンド, ジョイスティック, スレッド, タイマなどの機能を扱うことができ, ゲーム開発に多く用いられる
- Linux や Windows, MacOS など主要な OS で使用可能
- C 言語だけでなく, C++, Java, Perl といった多くのプログラミング言語にも対応している

などが挙げられる.

#### 1.2 レンダラー

レンダラーとはレンタリングコンテキストのことであり, ウィンドウに 1 対 1 で対応づけられて生成される. ウィンドウに何かを描画する場合はレンダラーを生成することが不可欠である.

#### 1.3 テクスチャ

ウィンドウ内に文字を描画する場合, 画像を文字化してウィンドウ内に描画する. しかしレンダラーには画像を直接描画することはできない. そのためテクスチャを生成し, テクスチャに対して画像を描画する必要がある. テクスチャは画像を描画する対象であり, VRAM 上で GPU により高速かつ柔軟な描画処理を行うことができる. また, 1 つのレンダラーに対して複数のテクスチャを生成することができる.

## 1.4 サーフェイス

サーフェイスは画像を描画する対象であり, サーフェイスの描画データはそのまま表示することができない. 画像ファイルを読み込む場合はサーフェイスからテクスチャに描画データを転送し, テクスチャからレンダラーに描画データを転送することでウィンドウに画像を表示することができる.

## 1.5 ソフトウェアにおける CUI と GUI

CUI(Character User Interface) と GUI(Graphical User Interface) は以下のような特徴がある.

- CUI
  - － 利点:複雑な処理をまとめて実行できる
  - － 欠点:コマンドを覚える必要がある
- GUI
  - － 利点:アイコンやボタンの採用により直感的で専門家でなくてもわかりやすい
  - － 欠点:特になし

認知工学に基づいた UI 設計を行うことで”使ってみたい”と思わせることができる. そのためには分かりやすく, かつ利便性の高い GUI などが不可欠である.

# 2 課題内容

## 2.1 ダイレクトマニピュレーション

ウィンドウの生成と表示, 図形描画, マウス操作によるイベント駆動型プログラミング習得を目的として, 次の要件を満たすプログラムを作成せよ.

- ウィンドウに円と四角を描画する
- マウス (カーソル) の現在座標をウィンドウ上に表示する
- 円をマウスでドラッグアンドドロップできる
- 円を四角の中にドロップすると”Good job!”という文字列をウィンドウの中央付近に表示する
- プログラム終了条件を入れるなど, 自分なりのアレンジを加えても良い

ダイレクトマニピュレーションとは,Human Machine Interaction(HMI) をデザインとした設計スタイルのことであり, コンピューターのような仲介者を介さずに自発的にタスクを実行することを促進する行動を特徴とする. 対話型システムのように操作が容易で誰にとってもわかりやすいシステムのことをいう.[1]

## 2.2 アフォーダンスを考慮した GUI 設計

画像描画, 文字描画, キーボード操作によるイベント駆動型プログラミングなどの習得を目的として, 次の要件を満たすプログラムを作成せよ.

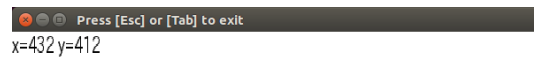
- ウィンドウの上部にアフォーダンスに満ちたボタン 3 つを描画する. ボタンは図形描画または画像で作成すること.
- 3 つのボタンに文字を描画する. 文字内容やフォントは各自で決める
- キーボードの”←”, ”→”キーを押すことで, 3 つのボタンから 1 つ選べるようにする. その際どのボタンが現在選択されているかわかるような視覚的效果を加えること.
- ”Enter/Return”キーを押すことで, 選択中のボタンに対応して以下の処理を行う.
  - － ウィンドウ下部に画像を描画する.
  - － 自分の好きな処理を行う.
  - － プログラムを終了する.

アフォーダンスとは”環境や物は元から様々な使い方をアフォード (提供) しており, 人や動物はその使い方をピックアップ (受取る) する”というものである. 例えば, 何も付いていない板だけのドアがあった場合, 人はドアを開けることができれば中に入ることが可能だと知覚できるが, どのようにしてドアを開ければいいのかを瞬時に判断できない. しかし, ドアに平らな板が貼ってあれば押すことによって, ノブが付いていればノブを回して押すか引くかすることによって, ドアを開けることができると知覚しやすくなる. このように, ドアに分かりやすい特徴を持たせることによって, 説明なしで誰でも使えるようになる.[2]

## 3 課題結果

### 3.1 ダイレクトマニピュレーション

2.1 節で与えられた課題の結果を以下図 1 に示す.



Press [Esc] or [Tab] to exit  
x=432 y=412

Good Job!



図 1: 課題 1 の結果

x 座標と y 座標を表示し、円と四角を表示することはできた。また四角の中に円が入ると”Good Job!”と表示することもできたが、円をドラッグアンドドロップして移動する部分ができずに、クリックで移動する円になってしまい不完全な状態になった。

### 3.2 アフォーダンスを考慮した GUI 設計

2.2 節で与えられた課題の結果を以下図 2 に示す。



図 2: 課題 2 の結果

好きな処理はウィンドウの色を変える動作にした。

## 4 考察

### 4.1 課題 1 について

課題 1 についてはイベント処理の挙動をしっかりと知ることが大切だと思われる。私ははじめ  $x$  座標と  $y$  座標の表示の部分でレンダラーに表示する部分の書き方を間違えて、何度も  $x$  座標と  $y$  座標の表示が何度も重なってしまい、左上の部分が真っ黒になっていた。また全体的に処理が早いいためイベント駆動型プログラムの処理の順序や構造を理解し使えることが重要であると思われる。

### 4.2 課題 2 について

課題 2 については大規模なプログラムになるので変数名の定義などが重要だと思われる。SDL では Surface と Texture と Renderer という順序をふむため、適当な変数を定義してしまうとどの Texture にどの Surface のデータを転送するかわからなくなってしまう。実際に自分もそうだったためあとから定義しなおし整理した。

## 5 感想

全体的に時間が少なく満足のいくプログラムを書くことができなかった。スライドや授業での解説は丁寧で分かりやすかったが、単純に覚える量が多すぎてキャパオーバーといった感じになってしまいかなり混乱した。私は編入生でシステム実験もあるためなかなか時間が取れなかったのも原因だと思う。次回からはある程度予習などもしていきたいと思う。

## 参考文献

- [1] B.Shneiderman(1992),”Designing the User Interface: Strategies for Effective Human-Computer Interaction – 2nd Edition”,1992
- [2] アフォーダンスとは,<https://boxil.jp/mag/a3059/>