

ソフトウェア設計及び実験

第6回レポート

6119019056 山口力也

2019/05/28 日提出

1 原理

1.1 サウンド処理

SDL2では, サウンドデータの種類の種類に Music 型と Chunk 型の二つがある. 以下にそれぞれの特徴を示す.

1.1.1 Music 型

Music 型には以下の特徴がある

- サウンドファイルのデータをメモリに全て読み込まず, ファイルからデコードしながら再生
- 同時に複数の Music 型サウンドは再生できない
 - － BGM(ある程度長いサウンド) 再生に用いられる
- Mix_Music 構造体 (不透明構造体のためメンバは不明)

1.1.2 Chunk 型

Chunk 型には以下の特徴がある.

- サウンドデータをメモリに読み込んで再生
 - － 容量の小さいサウンドファイルを対象とすることが多い
- 複数のサウンドを同時に再生できる
 - － 効果音 (即音声の高いサウンド再生) に用いられる
- Mix_Chunk 構造体

1.2 アニメーション処理

SDL でアニメーション処理を行う時は, いわゆるパラパラ漫画の要領で行う. 複数の画像を用意し, 一定時間間隔で次々に表示することによってアニメーション処理を実現できる. しかしながら, 一定時間間隔で呼ぶ時, while ループの中で描画関数を呼び出すと, 性能の低い CPU やプロセスが多く走っている場合アニメーションがゆっくりとなる可能性がある. また, Delay 関数を入れると, その間に他の処理を行うことができない. その解決策はタイマー割り込みである. OS により割り込みのタイミングが管理されているため, 上記で述べた問題を解決できる. SDL2 でのアニメーション処理の流れは以下の通りである.

1. キャラクタ画像や背景画像をサーフェイスに読み込む (IMG_Load 関数など)
2. タイマーを設定, 起動させる (SDL_AddTimer 関数)
3. イベントループ (無限ループ) に入る
4. 設定した時間間隔でコールバック関数が呼び出される
5. 描画すべきキャラクタの動作パターンに対応する画像領域を抽出する
6. 抽出した領域の画像データを描画対象 (サーフェイスやレンダラー) に転送する
7. 描画対象 (レンダラー) をディスプレイに出力する
8. 現在または次に抽出する画像領域を変数に格納する
9. コールバック関数からイベントループに戻る

2 課題内容

オリジナルのアニメーションを作成せよ.

3 課題の結果

以下図 1, 図 2, 図 3, 図 4 にアニメーションの結果画像を示す.



図 1: 課題 1 の出力画像の一例

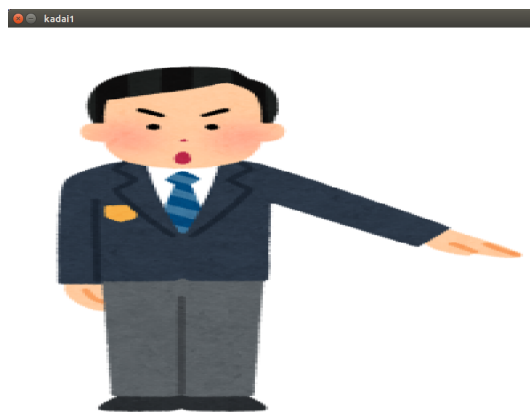


図 2: 課題 1 の出力画像の一例

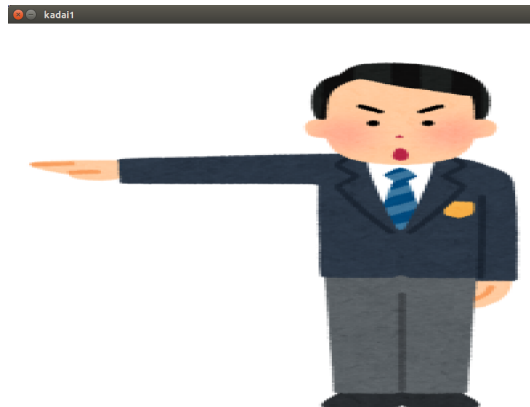


図 3: 課題 1 の出力画像の一例

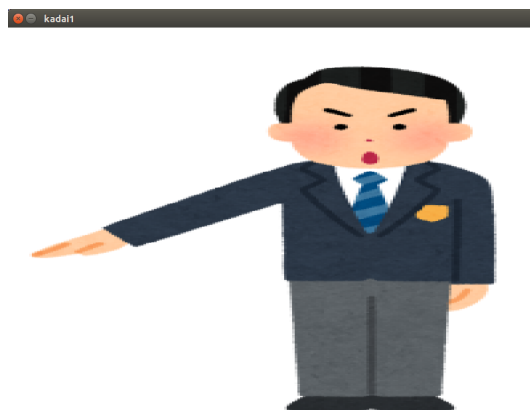


図 4: 課題 1 の出力画像の一例

4 工夫した点

アニメーションをただ見るだけではつまらないので,BGMを付け加えた.BGMは2種類用意し,それぞれキーボードの”1”キーと”2”キーで切り替えられるようにした.BGMとしてはTELL(KATAMORI)とPerfect10(Unknown Brain,Heather Sommer)を用いた.TELLはFreeDownloadであったためCreative Commonsに従って使用する分には著作権には問題ないようであった.Perfect 10はNCS(No Copyright Sounds)であるためこちらも著作権には問題ないと思われる.

5 考察

アニメーションを行うときに画像をずらしていくが、ピクセルが少しでもずれていると描画対象がうまく写らない場合があった。そのため作成する画像の大きさ (ピクセル) には十分気をつけるべきだと思われる。また、アニメーションはタイマ割り込みの中で行われているため、呼び出す間隔には気をつけるべきだと思われる。実際タイマ割り込みの間隔を 1ms にすると処理が重すぎて PC が落ちてしまった。適当なタイミングでタイマ割り込みを行うことも大事な要素の一つだと思われる。

6 感想

今回の実験は前回に比べると簡単なものであったが、学んだことには、ゲームに必要な要素が多くあり、これから行うゲーム開発に欠かせない知識だと感じた。