

近未来マルチメディア 2007

ビデオモザイク班 開発プロジェクト報告書

Project: 実時間動画分割絵遊戯

情報学類 200611248 中山 空星
2008/02/14

INDEX

- プロジェクト概要 ... 2
- 開発経緯
 - 目的 ... 2
 - 動機 ... 2
 - 経過 ... 3
- アプリケーション説明 ... 5
- システム構成概略 ... 6
- 先行研究・関連技術 ... 7
- 開発結果報告 ... 8
- 考察反省 ... 8

● プロジェクト概要

このプロジェクト「実時間動画分割絵遊戯」は、身近なマルチメディアツールでの手軽なエンターテインメントコンテンツの実現として、ウェブカメラから取得した動画でパズルが楽しめるゲーム・アプリケーションの開発を目指したものである。

● 開発経緯

➤ 目的

1. *自身のアプリケーション・クリエイティング能力の修得*
今までのプログラミングの知識と経験を活かし、独力でアプリケーションを開発する能力を身につける。
2. *コンピュータにおける動画再生処理の基礎の理解*
普段は既成アプリケーションで何気なしにコンピュータ上で再生できる動画は、いかにして再生されるのか。その根幹技術の基礎を理解する。

➤ 動機

1. 手軽なマルチメディアとして、音声・映像チャットの登場により普及したウェブカメラを使用したアプリケーションの制作を考案。
2. ユーティリティよりはエンターテインメントの開発に意欲があった。
3. 開発人数に見合った規模のプロジェクトとして、小規模ゲーム・アプリケーションの開発を選択。

上記三項を考慮の結果、ウェブカメラによる映像を用いた動画ジグソーパズルの開発を決断。

➤ 経過

1. 動画再生の実現

今回は C++ で Windows アプリケーションを開発するので、多くの形式の動画を簡単に扱える DirectShow を使用。しかしその機能を使うには Platform SDK を導入する必要があったり、その導入がいささか複雑であったりと思っていた程容易ではなかった。

Microsoft の提供する技術資料は、リファレンスとしては非常に重宝するのだが、入門書には向かない。そこで DirectShow 関連の資料を検索したのだが、DirectShow の使用をためらわせる程に情報量が少なかった。ここで ffmpeg という有名なフリーソフトを TA に紹介されたが、その機能は DirectShow の使用を前提としたものであったため、DirectShow の使用はやむを得なかった。

参考資料は少なかったが、簡単なソースを公開しているウェブサイトなどもあったので、理解はともかく動画を再生するプログラムを組むことは出来た。

2. DirectShow 機能のクラス化

コードの理解のために独自のクラス化を行ってみた。DirectShow は関数群や定数名が複雑な反面、特別な操作を行わない限りほぼ固定の使い方がるので、以降のコードが書きやすくなった。

3. ビットマップ仕様の理解・ビットマップ操作のクラス化

パズルという性質上、動画を分割して再生する必要があったため、ビットマップをキャプチャリングする必要があった。そこでビットマップ操作をクラス化するため、ビットマップの仕様を研究・理解に努めた。

4. プロジェクト規模縮小

ここに来て、このプロジェクトが時間的に、また自身の能力的に完成の目処が立たなかったため、プロジェクトの規模を縮小することにした。目的としては副次的であるゲーム・アルゴリズムを簡素化するため、ジグソーパズルではなく 15 ゲーム (スライドパズル) に変更した。

5. 15 ゲーム・アルゴリズムのクラス化

15 ゲームは結構古典的なパズルで、数学的にも興味深い性質を持っていたためか、調べればすぐに大量の資料が手に入った。C で記述されたアルゴリズムも見つかり、その簡単さに驚きつつ参考にし、C++ でクラス化した。

6. マルチスレッド化

上記までで動画 15 ゲームは一応の完成を果たした。しかしパネルスライド中のアニメーションはなく、当然ゲームはシャッフルされた状態から始まった。

そこでパネルスライドにアニメーションを付けるために、マルチスレッド・プログラムを組むことにした。ウインドウ処理と動画再生処理を分離することでアニメーションを実現しやすくし、複雑化した処理を簡素な 2 つの処理に分離することができた。

7. パネル・シャッフル

アニメーションが実現したので、パネル・シャッフルもそれを利用し、アニメーションさせようとした。アニメーション自体は簡単に実装できたのだが、ここで一つ問題が起こった。

乱数の種として時間を与えるという、乱数の再現性をなくすためによく用いられる手法を取ったにも関わらず、常に同じシャッフル結果が現れるという現象が起こった。

特に複雑なコードを書いたわけではなく、ミスの可能性が低かったため、コンパイラ標準の乱数生成の精度を疑った。そこで乱数生成器として、世界的に高性能と評価のある SFMT(SIMD-oriented Fast Mersenne Twister)を導入。問題は解決された。

8. ビデオ入力デバイスからの映像再生

動画再生とウェブカメラからの映像再生では、同じ再生でも処理が異なる。ウェブカメラからの映像再生にも対応するため、DirectShow クラスにあらたに関数を加え、メイン関数に必要な変更を加えてムービー再生とライブ再生の両方に対応した。

9. プロジェクト完了

● アプリケーション説明

- 二種類の方法で動画の 15 ゲームができる。
 - 通常起動ではウェブカメラから取り込まれた動画が使用される。
 - 動画ファイルを D&D で起動するとその動画が使用される。
 - 発見されなかった場合、起動しない。
- 起動後、動画の再生が開始すると共にパネル・シャッフルが行われ、ゲームスタート。
- 16 分割された動画のパネルをクリックすると、四方のいずれかに空きがある場合そこにスライドする。
- 15 枚のパネルが全て正しい位置に揃えられたら、ゲーム完了。そこから開いた最後のパネルも含めて続きの動画が再生される。

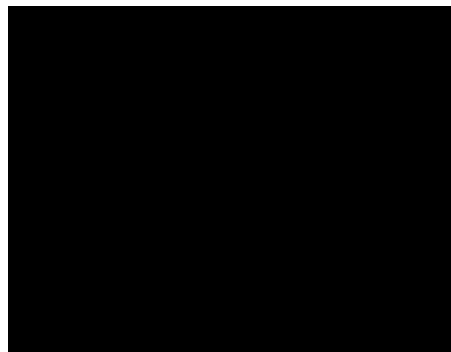
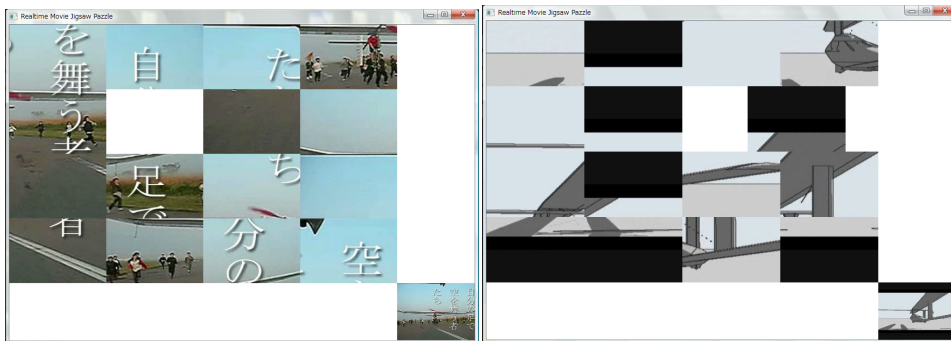
※ 動画の対応形式は **Windows Media Player** で再生可能な全ての形式

※ パネルスライド中もムービーはノンストップ。再生しながらのアニメーション。

※ 右下に幅 160pixel (アスペクト比維持) の完成動画付き。

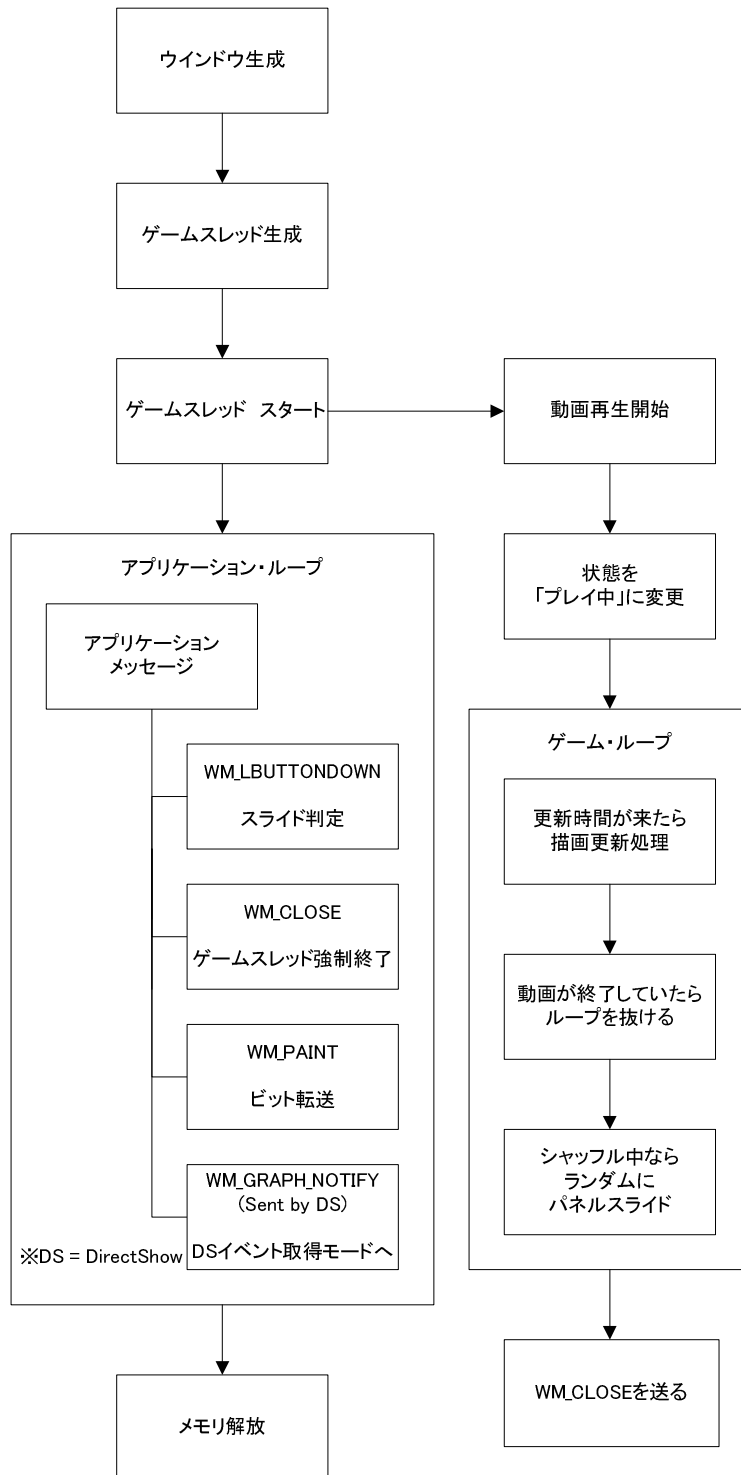
※ FPS30 以下。

スクリーンショット

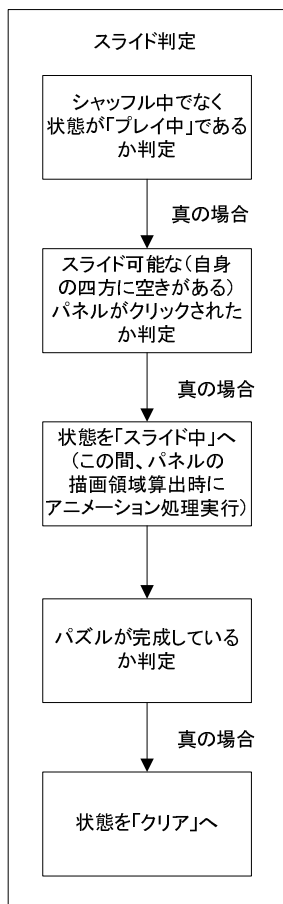


● システム構成概略

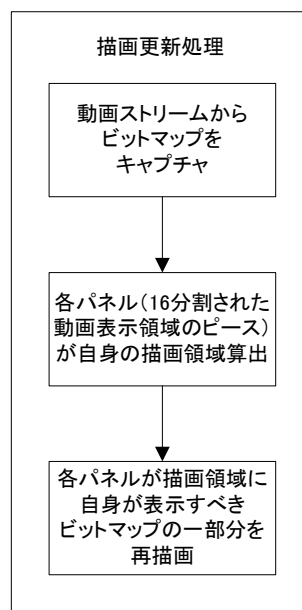
メインの流れ



スライド判定



描画更新処理



● 先行研究・関連技術

➤ eMotion Puzzle (URL: <http://www.emotion-puzzle.com/>)

Signal Talk が開発した動画パズル。販売促進ツールとして開発され、ニッセンや第2日本テレビなどが採用。特許出願済。

Windows Media Player で再生可能な動画ファイルを 16 分割し、正しい位置に並べ替えて動画を完成させる簡易ジグソーパズル。

➤ 15 パズル (Web カメラ) (URL: <http://playon.jp/photo/camera/>)

このプロジェクトと同じく、Web カメラからの映像を用いた 15 ゲームである。

Flash (FCS)を用いたウェブアプリケーションで、動画ファイル再生はサポートしない (別アプリケーション)。

● 開発結果報告

- プロジェクトの目標である「ゲーム・アプリケーション完成」を達成。
- 目的である「アプリケーション・クリエイティング能力の習得」と「コンピュータにおける動画再生処理の基礎の習得」も達成。
- プロジェクトを通して 15 ゲームの数学的特性や乱数生成の知識を習得。「学習」の意味で予想以上の収穫。

● 考察・反省

規模縮小は的確な判断ではあったが、残念なことこの上ない。しかしプロジェクトの完遂は大きな意味を持つだろう。

必要なものに的を絞ったため、非常に簡素なアプリケーションになったが、**Messenger** や **Skype** などのゲームコンテンツとして提供されると面白いのではないかと思われる。

このプロジェクトは自身の学習のためのステップと割り切り、発展させる予定はない。しかし改良できる余地が多く残るのも確かである。

次の機会があれば、映像以外のマルチメディアにも触れたいと思っている。