


某有名落ちもののゲーム っぽいものを作ってみ る



 ぽぴ王子@わんくま同盟

popi@wankuma.com

<http://www.wankuma.com/popi/>

<http://www.ne.jp/asahi/popiprince/>



わんくま同盟 東京勉強会 #35

アジェンダ

- このセッションのきっかけ
- 題材に選ばれる理由
- 今回のゲームで使われているテクニック
 - スレッド
 - 画面表示
 - キー入力
 - 等々…
- まとめ

きっかけ(1)

- 2008年12月の大阪勉強会#25 Game Day
- 最後にみんなでプレイしていた「テ○リスっぽいもの」(バグ付き)がすごく楽しそうだった
- [Wikipedia](#)で調べた

なお、『テトリス』の持つ数学性、動的性、知名度、並びに実装の平易性から、『テトリス』を**ゲームプログラミングの練習題材**として用いられる例がしばしば見られる。

- Ω ΩΩ くな、なんだってー!!!
- ということで『後で作るリスト』に登録

きっかけ(2)

- 今回スピーカーに立候補したものの、何を話すかは全く未定だった
- こんな感じで考えた ↓

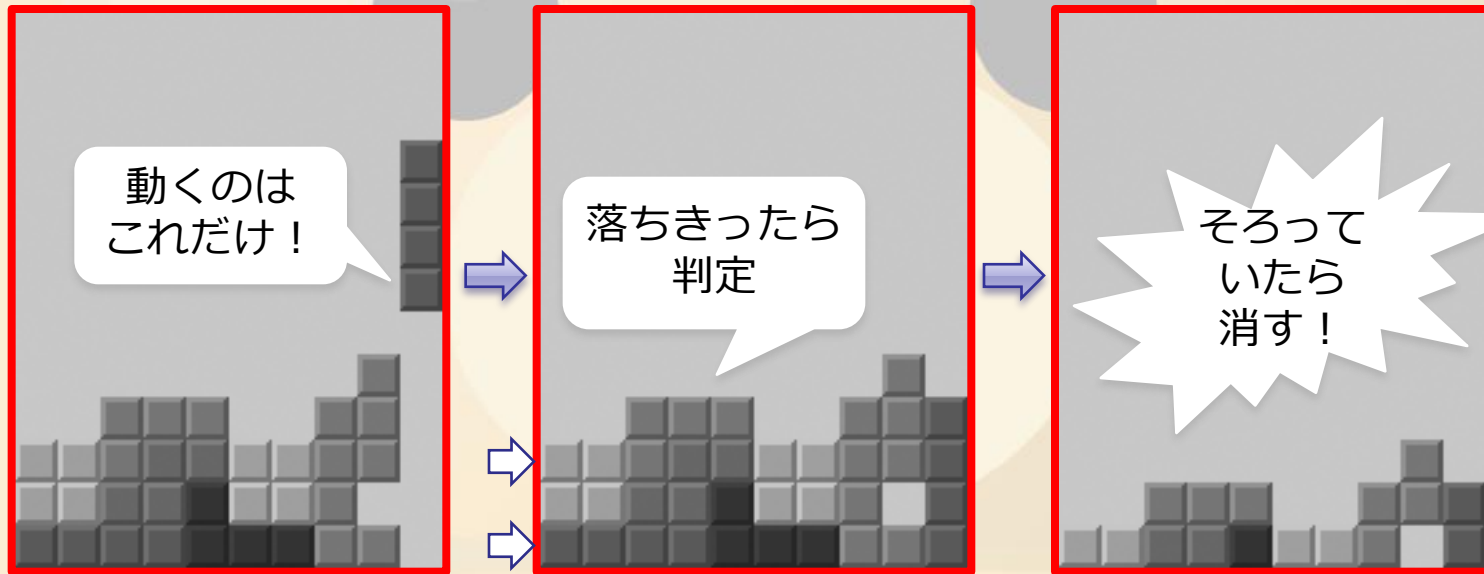
- 電卓の作り方をもう少し詳しく！
- 某世界的に有名な落ちものゲーム
- ぽぴBASIC(笑)
- VS2010とかC# 4.0の新機能について
- Webサービスを使ってごそごそ (Twitterをいじったりとか)

使ったこと
ない☺

Twitterはユーザーでないと
ウケが悪い (中さん談)

“某落ちものゲーム”が題材に選ばれる理由

- 画面上で動くオブジェクトが1個だけなので、当たり判定が簡単
- 動かないブロックの判定は落ちきった時点でOK

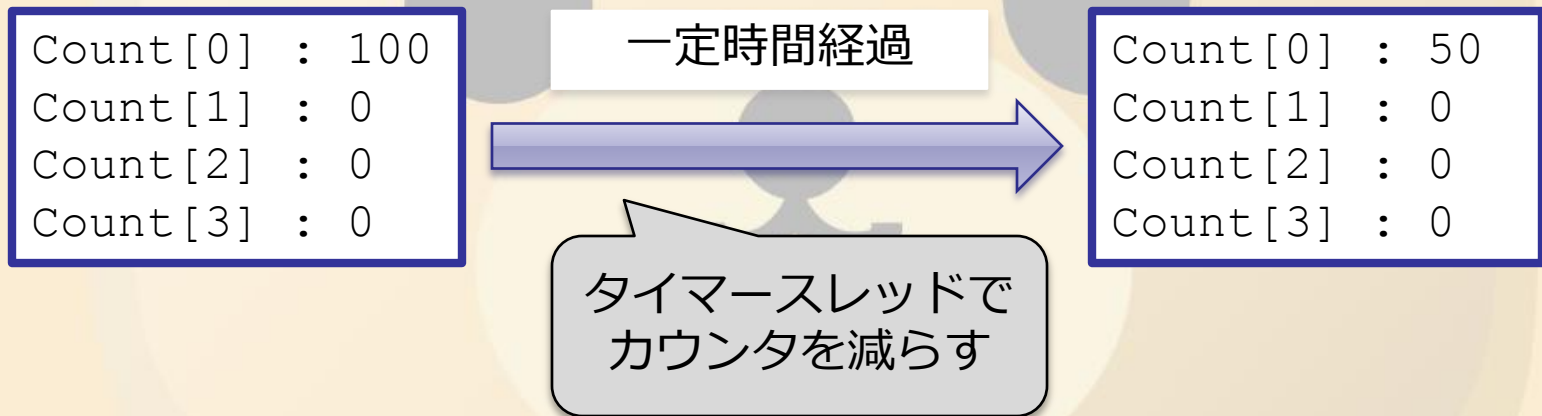


スレッド

- 画面描画はメインスレッドでしか動かせない
- メインスレッドのみで全ての処理を動かすと、時間のかかる処理を行ったときに全体が止まってしまう
- メインスレッドは描画のみ使用し、ゲームの処理はサブスレッドで行う

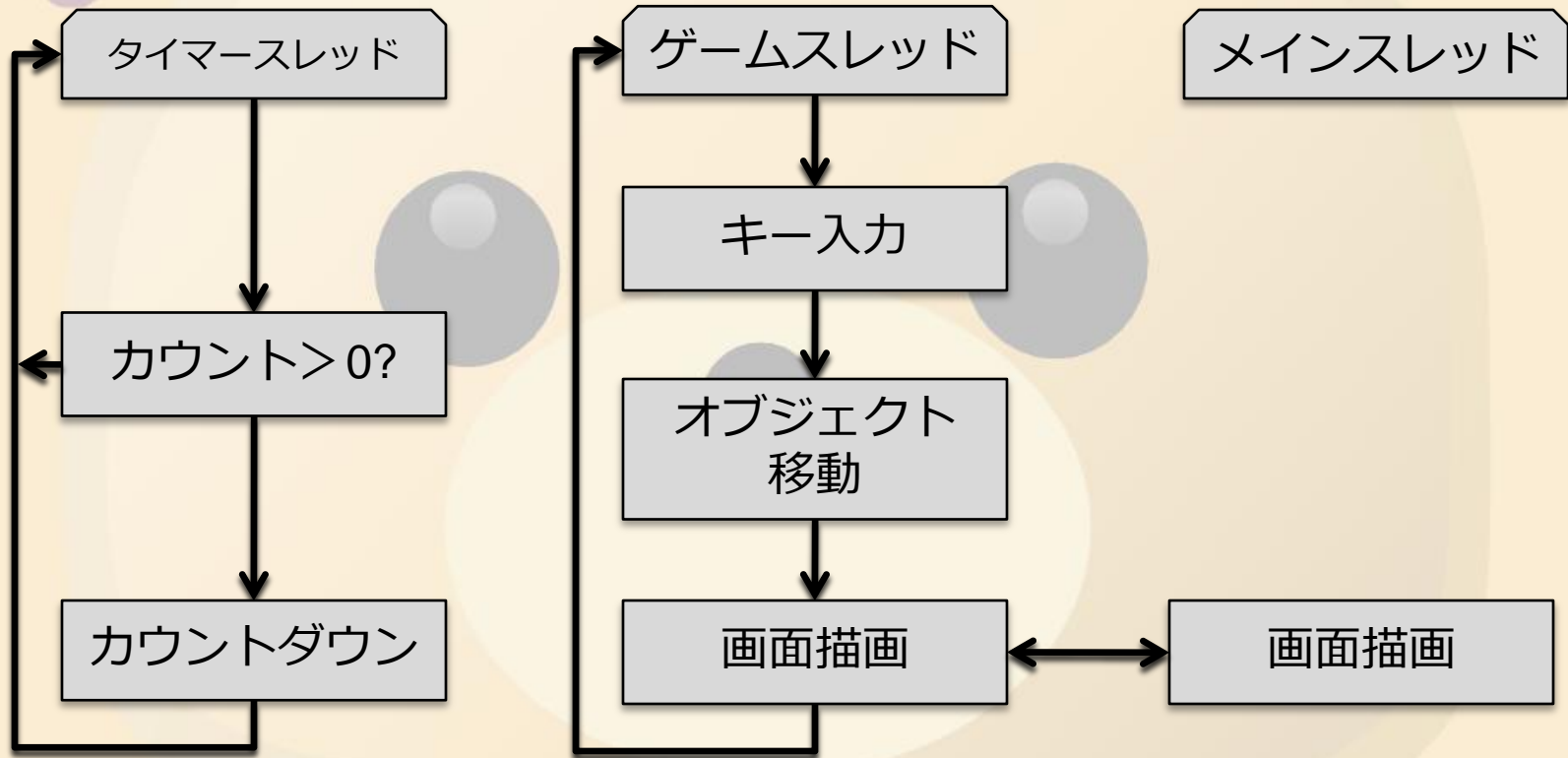
タイマー

- ゲーム用のスレッドともう一つ、タイマー用にスレッドを用意する
- `Environment.TickCount`を見てタイマーを減らす“だけ”の処理を行う



ゲームループ

- 全体的にこんなイメージで動いています



モード構成

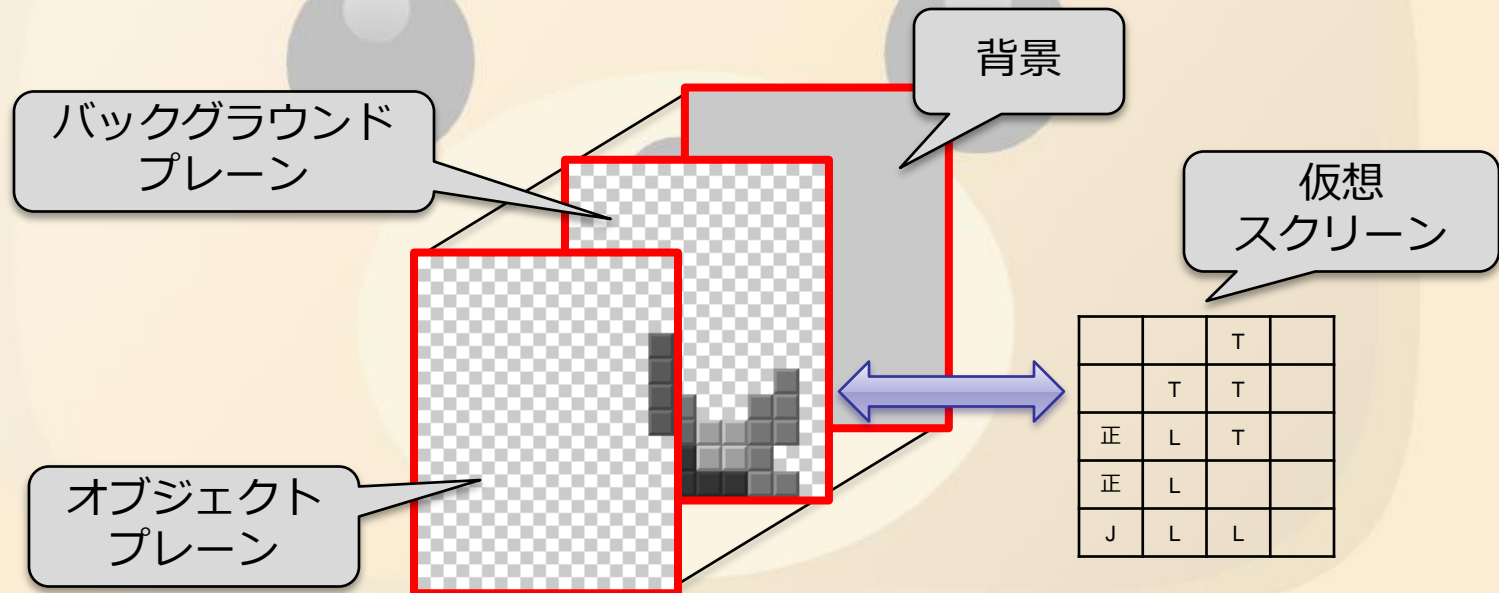
- 以下の3モードを用意する
- デモモード
 - キーを押されるまで待機
- ゲーム中動作モード
 - ブロックを動かす
 - そろったらブロックを消す
- ゲームオーバーモード
 - ブロックをグレー表示 → デモに遷移

画面表示

- フォーム上のPictureBoxをメインスクリーンとして、そこに画像を作成する
- 落ちるブロックを描画するオブジェクトプレーンと、落ちて固定されたブロックを描画するバックグラウンドプレーンを用意

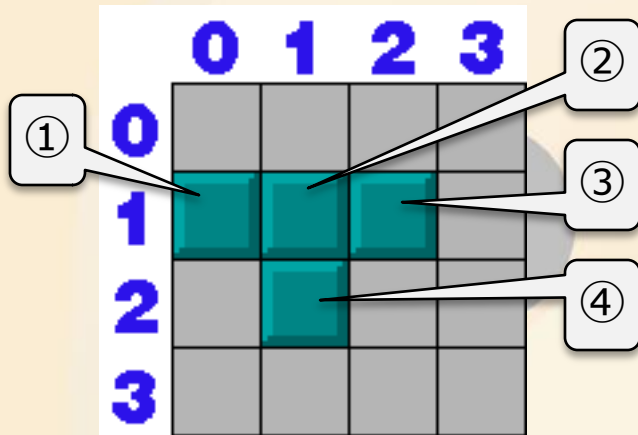
画面表示のイメージ

- 背景とバックグラウンドプレーン、オブジェクトプレーンを合成して表示する
- 固定ブロックがどこにあるかを調べるため、仮想スクリーンで管理する

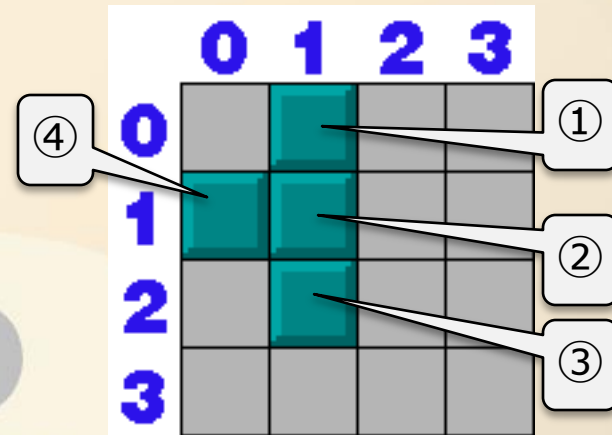


ブロックの表示

- Point型の配列を用意し、4×4のどの位置にブロックを描くかを決めておく



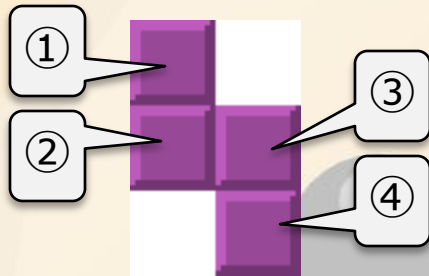
```
Point[] p = new Point[] {  
    new Point(0,1),  
    new Point(1,1),  
    new Point(2,1),  
    new Point(1,2)  
};
```



```
Point[] p = new Point[] {  
    new Point(1,0),  
    new Point(1,1),  
    new Point(1,2),  
    new Point(0,1)  
};
```

移動と回転

- ポイントごとに移動と回転の可否をチェックする



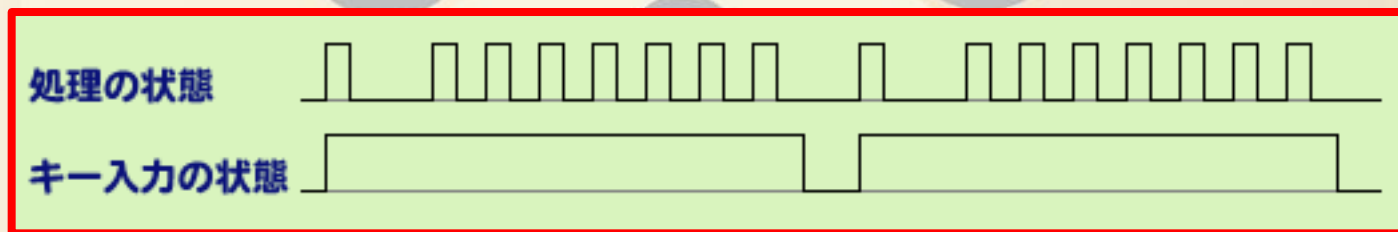
- このブロックが右移動する場合②の右は③のブロックが存在することになる
- 仮想スクリーン上はブロックが存在していないことになっているので問題ない

キー入力について

- キー入力はWin32APIのGetKeyStateを使用
- 次の3パターンのキーリピート制御を行う
 - 横移動
 - ブロック落下
 - 回転

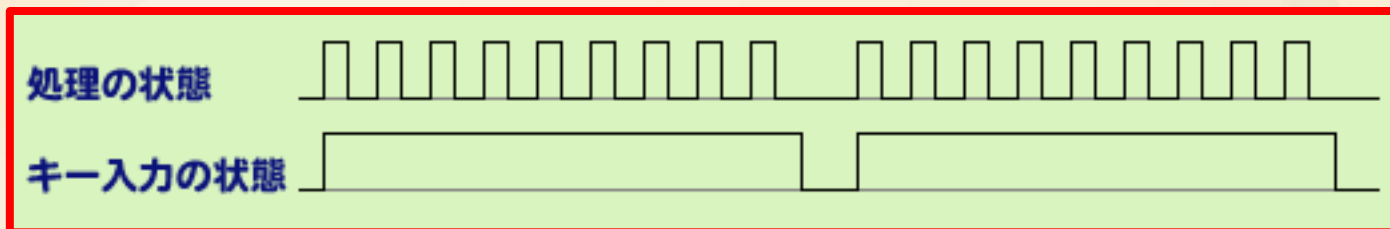
キーリピート（横移動）

- 1度目に押したときに移動
 - 押したままであればある程度待つ
 - それ以降は同じ速度で移動
- 図にあらわすところんな感じ

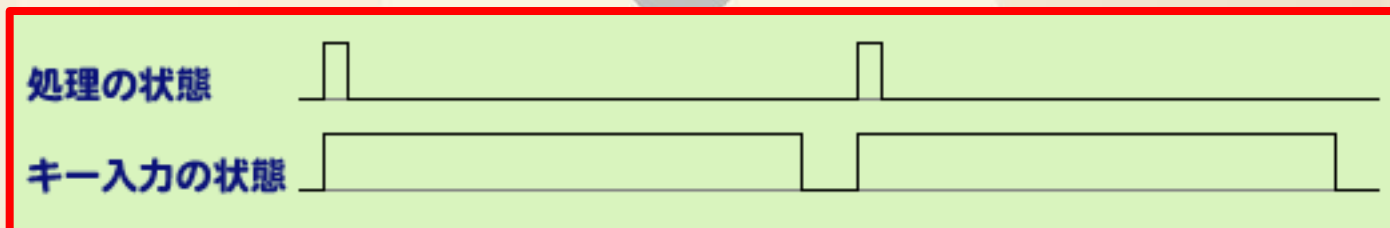


キーリピート（ブロック落下&回転）

- ブロック落下は押された状態である程度間隔を置いてリピート

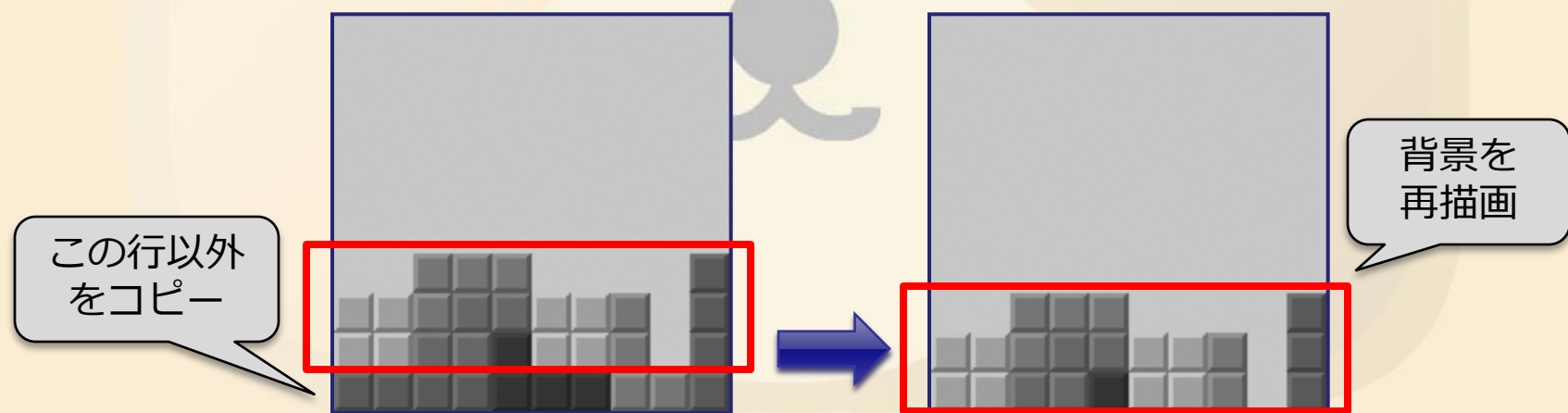


- 左右回転キーは一度押したら離すまでそのまま



ブロック落下判定

- ブロック落下時に下にブロックが存在するか一番下まで到達した時に判定を行う
- 新しい仮想スクリーンを用意し、そろっていないブロックのみコピー
- コピーしたら仮想スクリーンを置き換える



まとめ

- Wikipediaにあるように、他のゲームに比べると実装は簡単
- でも、基礎がしっかりできている人でないと（テト○スを作るのは）難しい
- Windows FormよりWPFの方が簡単！
 - でもおいらはWindows Form大好きだぜ！
- 今回のセッションが役立つかはわからないけれど、ゲームの作り方は自分なりに考えてみるといいヨ！