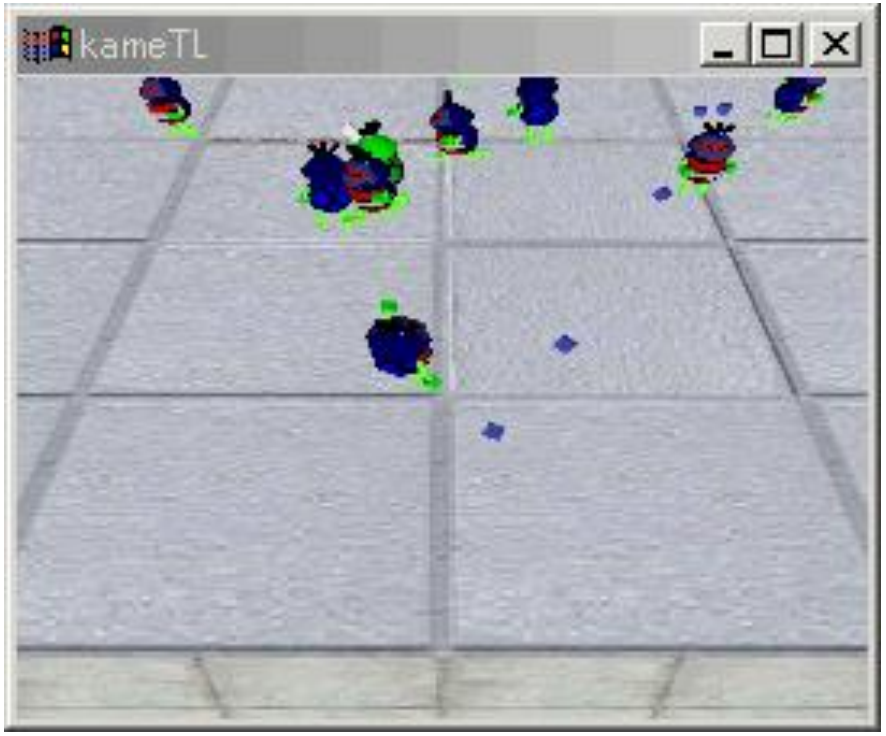
階層型ノンプリエンプティブスレッドによるゲームシステム記述言語



2007年　2月9日（金）

岩崎研究室

丹野 治門

ゲーム開発

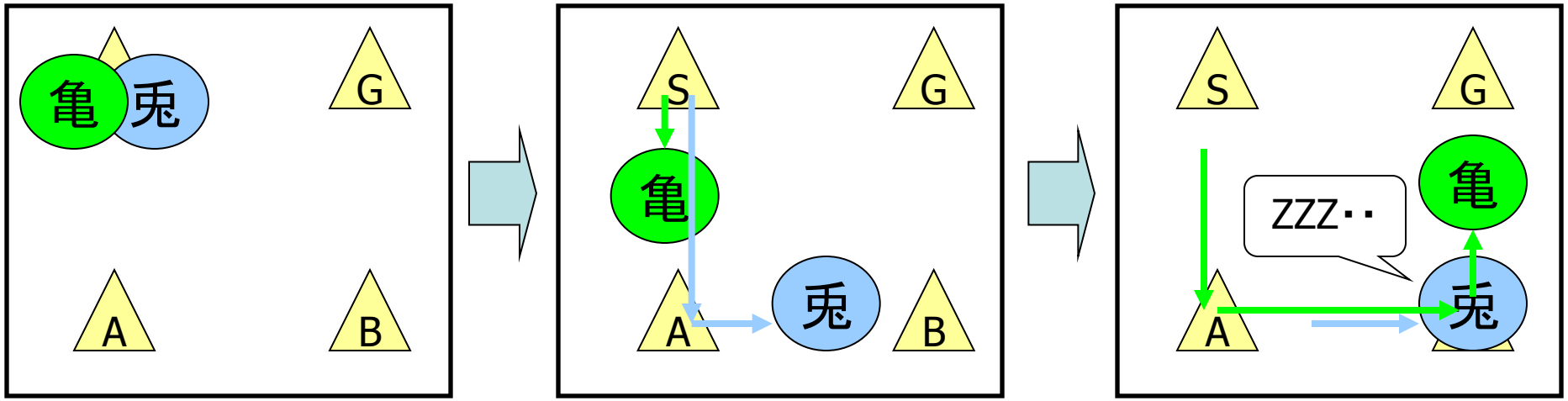
- 楽しい
  - プログラムが動いていることを実感しやすい
    - 視覚、聴覚にうったえる
  - 開発したもので遊ぶことができる

- 難しい
  - ゲームの進行制御
  - 多数のオブジェクトを扱う並行処理

  ➔ 既存言語では記述しにくい

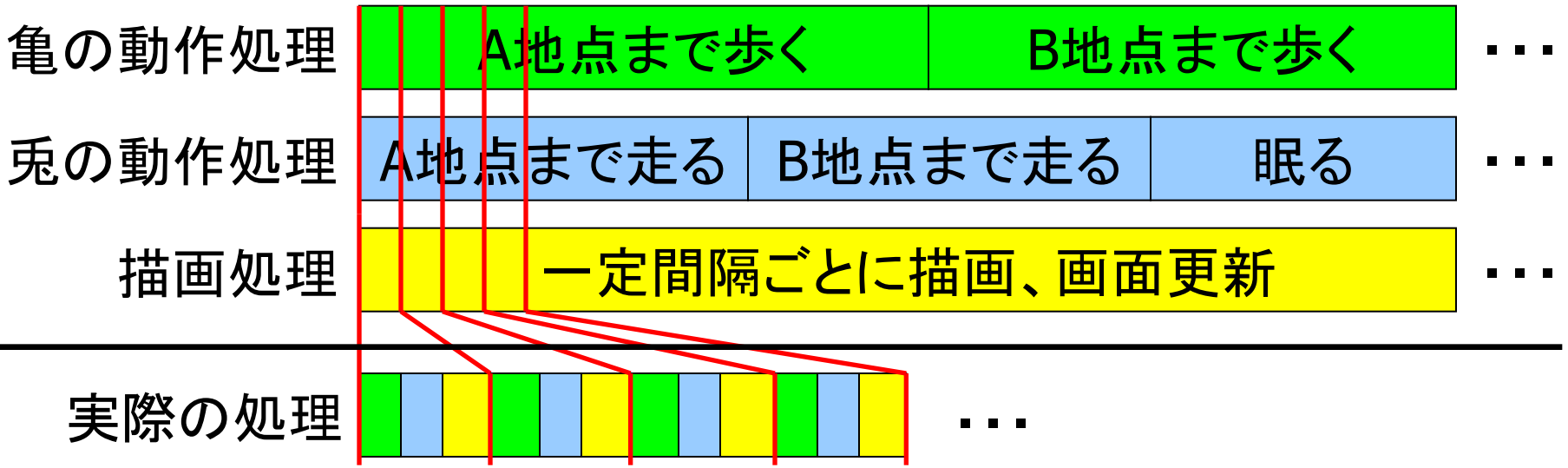
並行処理の例題

- 亀と兎の競争
  - A地点、B地点を経由してG地点へ移動
  - 亀は歩く、兎は走る
  - 兎はB地点で一分間眠る



処理の流れ

- 描画命令を毎フレーム呼ばなくてはならない



- フレームごとの処理単位を明示的に指定したい
  - ➔亀の座標を一定量変更したら切り替えるなど
- 細切れにした処理をどう並べるか指定したい
  - ➔亀、兎、描画・・の順で実行を繰り返したい

問題点

- 既存言語のスレッド（C/C＋＋、Javaなど）
  - プリエンプティブ
    - スレッドの切り替えをきめ細かに指定できない
  - スレッドスケジューリングが環境依存

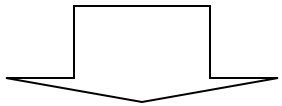
➔ゲームプログラム特有の並行処理に向かない

- 擬似マルチタスク処理（既存手法）
  - フレームごとの処理をする関数へのポインタのリストを用いる
    - 非直感的で複雑

➔プログラマの負担が大きい

提案言語　kameTL

- ゲームに適したスレッドを備える
  - 明示的なコンテキスト切り替え（ノンプリエンプティブ）
  - スレッドは木構造を形成し、階層化される
    - 生成されたスレッドは指定したスレッドの子になる
    - 1フレームごとに階層木を深さ優先で巡回し、実行していく



①キャラクタの動作の流れを自然に記述できる
②ゲームの進行制御が容易に行える

- オブジェクト指向言語
  - Javaに類似した馴染みやすい文法をもつ

①キャラクタの動作の流れを自然に記述

- 例：亀と兎の競争の記述

NPThreadクラスを継承

```
class Actor extends NPThread
  private Point pos;                                  // 位置座標
  public void moveTo(Point to,int velocity){
    while(!pos.equals(to)){                           // 目的地に達するまで
      updatePoint(pos,to,velocity);                   // 1フレーム分座標を更新
      yield();
    }
  }
}
```

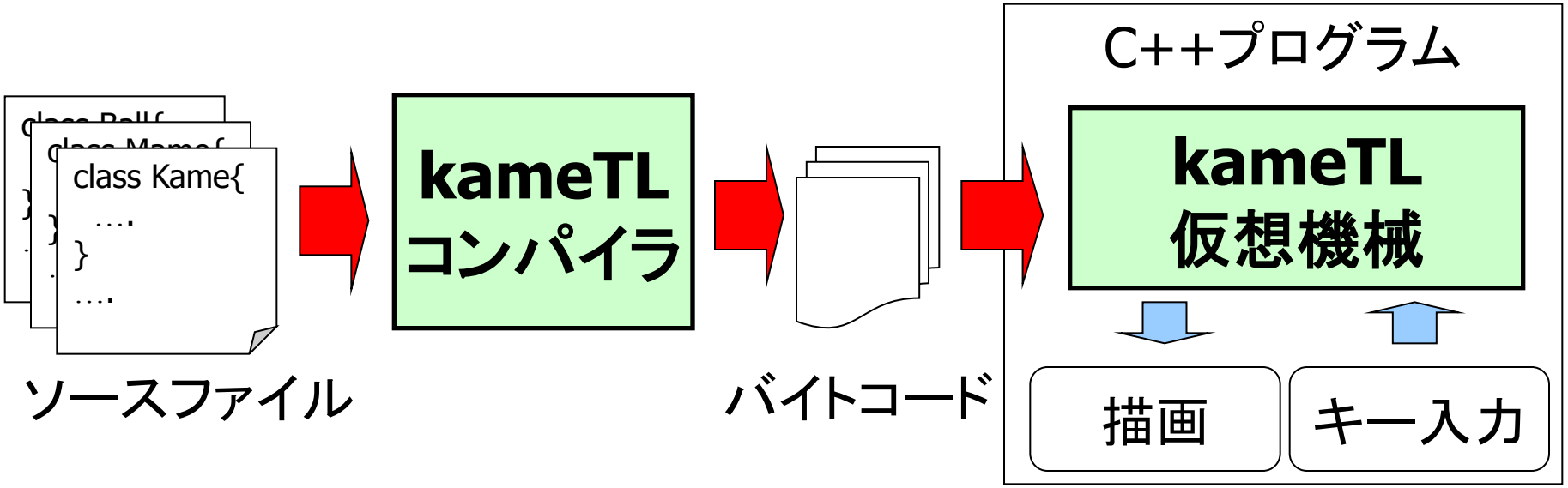
明示的なスレッド切り替え

```
class Kame extends Actor
  public void main(){
    moveTo(A,2); // A地点へ
    moveTo(B,2); // B地点へ
    moveTo(G,2); // G地点へ
  }
}
```

```
class Usagi extends Actor
  public void main(){
    moveTo(A,8);    // A地点へ
    moveTo(B,8);    // B地点へ
    sleep(60.0);    // 1分間眠る
    moveTo(G,8);    // G地点へ
```

動作の流れの自然な記述が可能

構成



- コンパイラ
  - Javaで作成（パーサはSableCCで自動生成）
  - Javaのソースコードと文法ファイルで、合わせて4800行程度
- 仮想機械
  - C++で作成
  - ソースコードは3800行程度

評価

- 仮想機械の基本性能評価（マイクロベンチマーク）

	C++（sec）	kameTL（sec）	実行時間比（倍）
tarai	3.7	161.5	43
dot	1.2	30.8	26
fib	0.9	30.9	34

- 実際のゲームで、1フレームの処理にかかる時間を測定
  - 約400ポリゴンのキャラクタを用意
  - １体あたり、動作制御用、アニメーション制御用の2本のスレッドをもつ

キャラクタ数	描画有（msec）	描画無（msec）	描画の割合（%）
128	32.8	1.4	96
256	63.4	2.9	95
512	125.8	5.8	95

実験環境　Pentium4 2.4GHz、768MB、ビデオカード GeForceTi4200、Windows2000

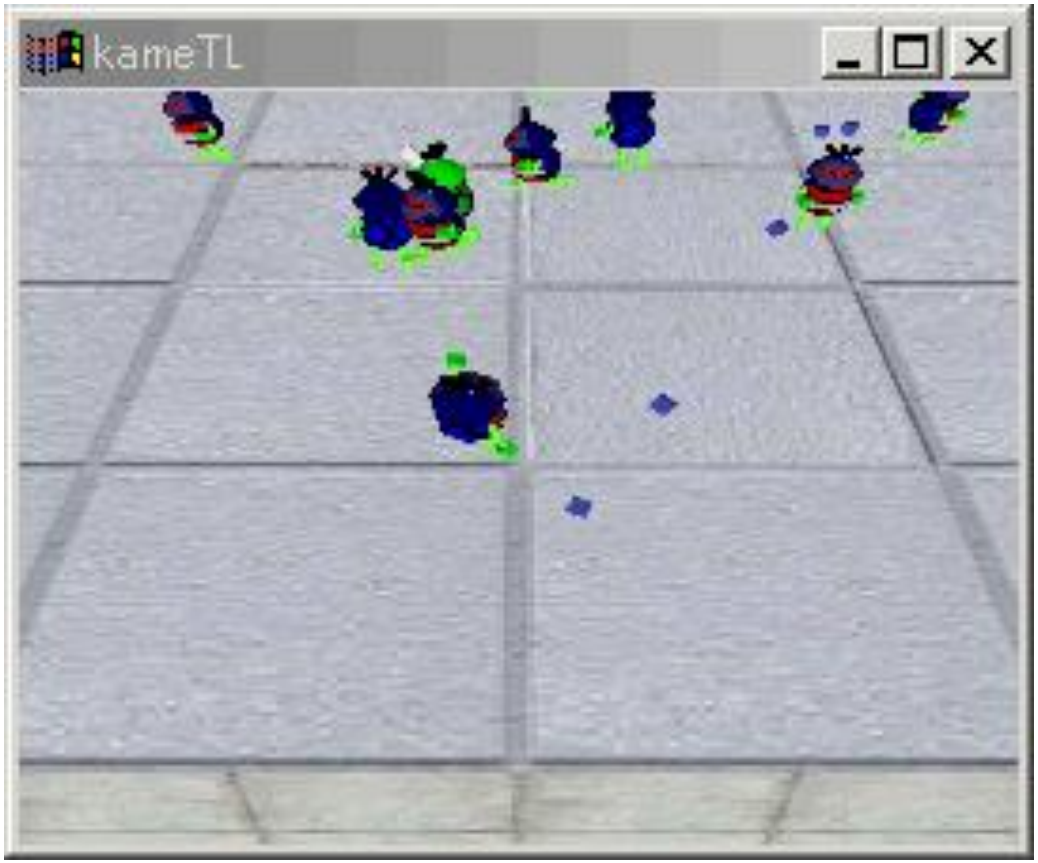
関連研究

- Hot Soup Processor
  - ゲームに必要な機能を文法レベルでサポート
    - 画像読み込み、音声再生等
  - 並行処理機構がない

- アクションゲーム記述に特化した言語[西森03]
  - 並行処理機構を備える
  - 文法が十分強力ではない
    - サブルーチン、配列、オブジェクト指向の機能がない

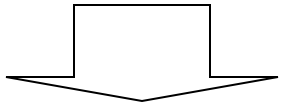
まとめ

- ゲームシステム記述言語kameTLを提案
  - キャラクタの並行動作を簡潔に記述できる
  - ゲーム進行制御を容易に行える
  - 評価
    - 仮想機械は、まだ最適化が不十分なので、C++プログラムに比べると低速
    - ゲームでは描画処理が大部分を占めるため、その影響は少ない


kameTL

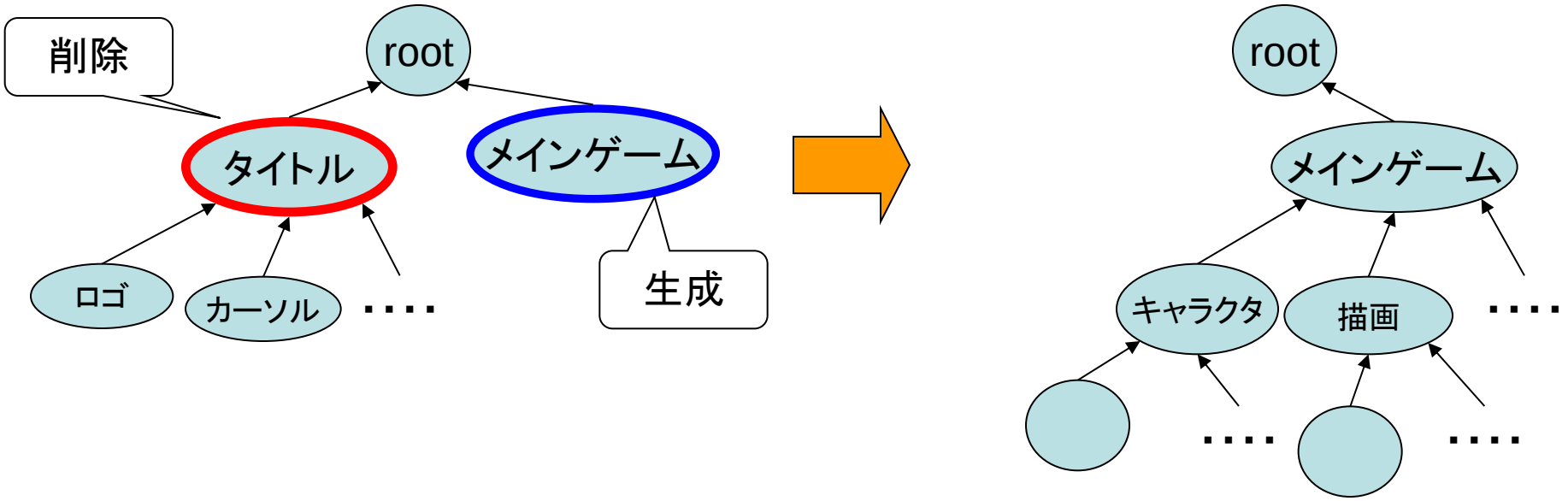
②ゲームの進行制御

- スレッド階層木の特徴
  - 親スレッドへの操作が、子スレッドへ伝播
    - 削除（kill）
    - 一時停止（suspend）と再開（resume）



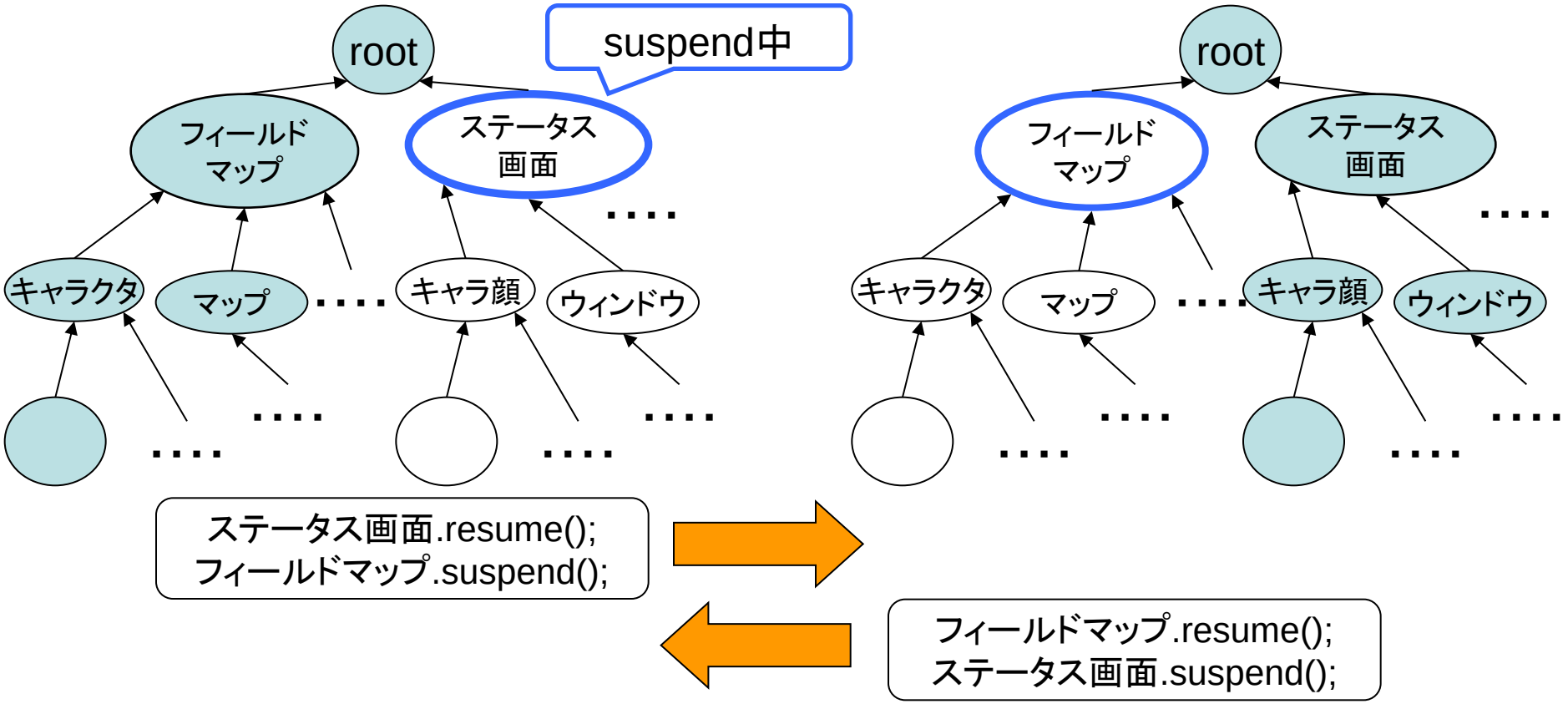
- ゲームの進行制御
  - 場面の遷移
    - ゲームタイトル→メインゲーム→エンディング
  - 場面の切り替え
    - RPGで、フィールド画面⇔ステータス画面の切り替え

場面遷移



例）ゲームタイトルからメインゲームへ

場面切り替え



例）フィールド画面とステータス画面の切り替え