　　　　　　　　　　　　サンプル調査(LockaheadPlayer/java)

　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　 山内 拓磨

* GameState.java , Point.java

このアルゴリズムに限らず、ゲームAIとして必要な処理が記述してあり、他のjavaで作成されたAIでも用いる。

詳細は下記、リンク先ドキュメント参照。

<https://docs.google.com/document/d/17H3m21G6vSNdS11ZOeMFqDgBSZLFr2JweWbSHb27a-U/edit>

* LockaheadPlayer.java

ざっくりいうと全通りの探索を行っていき、探索深さまで達した時評価を行う。

評価が最も良い手を更新していき、最終的に初めの一手で最良だと考えられるものを選択する

この時、相手の動きは考慮せず、現在の位置のみを参照する。

* + playRecursive()
    - この関数を再帰的に呼び出すことで先行探索を実現
    - 引数
      * GameState state : 現ステップのゲーム情報
      * Point position : 自分の座標

　　　 再帰的に呼ばれていく中で変更されていく

* + - * Point velocity : 自分の速度

　　　 再帰的に呼ばれていく中で変更されていく

* + - * opponentPosition : 相手の位置

　 同一ステップの探索の中では一定

* + - * int dist : 探索開始地点からの移動距離
      * Point bestAccel : 暫定的な最良の加速度

より良いものが見つかれば更新される

* + - 探索深さまで到達したときと未到達の場合で動作が異なる
    - 到達した場合(depth==0)、その座標から全通りの加速度で移動可能な組み合わせの数をカウントする

これと最終的なy座標とdistをもとにその葉ノードのscoreとして返す

* + - 探索途中の場合(depth!=0)、そこから派生しうる全通りの組み合わせで、各々再帰的にplayRecursive()を呼ぶ

衝突が起こる場合には再帰は発生しない

* + - playRecursive()の返り値はint型のbestsocreだが、引数のbestAccelにインスタンスを渡すようにしてあるため、bestAccelは更新されていく
    - bestAccelは再帰的に呼んだplayRecursive()の結果と、その時点のbestscoreを比較して更新を行う
  + play()
    - playRecursive()から得られたscoreが負の値である場合(立ち往生している場合)、速度から単純な条件で加速度を決定する

この手法も最終的に評価関数を使う

相手の動きを考慮せず自分の動きのみを考えるため、一概には言えないが計算量はαβなどに比べると少なく済みそう

しかしその分相手の動き次第では次ステップで先行探索した通りには動けない可能性もある

また、衝突以外の条件で枝刈りが発生しないため、明らかに有望でない手も探索してしまう

単純にこの手法を使うだけでは、それほどいい結果にはなりそうにはないと思った