c0a2400443 / **ProjExD_Group09**

Code Pull requests Actions Projects Wiki ···

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: main ▾ ← ... compare: C0A24004/King ▾

Discuss and review the changes in this comparison with others. Learn about pull requests

Create pull request

2 commits 1 file changed 1 contributor

Commits on Jul 8, 2025

キングの移動を実装
c0a2400443 committed 1 hour ago
8b77f40

ゲームの終了を実装
c0a2400443 committed 15 minutes ago
464175f

Showing 1 changed file with **37 additions** and **4 deletions**.

Split Unified

41 🟩🟩🟩🟩⬜ chess.py

```diff
     def get_possible_moves(self, board):
         """駒の可能な動きを取得（現在は全方向移動可能）"""
         moves = []
-        for row in range(8):
-            for col in range(8):
-                if board.is_valid_move(self.row, self.col, row, col):
-                    moves.append((row, col))
+        if self.type == PieceType.KING:
+            directions = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)]
+            for dr, dc in directions:
+                new_row = self.row + dr
+                new_col = self.col + dc
+                if 0 <= new_row < 8 and 0 <= new_col < 8:
+                    target_piece = board.get_piece(new_row, new_col)
+                    if not target_piece or target_piece.color != self.color:
+                        moves.append((new_row, new_col))
+        else:
+            for row in range(8):
+                for col in range(8):
+                    if board.is_valid_move(self.row, self.col, row, col):
+                        moves.append((row, col))
```

```diff
         return moves

     def move(self, new_row, new_col):
         self.current_turn = PieceColor.WHITE
         self.selected_piece = None
         self.selected_pos = None
+        self.winner = None
         self.possible_moves = []
         self.setup_initial_position()

         # 移動先が有効かチェック
         if not self.is_valid_move(from_row, from_col, to_row, to_col):
             return False
+
+        #target_pieceを定義
+        target_piece = self.get_piece(to_row, to_col)
+
+        # キングが取られたら勝敗を設定
+        if target_piece and target_piece.type == PieceType.KING:
+            self.set_piece(to_row, to_col, piece)
+            self.set_piece(from_row, from_col, None)
+            piece.move(to_row, to_col)
+            self.winner = piece.color  # 勝った側の色
+            return True
+

         # 移動実行
         self.set_piece(to_row, to_col, piece)
     def draw_info(self):
         """ゲーム情報を描画"""
         info_y = BOARD_SIZE * SQUARE_SIZE + 10
+
+        if self.board.winner != None:
+            winner_color = "White" if self.board.winner == PieceColor.WHITE else "Black"
+            result_text = f"{winner_color} wins!"
+            text = self.font.render(result_text, True, RED)
+            self.screen.blit(text, (10, info_y))
+            return  # 勝敗が決まったら他の表示は不要

         # 現在のターン（英語で表示）
         turn_text = f"Current Turn: {'White' if self.board.current_turn == PieceColor.WHITE else 'Black'}"

     def handle_click(self, mouse_pos):
         """マウスクリックを処理"""
+        if self.board.winner:
+            return # 勝敗が決まったらクリック操作無効
+
         row, col = self.get_board_pos(mouse_pos)
         if row is not None and col is not None:
             if self.board.selected_piece:
```