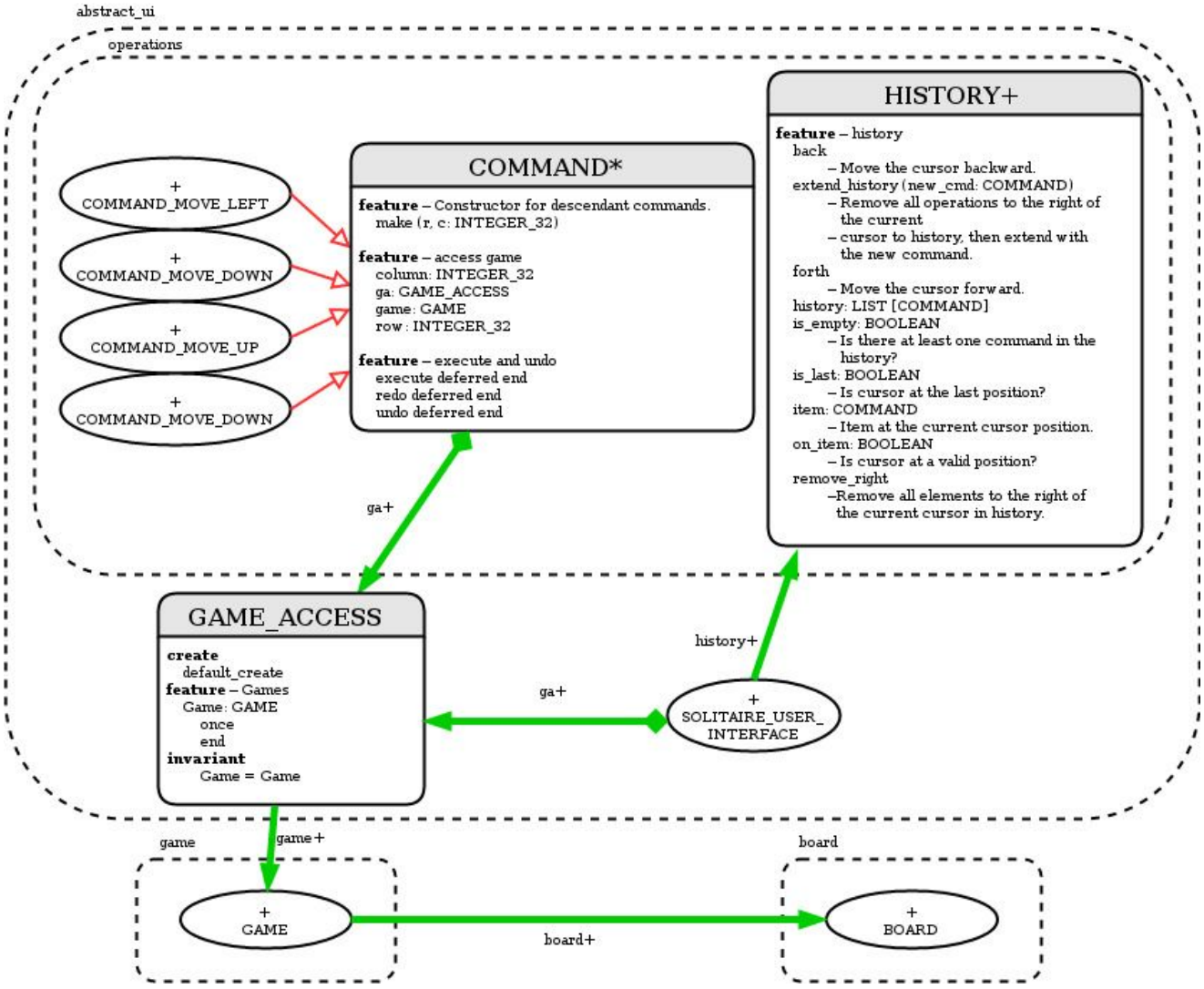
Lab 4: Report

EECS3311,
Fall 2017.
Michael Mierzwa.
213550702
mierzwam

Section 1: Architectural Diagram



Section 2: Undo Redo Pattern

The Undo Redo pattern in the case of peg solitaire, uses a list of commands in history to implement undo and redo as features. The history class modifies a list of commands called history, and provides the commands for traversing and editing a list of the commands used previously. The command class has two attributes for row and column, a game and game access attributes. The game is set to the game attribute in game access to maintain the singleton design pattern. The rest following from game access is identical to Lab 2.

The part unique to this lab however is the usage of the command as a deferred class to implement undo redo. Since the history class stores a list of commands, we can specialize the commands within the given history list. We do this by completing the deferred features from command, in all of the children of command. Each child of the command class represents a move command from game, and contains the two attributes that are shared through all the commands. Each command implements execute and redo as a single execution of the respective command, E.g. game.move_down(row,column). These children also implement the undo feature as setting the board state to be the same as was required by the preconditions in the respective feature in the game class.

Finally there is the SOLITAIRE_USER_INTERFACE class, which is simply a means of error checking and allows a client to test moves and receive appropriate error messages as to what went wrong. This class also has multiple constructors labeled by the board state that the user wishes to start the game in, and instantiates its variables accordingly. For each command, it has an error checking step that mirrors the preconditions for the feature in game, it also displays to the user why the exact precondition was violated. When all preconditions have been met, it executes the command and adds it to the history list. Undo and redo features in this class use features from history to check if they can undo/redo, and then call undo/redo on the specific command that history list is currently pointing to.

END