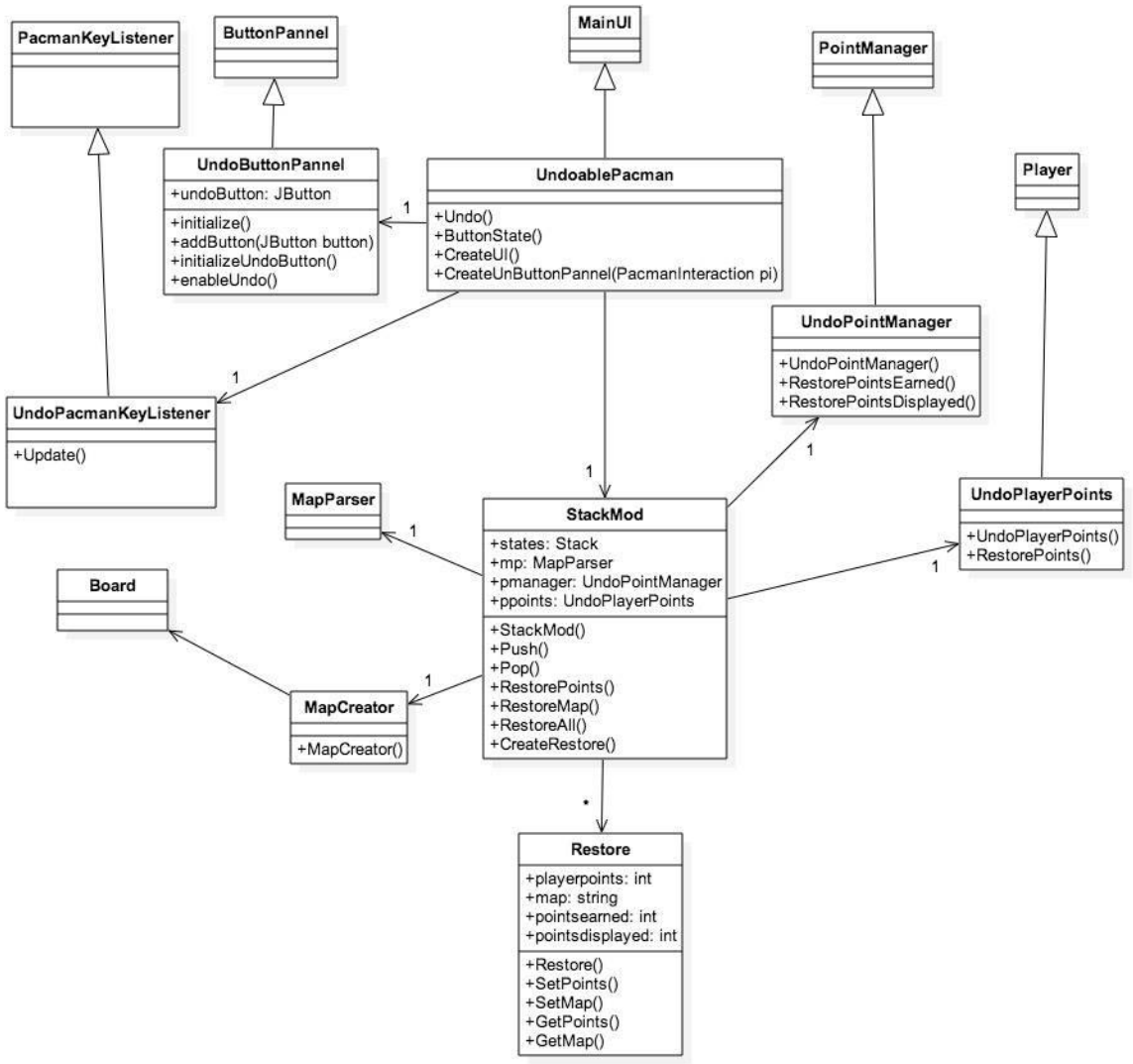


The basic idea behind our implementation of the UndoButton is to load a new game with the pervious state's value. We will save multiple "states", an abstract data type, Restore, whenever a key is pressed and access them using basic stack functions.

UndoablePacman is a class we created which extends MainUI class. UndoablePacmanis the extended UI class which will have an additional undo() method. UndoButtonPanel is a class created which extends ButtonPanel class. UndoButtonPanel will have additional features involving the new undo button. UndoPacmanKeyListener extends PacmanKeyListner and notifies UndoablePacman whenever a key event takes place.

UndoPointManager, UndoPlayerPoint are classes created which extends PointManager class, and Player class respectively. UndoPointManager, and UndoPlayerPoint will add additional method that allows direct modification of points earned and points to be displayed. StackMod is the class that stacks and manipulates the Abstract Data Type “Restore” which contains information of the states of the game when user makes a move. StackMod needs to know about an instance ofMapParser, UndoPlayerPoints, UndoPointManager, and a collection of “Restore”. StackMod will be the class that gets the information(states) out of restore and feeds the information into the corresponding class to allow the game to restore the previous state. For example, location of sprites will be fed to MapParser, and points will be fedinto UndoPointManager. MapCreator creates a mapfile to be parsed by accessing the Board class. When UndoablePacmanruns, it will need an instance of UndoButtonPanel to instantiate the button panel, and it will also need to know an instance of StackMod to be able to manage the undo process.