Dastan

Programming Tasks (Extension)

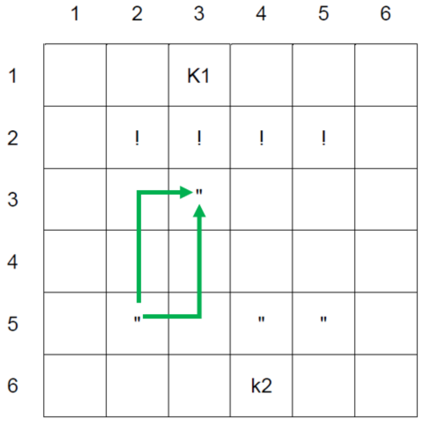
Extension 1

Introduce a health scoring system for pieces. Each piece (except the Kotla) has a starting score of 10 health points. Each time a piece is landed on, it incurs damage, reducing its health. Advise the player each time a piece's health is reduced. When a piece reaches 0 health points, it is removed from the board. Only one piece can attack another at one time. When a piece is being attacked the attacking player symbol should be shown on the left of the piece and the target piece symbol should be shown on the right of the square.

Damage is determined using this formula:

> *Position of move choice in the queue + Manhattan distance from the piece (number of rows different + number of columns different).*

Manhattan distance is a heuristic function for calculating distance between two locations, for example in a grid. In the case of Dastan it is calculated by counting the sum of the number of squares horizontally and then vertically (or vice versa) between a player starting location and the finishing location as shown in **Fig 1**.



	1	2	3	4	5	6
1			K1			
2		!	!	!	!	
3			"			
4						
5		"		"	"	
6				k2		

Fig 1

An attack using a move from position 1 in the move queue reduces health by 1 point.
An attack from position 3 in the move queue reduces health by 3 points. The distance from the piece is how far away the opponent is from the attacker. This is the sum of the row change and column change. An attack from further away, therefore, incurs a greater level of damage.

Extension 2

Create a new game square called 'Qunbila Ghayr Muwajaha' (Unguided Bomb). The Unguided Bomb has a 33% chance of appearing in any turn and is given to the current player in that turn. The bomb has a 10% chance of detonating. The player can either move away from the bomb or throw it across the board. When the bomb is thrown the player can choose any board location to throw it to except their own location or a Kotla.

The 'Throw bomb' option should be available through the **MoveOffer** menu.

If the bomb is thrown to an opponent's square, the opponent takes ownership of it and can then throw it back or move away from it. Each turn carries a 10% chance of the bomb detonating. If the bomb is thrown to a blank square, the bomb loses ownership from either player and changes its symbol to a '?'. The bomb remains at this location until a player moves to the square containing the bomb to pick it back up again and be able to throw it. Each turn carries a 10% chance of the bomb detonating.

If a player piece is captured while holding the bomb, the ownership of the bomb moves to the new owner of the square.

Extension 3

Introduce the concept of a ‘Makinat Taftish’ (Inspection Machine). This is a computer-controlled piece which does not belong to either player. After each player turn, the Inspection Machine should measure the distance from itself to all the other pieces on the board using Manhattan distance. The machine should then move itself towards the closest piece on the board, regardless of team. If two pieces are the same distance away, the machine should select one at random. The machine can move in any direction, but only one square at a time.

The machine should repeat this behaviour once a turn until it reaches a player piece and captures it. Neither player gains any points for a piece being captured.

	X	X	X		
	X	*	X		
	X	X	X		

Inspection Machine moves

Manhattan distance is a heuristic function for calculating distance between two locations, for example in a grid. In the case of Dastan it is calculated by counting the sum of the number of squares horizontally and then vertically (or vice versa) between a player starting location and the finishing location as shown in **Fig 1**.

The machine does not place any weighting on a ‘target’ to move towards and can capture a player piece or a Kotla.

A player loses the game if their Kotla is captured by the Inspection Machine.

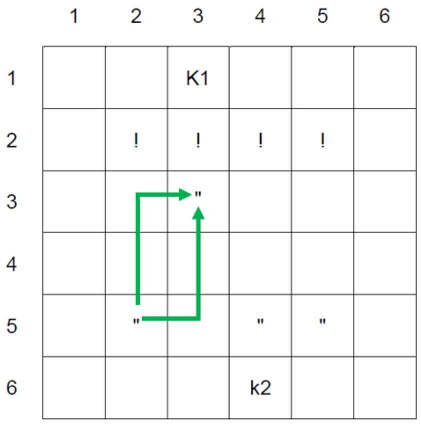


Fig 1

Extension 4

Introduce the concept of a ‘Multi-Move’. This allows a player to combine two move options into one, but at a significant points cost.

Introduce a new option 9 to the main game playing menu called ‘Multi-Move’. On selecting this option, the player can select two move options to execute sequentially. The player must choose move option 1, then move option 2, choosing a ‘move to…’ square reference for each option. The ‘move to…’ square reference for move option 2 must be a legal move based on the ‘move to…’ square reference from move option 1. Both moves must be legal. The program should use error handling to prevent the player from entering illegal references and allow them to re-enter.

Selected moves in a multi-move can be from any position in **MoveOptionQueue** without incurring a cost from the position of move in **MoveOptionQueue**.

On entering a legal multi-move, the game should move the selected player piece on the board. A multi-move should cost the player 25 points.

If the player lands on an opponent piece through either move 1 or move 2 or both, the associated opponent pieces should be captured as normal.

Extension 5

Introduce the concept of a 'Khalad' – a mole. Introduce a new submenu option when a player selects a move option from the main game playing menu. The submenu should offer the player the option to activate a 'mole' mode for the selected move option.

On selecting 'mole' mode, the move operates as normal; however, the player piece moves 'underneath' the board. A piece which is operating in 'mole' mode should be shown as an 'M' for player 1 and an 'm' for player 2, which is displayed on the left-hand side of each square instead of the normal piece symbol on the right-hand side of a square. This means that two pieces can occupy the same square, one in 'mole' mode and one on the board 'surface'.

A piece in 'mole' mode can move around underneath the board using normal move options, but cannot be captured by an opponent piece on the surface of the board. Once the player has selected a piece to be in 'mole' mode, the submenu should change to now give the player the option to resurface their mole piece after moving it. If a piece in 'mole' mode resurfaces in a square reference containing an opponent piece, the current player captures that opponent piece. Once a player resurfaces their 'mole' mode piece, the 'mole' mode submenu should no longer be offered to the player.

A mole cannot move onto the Kotla square as the foundations are too deep for it to dig underneath.

If an opponent also has a piece operating in 'mole' mode, one mole can capture another just like normal pieces on the board surface.

Extension 6

Introduce an option to 'preview a move' prior to making it. Add a new option 0 to the main game playing menu. On selecting this option, a player can select any move from position 1, 2 or 3 in their queue and a valid player piece as normal. The player is then shown a 'preview' copy of the playing board which shows an 'X' in all the squares which the selected move option can move to from the selected valid current player piece.

The player should then be given the option to enter in a valid 'move to…' square reference for the selected move option or go back to the main menu to choose a different move from their queue. When a valid 'move to…' square reference is selected, the game should make the move as normal.

The player can 'preview a move' as many times as they like during the game.

The 'preview a move' option should not attempt to show the player 'move to…' references which are outside the bounds of the board.

Extension 7

Introduce a new option at the start of the game to allow the players to place their pieces on the board in different formations prior to the game starting. Players can choose from any of the following.

All the positions are shown from the perspective of player 2.

Once the players have selected their chosen starting positions, the game should continue as normal.

	1	2	3	4	5	6
1						
2						
3						
4						
5			"	"		
6		"		k2	"	

‘Khandaq’ Option (Trench)

	1	2	3	4	5	6
1						
2						
3						
4	"					"
5			"	"		
6				k2		

‘Iltifaf’ Option (Pincer)

	1	2	3	4	5	6
1						
2						
3						
4						"
5			"	"	"	
6				k2		

‘Darba Rukniya’ Option (Corner Kick)

	1	2	3	4	5	6
1						
2						
3						
4			"	"		
5			"			
6			"	k2		

‘Khanjar’ Option (Dagger)

Extension 8

Introduce the concept of an ‘Al Amlaq’ (Giant), which is formed when a player lands their Mirza onto one of their **own** player pieces. A Giant is shown as a ‘G’ for player 1 and ‘g’ for player 2.

Once a Giant has been created by combining a player Mirza with a normal piece, it cannot be undone and remains as a Giant for the rest of the game.

A Giant can move around the board using the same move options as a normal piece; however, it only needs to land to within one square (in any direction) of any opposition piece to capture it, including the opponent Kotla and Mirza.

A Giant can be captured by any opponent piece as normal and is worth 20 points if captured.

Extension 9

Introduce the concept of an ‘Adra’ (Chainmail). Add a new option ‘C’ to the main game playing menu. On selecting this option, the player should be asked which piece they would like to add chainmail to. A player can only add chainmail to two pieces during the game. The chainmail is underneath a piece’s clothes and therefore a piece’s symbol doesn’t change when it has the chainmail. It is an invisible forward-facing barrier which means that the piece cannot be attacked from the front. Although an opponent piece can be one square in front of a current player piece, it cannot capture through the chainmail – it must approach the shielded piece from either side or behind.

A player can add a chainmail to any two pieces in the game including the Mirza, but not the Kotla.

Extension 10

Introduce the load and save features to the game. Add new options ‘L’ and ‘S’ to the main menu to offer the player options to load a previously saved game or save the current game.

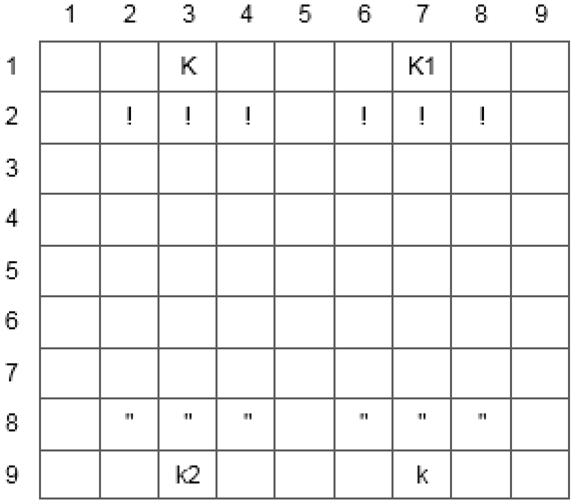
The load and save submenus should give the user the opportunity to enter the file path of a .txt file. The program should have appropriate error handling to prevent it from attempting to load corrupt or missing data or saving to an invalid location. The program should store appropriately formatted comma-separated values to store all of the program data required to rebuild a game. Again, appropriate error handling should be included when a game is being rebuilt to ensure that the positions of pieces, etc. are all valid within the bounds of the board.

Extension 11

Introduce a new feature to adjust the size of the playing board and pieces accordingly. At the start of the game, give the player the option to choose the size of the board. The dimensions do not need to be even; however, error handling should be included to prevent the board from being smaller than 6 × 6 or larger than 10 × 10. (Appropriate formatting needs introducing on a 10 × 10 board to ensure the squares line up correctly.)

For boards of 6 to 8 columns wide, ensure that both player Kotlas are placed centrally on appropriate top and bottom rows of the board. A 7 column wide board should have 5 pieces per player and an 8 column wide board should have 6 pieces per player.

For boards 9 and 10 columns wide, introduce a second Kotla for each player on the appropriate top and bottom rows of the board. The Kotlas should be evenly distributed across the board. The player should still only have 1 Mirza, which should be placed in either of the Kotlas at random. A 9 or 10 column wide board should have 6 pieces per player, 3 of which should be in front of one Kotla and 3 in front of the other, as per the example shown.

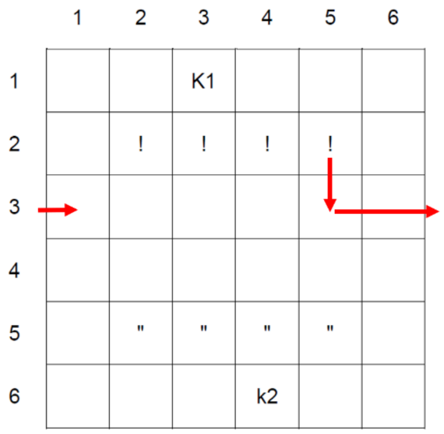


	1	2	3	4	5	6	7	8	9
1			K				K1		
2		!	!	!		!	!	!	
3									
4									
5									
6									
7									
8		"	"	"		"	"	"	
9			k2				k		

Extension 12

Adjust the playing board to allow the sides to wrap around. On making a move, a player can move off the left- or right-hand side of the board and land on the correctly associated square on the opposite side of the board as if the board was wrapped around.

For example, a player can select a Cuirassier move for the piece in square 2,5 and move to square 3,1 which is one square forward followed by two squares to the left (looking at the board from the point of view of the piece).



	1	2	3	4	5	6
1			K1			
2		!	!	!	!	
3						
4						
5		"	"	"	"	
6				k2		

Extension 13

Introduce the concept of an ‘Muraqib’ (Meerkat Lookout piece). At the start of the game, give each player the opportunity to place their Muraqib on any empty square on the board. The player 1 Muraqib is represented by an ‘M’ symbol and player 2 Muraqib is represented by an ‘m’.

The Muraqib is on constant lookout for the player it belongs to. For example, when player 1 makes a legal move and the board and player 1 queue are updated, the player 2 Muraqib should check every player 1 piece left on the board and test each of moves 1, 2, 3 from the player 1 move option queue to see if it could threaten to capture any player 2 piece. If such a threat is possible, the Muraqib should alert player 2 in case they have missed that possible threat.

A Muraqib cannot be captured. If either player lands on the square containing an Muraqib, it simply disappears down into its burrow underneath the board. While it is in its burrow, it cannot alert the player it belongs to of any threatening moves. When the player piece occupying the Muraqib square moves away from that square, the Muraqib should return to its lookout duties.

Extension 14

Introduce a new ‘Aqrab’ (Scorpion) option which can be added to any playing piece on the board. The Aqrab can only applied to one piece per player. Once applied, the piece symbol should change to a ‘$’ for a player 1 piece or ‘£’ for a player 2 piece. A piece chosen to be a Aqrab can move as normal around the board; however, when it is one square away from an opponent piece (in any direction), that opponent piece becomes paralysed and cannot move. This makes it vulnerable to be captured by any piece or by the Aqrab itself.

The Aqrab, however, can still be captured by any piece which can move from a location more than two squares away (in any direction). If the Aqrab moves away from a piece without capturing it, the piece is no longer paralysed and can move away as normal.

Extension 15

Introduce the concept of a ‘Quantum Mirza’. Add a new option ‘Q’ to the main menu to offer the player options to instantly swap their Mirza location with any of their other player pieces on the board. On selecting the Quantum menu, the player should be asked for the board location containing the piece to swap. The piece must belong to the current player. The program should use appropriate error handling to ensure that a valid piece is selected. The program should then swap the selected player piece and Mirza locations. The player turn should then continue as normal.