Project Micro-Charter

Project Name:
American Checkers Game

Vision:
To host the American Checker’s game on the public platform and make it accessible to everyone around the globe.

Project purpose:
To enable two players to play American checker’s game such that at the end either a player can win or lose or make a draw with opponent player.

Elevator pitch:
This application contains a unique and user-friendly GUI design which will make you feel comfortable while playing. It also allows users to use either drag and drop option or touch and place for making a move. And a selected piece gets highlighted based on selection.

Business value:
We implemented the project based on Object-Oriented Programming principles using Java Latest Version and Java Swing which makes this application sustain for a long run. We are implementing the design to handle heavy traffic to reduce crashes.

Customers and users:
Users – Players who wants to play Online American Checkers Game irrespective of age.
Customers – Anyone who is interested in hosting our application online (e.g : a website owner to host the game for users, professor for students, etc)

Metrics:
The success of this application will be measured using average numbers of games played per month.

Milestones:
Human playing American checkers with another human
Human playing American checkers with computer

Risks:
Competitive Risks (Having many competitors outside for American Checkers Game)
Operational Risks (IT System Risk by having poor internet connection)

User Stories

Describe all user stories using the following template: As a <role>, I want <goal> [so that <benefit>]

ID	User Story Name	User Story Description	Priority	Estimated effort (hours)	Actual effort (if completed)	Status (completed, toDo, inProgress)	Developer names
1	User Starting Checkers Game	The user starts the Application of the game then the board with checker's pieces placed in correct positions is displayed.	High	10	11	completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti
2	Player Moving Piece	The player makes a legal move.	High	10		inProgress	Sai Mukesh Parvathaneni, Hemanth Bandi
3	King promotion	When player's piece reaches last row, it promotes to king piece	Medium	10		toDo	
4	Capturing piece	When a player's piece hops over the opponent player's piece, he captures the opponent player's piece	High	10		toDo	
5	End of the game	When a player captures the opponent's last piece or when there is not valid move for the player or the players agree to a draw by mutual agreement, then the game should end	High	10		toDo	

Acceptance Criteria (AC)

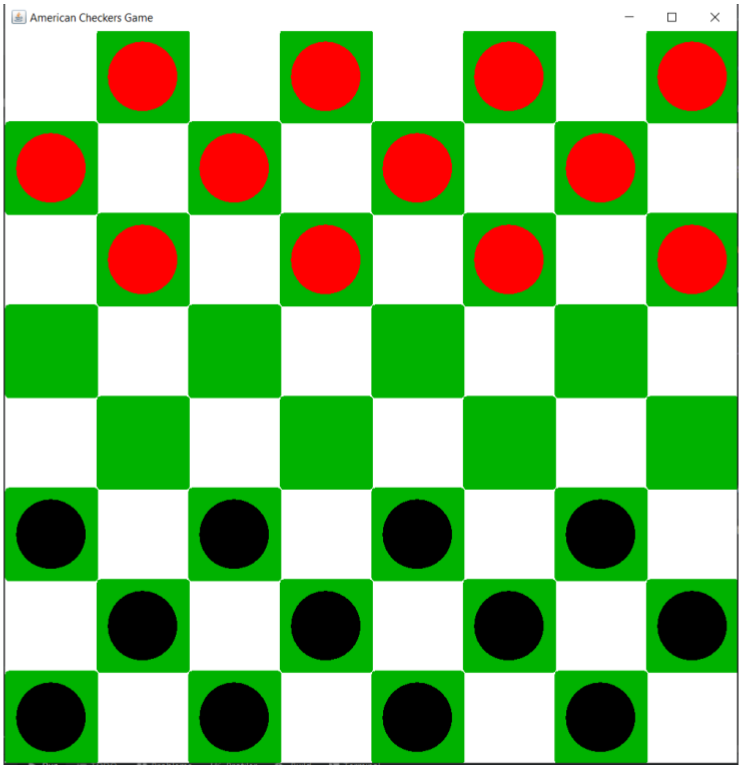
Describe all acceptance criteria using the Given-When-Then template.

User Story ID and Name	AC ID	Description of Acceptance Criterion	Status (completed, toDo, inProgress)	Developer Names
1 User Starting Checkers Game	1.1	**AC 1.1** Start of the game **Given** the game application **When** the user runs the application **Then** a checkers board is displayed with alternate white and green squares	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti
	1.2	**AC 1.2** Checkers piece placement on board **Given** the user runs the application **When** the checkers board is loaded **Then** the checkers pieces must be arranged in their correct position	Completed	Eeshwara Sai Tota, Venkata Subba Rao Inti

	1.3	**AC 1.3** Toss to determine who plays first **Given** the user runs the application **When** the checkers board is loaded **Then** the toss dialog box opens to determine who plays first	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi
2 Players Moving Piece	2.1	**AC 2.1** First Move by Black **Given** a checkers board with checker pieces **When** the black player tries to make the first move of the game **Then** the player black should be able move his piece to desired position	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi
	2.2	**AC 2.2** Invalid first move by red **Given** a checkerboard with checker pieces **When** player red tries to make the move **Then** player should not be able to move his piece to desired position	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti
	2.3	**AC 2.3** Making a Valid Move **Given** a player's turn to make a move **When** a player tries to make a valid move **Then** the player should be able to make a valid move.	Completed	Sai Mukesh Parvathaneni, Hemanth Bandi
	2.4	**AC 2.4** Making Invalid Move **Given** a player's turn to make a move **When** a player tries to make an invalid move **Then** the player should not be able to make an invalid move.	Completed	Sai Mukesh Parvathaneni, Hemanth Bandi
	2.5	**AC 2.5** Making Valid Move by **king** piece **Given** a player's turn to make a move **When** a player tries to make a valid move using king piece **Then** the player should be able to make a valid move.	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi
	2.6	**AC 2.6** Making an Invalid move by **king** piece **Given** a player's turn to make a move **When** a player tries to an invalid move by king piece **Then** the player should not be able to make an invalid move.	Completed	Sai Mukesh Parvathaneni, Inti Venkata Subba Rao
3. King Promotion	3.1	**AC 3.1** Promoting a normal piece to king piece **Given** a player's turn to move **When** the player's piece reaches last row **Then** the piece should be promoted to king piece	Completed	Sai Mukesh Parvathaneni, Inti Venkata Subba Rao
	3.2	**AC 3.2** Unsuccessful promotion of king **Given** a player's turn to move **When** the player's piece did not reach the last row **Then** the player's piece should not be promoted to king piece	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi
4. Capturing a Piece	4.1	**AC 4.1** Making a Hop over opponents' piece **Given** a player's turn to move **When** a player makes a valid hop over opponent's piece **Then** the player should be able to make the valid hop and capture the opponent's piece.	Completed	Sai Mukesh Parvathaneni, Inti Venkata Subba Rao, Hemanth Bandi

	4.2	**AC 4.2** Making an unsuccessful Hop over same color piece **Given** a player's turn to move **When** a player makes an invalid hop over same colored piece **Then** the player should not be able to make the invalid hop and capture the piece	Completed	Eeshwara Sai Tota, Hemanth Bandi
5. End of the Game	5.1	**AC 5.1** Winning the Game **Given** a player's turn to move **When** the player captures opponent's last piece **Then** the player wins the game and win message appears on the screen.	Completed	Sai Kishore Reddy Surabhi, Eeshwara Sai Tota
	5.2	**AC 5.2** Drawing the Game **Given** a player's turn to move **When** neither player captures the opponent's piece in last 40 moves **Then** the game ends in draw and draw appears on the screen.	Completed	Sai Mukesh Parvathaneni, Inti Venkata Subba Rao,
	5.3	**AC 5.3** Winning the Game with no legal moves left **Given** a player's turn to move **When** a player has no legal move to make **Then** the opponent wins and win message appears on the screen.	Completed	Eeshwara Sai Tota, Hemanth Bandi

Initial User Interface Design



Description:

- The above American Checkers Game Board contains 8x8 square board which includes 64 small grids.
- It contains 32 white grids and 32 green grids.
- It includes 12 Black pieces on one side of the board and 12 Red pieces on the other side of the board.
- All pieces should be placed in the dark green grids initially before starting the game.

Implementation Tasks

1. Summary of production code

Summarize how each user story/acceptance criterion is implemented in your production code (cl ass name and method name etc.)

User Story ID and Name	AC ID	Class Name(s)	Method Name(s)	Status (complete or not)	Developer Name(s)	Notes (optional)
1 User Starting Checkers Game	1.1	AmericanCheckersGUI	drawBoard(Graphics g)	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	Checks if board is visible or not
	1.2	AmericanCheckersGame, AmericanCheckersGUI	drawBoard(Graphics g), initGame()	Completed	Mukesh Parvatheneni, Hemanth Bandi	Checks if pieces are placed in correct position
	1.3	AmericanCheckersGUI	toss()	Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	To choose which player will start first

2 Players Moving Piece	2.1	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	Make sure first move is played by black

	2.2	AmericanCheckersGame AmericanCheckersGUI	validMove()	Completed	Mukesh Parvatheneni, Hemanth Bandi	When red tries to play the first move, it should not be allowed

	2.3	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	Any valid move should be allowed and turn should be changed.

	2.4	AmericanCheckersGame AmericanCheckersGUI	validMove()	Completed	Mukesh Parvatheneni, Hemanth Bandi	Invalid move should not be allowed and turn should remain same

	2.5	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			isKing()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	King piece should be allowed to move in all four diagonal ways

	2.6	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			isKing()
				Completed	Mukesh Parvatheneni, Hemanth Bandi	Making invalid move by King should be restricted

3. King Promotion	3.1	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			setKing()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	When normal piece reaches opponent first row, it should be promoted to king

	3.2	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	A normal piece not reaching opponent last row should remain a normal piece

4. Capturing a Piece	4.1	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
				Completed	Mukesh Parvatheneni, Hemanth Bandi	Killing a opponent piece should be allowed

	4.2	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			changeTurn()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	Killing self coloured piece should not be allowed

5. End of the Game	5.1	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
			setGameState()
				Completed	Mukesh Parvatheneni, Hemanth Bandi	When no opponent piece remains, he should be declared winner

	5.2	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
			setGameState()
				Completed	Eeshwara Sai Tota, Sai Kishore Reddy Surabhi, Venkata Subba Rao Inti	If no piece is captured in last 40 moves, then game should drawn

	5.3	AmericanCheckersGame
		AmericanCheckersGUI
			validMove()
			makeMove()
			changeTurn()
			setGameState()
				Completed	Mukesh Parvatheneni, Hemanth Bandi	When opponent has no legal moves, he should be declared win.

Summary of automated test code directly corresponding to some acceptance criteria

Summarize how user stories/acceptance criterion are tested by your test code (class name and method name).

User Story ID and Name	**Acceptance Criterion ID**	**Class Name (s) of the Test Code**	**Method Name(s) of the Test Code**	**Description of the Test Case (input & expected output)**	**Status**	**Developer Name(s)**
1 User Starting Checkers Game	1.1	CheckersBoardTest	testCheckersBoard()	Checks if board is visible or not	PASS	Eeshwara Sai Tota
	1.2	CheckersBoardTest	testCheckersBoardPiecePlacement()	Checks if pieces are placed in correct position	PASS	Sai Kishore Reddy Surabhi
2. Players Moving Piece	2.1	CheckersMoveTest	testFirstMoveByBlack()	Make sure first move is played by black	PASS	Eeshwara Sai Tota
	2.2	CheckersMoveTest	testFirstByRed()	When red tries to play the first move, it should not be allowed	PASS	Eeshwara Sai Tota

	2.3	CheckersMoveTest	testValidMove()	Any valid move should be allowed and turn should be changed.	PASS	Eeshwara Sai Tota
	2.4	CheckersMoveTest	testInvalidMove()	Invalid move should not be allowed and turn should remain same	PASS	Eeshwara Sai Tota
	2.5	CheckersMoveTest	testValidMoveByKing()	King piece should be allowed to move in all four diagonal ways	PASS	Eeshwara Sai Tota
	2.6	CheckersMoveTest	testInvalidMoveByKing()	Making invalid move by King should be restricted	PASS	Eeshwara Sai Tota
3. King Promotion	3.1	KingPromotionTest	testSuccessfulKingPromotion()	When normal piece reaches opponent first row, it should be promoted to king	PASS	Sai Kishore Reddy Surabhi
	3.2	KingPromotionTest	testUnSuccessfulKingPromotion()	A normal piece not reaching opponent last row should remain a normal piece	PASS	Sai Kishore Reddy Surabhi
4. Capturing a Piece	4.1	HopTest	TestValidHop()	Killing a opponent piece should be allowed	PASS	Inti Venkata Subba Rao
	4.2	HopTest	TestValidHop()	Killing self coloured piece should not be allowed	PASS	Inti Venkata Subba Rao
5. End of Game	5.1	EndGameTest	winPositionTest()	When no opponent piece remains, he should be declared winner	PASS	Sai Mukesh Parvathaneni
	5.2	EndGameTest	drawGamesTest()	If no piece is captured in last 40 moves, then game should drawn	PASS	Sai Mukesh Parvathaneni

6. Summary of manual test cases directly corresponding to some acceptance criteria

Summarize how user stories/acceptance criterion are tested manually

User Story ID and Name	Acceptance Criterion ID	Test Case Input	Test Oracle (Expected Output)	Status	Notes	Developer Name(s)
1	1.1	-	Checkers board	PASS	Checkers board should be display	Venkata Subba Rao Inti
	1.2	-	Piece arrangement	PASS	Checkers piece should be arranged correctly	Sai Mukesh Parvathaneni, Hemanth Bandi
	1.3	-	Player 1 or Player 2	PASS	Toss Dialog Box with toss winner	Eeshwara Sai Tota

7. Summary of other automated or manual tests not corresponding to the acceptance criteria

Number	Test Input	Expected Result	Class Name of the Test Code	Method Name of the Test Code	Status	Developer Name(s)
1	RED, BLACK	RED, BLACK	CheckersPiecePositionTest	testCheckersPieceColor()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
2	-	false, true	CheckersPiecePositionTest	testCheckersKingPiece()	PASS	Sai Kishore Reddy Surabhi, Hemanth Bandi
3	-	Move by black	CheckersMoveTest	testFirstMoveByBlack()	PASS	Sai Kishore Reddy Surabhi, Hemanth Bandi
4	-	No move by red	CheckersMoveTest	testFirstMoveByRed()	PASS	Sai Kishore Reddy Surabhi, Hemanth Bandi
5	-	Red piece killed	HopTest	BlackTestValidHop()	PASS	Sai Kishore Reddy Surabhi, Hemanth Bandi
6	-	Black piece killed	HopTest	RedTestValidHop()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
7	-	Red piece not killed	HopTest	BlackTestInvalidHop()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti

8	-	Black piece not killed	HopTest	BlackTestInvalidHop()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
9	-	Black king piece	KingPromotionTest	testBlackSuccessfulKingPromotion()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
10	-	Red king piece	KingPromotionTest	testRedSuccessfulKingPromotion()	PASS	Sai Mukesh Parvathaneni, Sai Kishore Reddy Surabhi
11	-	Black normal piece	KingPromotionTest	testBlackUnSuccessfulKingPromotion()	PASS	Sai Mukesh Parvathaneni, Sai Kishore Reddy Surabhi
12	-	Red normal piece	KingPromotionTest	testRedUnSuccessfulKingPromotion()	PASS	Sai Mukesh Parvathaneni, Sai Kishore Reddy Surabhi
13	-	Red wins	EndGameTest	redWinPosition()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
14	-	Black Wins	EndGameTest	blackWinPosition()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti
15	-	Game Draws	EndGameTest	drawGameTest()	PASS	Eeshwara Sai Tota, Venkata Subba Rao Inti

Meeting Minutes

Report the minutes of all meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming session.

Date	Time and Duration	Place	Participant Names	Purpose of the Meeting	Specific Action Items
10-14-2021	5:30 PM, 3 Hours	Zoom Video Meeting	Sai Kishore Reddy Surabhi, Eeshwara Sai Tota, Venkata Subba Rao Inti, Sai Mukesh Parvathaneni, Hemanth Bandi	Analyze and Design human vs human gameplay	• Piece should be highlighted when selected. • Normal Piece Movement for black and red
10-21-2021	5:30 PM, 4 Hours	520 E Study Room	Sai Kishore Reddy Surabhi, Eeshwara Sai Tota, Venkata Subba Rao Inti, Sai Mukesh Parvathaneni, Hemanth Bandi	Implement the production code in java for king promotion.	• Implement UI for king piece • Implement production code for king promotion when reached opponent's first row • Valid King piece movement
10-28-2021	5:30 PM, 4 Hours	Zoom Video Meeting	Sai Kishore Reddy Surabhi, Eeshwara Sai Tota, Venkata Subba Rao Inti, Sai Mukesh Parvathaneni, Hemanth Bandi	Implement production code for piece killing and end game.	• Implement production code for killing of opponent's piece • Show message when game has ended.
11-11-2021	5:30 PM, 4 Hours	Zoom Video Meeting	Sai Kishore Reddy Surabhi, Eeshwara Sai Tota, Venkata Subba Rao Inti, Sai Mukesh Parvathaneni, Hemanth Bandi	Implement toss functionality and test code	• Implemented production code for toss functionality. • Implemented test code and improved code coverage.

Buddy Ratings

Rating receiver

Rating giver

	Eeshwara Sai Tota	Inti Venkata Subba Rao	Sai Kishore Reddy Surabhi	Sai Mukesh Parvathaneni	Hemanth Bandi
Eeshwara Sai Tota	X	1	1	1	1
Inti Venkata Subba Rao	1	X	1	1	1
Sai Kishore Reddy Surabhi	1	1	X	1	1
Sai Mukesh Parvathaneni	1	1	1	X	1
Hemanth Bandi	1	1	1	1	X
Average	1	1	1	1	1