

		Sprint #	1	2	3	4	5	6	Total:
	Description	Estimate (hours)	13	11	19	15	17	17	92 hours
	Create a model consisting of top-level Block class	1	0	0	0	0	0	0	
	Render a blank window using slick2D	2	0	0	0	0	0	0	
	Create a view and BlockComponent class mapped to the model	1	0	0	0	0	0	0	
	Set up game states (from slick) that handle the main game loop & timer event	4	0	0	0	0	0	0	
	Add render() method to MainView to draw the Block in its current position	1	0	0	0	0	0	0	
	Code the logic to stop the block from falling once it reaches a barrier	2	0	4	0	1	0	0	
	Add Tetrimino model & component classes that hold references to 4 Blocks	2	0	0	0	0	0	0	
Sprint 1	<i>Render a single Tetrimino falling down the screen.</i>								
	Create a controller that checks user input using a redirected update() method	0	2	0	0	0	0	0	
	Add code that checks for L/R arrow keypresses and updates the Tetrimino position	0	2	0	0	0	0	0	
	Add rotate() method to Tetrimino by manipulating its array of points w/ pivot block	0	4	0	0	0	0	0	
	Add support for soft drops via dropSpeed variable in the model synced w/ timer event	0	2	0	0	2	0	0	
	Add support for hard drops by moving Tetrimino down as far as possible	0	1	0	0	1	0	0	
	Code ability to hold Tetrimino and switch out for next one in the lineup	0	0	0	0	4	0	0	
Sprint 2	<i>Control the Tetrimino as it falls.</i>								
	Add a "grab bag" that randomly generates the next Tetrimino	0	0	1	0	0	0	0	
	Create a GameBoard class and its component class to hold static Blocks	0	0	1	0	0	0	0	
	Logic to convert Tetrimino into its constituent Blocks and place in board's 2D array	0	0	1	0	0	0	0	
	Add render() method to GameBoardComponent called from MainView to draw Blocks	0	0	2	0	0	0	0	
	Add ScoreBoard class and component with a reference to the player's points	0	0	2	0	0	0	0	
	Add logic to GameBoard to check for lines to clear whenever Tetrimino is placed	0	0	4	0	0	0	0	
	Code logic to check for Tetriminos touching the top of the screen (Game Over state)	0	0	2	0	0	0	0	
	Render ScoreBoard text on the screen as a simple string of text	0	0	2	0	0	0	0	
Sprint 3	<i>Tetriminos stack and lines can be cleared. Scoreboard/points update accordingly.</i>								
	Create new MainMenu state (immediately loaded when the player launches the game)	0	0	0	4	0	0	0	
	Create a Resources class responsible for collecting game assets	0	0	0	4	0	0	0	
	Find .png files for game images (including colored Tetris blocks)	0	0	0	1	0	1	0	
	Add a "play" button that the controller checks for a click in order to change state	0	0	0	2	0	0	0	
	Draw Tetriminos using the new .pngs instead of old Slick2D rectangular shapes	0	0	0	1	0	0	0	
	Pressing ESC on game over screen takes user back to the main menu	0	0	0	1	0	0	0	
	Loop background music	0	0	0	2	0	0	0	
Sprint 4	<i>GUI polish: Main menu with button, prettier Tetrimino blocks, backgrounds, music.</i>								
	Create component class for the side panel to display nextTetrimino and heldTetrimino	0	0	0	0	4	0	0	
	Implement game difficulty scaling as the player progresses	0	0	0	0	2	0	0	
	Update logic of ScoreBoard to increase points for performing drops	0	0	0	0	1	0	0	
	Add quit button to the main menu	0	0	0	0	1	0	0	
	Play a sound whenever the user clears a line	0	0	0	0	1	0	0	
Sprint 5	<i>Advanced Tetris mechanics &amp; additional integration with the GUI.</i>								
	Add code to read final score into a text file once the game is over	0	0	0	0	0	1	0	
	Add a HighScoresState whose render() method sorts these scores and draws them	0	0	0	0	0	4	0	
	Add "high scores" button to the main menu and add controller logic to switch state.	0	0	0	0	0	1	0	
	Research native awt/Slick font compatibility and load custom fonts into the game	0	0	0	0	0	4	0	
	Improve high scores screen graphics, use arcade-esque font & custom background	0	0	0	0	0	2	0	
	Overhaul game's side panel to have nicer components for game elements	0	0	0	0	0	4	0	
Sprint 6	<i>Support for leaderboards (high scores) and GUI improvements.</i>								