Functional Requirements Tested

ID	Description	Test Class
FR_SCORE	A score must be calculated based on how well the player did in the game and in the end exam.	ScoreCalculatorTest
FR_HIGHSCORE	When a player achieves a new high score the current high score should be updated.	LeaderboardManagerTest
FR_TIME_PASS	There should be a passing of time while the player is interacting with the game or using energy.	GameStateTest
FR_FEEDBACK	Each interaction should show the user how this has affected their time and energy, with a brief pop up or an animation or sound cue.	Manual Test
FR_MOVEMENT	The map should be able to be moved around inside as an avatar representing the player.	Manual Test
FR_ENERGY_BAR	The game should have an energy bar displayed that contains a certain amount of energy in the day.	Manual Test
FR_ENERGY_USE	Energy can be spent on interactions which will make the energy bar total fall in increments for that interaction.	GameStateTest
FR_END_GAME	The system should allow the player to take the exam after 7 days and therefore obtain a score. The system should end the game in the process after the final exam showcasing a score.	GameStateTest + ScoreCalculatorTest +Manual Test
FR_ENERGY_EMPTY	Energy must be depleted after 16 hours, forcing player to sleep	Manual Test
FR_AVATAR_USE	The avatar should be able to interact with all other system functionality in order to reach the end game.	Manual Test
FR_CONTROLS	The system should allow the modifications of controls to play the game.	N/A
FR_SLEEP	The game should allow the player to sleep at a designated location in campus east.	Manual Test
FR_RECREATION	The game should allow the player to do recreational activities through interactions.	Manual Test
FR_STUDYING	The game should allow the player to study through interactions. There should be an expected amount of studying to do well in the final exam and this should be tracked in some way.	Manual Test + ScoreCalculatorTest
FR_FAILURE	The system should provide feedback if an error was detected or attempt to correct itself to keep running if something abnormal occurs.	N/A
FR_SCORE_CHANGE	The system should change the score based on specific conditions associated with what interactions were performed.	ScoreCalculatorTest
FR_TIME	The system should have a time displayed that passes after interactions.	Manual Test

FR_LEADERBOARD	Presenting the top 10 scores on the leaderboard with the playernames and keeping it up to date.	LeaderboardManagerTest
FR_STREAKS	Awards achievements for repeating an activity in the gameplay resulting in extra points.	ScoreCalculatorTest
FR_COUNTERS	Keep track of how many times each activity is completed	GameStateTest
FR_TIME_PASS	Some time shall pass with every interaction in a specific increment.	GameStateTest
FR_COUNTERS	Keep track of how much each activity is performed each day.	GameStateTest
FR_POINTS	Points are awarded through a set algorithm based on the activities completed in gameplay.	ScoreCalculatorTest
FR_IDLE	If the system is idle it should prevent a situation where the score of the player is affected.	Manual Test + ScoreCalculatorTest