

Manual Test Cases

We are unable to test elements of the game that attempt to load in shaders, thus methods in classes such as MenuScreen, MyGameGdx or GameScreen cannot be loaded. We extracted as many classes as possible from it.

Timer

The timer is tied to the UI and render function of the GameScreen class so cannot be automatically tested using unit tests. Therefore, we had to manually test it by loading the game and using an external timer to ensure that it was ticking at the correct rate.

During this, we also tested that the timer did not increase while the game was paused.
The timer worked as we were expecting it to so these tests passed (UR_TIMER)

User Interface

Main Menu UI

No save file



Visual inspection of the main menu UI (**UR_MENU**, **UR_ENDLESS_MODE** and **UR_SCENARIO_MODE**)

When the game is played and there is no save file, the 4 buttons in the above image are displayed and can be clicked by the user. (**UR_SAVE_GAME**)

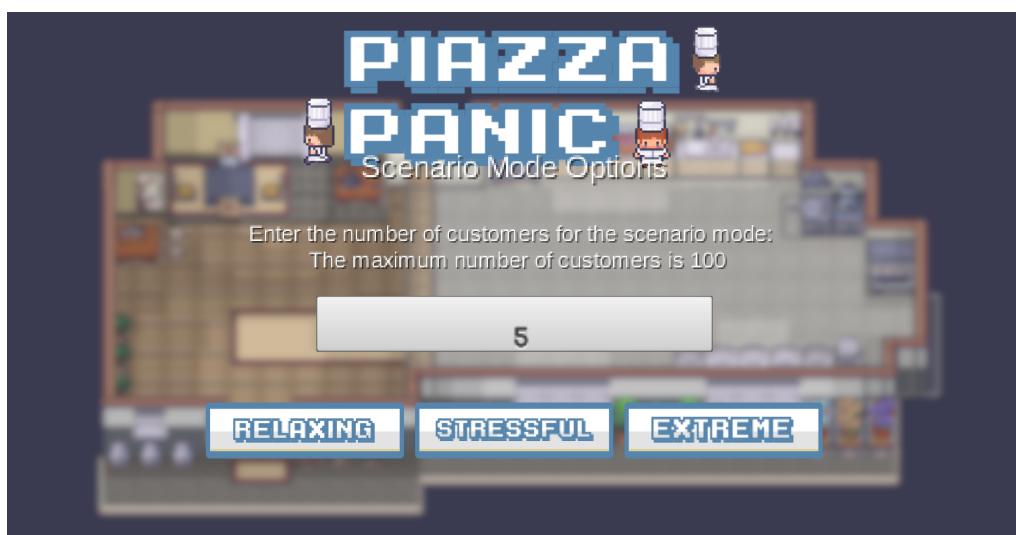
Save file



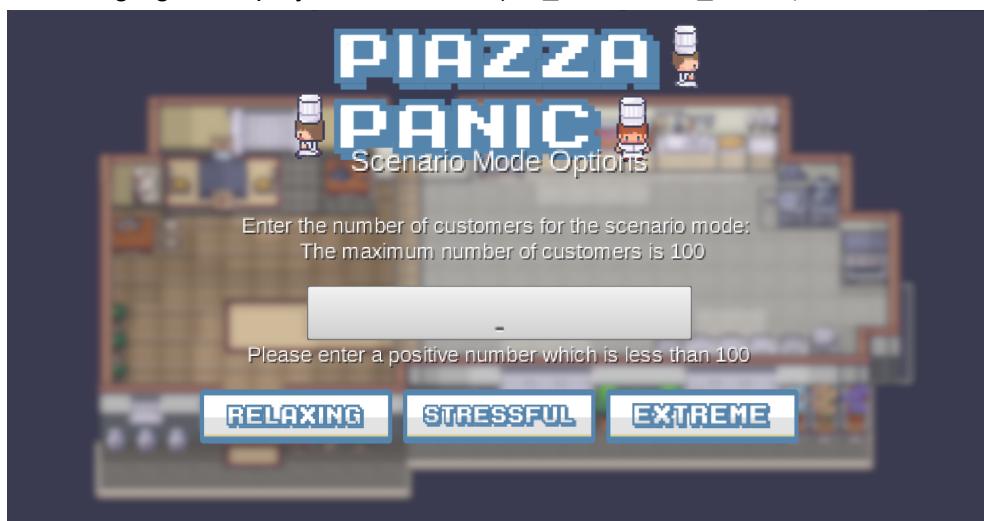
When there is a save file, the load game button is displayed which will cause the saved state of the game to be loaded. (UR_SAVE_GAME, FR_SAVESTATE, NFR_SAVE_GAME)
The game is able to be paused and saved at any time from the pause menu relating to UR_SAVE_GAME



Scenario Mode Configuration Screen UI



When the user clicks the Scenario Game button on the main screen, they are taken to the above screen where they can enter the number of customers for the scenario mode. If their input is invalid (negative number, number greater than 100, non-numeric characters), a message gets displayed on screen: (UR_SCENARIO_MODE)



Unit tests have been written for the function which checks this input.

Gamescreen UI



The map must be verified to be correct visually. (**UR_GAME_MAP**)

Game loads up correctly and all classes are loaded correctly. UR_WORKSTATIONS.

Masterchef, SaveState



Chef movement and swapping must get swapped manually by clicking around and hitting number keys. (**UR_COOK_CONTROLS**, **UR_CHEF_CONTROLS**, **UR_NUM_OF_CHEFS**)
Not all parts of the loading in the save can be tested so manual testing is required later.

Difficulty must be playtested to make sure its playable and setting changes have gone through. (**UR_DIFFICULTY**).



Pause menu must be visually inspected and the buttons must be pressed to verify they are working. (UR_PAUSE_MENU)



Customers leave the play space and despawn correctly.
Not practical to test and is easy to discover in play testing

NFR_SCENARIO_TIME was tested manually by playing the game several times and getting an average.

NFR_INSTRUCTIONS was done by asking friends and family to read the instructions and verify if they could play the game with little hints afterwards

NFR_PLATFORMS was done by loading the game up on different devices.

NFR_LAUNCH_TIME was done by timing the jar files launch several times

Frustration PowerUp

The powerup must stop the frustration from going up, and thus stop the patience going down for a total of 60 seconds, as testing does not allow time waiting or scheduling it cannot be tested

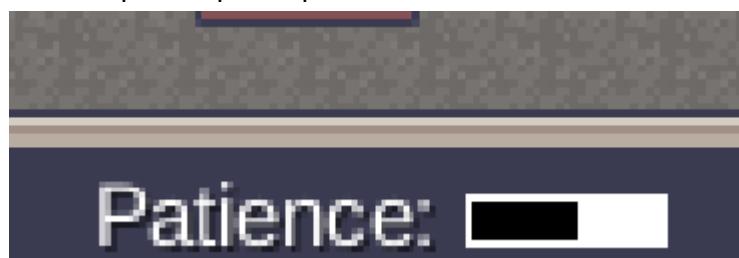


- The patience at the time the powerup was purchased



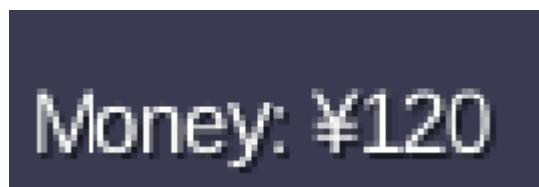
- The patience 60 seconds

after the powerup was purchased

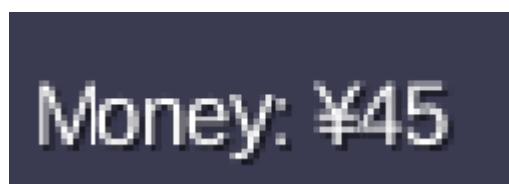


- The patience 120 seconds after

the powerup was purchased, and 60 seconds after the powerup should have ran out



- The money prior to purchasing the powerup



- The money after purchasing the powerup, 75 coins were taken from the total, which is the cost of the frustration powerup

From this you can see the powerup is working, as the patience remained the same for the first 60 seconds, and then continued to decrease once the powerup had expired

UR_POWERUP This was tested manually by timing 1 minute while the powerup was running, and watching the patience metre

UR_SPEND_EARNINGS This was manually tested by watching the money counter at the bottom while purchasing the powerup

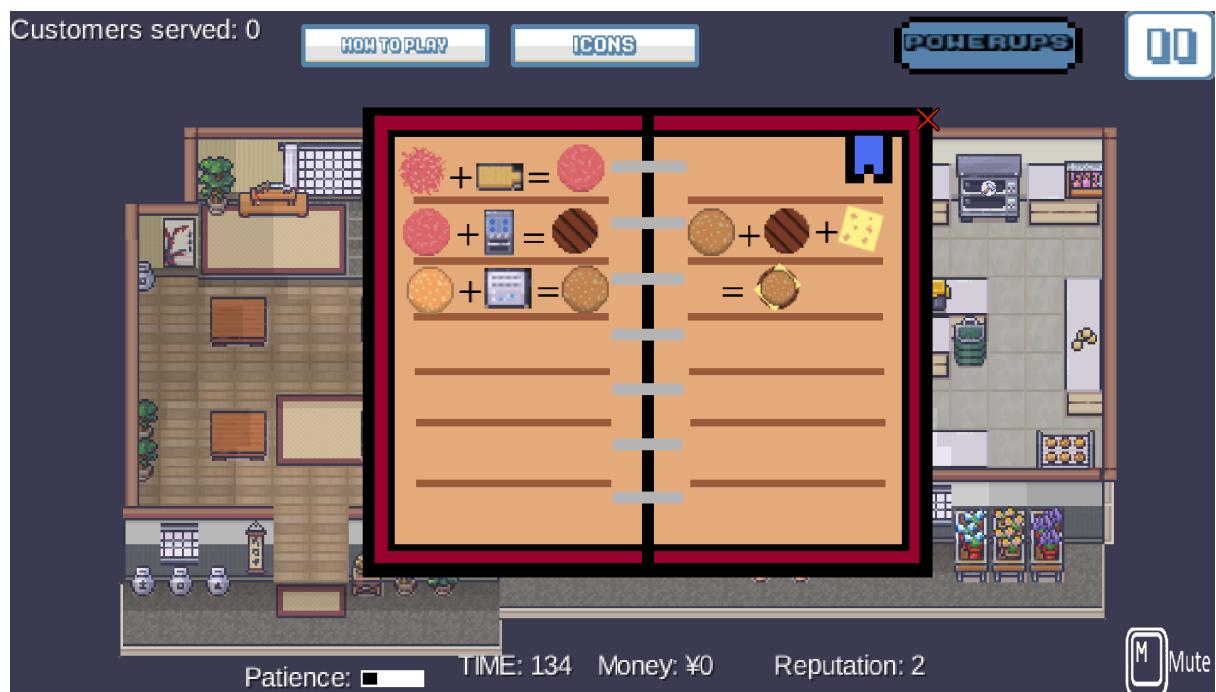
NFR_FRUSTRATION_STALL_POWERUP This was tested manually by watching the patience bar, denoting the frustration level

Test Show Recipe Screen

When the orders presented by the customers are clicked a recipe book should be displayed, showing the steps to make the order.



- The order icons appear when the customers approach the counter



-After the burger icon is clicked the recipe book for the burger appears on the screen



- After the recipe close button is clicked, the recipe book disappears from the screen

UR_HOW_TO_PLAY This is tested by checking every recipe book asset and verifying with other group members that they are correct and direct the user in the right direction. This is a part of the UR_HOW_TO_PLAY requirement with the other part being the instructions and controls

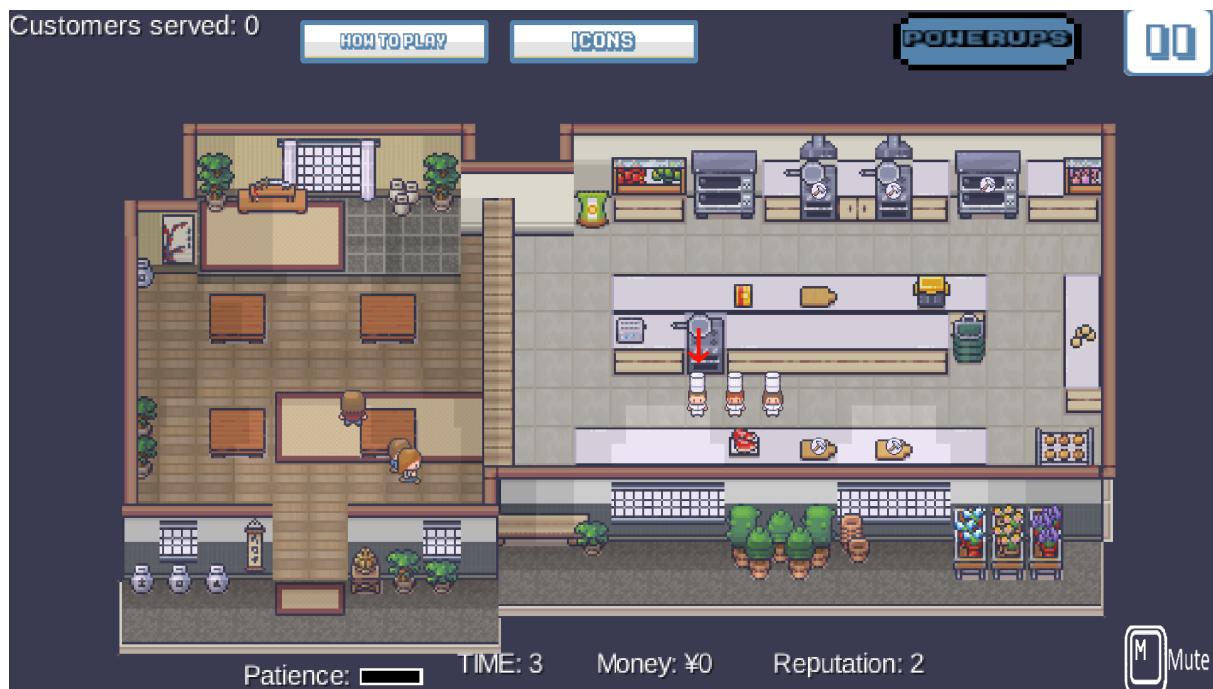
FR_HOW_TO_PLAY_BUTTON This is tested as shown in the images, using the icon buttons, when they are pressed the recipe book images appear

FR_RECIPES This is tested by confirming all the recipes were accurate and the correct recipes were displayed when the order icons were clicked

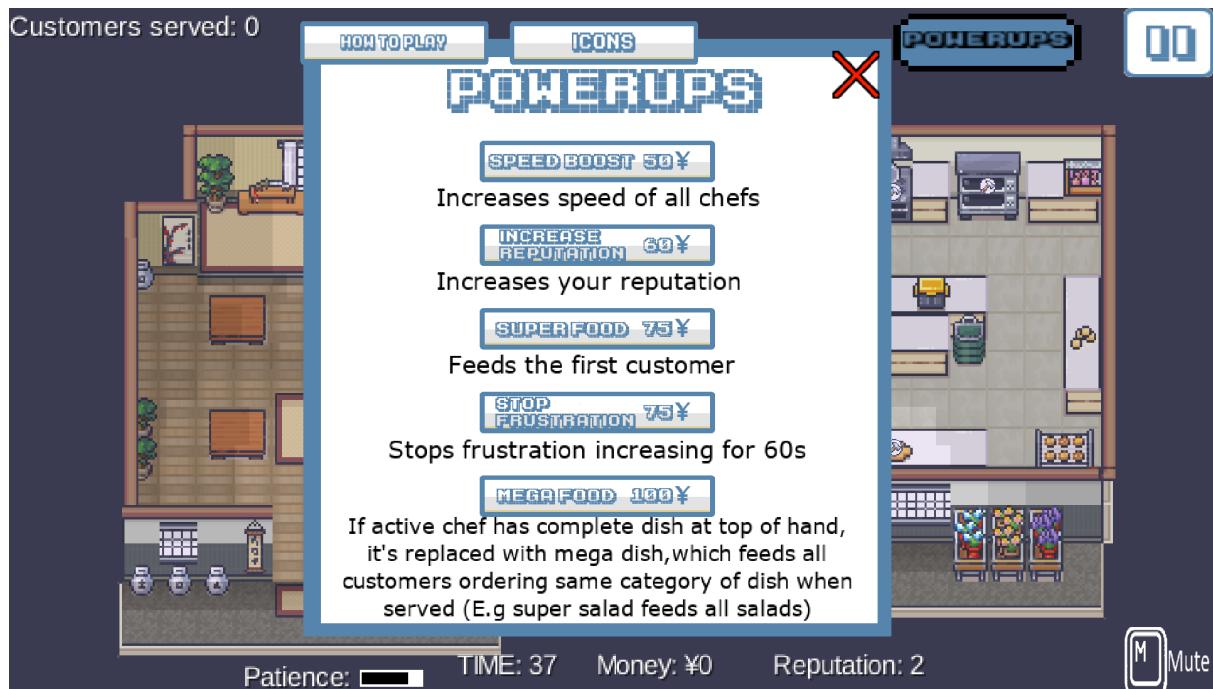
FR_ASSETS The assets are all hand created and implemented correctly into the game to make it more enjoyable

NFR_INSTRUCTIONS The recipe assets provide accurate and complete instructions on how to create all of the different recipes

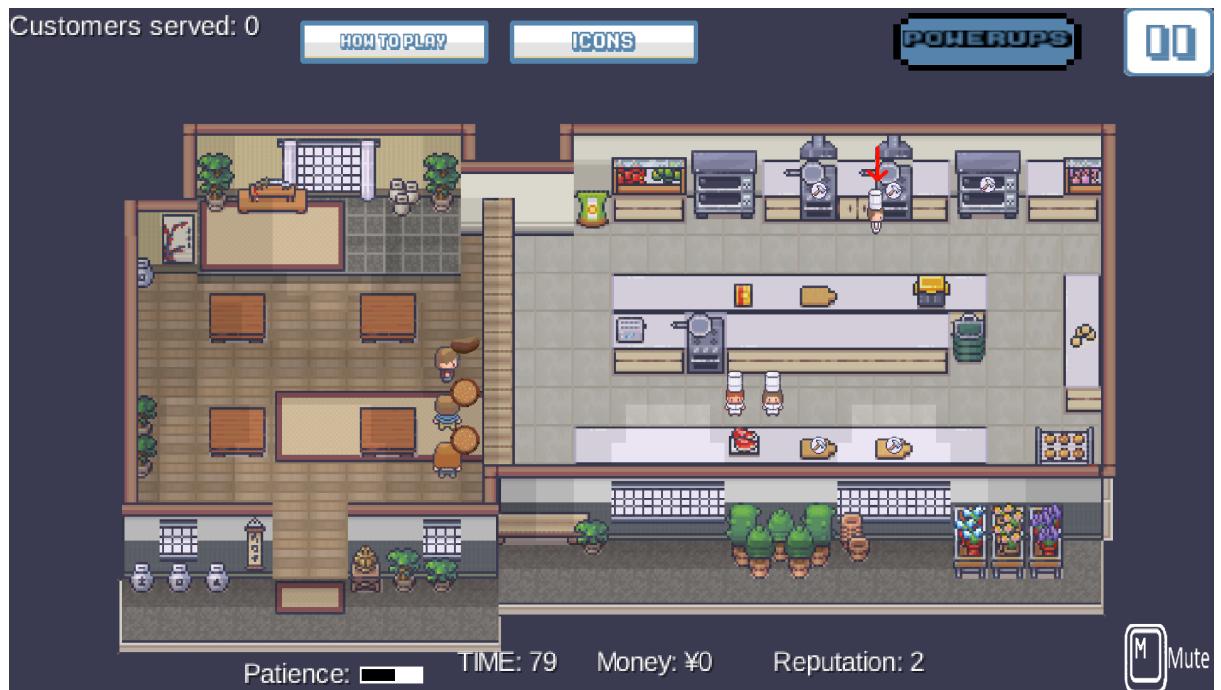
Test Powerup Menu Shown



- The game screen prior to the powerup button being pressed



- The game screen after the powerup button has been pressed



- The game screen after the exit button for the powerup menu has been pressed

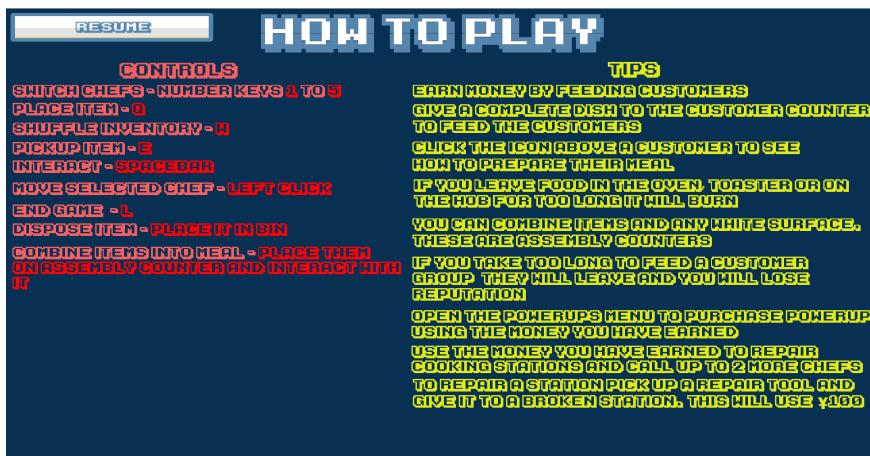
UR_POWERUP This part of the powerups was tested by opening the menu, and verifying every asset loaded in correctly and in the right position, with no errors being caused.

When the game screen is loaded, the UI features (text labels, buttons, and progress bars) all load and are displayed in the correct places on the screen as seen below:



How to play image

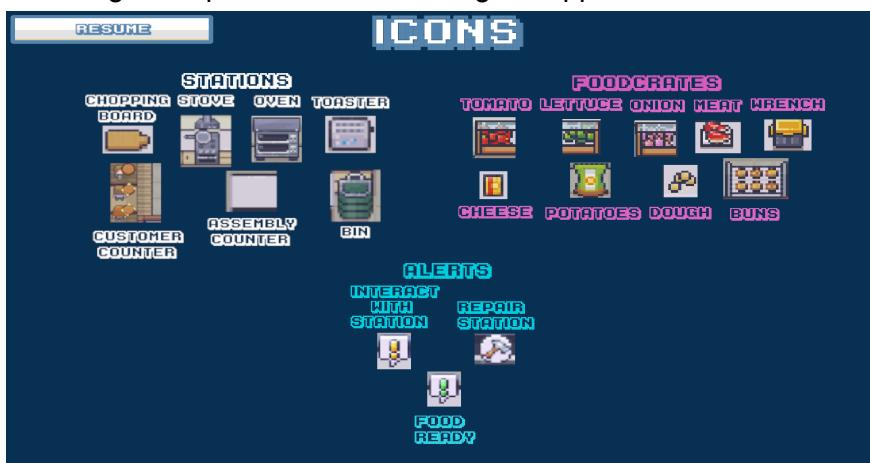
When the how to play button is clicked, the correct image is displayed on screen and the game update functions do not get run so customer patience stops like it does on the pause screen:



Pressing the resume button closes the image.

Icons image

When the icons button is clicked, the image showing the meaning of each icon is displayed and the game update functions also get stopped while it is visible:



Additional Chef Button and Label

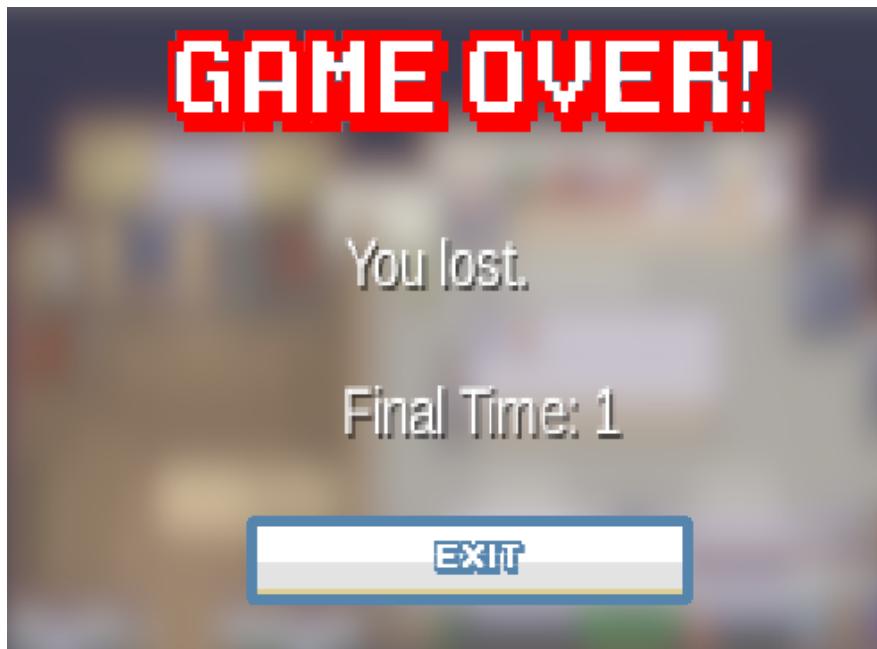
If the user attempts to call more chefs when they cannot afford them, an error message gets displayed beneath the call chef button:



If the user reaches the chef limit (5) the message below the button tells them that they cannot purchase any more chefs:



End Game Screen



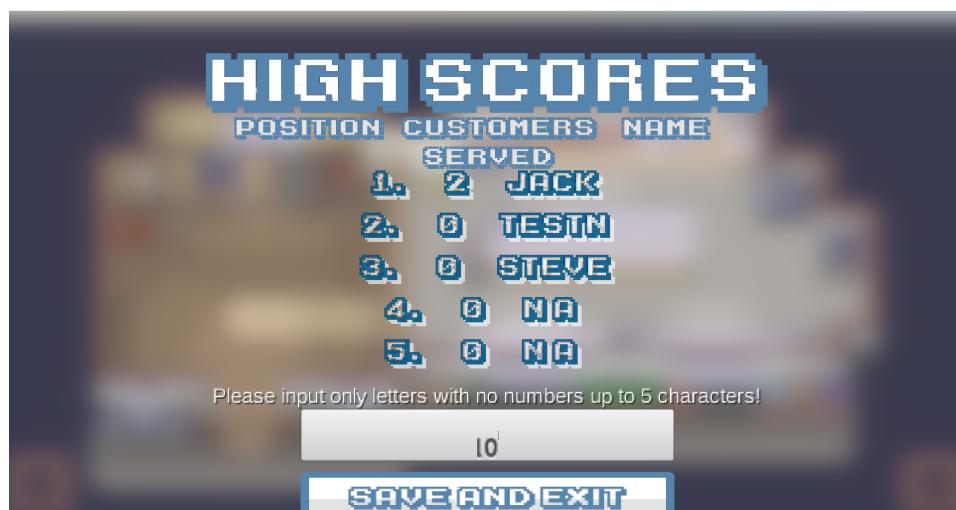
Loss screen. This gets displayed when the user runs out of reputation points.



Victory screen. This screen is displayed if the user completes the scenario mode.

Leaderboard UI

When the player loses all their reputation points in the endless mode and presses the exit button on the end game screen, the leaderboard screen is displayed for them:



When they enter an invalid name (non alphabetic). Unit tests have been written which check this input..