
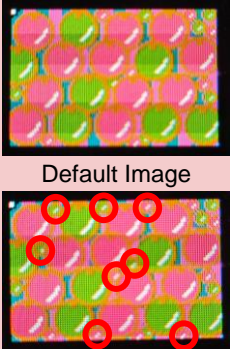

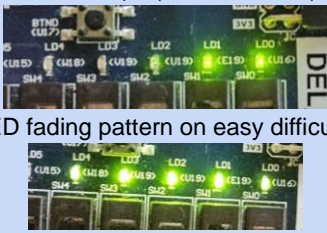




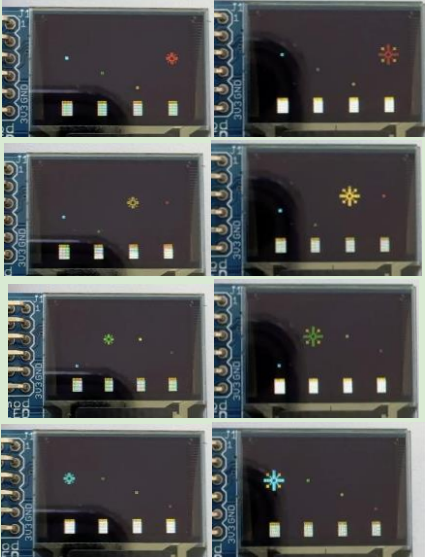


Group ID:	Members	Matriculation Number
S2_15	A. Tang Yuan Xing Aloysius B. Goh Eng Hui, Jeremy C. Neo Jun Qiao D. Tan Jie Hao	

PERSONAL AND TEAM IMPROVEMENTS		
Student and Improvement Name	Improvement Description	Images / Photos
S2_15 "Spot It!"	<p>"Spot It!" is a simple yet mind engaging game of "spot the difference". It starts from the main menu where the user can either enter the <b>difficulty settings</b> or start the game immediately (easy by default). In the game itself, the right OLED will display the user screen while the left OLED will display the default image. A <b>random number of items</b> will be <b>different</b> from the default image whenever the game starts and this number is <b>displayed</b> on the <b>7-segment</b>. On easy mode, there is a <b>hot and cold system</b> displayed on led[0] to led[4]. When the user clicks on the anomaly, the leds will fade away and the 7-segment will show a scrolling "correct". Once all the <b>anomalies</b> are <b>found</b>, an <b>ending screen</b> will appear and the user can press btnC to go back to the main menu. btnC can also be pressed anytime during the game to go back to the main menu.</p>	
Student A: Tang Yuan Xing Aloysius "Game Graphics"	<p>Aloysius coded the game screen, the hiding/showing of anomalies and the default screen. Both screens contain multiple layers of big and small apples. On the game screen, the anomalies include the <b>changing of apple colours</b>, two from red to green and two from green to red, and the <b>displaying of extra apples</b>, two of them are green and two of them red. When the user selects start, the <b>game screen (right OLED)</b> will be displayed. At the same time, the random 8-bit number generated by Jun Qiao will be fed into the module, causing the 8 differences to be present (0) or absent (1) on the game screen depending on the value. For example, if the 8-bit number is <b>0010_1110</b>, only the <b>anomalies</b> corresponding to the <b>0th, 4th, 6th and 7th bit</b> will be <b>present</b>. The <b>default screen</b> will also be shown on the <b>left OLED</b> for the user to refer to when playing the game.</p>	 <p>Default Image</p> <p>All Anomalies Shown</p>
Student B: Goh Eng Hui, Jeremy "Cursor & LED Functionality"	<p>Jeremy coded the cursor movement and the functionality of the LEDs. The cursor movement is tied to the position of the mouse, with the limits of the mouse movement mapped to the pixel coordinates on the OLED. Upon a <b>mouse click</b>, the <b>cursor turns yellow</b>.</p> <p>When an unselected <b>anomaly is selected</b>, <b>LEDs[0] to [4]</b> progressively <b>fade</b> out from left to right (on easy difficulty) or fade in and out, in a different pattern on hard difficulty. The duration of the pattern is equal to that of the scrolling on the 7-segment display. The fade in/out is controlled by an 8-bit 25MHz PWM signal.</p> <p>In easy mode, the LEDs[0] to [4] are used to display the <b>"hot and cold"</b> system. When the cursor is moved <b>closer to</b> an unselected <b>anomaly</b>, the <b>LEDs</b> progressively <b>light up</b> more brightly, and <b>start flashing</b> when the cursor is on the anomaly. Once an</p>	 <p>Default cursor (left) and on click (right)</p>  <p>LED fading pattern on easy difficulty</p> <p>LED fading pattern on hard difficulty</p>

	<p>anomaly has been selected, the LEDs will no longer light up if it is selected again.</p> <p>Jeremy also implemented the <b>multiplexing</b> module for handling the various screens in use throughout the bitstream/program.</p>	 <p>LEDs lighting up for hot-and-cold system</p>
<p><b>Student C:</b> <b>Neo Jun Qiao</b> “Random Number Generation &amp; 7-segment”</p>	<p>Jun Qiao coded the <b>pseudo number generator</b> that generates random 8-bit values that will <b>never be 1111_1111</b> when the user starts the game. This number is also fed into the <b>seven-segment module to count</b> the number of <b>differences</b> left for the user to find. After the users find a difference, the number on the seven-segment module is <b>updated in real time</b>.</p> <p>Jun Qiao also implemented <b>rolling messages</b> on the seven-segment display to keep track of the status of the game. At the main menu, the message <b>‘spot it’</b> will be displayed as it is the name of our game. When the game starts, the seven segment will change to display the number of differences left in the game for the users to find. When the users select a <b>wrong spot</b>, a message <b>‘try again’</b> will be displayed and the number of differences left will remain the same. When the users select a <b>correct difference</b> spot, a message <b>‘correct’</b> will be displayed and the <b>number</b> of differences left will <b>decrease by 1</b>. When the number of differences left hit 0, the game will end and a message <b>‘good job’</b> will be displayed.</p> <p>When the reset button is pressed, the seven segment will change the display to ‘spot it’ as the game <b>returns to the main menu</b> and the pseudo number generator will start to <b>randomise</b> the number for the <b>next game</b>.</p>	 <p>Message “SPOT IT”</p> <p>Message “TRY AGAIN”</p> <p>Message “correct”</p> <p>6 differences remaining</p> <p>Message “GOOD JOB”</p>
<p><b>Student D:</b> <b>Tan Jie Hao</b> “Menu, Difficulty Screen Designs; End Screen Animation; Button Presses &amp; Screen Navigation logic”</p>	<p>Jie Hao coded the respective transition <b>screens</b> outside of the game itself. The main screen incorporates both handwritten and typewritten fonts. Before Gameplay, users are prompted to set the desired game difficulty. <b>PushButton (btn) inputs</b> were programmed for navigation. To save on LUTs, both the Main Screen and Difficulty Setting Screens have highlighted options for an ease of Gameplay and potential for expansion. Jie Hao implemented the multiplexing module for handling the various screens in use throughout the bitstream/program. When the game is completed, Jie Hao has also implemented an <b>ending animated fireworks display</b> that varies based on the selected game difficulty setting.</p>  <p>Main game screen</p>  <p>Difficulty Setting screen with Easy(left) &amp; Hard(right) settings selected</p>	 <p>Fireworks for Easy(left) &amp; Hard(right) settings respectively</p>

Group ID:	Members	Matriculation Number
S2_15	E. Tang Yuan Xing Aloysius F. Goh Eng Hui, Jeremy G. Neo Jun Qiao H. Tan Jie Hao	

References:

1. <https://www.jagranjosh.com/general-knowledge/super-fun-spot-the-difference-puzzles-to-test-your-visual-skills-1693331613-1>
2. [https://pikbest.com/png-images/qianku-color-firework-pixel-painting\\_2250998.html](https://pikbest.com/png-images/qianku-color-firework-pixel-painting_2250998.html)
3. [https://github.com/newdigate/rgb565\\_colors?tab=readme-ov-file](https://github.com/newdigate/rgb565_colors?tab=readme-ov-file)