|  |
| --- |
| Task Complete  Task needs work – may be bugged  Task not reached   * Task details and notes   Task reworked |

|  |  |
| --- | --- |
| Bitmaps  (Week One) | * Research generic face   + Face found and made into bit/depth map * Determine how differences between face proportions will affect comparison percentage   + Depth map helps deal with proportions, depth map will be tweaked in the future through code to deal with faces of varying size * Determine how facial features should be dealt with   + Facial features appear brighter in a depth map, faces with glasses will not be considered in the facial recognition system until after week 2 * Create forward facing generic face bit/depth map   + Sample face in documentation directory * Determine how much of the face must be used to compare each face, and how to isolate the face from its background   + Top of image – Upper extremity of the forehead   + Bottom of image – Lower extremity of chin   + Left of image – Rightmost extremity of cheekbone, does not include ear   + Right of image – Leftmost extremity of cheekbone, does not include ear * Determine how to convert an image into bit/depth map * Determine how different types of images compare to created bit/depth maps * Test bitmap comparison percentages, determine appropriate percentage threshold to establish similar images |
| ~~Comparisons~~  Searching for new method  (Week Two) | * Test generic facing face against person test face * Test algorithm of bitmap face comparisons until percentage threshold is met * Test face comparison algorithm with inanimate objects to see how it responds * Find new method of identifying face   + Created method to identify pixels with an RGB value matching a skin tone   + Highlighted the skin-toned pixel with a solid color to indicate pixel identified |
| Week Three  The week pixels were bounded  (Not much accomplished) | * Isolate “face” through color clusters in the image and surround with a box   + Bounding box is able to bound the colors but not the face, must make algorithm that catches clusters and not just individual pixels * Use box to crop image and create new image to be analyzed   + Image is cropped from original and is analyzed for presence of a face   + If background holds skin pixels, it is possible tests will fail due to those extra pixels` |
| Week Four  The week of revisions | * Get images of celebrities/people * Make image iterator * Iterate through every picture and apply face recognition tests * Identify unique features of a face (Not started) |
| Week Five  Confused | * Identify unique features of a face (Start) * Improve tests to better face recognition (there’s a bug where it’ll box everything, not just the face) * Begin creating tests to search for unique face features |
| Week Six  To get back on track | * Have concrete methods of identifying uniqueness of a face in code * Begin tests to remember faces * Have a good amount of tests pass face identification * If time allows it, improve face identification algorithm |
| Week Seven  Cutting it close | * Have a good amount of progress into remembering a face * Have face identification working to its best * Have at least one face be identified across pictures |
| Week Eight  Time to wrap up | * Complete face identification * Identify specific face across faces * Begin presentation |