Henry Dyer
3010 Project Design Document

1. For my project I plan on implementing a more advanced sudoku game using a graphical user interface as well as a computer vision framework. The game will have all the standard features, different difficultly levels can be selected without uploading an image however, these options will be hardcoded in due to the difficulty of having the computer efficiently generate boards. Users will have the ability to add notes to each square, ask for a hint, and will get visual feedback if they input a move that is incorrect. For the computer vision part of it users will have the ability to select a file of a photo of a sudoku game, be it from a newspaper or a screenshot of a website and the application will read the image and give the user the option to either continue playing from the state of the game uploaded in the photo or to have the computer solve the rest of it and return the image with the correct values filled in.

2. I will implement this project completely in C++ using OpenCV as my computer vision framework and Qt for the GUI. I would consider myself very familiar with C++ and will continue to learn more of its intricacies over the course of this class. Currently I am very familiar with OpenCV, but I am taking an online course on it now and have a pretty good idea of how I will be able to implement it for this project. I have never used Qt and adding the GUI will be the last big part of my project but since we will be covering it in class, I am confident I will be able to pick up on it and will also have resources available should I run into any issues. To test my application, I will use catch.hpp like we did in class. I will have a test board that will have certain squares that are “legal” moves and some that are “illegal”. I will test these different squares against the methods that will determine whether the row, column, and box finds it to be a legal move.

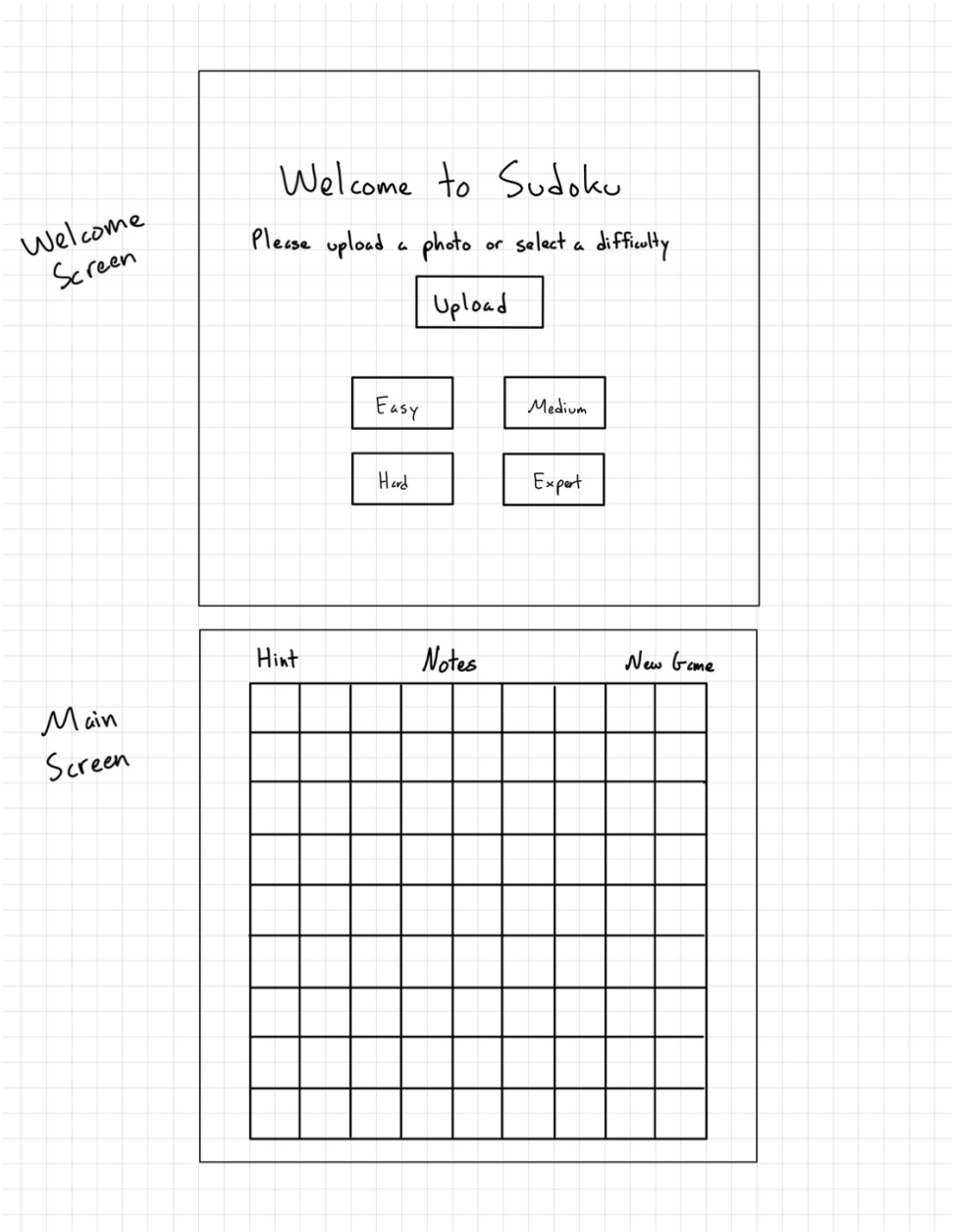
3. My project will not work if I do not have a correct sudoku solving algorithm implemented. This is essential because the program will not know what the correct solution to each game is and it is the base with which everything will be built off. While this algorithm is not extremely basic, I do not see this causing much of a hold up in the development of my project. With the research I’ve done into it up to this point it is a basic backtracking algorithm which I will look more into to understand before implementing it.

   My project will also not work if uploading the images and scanning the game using computer vision does not work. This is essential because this is the standout feature of this project that makes it something that I would be proud of turning in. I fully plan on investing the time into making this work but if for some reason I was not able to implement this feature I would spend more time on the GUI and make it more of a full-fledged game with more structured levels and likely a leaderboard which would be kept using a CSV file instead of a database. In this worst case scenario since I wouldn’t be spending time on the computer vision aspect IW would be able to put more time into creating algorithms that generate boards of different difficulties.

4. The outside resources I will need are just images of sudoku games I will use to help test and implement the computer vision aspect of the game. I can easily obtain these images from the internet and newspapers, and this will not cause any hiccup's in my development. The link below provides 20 free printable sukoku boards. I will print these out and write a few squares in with handwriting as well to make sure the computer vision algorithms can correctly detect this as well.
   20 Free Boards

5. For my architecture I will have two main classes and two subclasses. The first class will represent a single square on the sudoku board. Each square will hold values for the notes left on each square, the actual value of the square as well as the current value of the square. The next class will represent the whole board and the game, it will contain methods that check whether a move is valid or not and contain the method which solves the board on the backend. It is also where moves made by the user will be changed on the backend. The board class will have two subclasses, one which is for boards generated through an image which will hold the computer vision code and one which will hold the code for generating the board. The other subclass will be for boards when the user selects a difficulty and it will pull a hard coded board for the user to play. As for the user interface I do not really know how Qt ties into C++ but a combination of buttons and selecting a certain square on the board with your mouse will determine which methods are called.
   For my first design pattern I will use flyweight for the GUI, specifically for the squares on the sudoku board. Since they will all have the same style and user feedback when interacted with besides the numbers printed out in each square it will be faster and less redundant to move all of these stylistic elements into a single object.
   For my second design pattern I will be using singleton for the board class. Since I will only need one instance of the board class it will make sense to use singleton and have a global access point so other classes can interact with the board in an easier manner.

6.


Welcome Screen
Welcome to Sudoku
Please upload a photo or select a difficulty
Upload
Easy
Medium
Hard
Expert
Main Screen
Hint
Notes
New Game

7. Homework 2 Due Date: For this due date I will set up all the classes and integrate the OpenCV library as well as the Qt Library., I will also design a sudoku board in the terminal so that I can make progress until I eventually implement the GUI.

   Homework 3 Due Date: For the homework three due date I will still be running my sudoku game through the terminal but will have a fully functional game in which the computer can generate a game of varying difficulty and the user can solve the game. All the backend logic for the gameplay will be implemented by this deadline.

   Homework 4 Due Date: For this homework due date I will have fully implemented the Computer Vision Aspect of the Game. It will still be run through the terminal so the program will take in the name of a photo which will be scanned by the computer vision functions and converted to a playable sudoku game.

   Homework 5 Checkpoint 1: I will implement the gameplay part of the GUI, so users will be able to click on certain squares to add numbers instead of using a coordinate system in the terminal.

   Homework 5 Due Date: By this point my game will be complete and users will be able to select files from the GUI, select difficulty levels from the GUI and add notes to each of the sudoku squares.

8. Since I will be using C++ and Qt staying engaged in class shouldn't be an issue. The topics covered in class will be relevant to my project so it will be in my best interest to continue coming to class and taking as much away from it as possible.