Projeto MasterMind – Checklist.

Para que consigamos atingir o objetivo final de entregar nosso produto (jogo MasterMind), devemos, tal como os romanos, dividir para conquistar (*divide et impera*). Assim, o trabalho total será dividido em tópicos que serão trabalhados individualmente para, *a posteriori*, serem unificados no arquivo final.

Dito isso, cada integrante ficará responsável pela solução (isto é, criação) de um ou alguns tópicos pendentes que serão chamados de *checkpoints*. A fim de dinamizar a produção, para cada *checkpoint* haverá um prazo de entrega estipulado pela equipe, sendo seus parâmetros: grau de dificuldade e "necessidade para entendimento geral do código". Quanto a este último, seria um exemplo: X ficou responsável por um trecho de código que realizará a criação dos tabuleiros printados na tela e Y pela geração aleatória das senhas. Para que Y consiga concluir seu trabalho, precisará de um entendimento do que X está produzindo, assim, Y tem um tempo limite de produção de sua parte, para que o projeto dê sequência.

Os seguintes *checkpoints* serão abordados em nosso projeto[1]:

1. **Tela de Boas vindas ao jogador/Menu inicial** (Um menu com opções que compreendem as opções de jogo incluindo: 'Jogar'; 'Instruções'; 'Sair');
2. **OPÇÃO: 'Instruções'** (Praticamente nada de demais aqui. Será mais de exposição do jogo do que qualquer outra coisa. Lembrando que uma coisa imprescindível é a opção de retorno pra tela do Menu inicial).
3. **OPÇÃO: 'Jogar'** (Onde será introduzido diretamente a seleção do nível de dificuldade obviamente -_-. É interessante [este ponto vale 10% do trabalho] que haja uma interação entre o jogo e jogador expressa com o jogo pedindo o nome do jogador antes de começar a partida, e no momento que o jogo for se referir ao jogador, faça-o pelo nome dado).
   - 3.1. **Nível de dificuldade**: Deverá ser apresentado ao jogador as opções de jogo, seja por meio de uma tabela, seja pelo jeito que fique mais adequado. O relevante aqui é que deverão ser mostrados os três níveis de dificuldade com seus dados correspondentes: O 'Fácil' com 4 pinos (senhas que o jogador colocará), uma variação de 6 cores possíveis e um limite de 10 jogadas; o 'Médio, com 5 pinos, 7 cores e limite de 12 jogadas; e o 'Difícil', com 6 pinos, 8 cores e limite de 14 jogadas.
     ** Enfatizando que é um hub de escolhas, logo, uma sugestão é o tratamento por switch.*
   - *3.2.* ***Modo multiplayer****: depois ou antes da escolha da dificuldade deve ser garantida a opção de poder jogar ou não com outra pessoa.*
   - *3.3.* **Pontuação do jogador**: Apresentará ao player um valor de acordo com a quantidade de jogadas necessárias para que o player tenha terminado o jogo. A lógica é que, quanto menos jogadas melhor, assim, uma sugestão é que a pontuação máxima seja 10 (chute na primeira jogada), e a mínima 0 (não conseguiu acertar a senha).
   - *3.4.* **JOGAR NOVAMENTE:** No caso da partida em que o player estar jogando acabar, ser apresentada a opção de jogar novamente, bem como uma de terminar o código ou voltar pro menu iniciar (essas opções devem ser apresentadas depois da pontuação [a do menu inicial talvez não seja tão relevante).

4. **OPÇÃO: ‘Sair’** (A execução do código termina por aí [poderíamos inclusive por créditos bem breves aí]).
5. LÓGICA DO JOGO:
    5.1. **Interface do tabuleiro na tela:** entenda como interface o “printar” a cada nova jogada do tabuleiro na tela, que considera, claro, o tamanho do tabuleiro de acordo com a dificuldade escolhida (Usar matrizes para representar os tabuleiros é o caminho mais fácil).
    5.2. **Definição das cores das senhas e pinos:** não cabe ao player esta parte. Mas sim a nós, na criação, decidir como serão representados os pinos e as cores que serão escolhidas.
    5.3. **Definição da senha:** Não será apresentada ao jogador, nem aparecerá na tela. Será gerada aleatoriamente pelo programa e deve variar de tamanho de acordo com a dificuldade. Em resumo, é o valor comparativo toda vez que o tabuleiro printa, de forma a decidir sua vitória, derrota, assim como os tipos dos pinos que indicarão: cor certa na posição correta ou cor certa na posição errada (sem pinos indica que o jogador/player errou).

***Uma dica (ou não, sei lá kkkkkk) seria interpretar o espaço dos pinos como sendo duas casas da matriz do tabuleiro:
exemplo: Tabuleiro de dificuldade facil: 8 espaços para as adequações da escolha do jogador (entenda como 4 espaços para as cores e 4 para os pinos), e 10 para o número de jogadas.

Cor cor cor cor pino pino pino pino
Cor cor cor cor pino pino pino pino
Cor cor cor cor pino pino pino pino
Cor cor cor cor pino pino pino pino

.
.
.

********Outra dica é pensar também que o jogador não arrastará nenhum pino, então talvez dividir cada jogada em quatro perguntas sobre a posição de cada pino seja a ideia mais prudente.

[1] → uma coisa que seria relevante aqui, é encarar cada novo *checkpoint* como o ramo de uma árvore, ou, em alguns casos, como uma funcionalidade/ferramenta que será utilizada em um ramo da árvore. Uma analogia para isto seria que, ao criarmos uma casa, fazemos a fundação primeiro, então subimos as paredes, o teto, depois o piso, a instalação elétrica e a alocação de azulejos são realizadas; no entanto, durante o processo precismos “criar” o concreto, a massa corrida, bem como preparar as ferramentas, tal como o martelo, a enxada, a pá, etc. Estes últimos são necessários para que a casa seja criada, no entanto, não para mantê-la em pé. Da mesma forma serão alguns *checkpoints* do código, haverão alguns, tal como a seleção do nível de dificuldade e criação das matrizes do tabuleiro, que servirão mais como ferramentas do que o cerne da lógica de execução do código em si.

No geral é isso pessoal, com certeza ainda modificaremos algumas ideias e tudo mais, mas acredito que isso aí é o necessário para que consigamos fazer o jogo tranquilamente. Quanto as datas, assim como disse lá em cima, decidiremos de acordo com o tipo de checkpoint e o progresso do trabalho.

Link do arquivo no Github:

https://github.com/TheAssis/MastermindProject