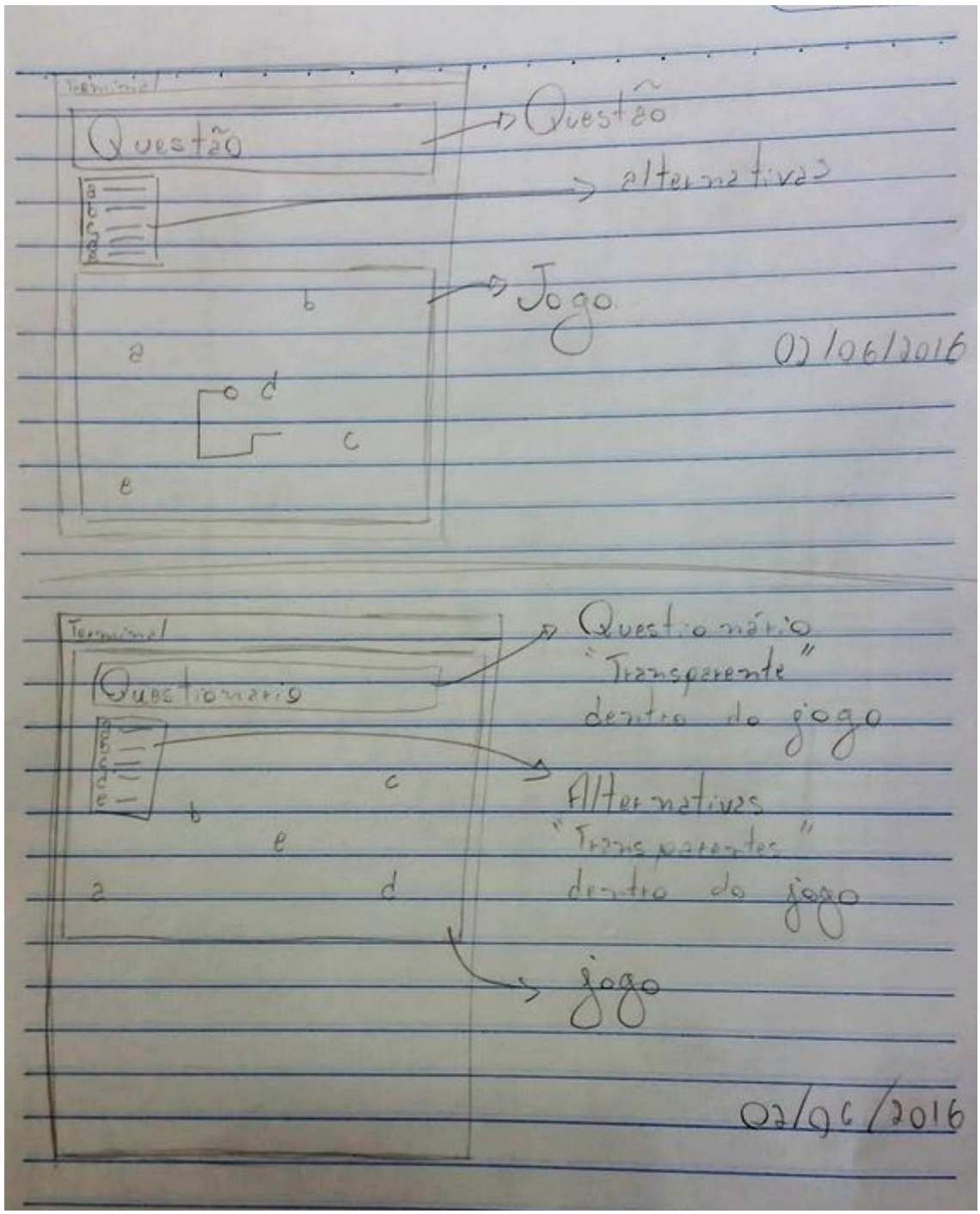
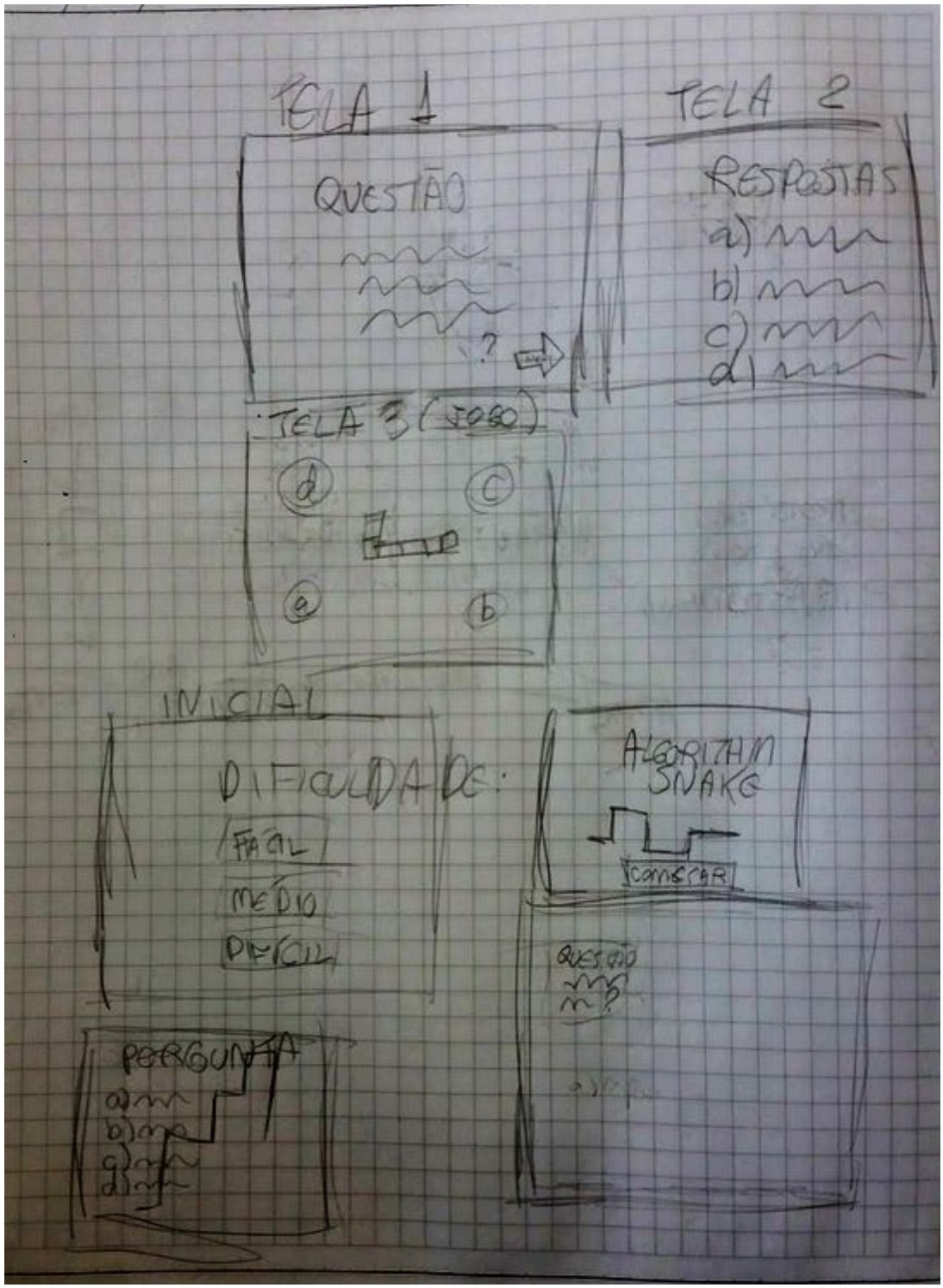
Protótipos:

Foram feitos alguns protótipos de baixa fidelidade para elucidar qual a melhor interface para o programa.





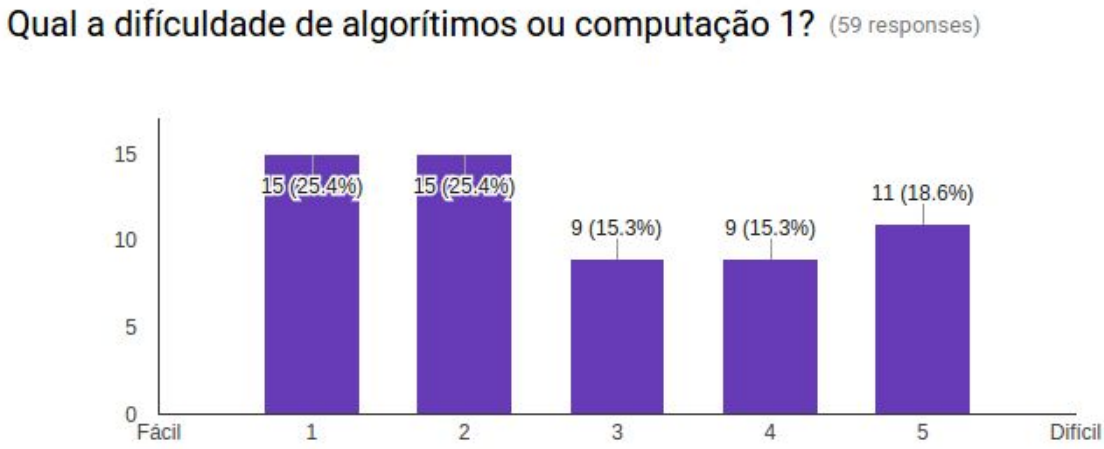
Pesquisa de Campo:

Para elucidar quais as dificuldades dos estudantes de computação foi utilizado de um questionário via o google forms, listando os grandes tópicos de conhecimento abordados pelas disciplinas.

Os resultados foram compilados abaixo:

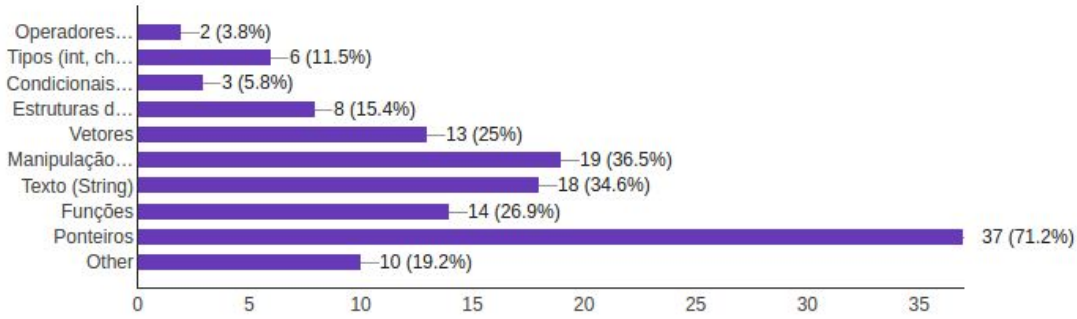
Respostas do “outros”:

- Eu não tenho dificuldade em nada, mas sei que ter dificuldades com ponteiros é comum.
- Denis
- Lógica (Eng. de Prod. não aprende lógica, deveria ser inclusa alguma matéria do gênero antes de computação)
- Denis, Denis, Denis, eu ja falei Denis, Denis de novo
- instalar o programa para testar
- Denis
- Na verdade eu sou bom, a professora que não é boa
- Árvore AVL
- aguentar os caras chatos do quarto período



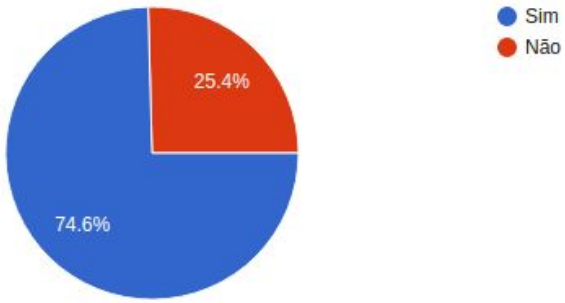
Algorítimos ou computação 1

Quais áreas você tem dificuldade? (52 responses)

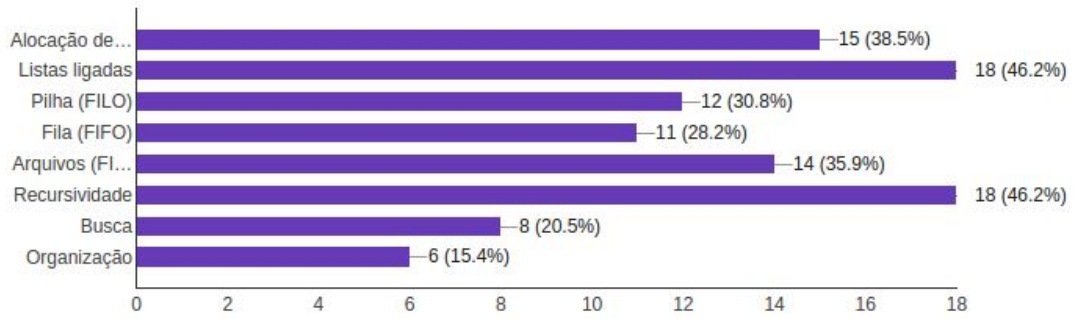


Algoritimos e estruturas de dados 1 ou Computação 2

Você tem Algorítimos e estruturas de dados 1 ou Computação 2? (59 responses)



Quais áreas você tem difículdade? (39 responses)



Eis o link com todos os dados, para referencia futura:
https://docs.google.com/spreadsheets/d/1tmoqaj4v0ZRRH_HigD2GExJtnLhv3q4rpBo6yPOam-4/edit?usp=sharing

Utilizaremos tais dados para realizar a elaboração do questionário apresentado no jogo, cujos assuntos serão primariamente ponteiros, manipulação de vetores e strings, alocação de memória, recursividade e listas ligadas.

Requisitos:

Programa em geral

- Linguagem C: Uso da linguagem de programação C em todo o projeto.

- Linguagem de comunicação com usuário: português

- Atualizar terminal de forma eficiente: para manter uma visualização adequada e uma jogabilidade fluida, será necessário eficiência na atualização do que é exibido no terminal.

- Input assincrono: O programa precisa receber informação a qualquer instante sem a necessidade de confirmação (apertar enter para cada comando) para que o comando do usuário seja efetuado de forma eficiente e em tempo real.

- Sleep com precisão boa: Controlar a velocidade do jogo de maneira exata.

- Variar tamanho do campo em função da janela: Preencher a janela com o campo.

Jogo

- Dificuldade: O jogo terá 3 dificuldades(fácil, médio e difícil) referentes a velocidade da cobra, quanto mais difícil mais rápida ela se moverá. O usuário irá selecionar a dificuldade a seu agrado no inicio do jogo e a partir disso a cobra deve corresponder a dificuldade selecionada. Caso o jogo reinicie (como por exemplo no caso de perder ou terminar o jogo) será reapresentada a opção de alterar a dificuldade.

- Design da cobra: para facilitar a compreensão do usuário, a cobra deve conter uma parte correspondente a sua cabeça e outra correspondente a seu corpo distintas entre si.

- Design da área de jogo (limites/borda): para manter claro os limites do jogo e por ser uma parte decisiva no jogo (por determinar um dos tipos de como perder no jogo), os limites do jogo devem ser claramente estabelecidos para o usuário evitá-los com eficiência.

- Comer: A cobra deve ser capaz de, quando encostar sua cabeça em uma “comida”, comê-la.

- Andar (cobra indo na direção certa): a cobra ao ser colocada em uma direção, deve continuar nela até trombar em algo (e consequentemente perder o jogo) ou até que seja dada outra direção.

- Mover (alterar o caminho da cobra): através das teclas w,a,s,d (onde “w” corresponde a cima, “a” corresponde a esquerda, “s” corresponde a baixo e “d” corresponde a direita) será inserido a informação da direção que a cobra deve se virar.

- Checagem para caso a cobra trombe em algo: quando a cobra entra em contato com as bordas externas do jogo, ou com o seu próprio corpo, ou com um obstáculo colocado no mapa (no caso deste jogo as respostas incorretas para a questão colocada), o jogo termina, mostra a pontuação e volta para a tela inicial.

- Crescer cobra: Ao comer, o tamanho da cobra deve aumentar em uma unidade, aumentando a complexidade do jogo.

- Colocar comida: Ao trocar de pergunta, serão colocadas comidas no mapa, em locais aleatórios, que não contém nenhuma outra coisa.

Questionário

- Dificuldade: o questionário deve ser crescente em sua complexidade, sem manter sua didática.

- Parte Algoritmo/Comp 1: Parte do questionário baseada na matéria correspondente a Algoritmos e Computação 1;

- Parte AED/Comp 2: Parte do questionário baseada nas matérias de AED e Comp 2. Dificuldade mais alta como resultado da matéria mais avançada.

- Questões em si: elaborar as questões com base no questionário realizado com os alunos.

Fusão de Questionário com Jogo

- Incrementar colocar comida em colocar alternativa: A “comida” colocada de forma aleatória deve ter a forma de uma alternativa e corresponder a mesma.

- Fazer questão ficar dentro da borda: A questão deve ficar dentro do campo do jogo, ou seja, dentro de onde a cobra pode andar, sem ficar em cima da borda ou fora da área do jogo.

- Fazer a cobra passar por cima do texto de questionário: De modo a não impedir a jogabilidade, mesmo tendo o espaço compartilhado entre eles

- Fazer o item comestível representar uma alternativa: A resposta estará junto às alternativas/itens comestíveis.

- Fazer itens comestíveis do número exato de alternativas.

- Atualizar questão (caso de acerto): Limpar a tela do que foi escrito e trocar a questão

- Finalizar jogo (caso de responder errado), com possibilidade de reiniciar: Deixar o usuário escolher o que vai fazer.

- Somar pontuação: Cada resposta correta acarretará em 1 ponto