Especificações Gerais

Henrique de Andrade Assme

Lucas Parra Sgarbosa

1. **Brainstorm e definição do tema**

Ao longo da semana passada, nós dois conversamos e discutimos sobre o que fazer para o projeto e chegamos no seguinte consenso:

Vamos adicionar funcionalidades novas para o modo de jogo que já existe e depois adicionar um modo novo de jogo.

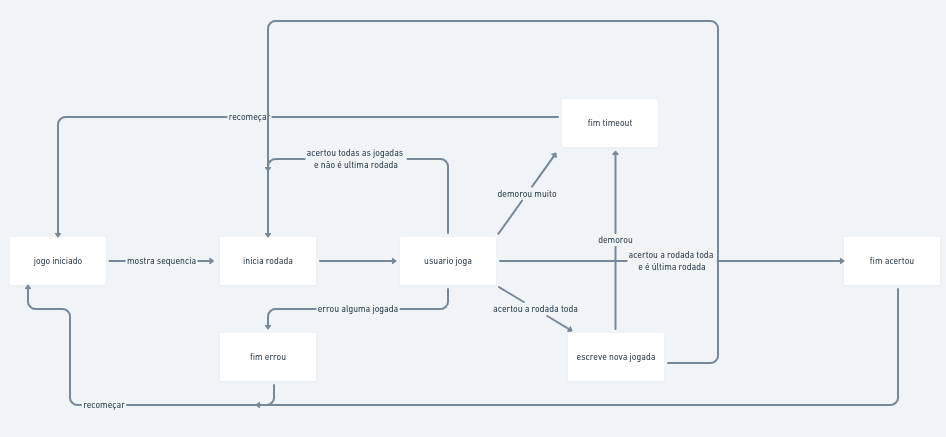
As funcionalidades adicionais serão:

* Sistema de vida para o jogador
* Sistema de nível
  + Cada nível irá definir a quantidade de vidas e rodadas do jogo
* Adicionar o timer da rodada num hexadecimal para ficar visível ao jogador quanto tempo ele ainda possui

Depois de adicionar isso ao modo atual, vamos criar um novo modo de jogo baseado no modo 1 do Genius. O jogador precisa fazer as jogadas na sequência que aparecem, porém, a cada nova rodada a quantidade de jogadas aumenta e o tempo total para fazer todas as jogadas diminui. Se possível, vamos adicionar o sistema de nível e vida nesse novo modo também.

1. **Recursos disponíveis**

Possuímos ao todo 4 semanas para fazermos o projeto. As restrições do que é necessário fazer já foram passadas para o grupo através das aulas online.

1. **Diagrama de blocos**

Acreditamos que para as mudanças propostas será necessário, antes de iniciar o jogo, selecionar o nível. No caso do usuário perder, tanto por algum timeout quanto por erro, o jogo não mais irá encerrar, mas apenas o jogador perderá uma vida e o jogo volta da rodada em que perdeu. Basta então adicionar uma condição nessas transições de estado para caso a vida esteja zerada, aí sim o jogador perdeu de fato o jogo. Do contrário, ele perderá uma vida e a rodada volta do ponto que perdeu.

1. **Especificação de Requisitos**

| **Código:** 0 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** Sistema de vida | |
| **Descrição:** Implementação de uma maneira do jogador continuar jogando mesmo que tenha perdido o jogo através de um sistema de vidas. Caso as vidas tenham acabado, o jogador perde de fato o jogo. Se não, perde uma vida e continua jogando do início da rodada perdida. A vida aparecerá em um display hexadecimal. | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** | |

| **Código:** 1 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** Sistema de nível | |
| **Descrição:** Implementação de uma maneira do jogador continuar jogando mesmo que tenha perdido o jogo através de um sistema de vidas. Caso as vidas tenham acabado, o jogador perde de fato o jogo. Se não, perde uma vida e continua jogando do inicio da rodada perdida. | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** sistema de vida | |

| **Código:** 2 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** Display hexadecimal para o timer da rodada | |
| **Descrição:** adicionar no display hexadecimal quanto tempo falta para terminar uma jogada, para que o jogador consiga ver na placa o tempo decaindo de 5 a 0 segundos. | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** | |

| **Código:** 3 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** Memórias para modo novo | |
| **Descrição:** no novo modo do genius, a memória não possui escrita, logo, será necessário adicionar uma memória com os dados pré colocados para o jogador colocar a sequência correta. Além disso, uma memória guardará o tempo respectivo de cada rodada, uma vez que nesse novo modo o tempo será diferente para cada uma delas (diminuindo conforme o jogo se aproxima do final). | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** | |

| **Código:** 4 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** Chave para escolha entre os dois modos e chave para escolha de nível | |
| **Descrição:** uma das chaves da placa será usada para escolher qual modo de jogo será desejado e chaves também para escolher qual nível escolhido (fácil, médio ou difícil) | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** | |

| **Código:** 5 | ◻ Funcional ◻ Não Funcional |
| --- | --- |
| **Requisito:** som para indicar que o jogador errou (perde vida), ganhou e perdeu | |
| **Descrição:** pensamos em adicionar um sinal sonoro para indicar quando o jogador perde uma vida, perde ou ganha o jogo. (ainda não decidimos se vamos realizar essa modificação mesmo) | |
| **Prioridade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Estabilidade:** ◻ Alta ◻ Média ◻ Baixa | |
| **Requisitos associados:** | |

1. **Arquitetura**
2. **Cronograma**

Semana 1:

Adicionar o sistema de vida, timer no hexadecimal e começar implementar o sistema de níveis e som

(0, 1, 2 e 5)

Semana 2:

Implementação completa do sistema de níveis. 3 níveis: fácil, médio e difícil

Fácil: apenas 4 rodadas e 5 vidas disponíveis para o jogador

Médio: 10 rodadas e 4 vidas

Difícil: 16 rodadas e apenas 2 vidas disponíveis

(término do requisito 2)

Semana 3

Implementação do novo modo. Será necessário uma memória nova, uma maneira de escolher o modo também (possivelmente uma chave fará a escolha) (3 e 4)

Semana 4

Término da implementação do novo modo, última bateria de testes para verificar se, tanto o novo modo quanto o modo antigo, estão funcionando corretamente e implementação do sistema sonoro.