

Programação de Baixo Nível - Relatório Trabalho Final (Guitar Hero)

Dhruv Babani, João Pedro Sostruznik Sotero da Cunha e Vítor Aguirre Caús

Escola Politécnica - PUCRS
Porto Alegre - RS

07 de junho de 2023

1. O Jogo

A proposta de jogo para este trabalho consiste em uma adaptação do *Guitar Hero* para a plataforma *SimulIDE*. Esta versão consiste de uma tela com 4 colunas onde notas surgem aleatoriamente, com limite definido por números pseudo-aleatórios ao iniciar um novo jogo. As notas disponíveis podem ser de dois tipos: vazia, que pontuam pelo pressionamento de botão, ou cheia, que pontuam pelo não pressionamento de botão.

Observação: Realizar o comando *make* apenas no Windows. A execução do jogo apresentará diversos erros se realizada com arquivos executáveis gerados em Linux (os arquivos executáveis do Windows estarão enviados junto ao código fonte).

2. Regras do Jogo

1. Notas vazias devem ser lidas apenas se o botão correspondente à sua coluna estiver pressionado, enquanto notas cheias são lidas se o botão não estiver pressionado. A leitura de ambos tipos de nota deve ser feita ao atingirem a linha de leitura;
2. Caso uma nota vazia atinja a linha de leitura e o botão correspondente não estiver pressionado, o jogador perde uma vida e tem sua *streak* atual zerada. O mesmo ocorre caso uma nota cheia atinja a linha de leitura e o jogador tenha pressionado o botão;
3. O jogador perde se cometer um total de erros igual ao número de vidas + 1.

3. Funcionamento dos Componentes

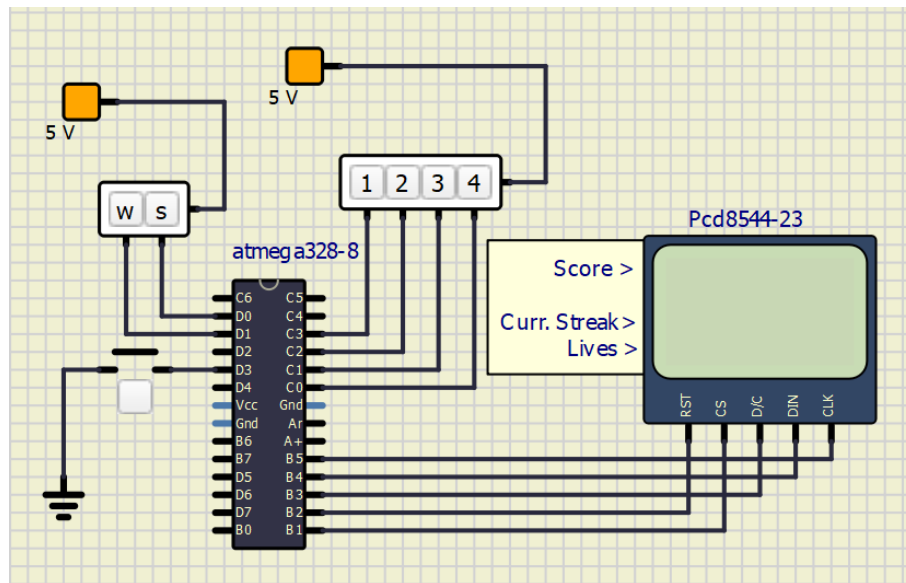


Figura 1. Circuito construído no simulador

A Figura 1 apresenta a estrutura do circuito desenvolvida para o jogo. Dois conjuntos de botões (*keypads*) e um botão individual são disponibilizados:

- Conjunto 1 (2 botões): permitem ao jogador selecionar uma ação no menu inicial do jogo (iniciar um novo jogo ou encerrar o programa). A escolha definitiva é feita pelo botão individual, explicado abaixo;
- Conjunto 2 (4 botões): permitem, conforme restrições de cada tipo de nota, a leitura delas em suas colunas (da esquerda para a direita), quando elas atingirem a linha de leitura na tela. Pode-se pressionar mais de um botão ao mesmo tempo, caso 2 ou mais notas cheguem na linha de leitura ao mesmo tempo;
- Botão individual: Quando pressionado, realiza a ação selecionada pelo jogador com o uso do Conjunto 1. Caso escolha iniciar um novo jogo, uma tela com a mensagem “Get Ready!” aparece por 5 segundos (conforme a Figura 2), então a geração de notas é iniciada (conforme Figura 3). Nesse momento, o número de vidas também é informado. Caso decida encerrar o jogo, a tela é desligada, e o reinício só é possível com uma nova ativação da simulação.

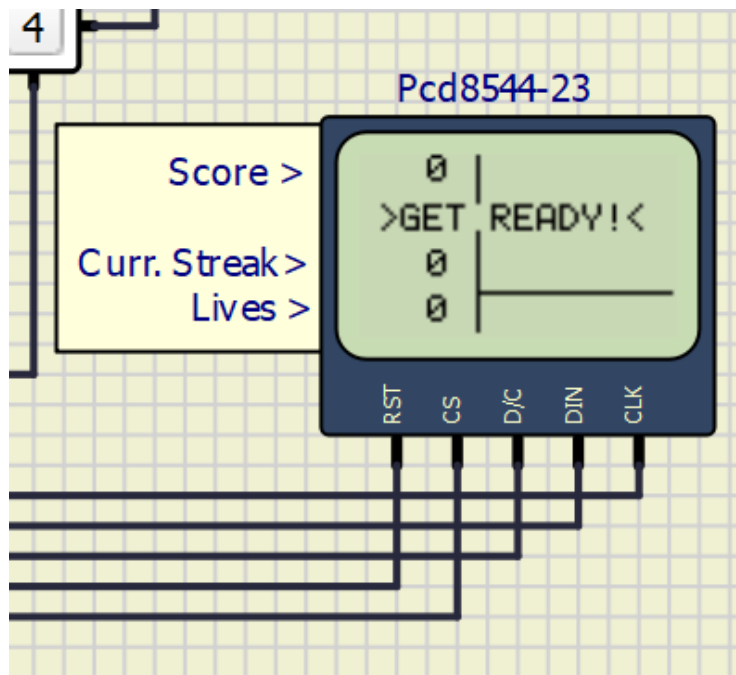


Figura 2. Tela na espera de 5 segundos

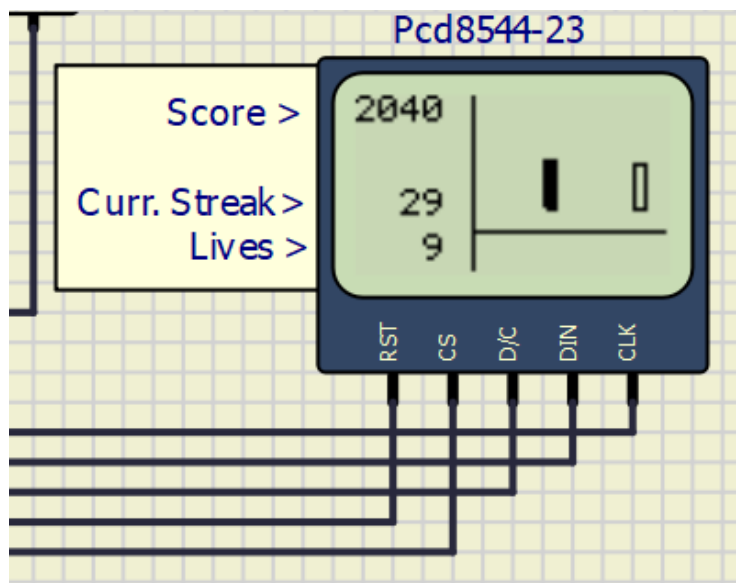


Figura 3. Jogo em andamento

Na Figura 3, são apresentadas as 3 informações sobre o *status* do jogador:

1. *Score*: Pontuação atingida pelo jogador até o momento na execução atual. É incrementada para cada nota lida corretamente (caso mais de uma nota seja lida ao mesmo tempo, cada uma delas incrementará a pontuação);
2. *Curr. Streak*: Quantidade de notas lidas em sequência pelo jogador sem a ocorrência de penalidades;
3. *Lives*: Quantidade de vidas que o jogador possui. O valor é inicializado com 10% do total de notas que a execução atual possui.

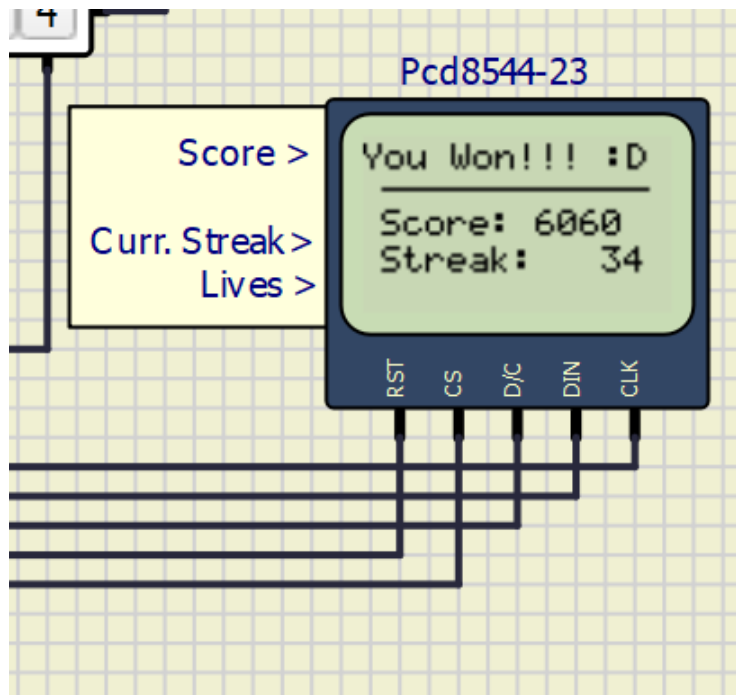


Figura 4. Tela de vitória

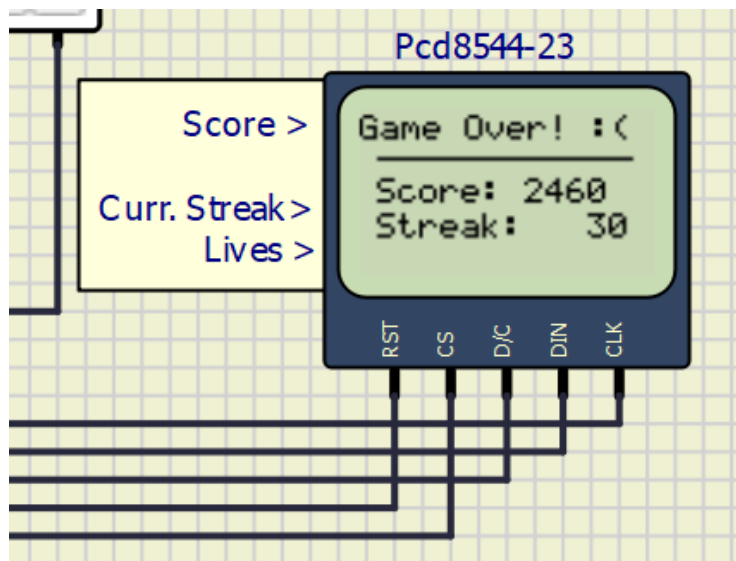


Figura 5. Tela de derrota

Por fim, o jogo possui duas telas para os resultados de vitória ou de derrota do jogador (Figuras 4 e 5, respectivamente). Ambas as telas indicam se o jogador ganhou ou perdeu, além de informarem a sua pontuação total e a *streak* recorde (maior valor de *streak* adquirido na execução atual).

4. Dificuldades Encontradas

Inicialmente, a leitura das notas em suas respectivas colunas foi um desafio. Nas primeiras versões, um laço dentro da iteração principal foi criado, onde cada nota era manipulada

a partir de números binários de 4 bits em um único vetor, de modo a armazenar todas as notas em uma só estrutura. A ineficiência dessa solução levou à sua substituição por uma lógica envolvendo 4 vetores (1 para cada coluna), sem manipulações de bits.

Entretanto, o laço extra ainda existia, e *bugs* relativos ao duplo incremento da *streak* e ao duplo decremento das vidas eram frequentes. Tentou-se reorganizar os testes dentro dessa iteração auxiliar, mas isso apenas tornou os erros menos frequentes.

Assim, dois vetores de 4 posições inteiras (1 para cada coluna) foram criados:

- **leituraInicialNotas**, inicializado em 1. Permite a leitura das notas em suas respectivas posições no momento exato em que a linha de leitura é atingida, tendo o valor 1 substituído por 0 logo após. Quando a nota chega ao fim, o valor associado à sua coluna volta a 1.
- **leituraContinuaNotas**, inicializado em 0. Caso uma nota tenha sido lida no momento exato, o valor associado à sua coluna torna-se 1. Assim, enquanto esse valor manter-se em 1, a pontuação do jogador aumentará conforme a nota a lida. O número associado à coluna retorna a 0 ao fim da nota.

Com esses vetores auxiliares e uma nova organização dos testes, foi possível evitar a atualização duplicada dos valores de *streak* e vidas.

Além disso, nessas versões anteriores o jogo possuía apenas notas vazias, mas com as mesmas lógicas de leitura e penalidades ao jogador que as atuais. Entretanto, existia a possibilidade de o jogador apenas segurar os 4 botões de leitura de notas durante todo o jogo, o que excluía a necessidade de alternância entre as notas e, consequentemente, eliminava a dificuldade do jogo.

Assim, decidiu-se adicionar notas cheias ao jogo, com lógica de leitura inversa àquela atribuída às notas vazias. Essa solução foi feita com a adição da função **nokia_lcd_fullrect** ao conjunto de funções disponibilizado para o trabalho (**nokia5110.c** e **nokia5110.h**), além da adição da condição de não pressionamento do botão aos testes condicionais realizados durante o jogo. Também foi adicionado um vetor de 4 inteiros (um para cada coluna), que definem, com base no Timer, o tipo de nota cada coluna terá em determinado momento do jogo.

Outra dificuldade encontrada foi a criação de uma condição que penalizasse o jogador por tentar ler uma nota antes que ela atingisse a linha de leitura, de modo a permitir a leitura apenas quando a nota atingisse exatamente a linha de leitura (conforme proposto no fórum de indicação do jogo). Porém, devido ao tempo restante para realizar o trabalho, essa funcionalidade foi descartada.