Uma Proposta de Produção de Jogos Estilo Quiz na Plataforma Arduino

Gabriel Silva de Azevedo[1], **Victor Travassos Sarinho**[1]

[1] Laboratório de Entrenimento Digital Aplicado (LEnDA)
Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, s/n - Novo Horizonte, Feira de Santana - BA, 44036-900

gabrielsilvadeazevedo@gmail.com, vsarinho@uefs.br

Abstract. *Different solutions for the production of digital games have been made available on the Arduino platform. This article presents Quizduino, a hardware/software suite proposal that sets a standard for the development of quiz games on the Arduino platform. For this, work related to the production of games in Arduino, standard hardware/software scheme, game loop execution of the Quizduino, and results obtained with the production/execution of the developed GuessMyNumber game, are described.*

Resumo. *Diferentes soluções para a produção de jogos digitais têm sido disponibilizadas na plataforma Arduino. Este artigo apresenta o Quizduino, uma proposta de conjunto hardware/software que define um padrão para o desenvolvimento de jogos estilo Quiz na plataforma Arduino. Para tal, são descritos trabalhos relacionados a produção de jogos em Arduino, esquema de hardware/software padrão aplicado, game loop de execução do Quizduino, e resultados obtidos com a produção/execução do jogo GuessMyNumber desenvolvido.*

1. Introdução

Baseado em um microcontrolador Atmel AVR e na linguagem de programação C/C++, Arduino é uma plataforma de prototipagem eletrônica, de hardware livre e de placa única, cujo objetivo consiste em ser acessível, flexível, de baixo custo, e de fácil utilização para principiantes e profissionais dentro do movimento *open hardware* [Arduino 2018].

Se tratando de jogos digitais, estes possuem uma notável participação no desenvolvimento das gerações atuais, abrindo espaço para o desenvolvimento e o uso dos mesmos em atividades e plataformas diversas. Neste sentido, diferentes soluções para a produção de jogos digitais têm sido disponibilizadas na plataforma Arduino, a exemplo do Gamebuino [Gamebuino 2016] e do MAKERbuino [MAKERbuino 2016] que fornecem API e ciclo de execução para produção de jogos embarcados diversos.

Este artigo apresenta o Quizduino, uma proposta de conjunto hardware/software que define um padrão para o desenvolvimento de jogos estilo Quiz [Wolf 2001] na plataforma Arduino. Para tal, são descritos trabalhos relacionados a produção de jogos digitais com Arduino, esquema hardware/software proposto para produção de jogos estilo Quiz, resultados obtidos com a implementação de um jogo demo para fins de validação do conjunto proposto, e, finalmente, conclusões e trabalhos futuros esperados para este projeto.

2. Trabalhos Relacionados

Atualmente, existem alguns trabalhos que demonstram o uso do Arduino na produção de jogos digitais diversos. Como exemplo de desenvolvimento de um jogo específico, tem-se a produção de um jogo de memória baseado em LEDs e botões de pressão (Figura 1(a)) aplicando o Arduino de forma isolada sem o apoio de recursos externos [Gomes 2011].

Com relação a plataformas Arduino para produção de jogos, MAKERbuino (Figura 1(c)) é um dispositivo educacional disfarçado de console de jogos que busca motivar as pessoas a explorar, aprender e criar algo novo de maneira divertida e interessante [MAKERbuino 2016]. Já o Gamebuino (Figura 1(b)) é um projeto de console de jogos portátil retro baseado no Arduino que se dispõe a ser uma solução pronta para começar a criar jogos 8 bits de uma maneira rápida e simples, mesmo com poucos conhecimentos em programação [Gamebuino 2016].

(a) Jogo de memória (b) Gamebuino (c) MAKERbuino

Figura 1. Trabalhos relacionados a produção de jogos digitais com Arduino.

3. Metodologia

3.1. Esquema de Hardware

Para o projeto foi utilizado um Arduino Due em conjunto a um sensor infravermelho, e uma tela de LCD para um Arduino Mega (Figura 2), sendo que tanto o receptor quanto a tela usam bibliotecas em código C para fins de monitoração dos dados recebidos. O código do infravermelho busca a decodificação dos sinais analógicos em hexadecimal, e com esses valores informar quais botões estão sendo pressionados de um controle infravermelho. Já o código da tela de LCD permite diretamente renderizar cores e letras em posições da tela de forma matricial, com colunas e linhas.

3.2. Software de Gerenciamento

Um microcontrolador Arduino trabalha com duas funções principais de execução: ***setup*** e ***loop***. A função ***setup*** é apenas executada uma vez no inicio do funcionamento da placa. Já a função ***loop*** executa um conjunto de comandos em cada ciclo contínuo de funcionamento do microcontrolador.

Para a tela de LCD, por conta de ser um *shield*, não precisa atribuir pinos de leitura na inicialização, sendo necessário apenas inicializar as bibliotecas C de apoio na função ***setup***. Utilização de métodos de posicionamento de cursor, renderização de cores na tela inteira, e variante do *println()* serão executados na função ***loop***.

Com relação ao sensor infravermelho, este deve ser conectado aos pinos de energia, terra e uma entrada digital pela função ***setup***, sendo esta entrada digital atribuída

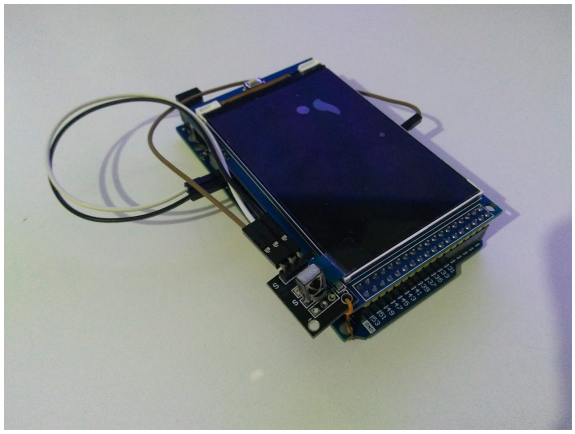

Figura 2. Protótipo em hardware inicial do Quizduino.

ao sensor via biblioteca C inicializada para o infravermelho. Já a leitura contínua para decodificação de sinais infravermelho no Arduino em teclas equivalentes pressionadas é efetuada pela função ***loop***.

3.3. *Game Loop* Padrão

Com a captura de entradas do usuário via sensor infravermelho definida, e a saída do status do jogo exibida em tela de LCD configurada, o passo seguinte consiste na definição de um *game loop* padrão de execução para o Quizduino (Figura 3). Por *game loop*, trata-se de um "algoritmo que relaciona o estado atual do jogo e as propriedades dos objetos com um número de condições que podem modificar o estado do jogo"[Sicart 2008].

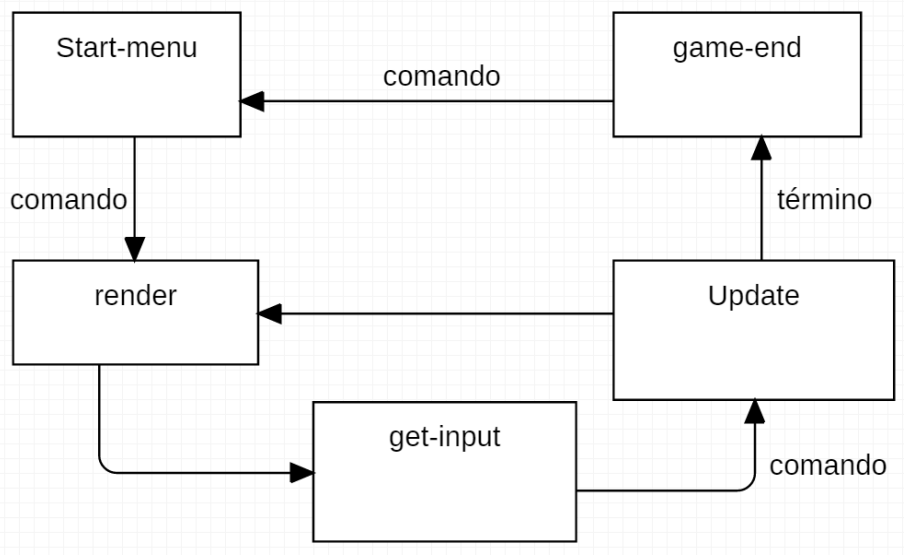

Figura 3. Ciclo de execução padrão de jogos no Quizduino.

Para o game loop padrão, foram definidas alguns estados base, responsáveis por ações básicas de inicialização, renderização, captura de entradas do usuário, atualização do status do jogo, e finalização de uma partida. Alguns destes estados executam algumas funções base que cada jogo deve fornecer, a exemplo da função *startGameStatus* que

é executada tanto na inicialização como no ”*Try Again?*”ao final de cada partida. Para o estado *render*, este procura verificar continuamente o *status* corrente do jogo e exibe a mensagem corrente na tela do LCD. Para a captura de dados do jogador, o estado *get-input* trabalha com os dados fornecidos pelo infravermelho, indicando os valores de comando a serem passados para o estado *update*, o qual executa a função *updateGameStatus* que atualiza o status corrente do jogo e finaliza a execução de um ciclo do *loop* (Figura 3).

Para fins de compatibilidade do *core* dos jogos desenvolvidos com demais plataformas de execução a serem futuramente implementadas, desenvolveu-se o ciclo de execução do *game loop* em Javascript. Trata-se de um ganho do ponto de vista de reuso de lógica do jogo e de independência de hardware para execução do mesmo [BinSubaih and Maddock 2008], apesar da dependência no caso do Arduino de um PC para execução de códigos Javascript.

4. Resultados Obtidos

Para fins de validação do conjunto hardware/software produzido, desenvolveu-se uma versão Arduino do clássico jogo de adivinhação de números *GuessMyNumber* (Figura 4). Para tal, efetuou-se o carregamento inicial da configuração do jogo no *framework* Javascript responsável pela execução do *game loop* proposto. Esta configuração define duas funções base para a execução do jogo (*startGameStatus* e *updateGameStatus*), bem como dados JSON referentes ao menu inicial do jogo.

Uma vez carregado o arquivo de configuração, o *game loop* verifica em qual estado o jogo se encontra, apresentando na tela de LCD a mensagem de *status* corrente do jogo. Tal mensagem fica carregada em um *buffer* no microcontrolador para indicação de quando a tela deve ser atualizada ou não para o caso de mudança de valor. A medida que informações de comando oriundas do sensor infravermelho são encaminhadas para o *framework*, ocorre a alteração de estados indicados no *game loop* (Figura 3), bem como a devida execução das funções configuradas no *core* do jogo, garantindo assim a devida execução do mesmo conforme programado.

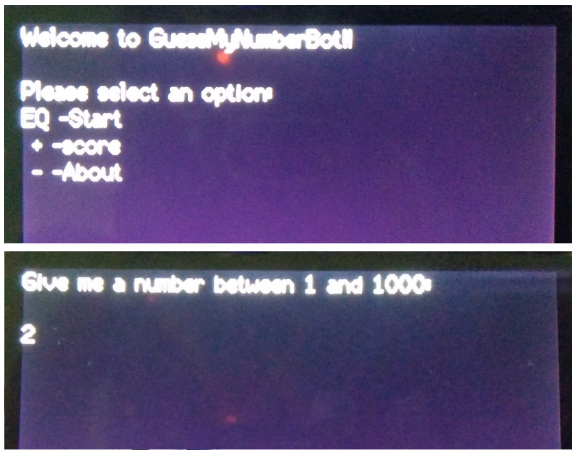

Figura 4. Execução do jogo GuessMyNumber na plataforma Quizduino.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou o Quizduino, uma proposta padronizada de produção de jogos estilo Quiz na plataforma Arduino. Para tal, são descritos trabalhos relacionados a produção

de jogos em Arduino, esquema de hardware/software padrão aplicado, *game loop* de execução do Quizduino, e resultados obtidos com a produção/execução do jogo *GuessMyNumber* desenvolvido.

Quizduino representa um conjunto hardware/software capaz de fornecer jogos Quiz de uma maneira rápida, de fácil manutenção, e com a lógica principal do jogo definida sem dependências com a plataforma de execução alvo. Contudo, existe uma limitação com relação a entradas textuais por parte do usuário, algo que deve ser compensado no futuro com a introdução de outras abordagens de captura de dados textuais, tais como teclado, reconhecimento de voz, entre outras.

Como trabalhos futuros, pretende-se estender a abordagem proposta no Quizduino para demais plataformas alvo, a exemplo dos jogos LibrasZap [Sarinho 2017] e BodyZap [Sarinho et al. 2017] desenvolvidos para plataformas de mensagens instantâneas e que também serão adaptados para a plataforma Arduino. A codificação em Javascript também é uma limitação que precisa ser resolvida em um futuro próximo, uma vez que ela exige a conexão com um PC para executar os jogos Quizduino proposto. Finalmente, a montagem de um circuito em placa junto com uma *case* customizada do Quizduino também será efetuada em um futuro próximo, de modo a produzir uma interface ergonômica para testes com usuários de diferentes perfis.

Referências

Arduino (2018). Arduino. https://www.arduino.cc/. acessado em 18/06/2018.

BinSubaih, A. and Maddock, S. (2008). Game portability using a service-oriented approach. *International Journal of Computer Games Technology*, 2008:3.

Gamebuino (2016). https://www.gamebuino.com/. Accessed: 2018-06-16.

Gomes, V. K. L. (2011). Desenvolvimento de um jogo de memorização luminosa na plataforma arduino.

MAKERbuino (2016). https://www.makerbuino.com/. Accessed: 2018-06-16.

Sarinho, V. T. (2017). Libraszap-an instant messaging game for knowledge assessment in brazilian sign language. *Brazilian Journal of Computers in Education*, 25(01):44.

Sarinho, V. T., Lima, C. O. C., and Granjeiro, E. A. (2017). Bodyzap: Um jogo de im para o ensino de fisiologia humana. In *II Workshop de Jogos e Saúde, 2017. XVI SBGames*. SBC.

Sicart, M. (2008). Defining game mechanics. *Game Studies*, 8(2):1–14.

Wolf, M. J. (2001). Genre and the video game. *The medium of the video game*, 1.