

Tamagotchi embarcado no Raspberry Pi

Pedro Eugênio Machado de Lima
Engenharia Eletrônica
Universidade de Brasília - Campus Gama
Brasília, Brasil
peugenio95@gmail.com

Pedro Henrique Trindade Andrade
Engenharia Eletrônica
Universidade de Brasília - Campus Gama
Brasília, Brasil
pedrohenriqueatrindade@gmail.com

Resumo—Esse trabalho se propõe a apresentar uma versão contemporânea de um popular brinquedo eletrônico da década de 90 usando um Raspberry Pi como plataforma de desenvolvimento e implementação.

Palavras-chave—Raspberry Pi, Tamagotchi.

I. INTRODUÇÃO

A-REVISÃO BIBLIOGRÁFICA

O Tamagotchi foi um brinquedo eletrônico lançado inicialmente pela Bandai em 1996 no Japão, não levou muito até tomar conta de todo o mundo. O princípio básico do brinquedo se baseia em criar e cuidar de um animal de estimação virtual, que se relaciona com o usuário a partir de uma tela geralmente de cristal líquido (LCD) e botões coloridos. A criatura digital possui necessidades diárias como fome, sono, felicidade e disciplina, podendo até morrer se negligenciadas. Todos esses marcadores definem a saúde geral do animal, criando uma sensação de progressividade e uma jogabilidade de alta liberdade ao jogador, fazendo cada gameplay criado especial e único.

O Raspberry Pi foi criado pela instituição de caridade Raspberry Pi Foundation com o objetivo de fornecer para estudantes e hobistas um computador de baixo custo e grande eficiência [1]. O Raspberry Pi é, de fato, um computador e, portanto, apresenta quase todas as funcionalidades de um computador desktop, como por exemplo embarcar um sistema operacional por exemplo um Linux [2].

Um sistema operacional como o Linux é formado pelo Kernel, drivers e outras aplicações como a interface gráfica e outros. Atualmente, existem diversas distribuições de Linux facilmente acessíveis e com características específicas para cada aplicação. Apesar da capacidade do Raspberry ser semelhante a de um desktop, ele ainda apresenta limitações, como pouca memória RAM, logo algumas versões adaptadas do Linux foram criadas para funcionarem melhor na placa [2].

B-PROPOSTA DE PROJETO

Justificativa:

O Tamagotchi foi uma febre dos anos 90 mesmo com seu sistema limitado tecnicamente, pois

a própria tecnologia da época limitava que o jogo fosse muito mais avançado, porque isso aumentaria o preço comercial, inviabilizando a produção. Com o passar do tempo a capacidade de computação e processamento evoluíram exponencialmente, como já é esperado se baseando na lei de Moore para os equipamentos eletrônicos, já a popularidade do Tamagotchi diminuiu, o que abre a oportunidade para a inovação e a criação de um sistema ainda melhor com mais funcionalidades e interação, aproveitando tanto um público já familiarizado com o funcionamento do jogo original, aproveitando a nostalgia, quanto apresentando para um público mais jovem a mesma frenesi que levou uma geração inteira a comprar milhões de unidades desses jogos eletrônicos.

Objetivo:

Desenvolver uma versão mais interativa e mais moderna do tamagotchi usando um sistema embarcado no Raspberry Pi.

Requisitos funcionais:

- Desenvolver ferramentas de interação.
- Implementar em um Raspberry Pi.
- Criar uma comunicação com um aplicativo.
- Garantir entretenimento e diversão.
- Apresentar uma interface amigável.
- Garantir a robustez do sistema.

Requisitos não funcionais:

- Raspberry Pi 3.
- Sistema operacional
- Sensores

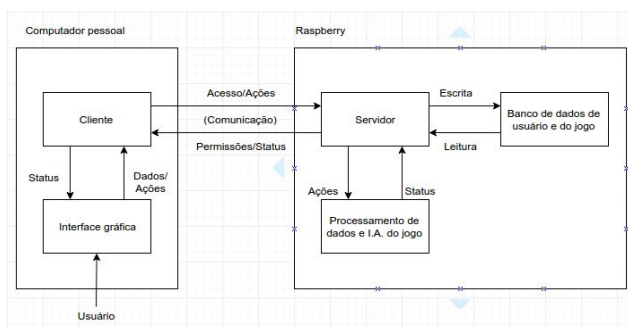
Benefícios:

Com a conclusão do projeto, os estudantes adquirirão os conhecimentos relativos à sistemas embarcados bem como terão a oportunidade de revitalizar um dispositivo eletrônico de entretenimento que foi marcante para uma geração inteira.

II. DESENVOLVIMENTO

A. Visão geral

O sistema se baseia no modelo de cliente/servidor, onde o cliente é o computador pessoal do usuário e o Raspberry é o servidor. Para o acesso do usuário foi desenvolvida uma interface gráfica usando a linguagem C#. O código do cliente recebe as informações da interface e as passa para o servidor usando o protocolo TCP/IP. O servidor recebe os dados e os valida ou registra no banco de dados. Com a permissão de acesso, o usuário pode realizar ações que serão enviadas para o servidor e processadas, retornando novos status. Além disso o servidor possui um algoritmo que representa a inteligência artificial do jogo, variando os status de forma independente e gerando respostas diferentes até para ações iguais do usuário. A figura 1 apresenta a visão geral do projeto.



B. Hardware

O desenvolvimento do hardware para o jogo se baseia em criar formas de interação lúdica com o bicho, assim como informar a ocorrência de eventos específicos.

Para a entrega do ponto de controle 3, optou-se em adicionar alguns elementos mais simples que pudessem ser controlados usando apenas os conhecimentos básicos sobre o GPIO do Raspberry.

Inicialmente adicionou-se um LED para a sinalização de solicitações recebidas. Optou-se em colocar um LED RGB para sinalizar as atualizações dos status, como os pinos de entrada e saída do Raspberry só funcionam com sinais digitais, o LED RGB, apresenta as cores azul, verde, vermelho e branco, para indicar quando cada status é modificado pelas solicitações do cliente.

Outro elemento importante é a utilização de comunicação sonora. Para isso usou-se um sensor piezoelétrico que ao sofrer uma pressão mecânica gera uma tensão correspondente que é interpretada de forma binária pelo Raspberry. Caso o Raspberry receba o valor '1' no pino de leitura do piezo, o servidor inicia a reprodução de um áudio que até o momento é uma música que será reproduzida por um período de tempo previamente definido.

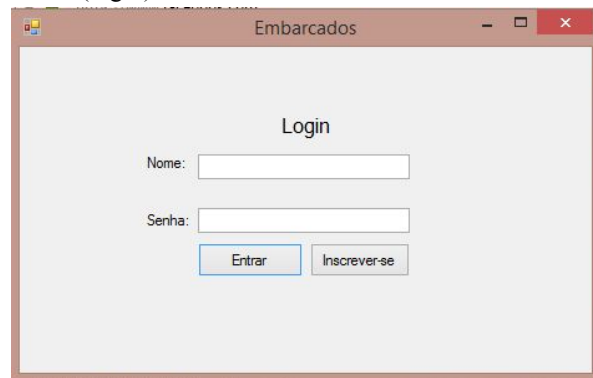
Os elementos apresentados são apenas a base teórica da interatividade do projeto, já planeja-se formas mais dinâmicas e lúdicas e aplicá-las, além disso, com o refinamento do projeto, novos elementos

interativos serão adicionados e que já estão sendo testados.

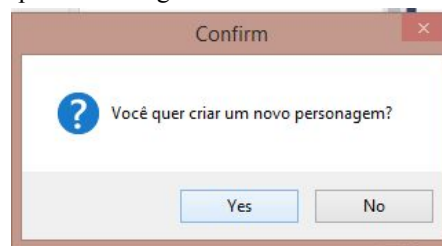
C. Software

Interface Gráfica:

Tela 1 (login):



Essa é a primeira tela que aparece para o usuário, nessa tela ele possui duas opções, o usuário pode tanto colocar seu nome e senha nos campos correspondentes e clicar "Entrar", assim o sistema mandará informação via tcp para o server checar se o nome e senha do usuário se encontram no banco de dados (olhar tópico "comunicação"), caso esteja ele irá diretamente para a Tela 2 (status), caso contrário aparecerá um aviso para o usuário se registrar. A outra opção possível é clicar em "Registro", nesse caso aparecerá o seguinte aviso:



Em caso de sim, o sistema avisará ao server que o usuário pretende apagar as informações do animal anterior e criar um novo do começo, indo para a Tela 3. Se não, o sistema apenas registra o nome e senha desse novo usuário que quer se cadastrar no banco de dados.

Tela 2 (status):

Exemplo:

Usuário alimentou o bicho com batata frita.

O cliente enviará:

2 1

Tanto ao entrar, registrar e na ação do bicho, a resposta de informação server será da seguinte forma, em ordem:

- 60 chars para o nome do bicho
- depois mais 1 int para o tipo do bicho
- depois 4 ints de status (para atualizar a tela)
- um inteiro para dizer se ação foi bem sucedida ou não

Servidor:

O servidor é responsável por receber as solicitações do cliente e através da lógica do jogo responder adequadamente a cada caso. Ele também é responsável pelo banco de dados dos jogadores, onde são armazenados os dados necessários para cadastro e acesso dos usuários, bem como algumas informações que conferem ao jogo uma ilusão de memória.

Para processar as solicitações dos clientes, o servidor cria um *socket* para receber essas solicitações seguindo o protocolo TPC. Após o recebimento dos dados do cliente, no caso as ações realizadas pelo jogador, o servidor modifica os *status* seguindo uma lógica pré estabelecida de valores em conjunto um fator de aleatoriedade. Em seguida, os novo *status* são armazenados em arquivos binários e enviados para o cliente.

Outra função do servidor é a lógica de inteligência artificial do jogo, ou seja, o modo como os *status* variam sozinhos para simular as necessidades do

bicho. Para isso usa-se um sinal que com uma certa frequência chama uma função que modifica os *status* seguindo uma lógica fixa.

A medida que os status se modificam novas funções são solicitadas para processar condições específicas, um exemplo disso é a função *morte()*. Essa função é chamada assim que o valor de um dos status atinge o seu limite e o bicho é considerado morte, assim o servidor informa a causa da morte e o jogador deve reiniciar o jogo criando um novo bicho.

III. RESULTADOS

IV. CONCLUSÃO

V. REFERÊNCIAS

[1] Raspberry Pi Brasil. “**O que é Raspberry Pi?**”. Disponível em: <<http://raspberrypibra.com/o-que-e-raspberry-pi-4.html>> acesso em 1º de abril de 2017.

[2] RICHARDSON, Matt; WALLACE, Shawn. “**Primeiros Passos com o Raspberry Pi**”. Disponível em: <<http://static.novatec.com.br.s3.amazonaws.com/capitulos/capitulo-9-788575223451.pdf>> acesso em 1º de abril de 2017.

[3] MAITA, aki. “**Who came up with Tamagotchi?**”. Disponível em: <<http://www.mimitchi.com/html/q10.htm>> acesso em 1º de abril de 2017.

[4] Mimitchi. “**Tamagotchi angel instructions**”. Disponível em: <<http://www.mimitchi.com/html/tainst.htm>> acesso em 1º de abril de 2017.

