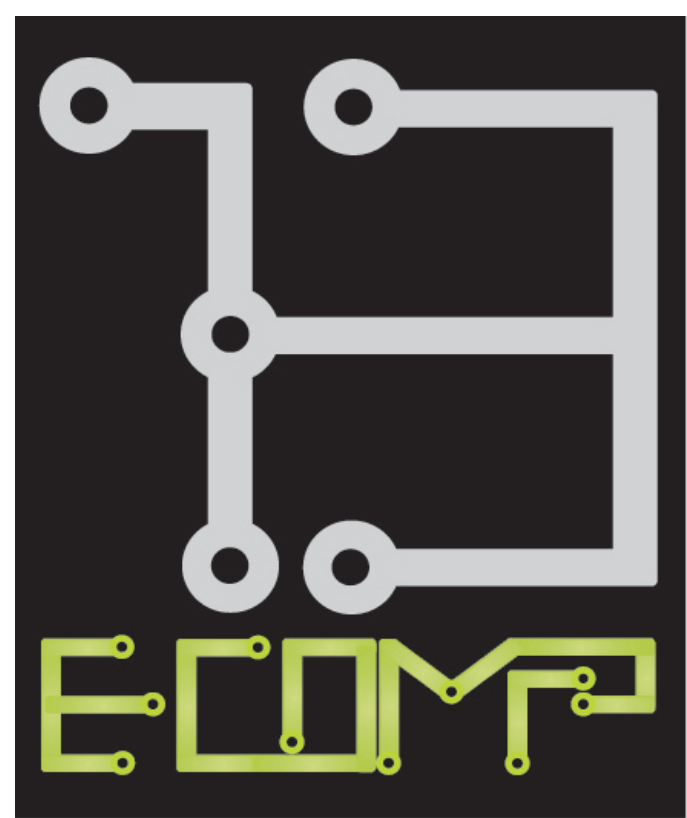


# LOGIKID - PROPOSTA DE UM JOGO EDUCATIVO PARA O ENSINO DE PORTAS LÓGICAS



Ezequiel França dos Santos, Gabriel Fontenelle, Tales Carlos de Pádua



Bacharelado em Ciência da Computação

Centro Universitário Senac - Campus Santo Amaro (SENAC-SP)

Av. Engenheiro Eusébio Stevaux, 823 – Santo Amaro, São Paulo – CEP 04696-000 – SP – Brasil

alebencz, ezefranca.br, colecionador.gabriel, talescpadua, (@gmail.com)

## Resumo

Este trabalho apresenta os resultados relativos ao desenvolvimento de um jogo eletrônico educativo, chamado LogiKid, voltado para o ensino e aprendizagem de portas lógicas e lógica booleana. Neste sentido, o presente trabalho enfoca a descrição de como foi incorporada ao jogo a apresentação de maneira visual dos conceitos de portas lógicas, conforme objetivos educacionais propostos. O desenvolvimento envolveu questões como tempo de aprendizado, abordagem lúdica e também questões pertinentes a um jogo, como seus desafios e aspectos técnicos na programação.

## 1. Introdução

Os jogos educativos digitais possibilitam a criação de ambientes de aprendizagem atraentes e gratificantes, constituindo-se num recurso poderoso de estímulo para o jogador, permitindo o desenvolvimento de inúmeras habilidades.

## 2. Objetivos

O significado atual do jogo na educação, sinaliza a existência de divergências em torno do jogo educativo, que estaria relacionado concomitantemente a duas funções: A primeira seria a função lúdica do jogo, expressa na ideia de que sua vivência propicia a diversão, o prazer, quando escolhido voluntariamente pelo indivíduo. A segunda seria a função educativa que representa a maneira de ensinar.[1] O jogo é mais do que um fenômeno fisiológico ou um reflexo psicológico, o jogo "é uma função significante, isto é, encerra um determinado sentido".[2]

## 3. Metodologia

O jogo foi desenvolvido utilizando a Linguagem de Programação C, padrão C99, e a biblioteca Allegro 5. A Allegro é uma das bibliotecas mais populares entre as pessoas que se iniciam na programação de jogos virtuais para computadores. Tal popularidade se justifica pelo fato dela proporcionar acesso fácil às rotinas comumente necessárias no desenvolvimento de jogos virtuais, tais como entrada de mouse, teclado e joystick, geração de gráficos, efeitos sonoros e gerenciamento de tempo (timers). Além disso, a biblioteca se destaca por ser livre, sem custos de licença de uso para o desenvolvedor.[3] O projeto foi versionado utilizando Git [4], e a hospedagem dos arquivos foi feita no Github [6], rede social integrada com sistema git, a qual proporcionou agilidade no gerenciamento do código fonte para grupo. A Figura 1 mostra um snapshot do repositório do grupo no Github.

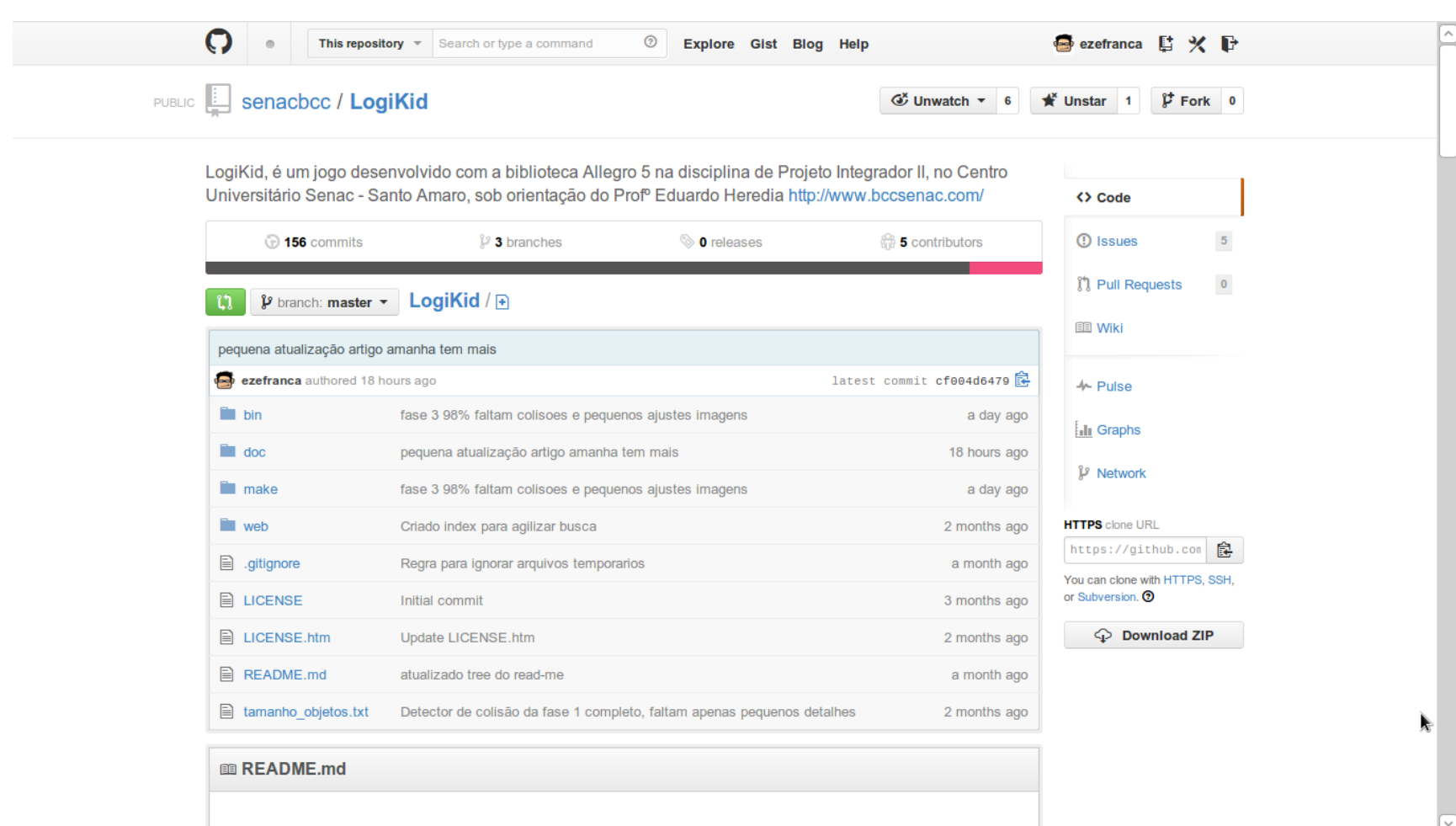


Figura 1: Repositório git do grupo hospedado no Github

Graficamente, usou-se elementos atrativos ao jogador, com a adição de diversos personagens, alguns dos personagens disponíveis são mostrados na Figura 2.

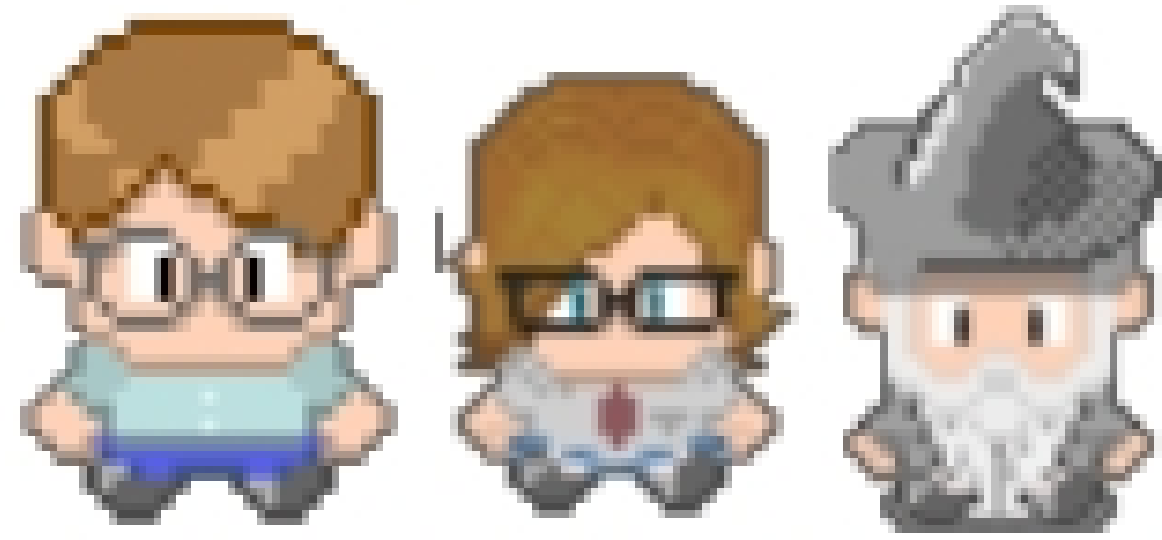


Figura 2: Alguns dos personagens disponíveis

A construção das fases contemplou um design didático, semelhante a programas de simulação de circuitos, como Logisim[5] por exemplo.

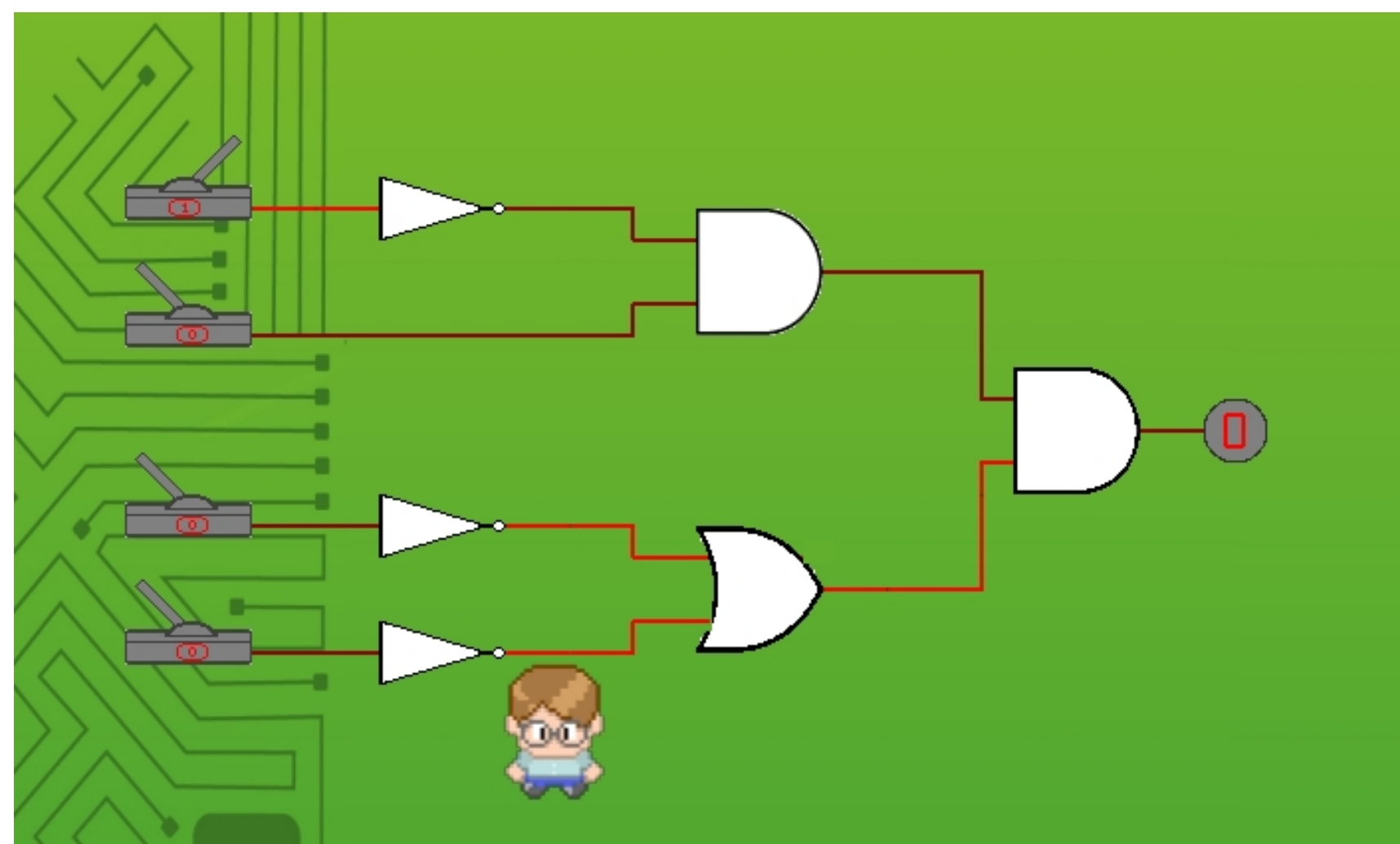


Figura 3: Exemplo de fase: composição de portas lógicas gerando um desafio

Na codificação do jogo, foi utilizada uma metodologia estruturada, com uso de funções e estruturas de dados, criando estruturas próprias para criação de telas e a game engine. A Figura 4 mostra diagrama da arquitetura adotada.

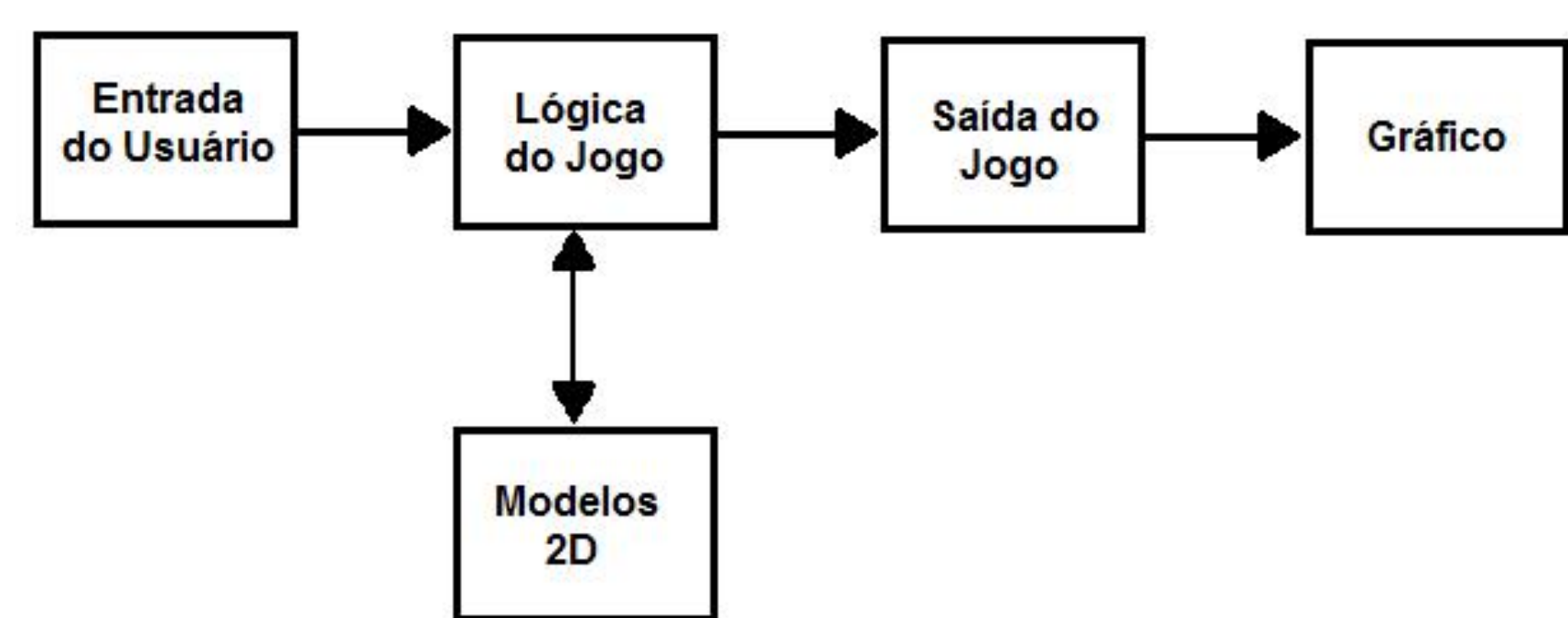


Figura 4: Arquitetura do jogo

O fato de trabalharmos com circuitos lógicos possibilitou o uso dos próprios operadores da linguagem C para criação da lógica do jogo, pois ela possui todos operadores lógicos necessários para este fim. O trecho de código abaixo exemplifica este uso.

```
void drawLogicLevel(bool gateOne, bool gateTwo, Level *level){
    if(!gateOne && !gateTwo)
        level->circ1 = al_load_bitmap("./data/00XX.png");
    if(!gateOne && gateTwo)
        level->circ1 = al_load_bitmap("./data/01XX.png");
    if(gateOne && !gateTwo)
        level->circ1 = al_load_bitmap("./data/10XX.png");
    if(gateOne && gateTwo)
        level->circ1 = al_load_bitmap("./data/11XX.png");
}
```

Listing 1: Exemplo de código para atribuição com Allegro 5, utilizando operadores lógicos da linguagem C

## 4. Resultados e Discussão

Para este trabalho foi muito importante o estudo dos conceitos de portas lógicas, necessários para planejar como seria o desenvolvimento do jogo Logikid. Foram abordados desde conceitos básicos até aos conceitos de circuitos avançados. A partir desse estudo foi possível criar o jogo Logikid, mediante os requisitos um jogo educativo. O resultado final, foi um jogo simples e interativo e de grande impacto visual, a Figura 5 mostra a explanação do conceito da porta NOT, (Inversora), ao manipular a alavanca o jogador pode observar o funcionamento básico desta porta.

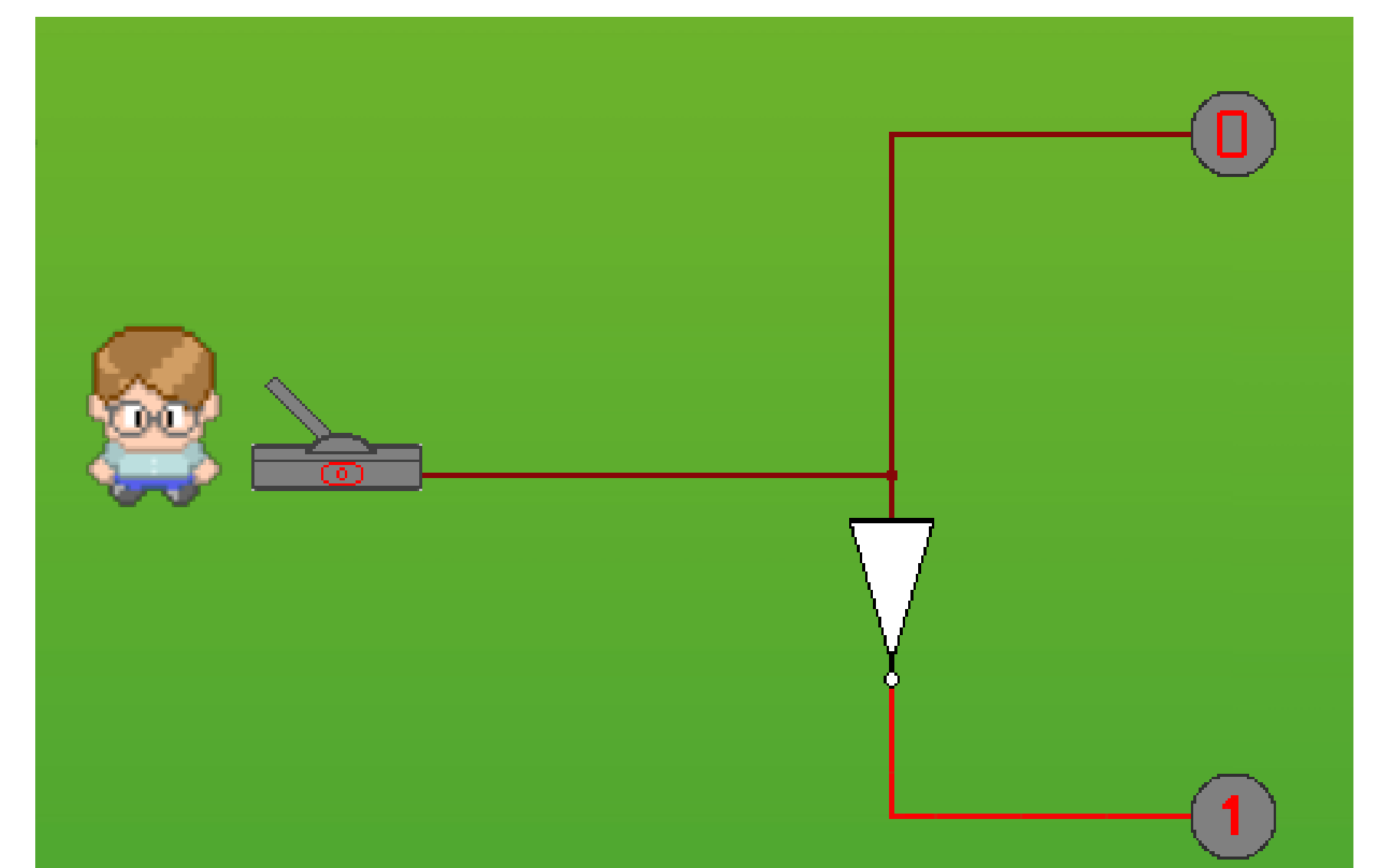


Figura 5: Fase 0: Tutorial da porta NOT, exemplo de funcionalidade do jogo

- O personagem puxa a alavanca
- A energização do circuito troca de caminho

## 5. Conclusão

Com os estudos realizados foi possível notar a vantagem do uso de softwares educativos no processo de ensino, pois o computador facilita o uso de atividades de ensino-aprendizagem pelo professor. Porém um jogo educativo deve ser projetado e desenvolvido com qualidade, pois sem uma boa base pedagógica e lúdica pode acabar servindo apenas de distração para o aluno e um jogo sem uma interface de qualidade pode fazer com que o aluno não se motive à utilizá-lo.

## Referências

- [1] Tizuko Kishimoto. *O jogo e a educação Infantil*, Cengage Learning, São Paulo, 2008.
- [2] Maria Carmen Silveira Barbosa. Jogo, brinquedo, brincadeira e a educação. *Educ. Soc. [online]*, Scielo, pg:398–404, 1997.
- [3] [Online]. Allegro 5.0 reference manual. <<https://www.allegro.cc/manual/5/>>, Último acesso em Novembro de 2013.
- [4] GIT. Manual de Referência. <<http://git-scm.com/docs>>, Último Acesso em Novembro de 2013.
- [5] Dr. Carl Burch. Documentação do Software LogiSim. <<http://ozark.hendrix.edu/burch/logisim/pt/docs.html>>, Acessado em Novembro de 2013.
- [6] Kris Shaffer. Push, Pull, Fork: GitHub for Academics. <<http://hybridpedagogy.com/Journal/GitHubforAcademics>>, Último Acesso em Novembro de 2013.