

# Porte do jogo *Traveling Will* para o *Nintendo Game Boy Advance*

Igor Ribeiro Barbosa Duarte e Vítor Barbosa de Araujo

Faculdade Gama  
Universidade de Brasília  
Brasil

10 de dezembro de 2018

# Objetivo geral

- O objetivo geral deste trabalho é reescrever o jogo *Traveling Will*, desenvolvido originalmente para PC na disciplina de Introdução aos Jogos Eletrônicos, para o *Nintendo Game Boy Advance*

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar o jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar métodos para carregamento do *level design* das fases do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar métodos para carregamento do *level design* das fases do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar métodos para carregamento do *level design* das fases do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar métodos para carregamento do *level design* das fases do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

# Objetivos específicos

Os objetivos específicos são:

- Comprimir imagens e músicas do jogo original para reduzir o uso de memória;
- Criar módulos para renderização de imagens e texto;
- Criar módulos para manipulação de *inputs* dos botões e carregamento de áudio;
- Criar módulos para detecção de colisões e manipulação de eventos;
- Criar métodos para carregamento do *level design* das fases do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.



# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- C++
- *devkitARM*
- *GIMP*
- *GRIT*
- *svconv*, *libressl* e *modplug tracker*
- *libretro*
- *SDL2*
- *LibRetro*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv*, *librosa* e *modplug tracker*
- *SDL*
- *SDL\_mixer*
- *SDL\_image*
- *SDL\_ttf*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv*, *librosa* e *modplug tracker*
- *VisualBoyAdvance-M*
- *libretro*
- *LibRetro*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv, librosa e modplug tracker*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *Libretro*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv, librosa e modplug tracker*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv*, *librosa* e *modplug tracker*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv*, *librosa* e *modplug tracker*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

# Ferramentas

Para o porte do jogo foram utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *GRIT*
- *avconv*, *librosa* e *modplug tracker*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*



# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** carregar e reproduzir arquivos de áudio;
- **Módulo de física:** simular a ação de gravidade e detectar colisões entre objetos do jogo.

# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** iniciar e parar músicas de fundo;
- **Módulo de física:** gerenciar a colisão de objetos e detectar colisões entre objetos do jogo.

# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** iniciar e parar músicas de fundo;
- **Módulo de física:** simular a ação de gravidade e detectar colisões entre objetos do jogo;

# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** iniciar e parar músicas de fundo;
- **Módulo de física:** simular a ação de gravidade e detectar colisões entre objetos do jogo;

# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** iniciar e parar músicas de fundo;
- **Módulo de física:** simular a ação de gravidade e detectar colisões entre objetos do jogo;

# Desenvolvimento da *engine*

Para este trabalho foi desenvolvida a *gbengine*.

- Módulos implementados: *input*, vídeo, gerenciador de memória, áudio e física;
- **Módulo de input:** detectar pressionamento dos botões do GBA.
- **Módulo de vídeo:** renderizar, animar e atualizar imagens;
- **Módulo gerenciador de memória:** garantir alocação segura e eficiente de memória;
- **Módulo de áudio:** iniciar e parar músicas de fundo;
- **Módulo de física:** simular a ação de gravidade e detectar colisões entre objetos do jogo;

# Módulo de *input*

# Módulo de vídeo



# Módulo gerenciador de memória

# Módulo de física

# Módulo utilitário

# Adaptação das músicas do jogo

# Abstração de níveis e objetos do jogo

# Adaptação das imagens do jogo

# Construção dos níveis do jogo

# Transição entre os níveis do jogo



# Comparação entre o porte e o jogo original

# Considerações Finais

Principais impedimentos durante a execução do trabalho:

- Entender detalhes do *hardware do GBA*;
- Testar o jogo no *console*;
- Adaptação de imagens e músicas do jogo;

# Considerações Finais

Principais impedimentos durante a execução do trabalho:

- Entender detalhes do *hardware do GBA*;
- Testar o jogo no *console*;
- Adaptação de imagens e músicas do jogo;

# Considerações Finais

Principais impedimentos durante a execução do trabalho:

- Entender detalhes do *hardware do GBA*;
- Testar o jogo no *console*;
- Adaptação de imagens e músicas do jogo;

# Considerações Finais

Pontos de melhoria pós-execução do trabalho:

- Melhor priorização das tarefas a serem executadas;
- Testes mais frequentes no *console*

# Considerações Finais

Pontos de melhoria pós-execução do trabalho:

- Melhor priorização das tarefas a serem executadas;
- Testes mais frequentes no *console*

# Considerações Finais

*É possível portar o jogo *Traveling Will*, desenvolvido para PC pelos autores deste trabalho, para o Nintendo Gameboy Advance, no contexto de um trabalho de conclusão de curso, com performance e jogabilidade próximos da versão para computador?*

R: **Sim, é possível.**

# Trabalhos futuros

- Melhorar módulo de áudio para carregar efeitos sonoros e pausar músicas;
- Implementar carregamento e utilização de fontes;
- Adicionar elementos de HUD e seleção de fases;
- Salvar o estado do jogo em memória;



# Trabalhos futuros

- Melhorar módulo de áudio para carregar efeitos sonoros e pausar músicas;
- Implementar carregamento e utilização de fontes;
- Adicionar elementos de HUD e seleção de fases;
- Salvar o estado do jogo em memória;
- Implementar um desfragmentador de memória na classe `MemoryManager`

# Trabalhos futuros

- Melhorar módulo de áudio para carregar efeitos sonoros e pausar músicas;
- Implementar carregamento e utilização de fontes;
- Adicionar elementos de HUD e seleção de fases;
- Salvar o estado do jogo em memória;
- Implementar um desfragmentador de memória na classe `MemoryManager`

# Trabalhos futuros

- Melhorar módulo de áudio para carregar efeitos sonoros e pausar músicas;
- Implementar carregamento e utilização de fontes;
- Adicionar elementos de HUD e seleção de fases;
- Salvar o estado do jogo em memória;
- Implementar um desfragmentador de memória na classe `MemoryManager`

# Trabalhos futuros

- Melhorar módulo de áudio para carregar efeitos sonoros e pausar músicas;
- Implementar carregamento e utilização de fontes;
- Adicionar elementos de HUD e seleção de fases;
- Salvar o estado do jogo em memória;
- Implementar um desfragmentador de memória na classe `MemoryManager`