

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

11 de julho de 2018

Contextualização do problema

- Uso de *engines* comerciais no desenvolvimento de jogos
- Impacto na performance dos jogos atuais
- Como lidar com cenários onde a performance é indispensável?

Contextualização do problema

- Uso de *engines* comerciais no desenvolvimento de jogos
- Impacto na performance dos jogos atuais
- Como lidar com cenários onde a performance é indispensável?

Contextualização do problema

- Uso de *engines* comerciais no desenvolvimento de jogos
- Impacto na performance dos jogos atuais
- Como lidar com cenários onde a performance é indispensável?

Introdução

- 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*

Introdução

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 código principal do jogo;
- Executar e testar o jogo desenvolvido na plataforma escolhida.

Introdução

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.

Introdução

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.

Introdução

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.

Introdução

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.

Introdução

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 serão utilizadas as seguintes ferramentas:

- C++
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
-
-

Ferramentas

Para o porte do jogo serão utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
- *Nintendo DS*
-

Ferramentas

Para o porte do jogo serão utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

Ferramentas

Para o porte do jogo serão utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

Ferramentas

Para o porte do jogo serão utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

Ferramentas

Para o porte do jogo serão utilizadas as seguintes ferramentas:

- *C++*
- *devkitARM*
- *GIMP*
- *VisualBoyAdvance-M*
- *Nintendo DS*
- *EZFlash II*

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** implementar a física de corpos rígidos;
- **Módulo de input:** receber o pressionamento de qualquer botão do *GBA*.

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** simular a ação de gravidade, detectar (opcionalmente) colisões entre objetos do jogo;
- **Módulo de input:** receber e processar entradas de qualquer botão do *GBA*.

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** simular a ação de gravidade, detectar (opcionalmente) colisões entre objetos do jogo;
- **Módulo de input:** receber o pressionamento de qualquer botão do *GBA*.

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** simular a ação de gravidade, detectar (opcionalmente) colisões entre objetos do jogo;
- **Módulo de input:** receber o pressionamento de qualquer botão do *GBA*.

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** simular a ação de gravidade, detectar (opcionalmente) colisões entre objetos do jogo;
- **Módulo de input:** receber o pressionamento de qualquer botão do *GBA*.

Desenvolvimento da *engine*

- Para este trabalho será desenvolvida uma *engine* que irá padronizar a utilização dos recursos providos pelo *GBA*;
- Os principais módulos a serem desenvolvidos serão: vídeo, áudio, física e *input*;
- **Módulo de vídeo:** renderizar imagens e textos, *horizontal scroll*;
- **Módulo de áudio:** iniciar, pausar e parar músicas de fundo e efeitos sonoros;
- **Módulo de física:** simular a ação de gravidade, detectar (opcionalmente) colisões entre objetos do jogo;
- **Módulo de input:** receber o pressionamento de qualquer botão do *GBA*.

Reescrita do jogo

- Todos os recursos do jogo como imagens, áudios e arquivos de fontes precisarão ser ajustados para um formato utilizável no GBA
- Os principais elementos do jogo podem ser divididos em: objetos do jogo e níveis

Reescrita do jogo

- Todos os recursos do jogo como imagens, áudios e arquivos de fontes precisarão ser ajustados para um formato utilizável no GBA
- Os principais elementos do jogo podem ser divididos em: objetos do jogo e níveis

Reescrita do jogo

Objetos do jogo

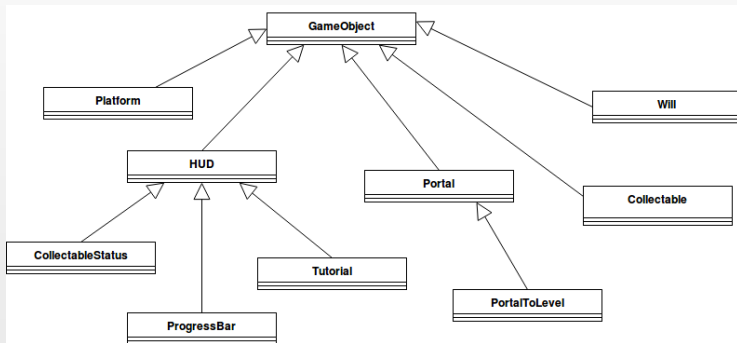


Figura: Modelagem inicial dos objetos do jogo

Reescrita do jogo

Níveis do jogo

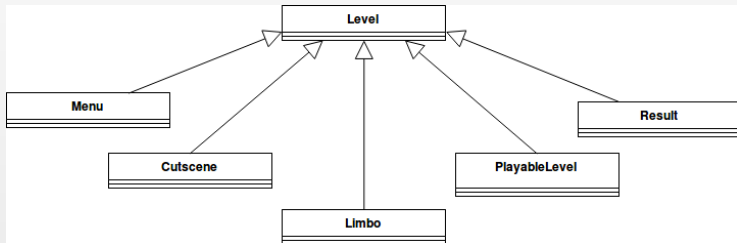


Figura: Modelagem inicial dos níveis do jogo

Desenvolvimento do protótipo

- A fim de atestar a viabilidade do porte do jogo *Traveling Will* para o *GBA*, foi feita uma versão funcional do menu original do jogo, já testada em um *Nintendo DS*;
- Para isso, a principal ferramenta utilizada foi a *libtonc*, que nessa versão inicial fez o papel de *engine* do jogo.

Desenvolvimento do protótipo



Figura: Jogo original sendo executado em um PC

Desenvolvimento do protótipo

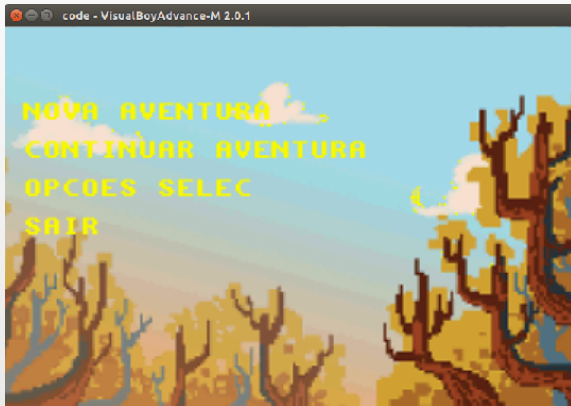


Figura: Protótipo sendo executado em um emulador de GBA

Desenvolvimento da *gbengine*

- Após a finalização do protótipo inicial, foi iniciado o desenvolvimento da *engine* que irá substituir a *libtonc* na versão final do jogo. Ela irá padronizar a utilização dos recursos providos pelo *GBA*;
- Até o momento, o módulo de *input* e parte do módulo de vídeo foram implementados.

Módulo de *Input*

- Registrador
- Checagem dos estados

Módulo de *Input*

- Registrador
- Checagem dos estados

Módulo de *Input*

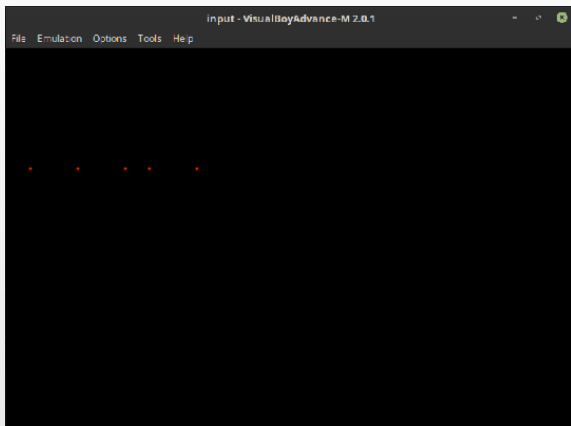


Figura: Teste de pressionamento de botões no emulador

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *sprites* em qualquer região válida de memória

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado
 - Escolha do *sprite* a ser utilizado
 - Escolha do *sprite* a ser utilizado

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado (incompleto)
 - Escolha do *background* a ser utilizado em qualquer região válida de memória

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado (incompleto)
 - Renderização de *backgrounds* apenas em uma região específica de memória

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado (incompleto)
 - Renderização de *backgrounds* apenas em uma região específica de memória

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado (incompleto)
 - Renderização de *backgrounds* apenas em uma região específica de memória

Módulo de Vídeo

- As principais funcionalidades que o módulo de vídeo deverá ter são:
 - Escolha modo de vídeo
 - Escolha do *background* a ser utilizado
 - Renderização de *backgrounds* em qualquer região válida de memória
 - Renderização de *sprites* em qualquer região válida de memória
- Funcionalidades da versão atual
 - Escolha do modo de vídeo (incompleto)
 - Escolha do *background* a ser utilizado (incompleto)
 - Renderização de *backgrounds* apenas em uma região específica de memória

Considerações Finais

- Para testar a viabilidade de portar o jogo, foi desenvolvido um protótipo usando a *libtonc*
- Início do desenvolvimento da *gbengine*, disponível em <https://github.com/traveling-will-gba>

Considerações Finais

- Para testar a viabilidade de portar o jogo, foi desenvolvido um protótipo usando a *libtonc*
- Início do desenvolvimento da *gbengine*, disponível em <https://github.com/traveling-will-gba>

Trabalhos futuros

As próximas tarefas a serem realizadas são:

- Conversão dos demais recursos do jogo;
- Finalização do módulo de vídeo da *engine*;
- Implementação dos módulos de física e áudio na *engine*;
- Reescrita do jogo utilizando a *engine* desenvolvida;

Trabalhos futuros

As próximas tarefas a serem realizadas são:

- Conversão dos demais recursos do jogo;
- Finalização do módulo de vídeo da *engine*;
- Implementação dos módulos de física e áudio na *engine*;
- Reescrita do jogo utilizando a *engine* desenvolvida;

Trabalhos futuros

As próximas tarefas a serem realizadas são:

- Conversão dos demais recursos do jogo;
- Finalização do módulo de vídeo da *engine*;
- Implementação dos módulos de física e áudio na *engine*;
- Reescrita do jogo utilizando a *engine* desenvolvida;

Trabalhos futuros

As próximas tarefas a serem realizadas são:

- Conversão dos demais recursos do jogo;
- Finalização do módulo de vídeo da *engine*;
- Implementação dos módulos de física e áudio na *engine*;
- Reescrita do jogo utilizando a *engine* desenvolvida;

Cronograma

| TAREFAS | JULHO | AGOSTO | SETEMBRO | OUTUBRO | NOVEMBRO | DEZEMBRO |
|---|-------|--------|----------|---------|----------|----------|
| Finalizar desenvolvimento do módulo de input | X | X | | | | |
| Finalizar desenvolvimento do módulo de vídeo | X | X | X | | | |
| Desenvolver módulo de áudio | X | X | X | | | |
| Desenvolver módulo de física | X | X | X | | | |
| Finalizar implementação do menu | | X | X | | | |
| Implementar rolagem infinita do background dos níveis | | | X | X | | |
| Implementar mecanismo de renderização das plataformas | | | X | X | | |
| Implementar movimentos do personagem | | | X | X | | |
| Implementar mecanismo de renderização dos coletáveis | | | X | X | | |
| Implementar telas de finalização do nível | | | | X | | |
| Carregar níveis a partir do level design | | | | X | X | |
| Implementar seletor de fases (Limbo) | | | | X | X | |
| Implementar opções do menu | | | | | X | X |
| Implementar tutorial | | | | | X | X |

Figura: Cronograma de desenvolvimento