

# 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



# 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 uma interface gráfica para o 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*

- A representação dos objetos do jogo será feita por meio da classe *GameObject*



# Desenvolvimento da *engine*

GameObject
- x: double - y: double
+ update(dt): void + draw(): void

Figura: Modelagem inicial da classe *GameObject*

# Desenvolvimento da *engine*

- Principais componentes: 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*:** ler o estado dos botões e o joystick;

# Desenvolvimento da *engine*

- Principais componentes: 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*

- Principais componentes: 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*

- Principais componentes: 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*

- Principais componentes: 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

# Reescrita do jogo

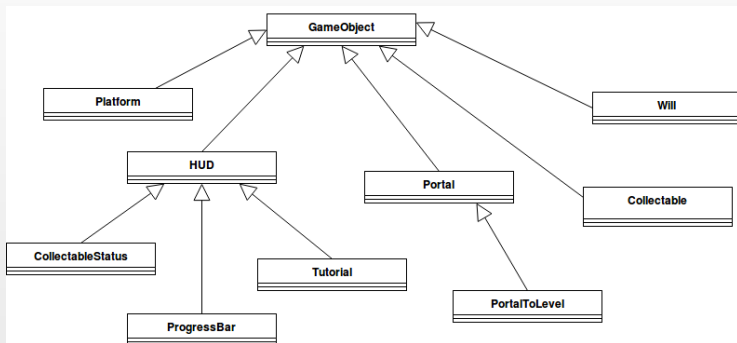


Figura: Modelagem inicial dos objetos do jogo



## Reescrita do jogo

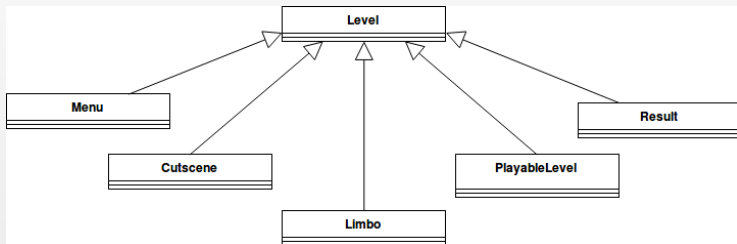


Figura: Modelagem inicial dos níveis do jogo

# Considerações Finais



# Próximas tarefas

As próximas tarefas a serem realizadas são:

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

# Próximas tarefas

As próximas tarefas a serem realizadas são:

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

## Próximas tarefas

As próximas tarefas a serem realizadas são:

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

## Próximas tarefas

As próximas tarefas a serem realizadas são:

- Conversão dos demais recursos do jogo;
- Aprimoramento 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