

Universidade Federal de São Carlos  
Engenharia de Computação

Laboratório de Arquitetura e Organização de Computadores 2

## **Segundo Relatório**

Alunos: Luan Vinicius Moraes da Silva (RA: 744342)  
Antônio Caique dos Santos (RA: 744332)

Professor: Dr. Luciano Neris

06/2019  
São Carlos - SP, Brasil

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Etapas de Desenvolvimento . . . . .	1
1.2	Divisão de Tarefas . . . . .	1
1.3	Controle de Versões . . . . .	2
<b>2</b>	<b>Descrição</b>	<b>3</b>
2.1	Estrutura de Dados . . . . .	3
2.2	Variáveis Principais . . . . .	3
<b>3</b>	<b>Elementos Específicos</b>	<b>4</b>
3.1	Procedimentos . . . . .	4
3.1.1	Procedimento Principal ( <i>Main</i> ) . . . . .	4
3.1.2	Movimentação . . . . .	4
3.1.3	Contador . . . . .	4
3.1.4	Imprime tela . . . . .	4
3.1.5	Movimentação das criaturas . . . . .	5
3.2	Configurações disponíveis ao usuário . . . . .	5
3.3	Diagrama de Estados . . . . .	6
<b>4</b>	<b>Análise Crítica e Discussão</b>	<b>8</b>
<b>A</b>	<b>Imagens dos níveis, de 1 a 4</b>	<b>9</b>
	<b>Referências</b>	<b>11</b>

## **Lista de Figuras**

1	Diagrama de estados - menu inicial . . . . .	6
2	Diagrama de estados - níveis do jogo e encerramento . . . . .	7
3	Mapa nível 1 . . . . .	9
4	Mapa nível 2 . . . . .	9
5	Mapa nível 3 . . . . .	10
6	Mapa nível 4 . . . . .	10

## **Lista de Tabelas**

1	Dados Armazenados . . . . .	3
---	-----------------------------	---

# **1 Introdução**

O presente relatório objetiva a apresentação mais detalhada do jogo a ser desenvolvido na disciplina Laboratorio de Arquitetura e Organização de Computadores 2, ministrada pelo docente Dr. Luciano Neris, no primeiro semestre de 2019, na Universidade Federal de São Carlos.

O jogo, intitulado "Super @ Bros", sera desenvolvido em linguagem Assembly e fazendo uso da biblioteca Irvine32 (1). Como objetivo secundário, destaca-se aprofundar o conhecimento na linguagem anteriormente citada e na referida biblioteca, assim como os recursos que esta oferece. Não obstante, ser a utilizado o montador Microsoft Macro Assembler (MASM) (2), o qual suporta as arquiteturas IA-32 e x86-64 para MS-DOS e Microsoft Windows. O ambiente de desenvolvimento - Integrated Development Environment (IDE) - sera o Visual Studio.

## **1.1 Etapas de Desenvolvimento**

As etapas de desenvolvimento do jogo estão enumeradas a seguir:

1. Planejamento da lógica do jogo
2. Construção das telas
3. Desenvolvimento dos procedimentos
4. Procedimento de Atualização
5. Procedimento de Movimentação do personagem
6. Procedimento de Movimentação das criaturas
7. Procedimento de Imprimir tela
8. Procedimento de Contador de Tempo Regressivo
9. Detecção de Colisão
10. Testes
11. Correção de Eventuais Erros

## **1.2 Divisão de Tarefas**

No desenvolvimento do jogo, ficou entendido que ambos os membros da equipe deveriam ter igual participação. Tornando, assim, o conhecimento integro das etapas e desenvolvimento do projeto.

Primeiramente, a dupla se reuniu para decidir a lógica e funcionamento do jogo, as variáveis principais, padronização de codificação e atribuição as atividades iniciais de cada um, resultando na divisão inicial entre desenvolvimento do Menu inicial e movimentação do personagem para um, e desenvolvimento dos níveis para outro.

### **1.3 Controle de Versões**

Afim de organizar e agilizar o desenvolvimento do jogo será usado um ambiente de controle de versão de software (github). O repositório do jogo juntamente com seu código e demais informações podem ser encontrados em [https://github.com/MoonLean/super\\_at\\_bros](https://github.com/MoonLean/super_at_bros)

## 2 Descrição

### 2.1 Estrutura de Dados

À princípio, as estruturas de dados a serem utilizadas no desenvolvimento do jogo são: Pilhas.

**Pilhas:** As chamadas de procedimentos são feitas pelas instruções "CALL" e "RET", que por sua vez fazem manipulação dos endereços das instruções seguinte e anterior.

### 2.2 Variáveis Principais

Os principais dados armazenados são:

Tabela 1: Dados Armazenados

Label	Tamanho	Conteúdo	Dado
posPers	DWORD	"Posição do personagem"	1450
mapa1	BYTE	"mapa1"	caracteres representando o mapa
mapa2	BYTE	"mapa2"	caracteres representando o mapa
mapa3	BYTE	"mapa3"	caracteres representando o mapa
mapa4	BYTE	"mapa4"	caracteres representando o mapa
tempoInic	DWORD	"Mantém guardado o tempo do início do jogo"	?
tempoAtual	DWORD	"Contém o tempo no instante do jogo"	?
criatVert	WORD	"Mantém a posição das criaturas que se movem na vertical salva."	?
criatHoriz	WORD	"Mantém a posição das criaturas que se movem na horizontal salva.", 0	?
pontuacao	DWORD	"Contém a pontuação na fase atual do jogador."	?
pontuaTotal	DWORD	"Contém a somatória das pontuações de todas as fases."	?
tempoTotal	DWORD	"Contém a somatória do tempo gasto em todas as fases."	?
missao	BYTE	"Variável destino dos caracteres "adquiridos" pelo personagem."	?,?,?

## 3 Elementos Específicos

### 3.1 Procedimentos

#### 3.1.1 Procedimento Principal (*Main*)

O procedimento *Main* além de guardar o início da execução do jogo, comporta também o menu inicial, e a interação do usuário, caso queira navegar para o procedimento "Instruções", "Atribuições" ou "jogar". O procedimento *Main* pode ser melhor representado pelo Diagrama de Estados, na Seção 3.3.

#### 3.1.2 Movimentação

Serão quatro procedimentos de movimentação, para cima (com o procedimento UP), para baixo (com o procedimento DOWN), para esquerda (com o procedimento LEFT) e para direita (com o procedimento RIGHT).

A movimentação depende do comando de teclado vindo do usuário:

- Tecla 'w': Indica movimento para cima.;
- Tecla 's': Indica movimento para baixo;
- Tecla 'a': Indica movimento para esquerda;
- Tecla 'd': Indica movimento para direita.

A movimentação será feita a partir do valor contido na variável "posPers", com o valor contido nele, é possível saber a posição atual, verificar a posição requerida e se possível, atualizar a tela com o personagem na nova posição

#### 3.1.3 Contador

O procedimento contador será utilizado para atualizar o tempo de jogo, que se inicia em 0 e vai até 120 segundos e é atualizado a cada chamada de procedimento.

#### 3.1.4 Imprime tela

A dinâmica do jogo só é visível com a alteração das posições dos personagens e criaturas, a partir disso, o procedimento "IMPRIME\_MAPA" é criado, para limpar o console, imprimir o mapa juntamente com as variáveis de pontuação e tempo.

### **3.1.5 Movimentação das criaturas**

As criaturas são obstáculos móveis no jogo que possuem posição e sentido de movimento (horizontal ou vertical) que possuem necessidade de terem suas posições alteradas constantemente.

## **3.2 Configurações disponíveis ao usuário**

Por ser obrigatório a utilização de sons no desenvolvimento do jogo, opta-se por fazer uma configuração disponível ao usuário a ativação ou não dos sons, através do pressionamento da tecla "P" ou "p" em qualquer momento da execução do jogo.



### 3.3 Diagrama de Estados

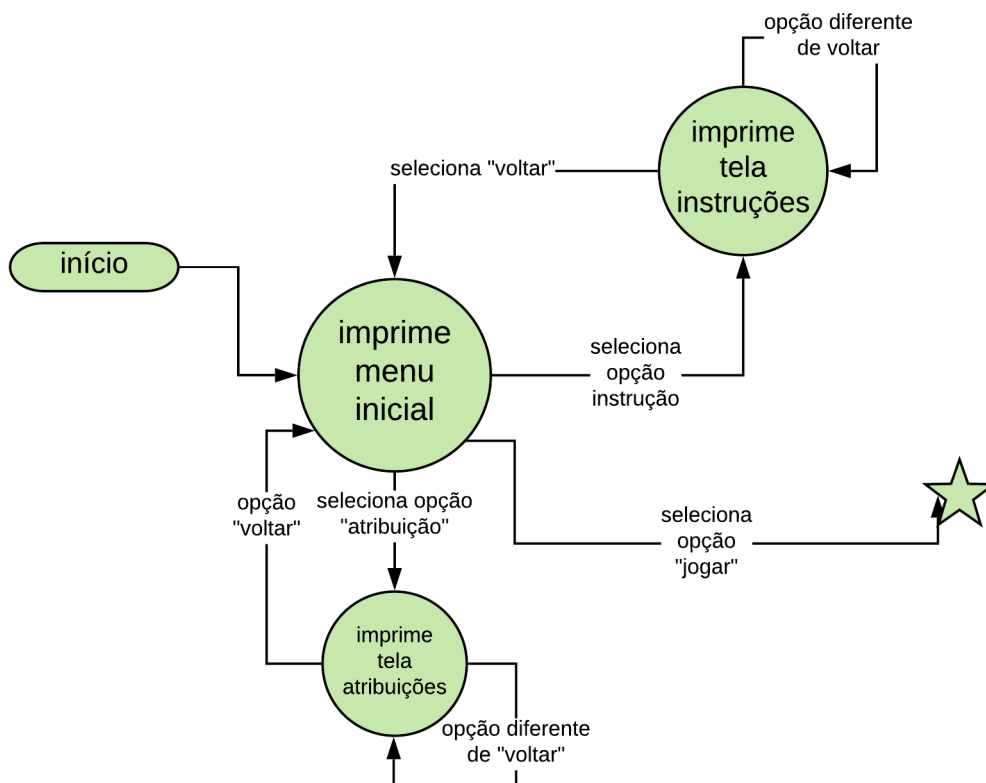


Figura 1: Diagrama de estados - menu inicial

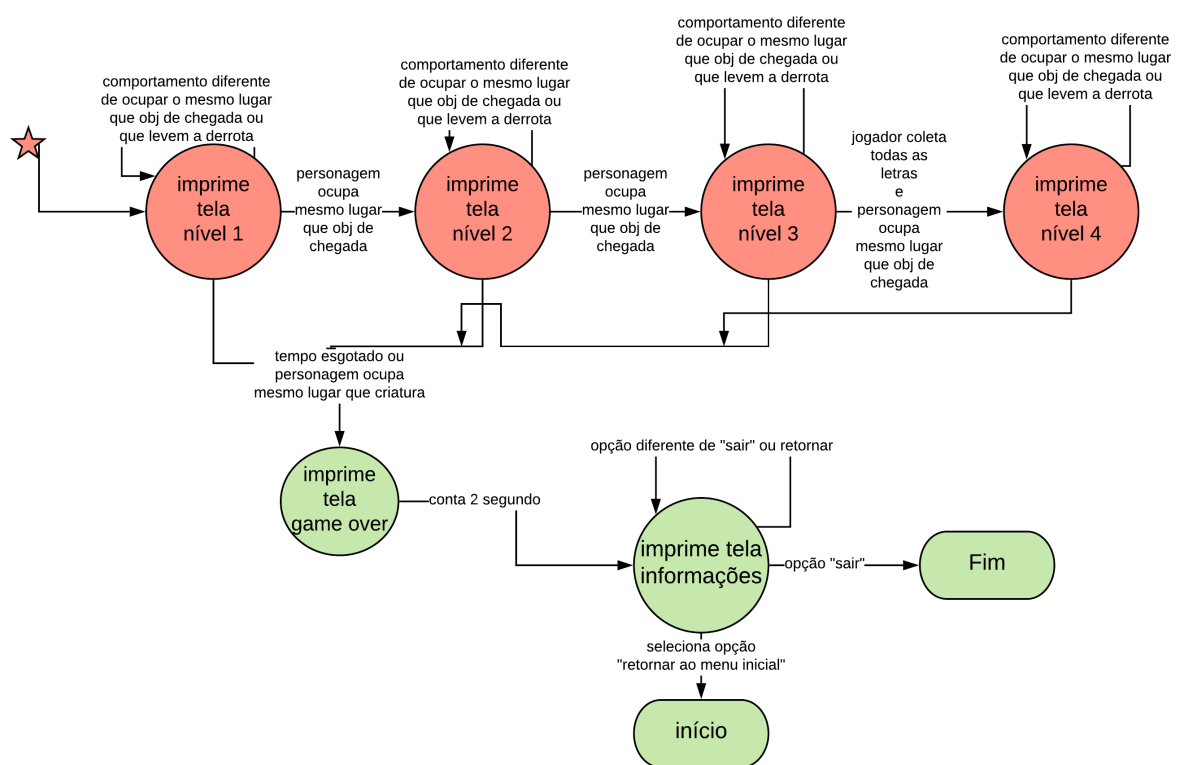


Figura 2: Diagrama de estados - níveis do jogo e encerramento

## **4 Análise Crítica e Discussão**

As maiores dificuldades encontradas até o presente momento do desenvolvimento do jogo foram relacionadas a pesquisa de formas de representação do mapa da forma mais otimizada possível, a movimentação tanto do personagem quanto das criaturas, a lógica para identificar colisões e quais tipos de variáveis melhor se adéquam a cada caso.

Foram feitas alterações em relação ao projeto inicial, primeiramente, a barra de informações apresentada durante o tempo de execução do jogo encontra-se na parte inferior da tela. Em segundo lugar, nos níveis 3 e 4 o jogador deve "coletar" as letras dispostas no mapa para formar uma palavra e "abrir" o portão que impede o acesso ao avanço de fase. Por fim, foi aumentado o tempo de 90 segundo para 120 (2 minutos) afim de deixar o jogo mais possível.

No geral, acredita-se que o desenvolvimento do jogo até o momento é aceitável.

## A Imagens dos níveis, de 1 a 4

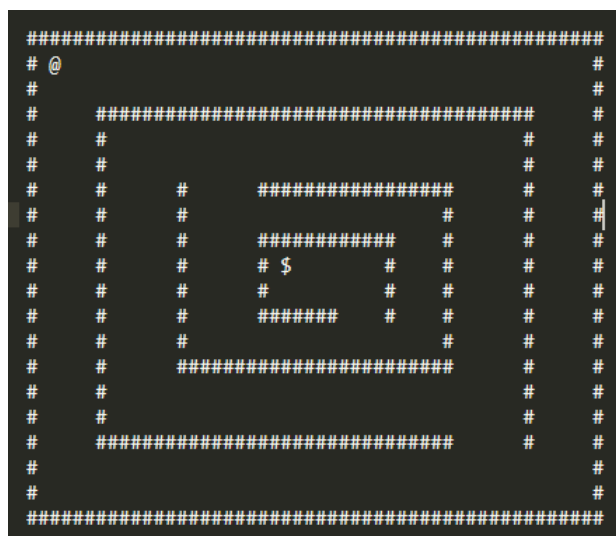


Figura 3: Mapa nível 1



Figura 4: Mapa nível 2

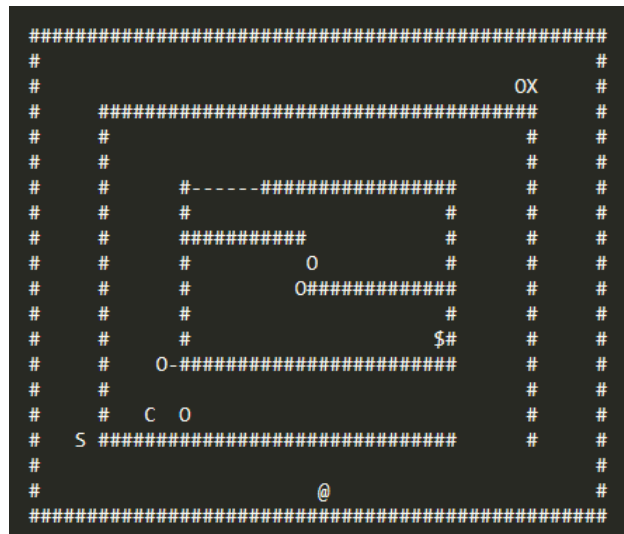


Figura 5: Mapa nível 3

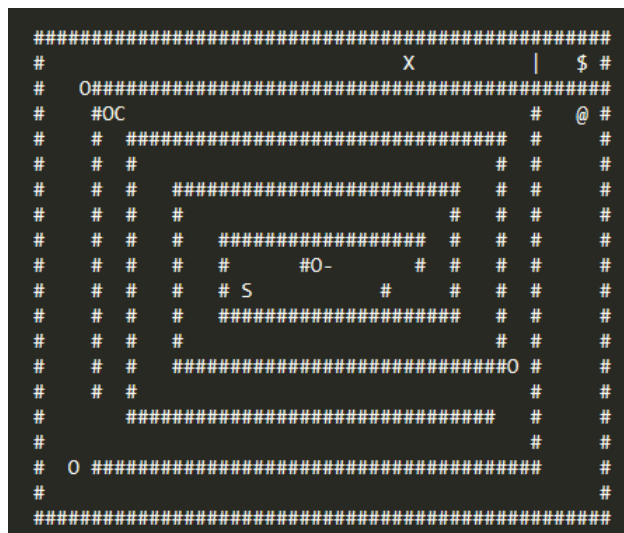


Figura 6: Mapa nível 4

## Referências

- 1 IRVINE Library Help. Disponível em: <http://programming.msjc.edu/asm/help/index.html?%20page=source>. Acesso em: 28 abr. 2019.
- 2 THE MASM32 SDK. Disponível em: <http://www.masm32.com/>. Acesso em: 28 abr. 2019.