

# Trabalho Prático – Etapa 1: Requisitos, Projeto e Interface

## 1. Especificação

Esta seção apresenta as características fundamentais do projeto Peck-Man em Java relativas à sua especificação.

### 1.1 Informações gerais

O objetivo desta etapa é definir as funcionalidades e restrições do jogo Peck-Man. O jogo será desenvolvido em Java e deverá conter um mapa com paredes, pastilhas e fantasmas. O jogador controla o Peck-Man com o teclado, devendo comer todas as pastilhas sem ser capturado pelos fantasmas.

### 1.2 Mudanças em relação à Etapa anterior

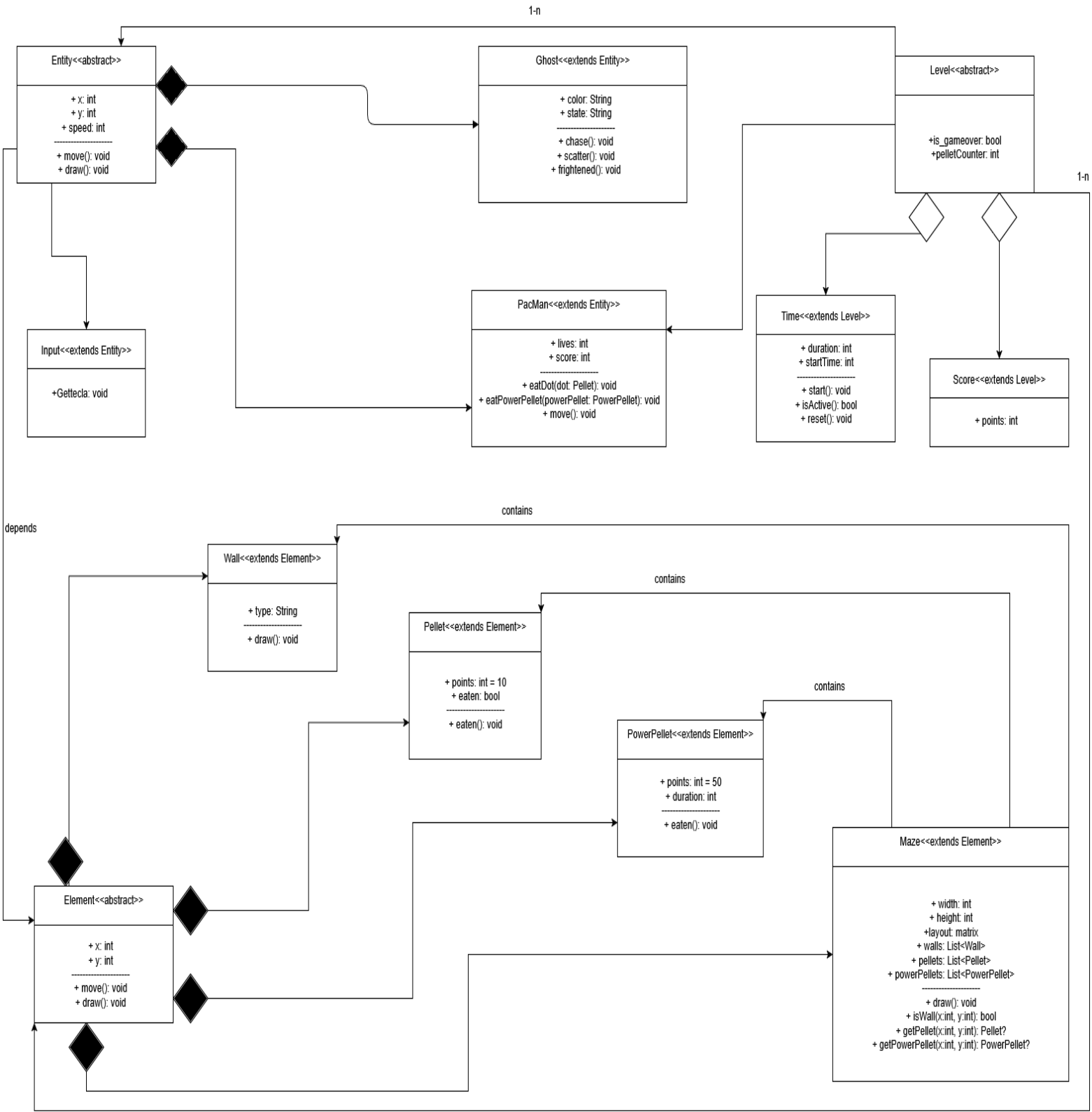
Não se aplicam mudanças, pois esta é a primeira etapa do projeto.

### 1.3 Requisitos

Abaixo estão listados os requisitos funcionais e não funcionais do sistema.

Requisitos Funcionais	Requisitos Não-Funcionais
<b>RF-1:</b> O jogo deve ser graficamente e mecanicamente atualizado a uma taxa constante que não sofre erros e atrasos (ex.: 60fps)	<b>RNF-1:</b> A pontuação do jogador deve ser atualizada conforme os itens relevantes são coletados por Peck-Man.
<b>RF-2:</b> O jogador deve ser capaz de movimentar o personagem (Peck-Man) de forma responsiva, que não apresente inconsistências e que se relacione 1-a-1 ao input do teclado.	<b>RNF-2:</b> O jogo deve ser graficamente e mecanicamente atualizado a uma taxa constante que não sofre erros e atrasos (ex.: 60fps)
<b>RF-3:</b> Peck-Man deve interagir com todos os objetos e entidades do jogo de forma fidedigna a sua versão original.	<b>RNF-3:</b> Cada um dos fantasmas deve ter comportamento específico não só relacionado ao fantasma em si como ao seu estado atual.
<b>RF-4:</b> As entidades não controláveis devem agir de forma natural tanto em resposta ao jogador quanto à parte de sua interferência em suas rotinas de comportamento.	<b>RNF-4:</b> As animações dos fantasmas e do jogador devem representar acuradamente o estado mecânico de ambos dentro do campo de jogo.
<b>RF-5:</b> O estado do jogo deve corresponder às ações de cada objeto/entidade conforme elas interagem com o sistema.	<b>RNF-5:</b> O jogo deve ser graficamente e mecanicamente atualizado a uma taxa constante que não sofre erros e atrasos.

1.4 Projeto



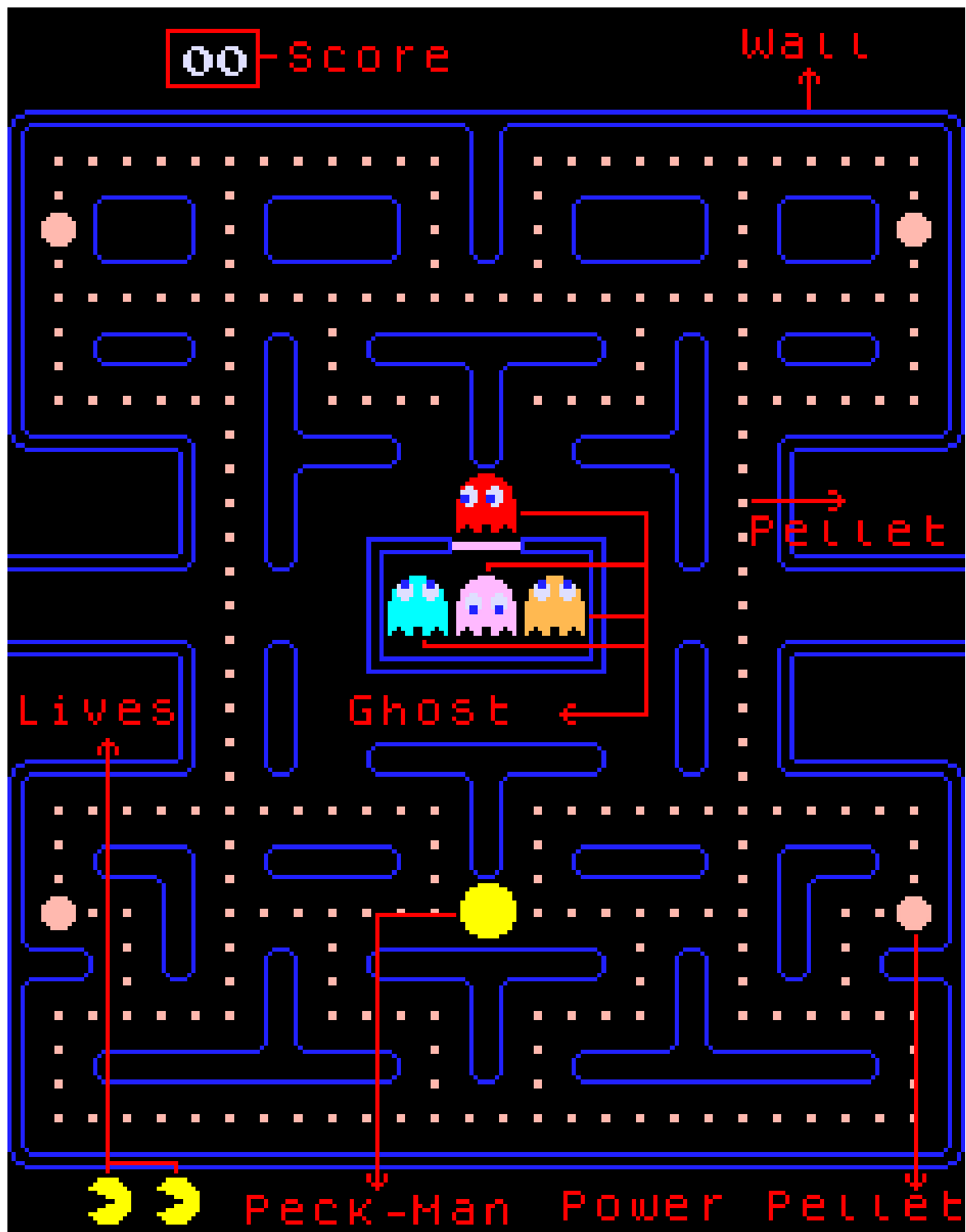
Esta implementação foi conceitualizada a fim de tentar imitar os comportamentos da versão original *arcade* de Pac-Man. Foi determinado que existe a necessidade de separar os objetos disponíveis no jogo em *entidades*, que possuem características de movimentação e podem ser relacionadas com objetos “vivos”, e *elementos*, que dizem respeito aos obstáculos e itens dispostos no layout do labirinto.

Além disso, é necessária a implementação de algum tipo de *manager* para as partes que controlam o estado de jogo dentro no nível, como o tempo e a pontuação do jogador.

Cada classe foi projetada para associar-se de forma simples com elementos claramente visíveis da interface do jogo original, de forma a facilitar a implementação das relações entre esses objetos como classes em Java.

Pac-man não possui um diagrama de relações demasiadamente complexo pois não existe uma grande variedade de elementos no cenário, logo os itens disponibilizados no diagrama de classes devem capturar uma fidedignidade alta em relação ao jogo original.

### 1.5 Interface com o Usuário



O usuário irá utilizar as setas do teclado para controlar Peck-Man por entre o labirinto, navegando os elementos listados na imagem com os nomes relacionados às classes conforme dispostas no diagrama da seção 1.4.