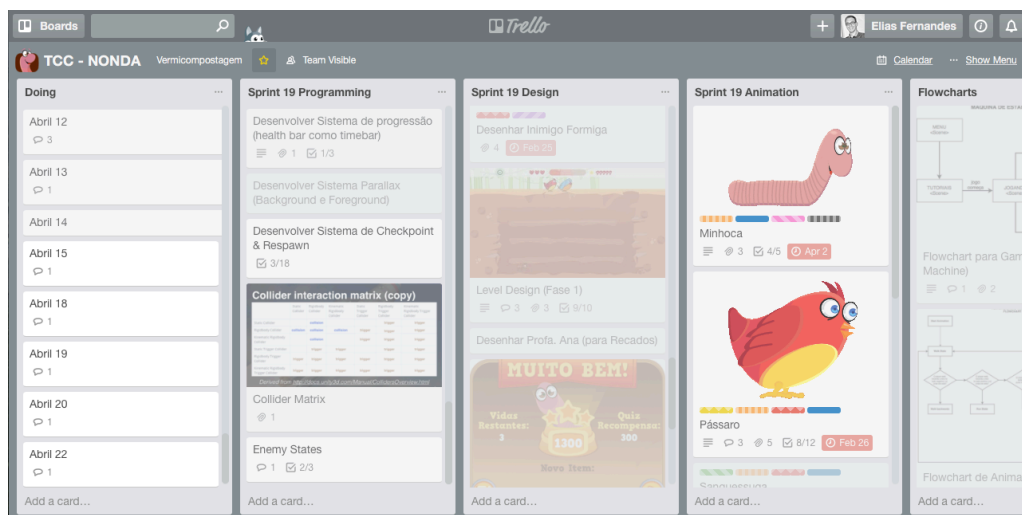


Para armazenar todos os products backlogs (ver Seção 5.2), será utilizado o Trello, que é uma ferramenta colaborativa de tarefas com versão grátis e paga. Como a versão grátis do Trello é suficiente para listar os requisitos do jogo, as atividades serão divididas em Cards (cartões) e Lists (listas). Na Figura 2, utiliza-se Lists contendo tarefas para um dia (Doing) e divisões entre diferentes áreas (Design, Animação, Programação, Testes e Bugs etc), sempre contendo a Sprint em questão, sendo que terá um List contendo o total de Sprints do jogo. Na Figura 12 (ver Seção 6), contém a lista completa de atividades.



**Figura 2** – Estrutura do Trello usando Scrum Solo

Fonte: Autoria Própria

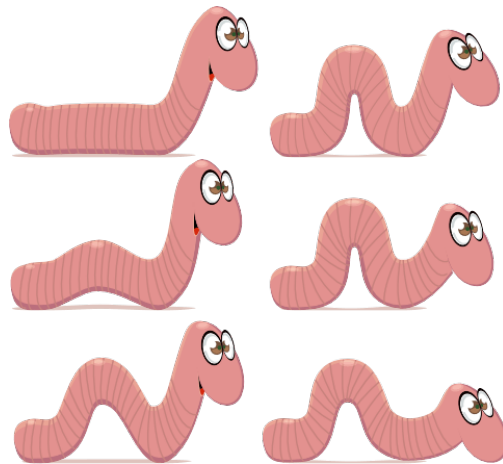


**Figura 3** – Fluxo de uma sprint específica de uma entrega parcial do projeto

Fonte: <https://www.scrumsolo.wordpress.com/>

## 5.2 Design Pattern MVC

Entre os engenheiros de software e arquitetos foi amplamente aceito que a concepção de aplicações em conformidade a esses *designs pattern* facilitariam a reutilização da experiência e conhecimento adquiridos por *experts* ao longo de exaustivos esforços em desenvolver um software de alto nível no mundo real (MASOVER, 2004). A arquitetura *Model View Controller* – MVC – justifica a afirmação do Masover (2004) e o fluxograma contido na Figura 4 possibilita a compreensão do reuso da experiência através da divisão das três camadas de aplicação Modelo de Negócio, Interface de Usuário e Lógica de programação. Em geral, o modelo gerencia os dados do software, a Interface é a camada de apresentação para o usuário e a Lógica trata os eventos para a interface (YONGLEI, 2002).



**Figura 7** –Personagem Nonda: Spritesheet com animação de Pular  
Fonte: Autoria Própria

### 5.11 Controle

Do Personagem:

O personagem estará sempre se movendo, nunca parado. Para controlar, é necessário deslizar o dedo sobre tela (*swipe*). Os controles disponíveis são:

Direcionais

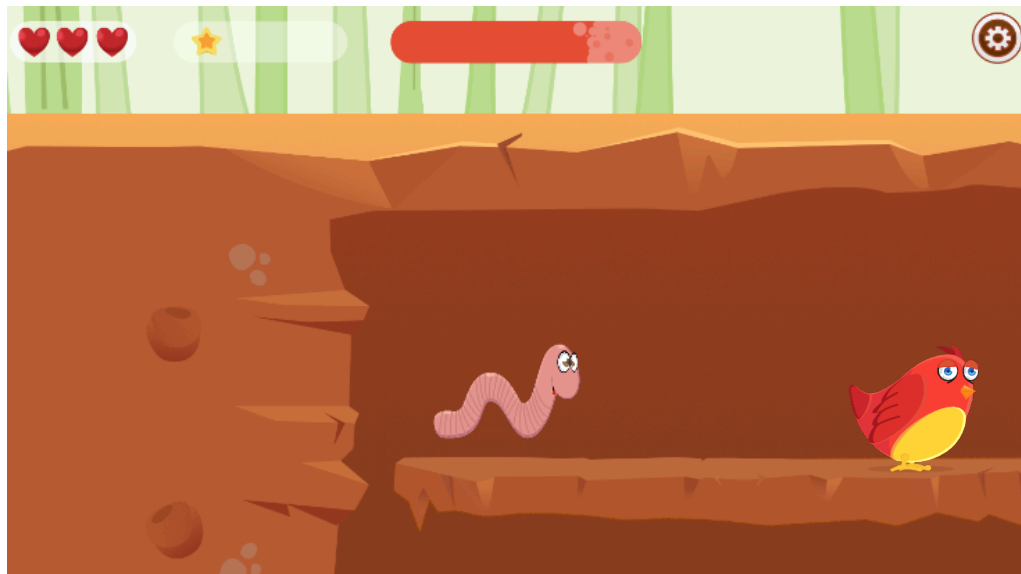
- Toque no lado oposto que o personagem está movendo para mudar direção.
- Cima (Tocar acima do personagem para pular – quando tiver inimigo próximo ou pular de uma plataforma para outra)
- Toque (várias vezes) em cima do inimigo para dar dano (Se Nonda tiver com bomba ácida, será lançada logo em seguida)

Do inimigo:

- Serão randômicos conforme a especialidade de cada um.

### 5.12 Interface

Na interface, o foco é no HUD – *head-up display*, termo utilizado para visualizar todos os elementos de interesse do jogador, tais como barra de progresso, tempo restante, quantidade de vidas, pontuações, indicações de localidade para atacar etc – do jogo, conforme ilustra a Figura 8 abaixo.



**Figura 8** – Nonda: HUD – Posicionamento do UI  
Fonte: Autoria Própria

### 5.13 Inimigos

O inimigo no contexto do jogo são espécies distintas que aparecem para atrapalhar o trabalho de Nonda no decorrer do jogo.

#### 5.13.1 Pássaros



**Figura 9** – Jogo Nonda: Inimigo Pássaros  
Fonte: Autoria Própria

### 5.13.2 Sanguessuga



**Figura 10** – Jogo Nonda: Inimigo Sanguessuga  
Fonte: Autoria Própria

### 5.13.3 Formiga



**Figura 11** – Jogo Nonda: Inimigo Formiga  
Fonte: Autoria Própria

## 5.14 Fases

As fases terão a mesma estrutura citada no Tópico 5.6.

## 5.15 Level Design

O jogo vai se passar somente em um cenário, pois se trata de um ambiente de cativeiro da minhoca, que é característica básica da vermicompostagem. Tem a possibilidade desse cativeiro mudar de cor, conforme a qualidade do adubo.

## 5.16 Arte

A arte é baseada no minhocário da UTFPR, tendo apenas como *background* essa mudança de local, conforme a fase. Foi escolhida o estilo *cartoon* para dar maior diversão ao jogador, pois não se sente na obrigação de estar em um simulador ou um jogo de primeira pessoa, por exemplo. Outro fator para escolha é pela habilidade artística adquirida com *cartoons* ao longo dos anos.