

# **PlantGoshi**

## **Projeto Integrador III - Sistema Autônomo**

**Anderson J. Silva, Felipe R. de Luca, Nelson J. Dressler**

<sup>1</sup>Bacharelado em Ciência da Computação – Centro Universitário Senac - Santo Amaro  
São Paulo - SP - Brasil  
2015

*Resumo. [ ... ]*

### **1. Introdução**

O Projeto consiste num jogo de simulação de criação de plantas e do seu cuidado contra a presença de invasores e/ou criaturas nocivas à sua sobrevivência no seu habitat natural, a terra. O jogador terá poderes por intermédio de uma varinha mágica e, ao longo do jogo, deverá utilizá-los para alimentar e aprimorar a planta ou a árvore e também combater as pragas aparentes através de interação com diferentes cores emitidas pelo LED.

Os frutos nascem verdes em pontos randômicos e demoram um tempo  $x$  para seguir para a segunda etapa. Se o fruto no estado verde não for "regado" (ou, poder de mágica) depois de um tempo determinado, ele vai ficar maduro com praga ou ficará estragado. Isso força o usuário a usar o poder de "regar" (ou, poder de música) no fruto.

Quando o fruto fica maduro, depois de um tempo  $x$  ele cria praga, forçando o usuário a usar o poder de "remover" para eliminar a praga do fruto maduro.

Com o fruto maduro e sem praga o usuário poderá usar o poder de "colher" para colher o fruto e somar pontos. Se o usuário colher fruto verde, maduro com praga ou estragado ele perde pontos. Caso deixe o fruto estragar também pontos são descontados. Depois que uma quantidade  $x$  de frutos estragar o jogo termina.

### **2. Estrutura principal do jogo**

#### **2.1. Layout**

Na parte inferior, aparecem os ícones com os poderes disponíveis, cada uma com uma cor diferente (azul: hidratar, vermelho: combater, verde: tocar a música); na parte superior esquerda, uma barra de progresso do crescimento da árvore (branco, verde claro, verde escuro); na parte superior central, uma barra de progresso da qualidade dos frutos; superior direita, uma barra de progresso da saúde da planta; esquerdo central, um painel de controle entre o calor, a hidratação da árvore e ocorrência de pragas.

#### **2.2. Componentes de Tela**

- Barras de progresso de crescimento da árvore, amadurecimento dos frutos e de saúde da planta;
- Painel de Controle com 3 direções (água, calor e praga);
- Ícones de poderes; Árvore (tronco central, galhos, raízes, folhas, frutos), Pragas (ervas daninhas e larvas) e Cesta;
- Ferramentas de interação (mãozinha, nota musical e gota d'água).

### 2.3. Jogabilidade

o jogador acessa os poderes com a sua varinha conectada ao Arduino encostando o objeto na tela e poderá aplicar o poder selecionado em qualquer parte da tela, encostando novamente. Para isso, estarão presentes duas câmeras: uma frontal apontado para o jogador (altura e largura) e outra lateral (profundidade). Para representar os poderes selecionados no momento, a varinha terá um LED correspondente a cor do poder (branco: nada selecionado, azul, vermelho e verde).

### 2.4. Etapas

1. **Nascimento e Crescimento da Árvore:** é contemplado pelo processo desde o plantio da semente até a árvore crescer e brotar atingindo um tamanho razoável e que já detém a capacidade de criar galhos e plantas;
2. **Amadurecimento dos Frutos:** é composto pelo momento onde a planta já começa a dar seus primeiros frutos;
3. **Colhimento dos Frutos:** é o momento no qual os frutos cresceram e amadureceram o suficiente para serem recolhidos e depositados numa cesta.

### 2.5. Simulação

Iniciando o jogo, a árvore está com suas raízes dentro do solo e um pequeno e fino caule brotando. Nesse momento, 2 poderes aparecem na tela: azul (lado esquerdo) e vermelho (centro) (O poder da nota musical fica desabilitado). Passa um pequeno período de tempo e o painel de controle começa a tender para a cor azul, indicando a necessidade de regar planta. O jogador seleciona com a varinha a gota d'água (encostando na tela), o LED da varinha troca de cor (de branco para azul) e começa a aplicar a água na planta para que cresça verticalmente (encostando na tela). A câmera lateral detecta a seleção da varinha e a aplicação. A câmera frontal detecta a posição (coordenada x, y) que a varinha tocou à tela. Após alguns segundos (10 segundos) o poder de água termina e os ícones ficam travados por um tempo (dependendo do nível) para recarregar e novamente estar habilitado na tela. A barra de progresso do crescimento da árvore avança um pouco (ainda permanecendo na cor branca) e a barra de controle centraliza novamente, indicando equilíbrio das forças. Ao mesmo tempo, a barra de saúde também avança um pouco.

Ao mesmo tempo, aparecem ervas daninhas crescendo junto a planta. O painel de controle indica a necessidade de aplicar fogo nas mesmas, tendendo para a cor amarela. Nesse momento, o jogador seleciona a mãozinha e aplica o fogo para queimar as ervas. Após alguns segundos (10 segundos), o poder termina e novamente todos ficam travados por um tempo (dependendo do nível) desabilitado. Ao final, o painel de controle é centralizado novamente.

Deve se ter o cuidado para não usar excessivamente o poder do fogo para não queimar a planta também. Isso será indicado no painel de controle com o indicador tendendo para a cor vermelha.

Assim que a árvore estiver crescido consideravelmente (a definir), os poderes ficam dispostos na parte inferior da tela da seguinte maneira: azul (lado esquerdo), verde (direito) e vermelho (centro).

Além disso, a medida que o jogo segue começam a aparecer algumas notas musicais fluando na tela. O jogador seleciona a nota musical e aplica na planta para melhorar

o crescimento e florescimento da árvore (espessura) e o crescimento qualitativo dos frutos. Novamente, o poder dura questão de segundos (10 segundos) até que termina e, logo em seguida, todos ficam travados por algum tempo (dependendo do nível).

Ao mesmo tempo, o jogador deve regar a árvore (azul) para que continue crescendo (verticalmente) e combater as pragas que começam a aparecer nas folhas e flores (vermelho).

Nesse momento, começam a crescer os primeiros frutos e, para isso, é necessário continuar regando a planta, combatendo as pragas e tocando as notas musicais (crescimento dos frutos).

Finalmente, o jogador atinge a última parte do jogo que é constituída pela colheita dos frutos, já em sua condição ideal para tal ação.

Nessa última etapa o jogador deve ser rápido para colher os frutos sem que apodreçam (azul, lado direito) e, ao mesmo tempo, combater as pragas que crescem e progridem a cada instante (vermelho, lado esquerdo). Essas deverão ser realizadas atendendo ao tempo limite imposto sobre aquela etapa (fácil: 2 min; médio: 1 min; difícil: 30 segs).

Novos frutos vão nascendo ao mesmo tempo que outros vão amadurecendo até apodrecer. Então, é necessária a percepção de tudo o que ocorre naquele instante, para saber qual força selecionar e qual jogada escolher, visando a saúde da árvore e a qualidade dos frutos.

Ao fim dessa etapa, é atribuída uma pontuação ao jogador pela atuação completa e o jogo se dá por encerrado.

### **3. Algoritmos**

#### **3.1. Visão Computacional**

Compreendendo a parte de visão computacional, está sendo feito um levantamento bibliográfico referente ao processamento digital de imagens, reconhecimento de padrões em imagens, operações aritméticas e um estudo aprofundado sobre o modelo HSV (Hue, Saturation e Value ou Brightness).

#### **3.2. Espaço de cores**

##### **3.2.1. Espaço de cor HSV**

Foi possível executar alguns testes sobre os pixels capturados da imagem de uma câmera, descobrindo o grau da cor pura (Matiz) e as faixas representada por cada cor, a porcentagem de saturação da cor (Pureza) e a porcentagem de brilho (Valor), apenas convertendo do modelo RGB para o HSV.

Foram realizadas também algumas operações sobre os pixels como a redução da quantidade das possíveis cores aparentes pela imagem, dividindo a composição do RGB por um fator (constante). Foi possível perceber que quanto maior o fator, menor a quantidade de cores possíveis.

### **3.2.2. Escala de cinza**

Utilizamos conversão do modelo RGB em escala de cinza, aplicando os métodos de suavidade  $((\text{máximo}(R, G, B) + \text{mínimo}(R, G, B)) / 2)$ , média  $((R + G + B) / 3)$  e de luminosidade  $(0.21 \times R + 0.72 \times G + 0.07 \times B)$  ou também  $0.2989 \times R + 0.5870 \times G + 0.1140 \times B$ .

### **3.3. Reconhecimento de objetos**

### **3.4. Simulação da árvore**

Para que simulação tenha um aspecto e comportamento mais próximo de uma árvore, elaboramos uma solução que inclui a matéria estudada na disciplina de Estrutura de Dados e alguns algoritmos para simular o crescimento da árvore, dos galhos e dos frutos. Basicamente a árvore cresce de acordo com um valor de energia de crescimento fornecida a ela logo no início do jogo. O jogador tem a oportunidade de fornecer mais energia ao longo da partida, o que irá proporcionar uma árvore mais desenvolvida e com mais frutos para serem colhidos.

#### **3.4.1. Estrutura da árvore**

Adotamos a estrutura de dados de árvore ternária para construir a árvore do jogo. Essa estrutura é constituída, basicamente, em pontos de crescimento que se dividem no máximo em três partes cada um. Essas partes dão origem aos galhos que crescem até determinado tamanho, decidido aleatoriamente. As pontas de cada galho são novos pontos de crescimento, que se dividem de novo e dão origem a novos galhos. Esse processo se repete até que determinado limite seja atingido.

#### **3.4.2. Crescimento**

Para que os galhos cresçam controladamente mas mantendo certa diferenciação em tamanho e direção entre um e outro, estabelecemos um critério que chamamos de energia de crescimento e energia limite. A energia limite determina o quanto cada galho da árvore vai crescer e é estabelecido através de uma porcentagem, calculada a partir do total de energia de crescimento fornecida para o galho. O valor da energia limite pode ser alterado de acordo com a interação do usuário ao longo do jogo, o que pode proporcionar uma árvore mais mais ou menos desenvolvida.

Enquanto os galhos crescem eles podem dar origem a novos galhos. Isso é determinado após ser consumida uma determinada quantidade de energia de crescimento fornecida ao galho. Isso permite que a simulação de crescimento da árvore fique mais natural. A única exceção é o tronco da árvore, que tem o crescimento mais controlado.

#### **3.4.3. Pontos de crescimento de frutos e folhas**

Os pontos de crescimento de frutos e folhas são determinados de acordo com o mínimo de energia que o galho tem para crescer. Caso esse valor seja igual ou abaixo de determinado

critério, o algoritmo de simulação assume que não irão surgir novos galhos a partir do galho atual, ainda durante a fase de crescimento. Com isso determinado, o galho passa a produzir folhas e frutos ao invés de novas ramificação de galhos.

### **3.5. Primeiros experimentos**

Para o reconhecimento de uma cor emitida por um LED do Arduino, fizemos uma otimização física com um pedaço de papel que reduzisse o brilho da luz, porém ainda não é suficiente para o reconhecimento total.

Como experimentos necessários, temos o reconhecimento da luminosidade para verificar a proximidade da luz com a câmera e uma otimização afim de isolar a cor do LED, ignorando as outras cores similares e aparentes na câmera.

## **4. Bibliotecas**

### **4.1. OpenCV**

### **4.2. Allegro**

### **4.3. Arduino-Serial**

## **5. Equipamentos**

- Câmera de captura de vídeo
- Placa controladora ARduino Uno
- LED RGB
- Computador (desktop ou notebook)

## **6. Bibliografia**