

Ficha de trabalho para os alunos Equipa:.....

Tempo para o brainstorming:

O que sabe sobre as aplicações de assistente de voz incorporadas nos automóveis? Sabe se foram utilizadas tecnologias de IA para otimizar estas aplicações? *Procurem informações online com a vossa equipa e escrevam as vossas respostas abaixo.*

Consegue pensar em alguns casos do mundo real em que os comandos de voz para controlar e/ou navegar num automóvel autónomo possam ser úteis ou necessários? *Discuta com a sua equipa e documente as suas ideias abaixo.*

Consegue pensar em algumas vantagens e desvantagens da utilização de comandos de voz para controlar e/ou navegar num automóvel autónomo? Como é que estas podem afetar o processo de conceção? *Discuta com a sua equipa e documente as suas ideias abaixo.*

Tempo para criar a aplicação para controlar/navegar o nosso carro robótico através de comandos de voz

Enriquecer a nossa aplicação integrando serviços de IA

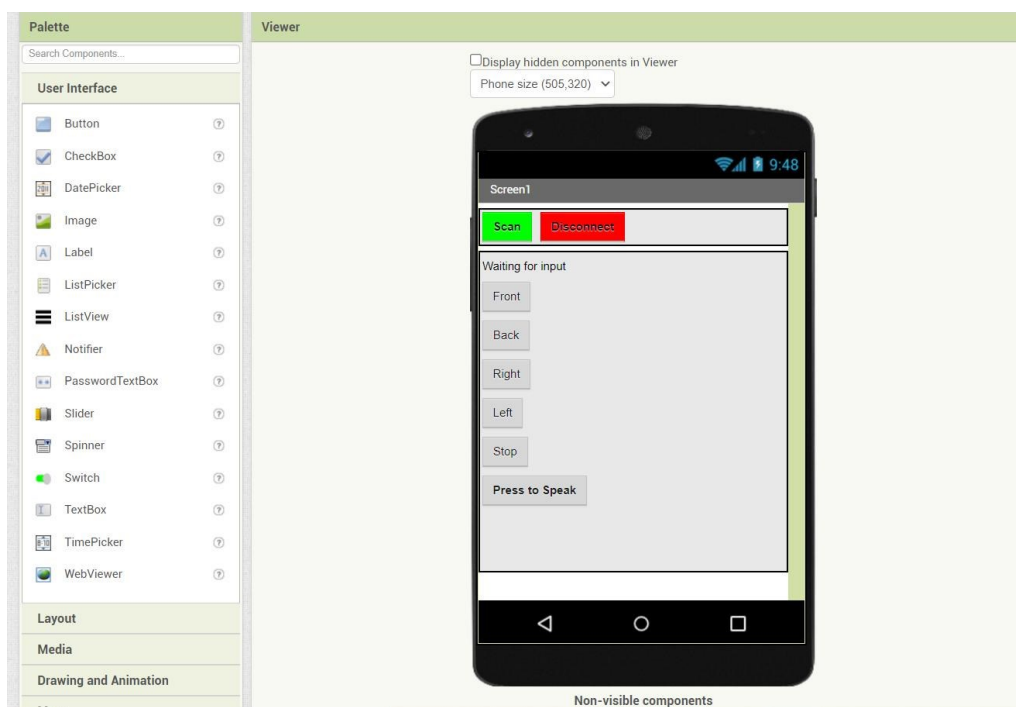
Para podermos navegar no nosso carro robótico utilizando a nossa voz, vamos adicionar um botão extra à nossa aplicação. Quando este botão for premido, o microfone do nosso dispositivo será acionado e, com a ajuda do serviço de IA Speech-to-Text da Google, os nossos comandos de voz serão convertidos em texto. De seguida, este texto será filtrado da seguinte forma: Se o comando de voz contiver a palavra "avançar", "recuar", "esquerda", "direita" ou "parar", a nossa aplicação transmitirá (através do Bluetooth) a mensagem correspondente ao nosso carro robótico e este actuará em conformidade.

Tenta preencher a tabela seguinte com base na descrição acima referida. Que mensagem será enviada para o nosso carro robótico e que movimento será consequentemente executado, quando cada um dos comandos de voz for dado? Dica: Podes também dar uma vista de olhos ao guião Makecode.

<i>Se o comando de voz contiver a palavra:</i>	<i>De seguida, envia ao carro robótico a mensagem:</i>	<i>Depois, o carro robótico vai:</i>
"avançar"		
"para trás"		
"esquerda"		
"direito"		
"parar"		

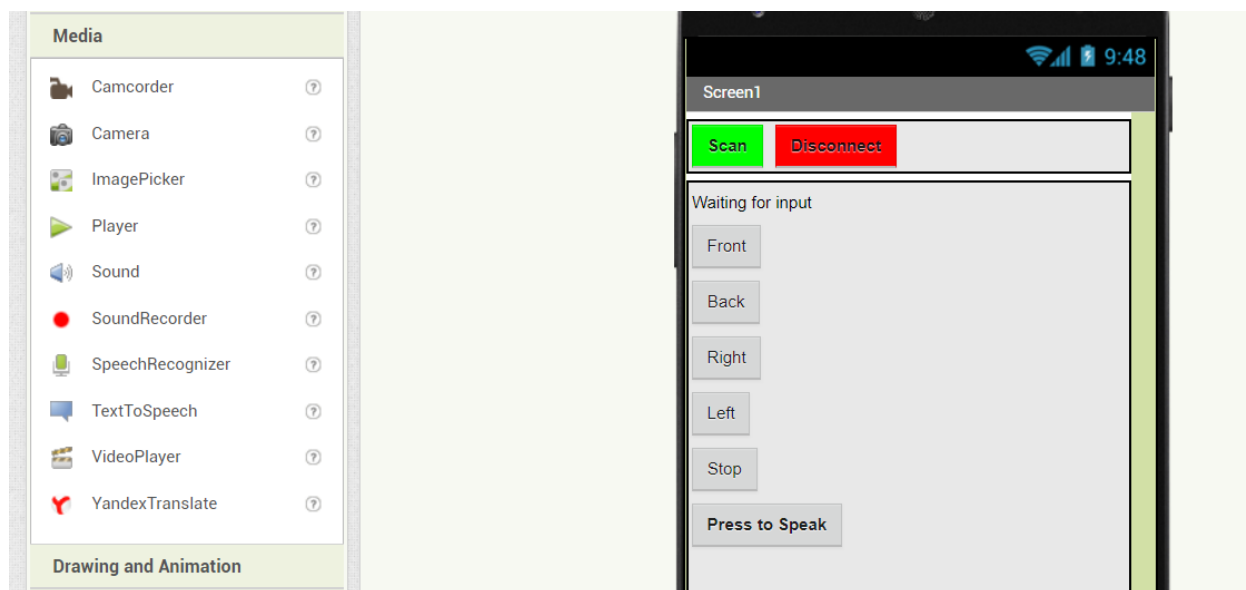
1) Adicionar o novo botão

Vá ao menu Designer e arraste um botão do separador Interface de utilizador para a área de design da interface da aplicação. Em seguida, altere o texto do botão e o nome do botão para algo com significado, como "Prima para falar" e "falar", respetivamente. Após este passo, a pré-visualização da sua aplicação terá o aspeto da imagem seguinte.



2) Adicionar o serviço de IA de voz para texto

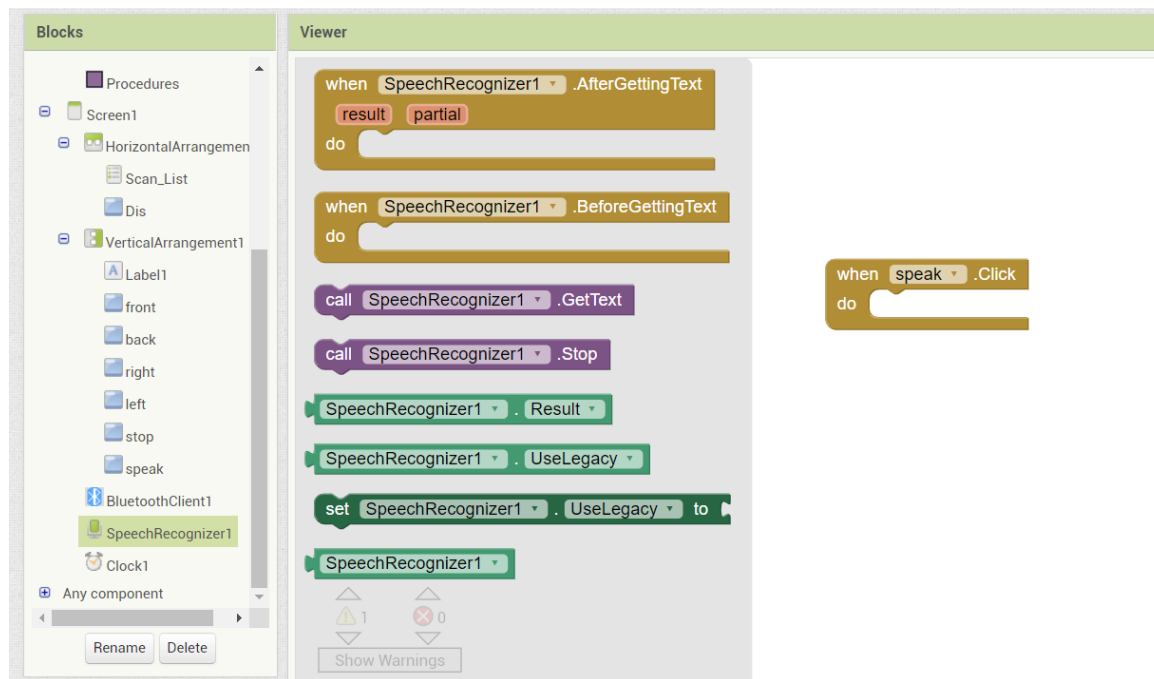
Agora, é altura de adicionar um serviço que permitirá à nossa aplicação reconhecer os nossos comandos de voz. Clique no separador Media. Qual dos serviços contidos acha que precisamos de utilizar? *Escreva a sua resposta abaixo.*



Em seguida, arraste este serviço para o ecrã da aplicação.

3) Criar o guião para o botão "Premir para falar"

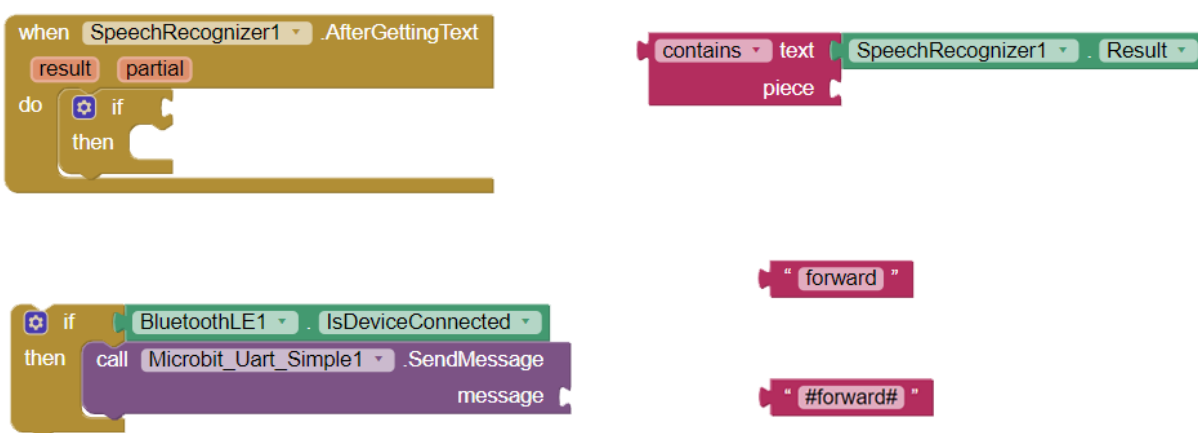
Como já foi referido, quando o botão "falar" é premido, os nossos comandos de voz são convertidos em texto. Dos comandos contidos na imagem seguinte, *que comando do SpeechRecognizer devemos colocar dentro do manipulador do evento "when_speak_click", para permitir que a nossa aplicação **obtenha** um pedaço de **texto**?* Discuta com a sua equipa e escreva as suas ideias em baixo.



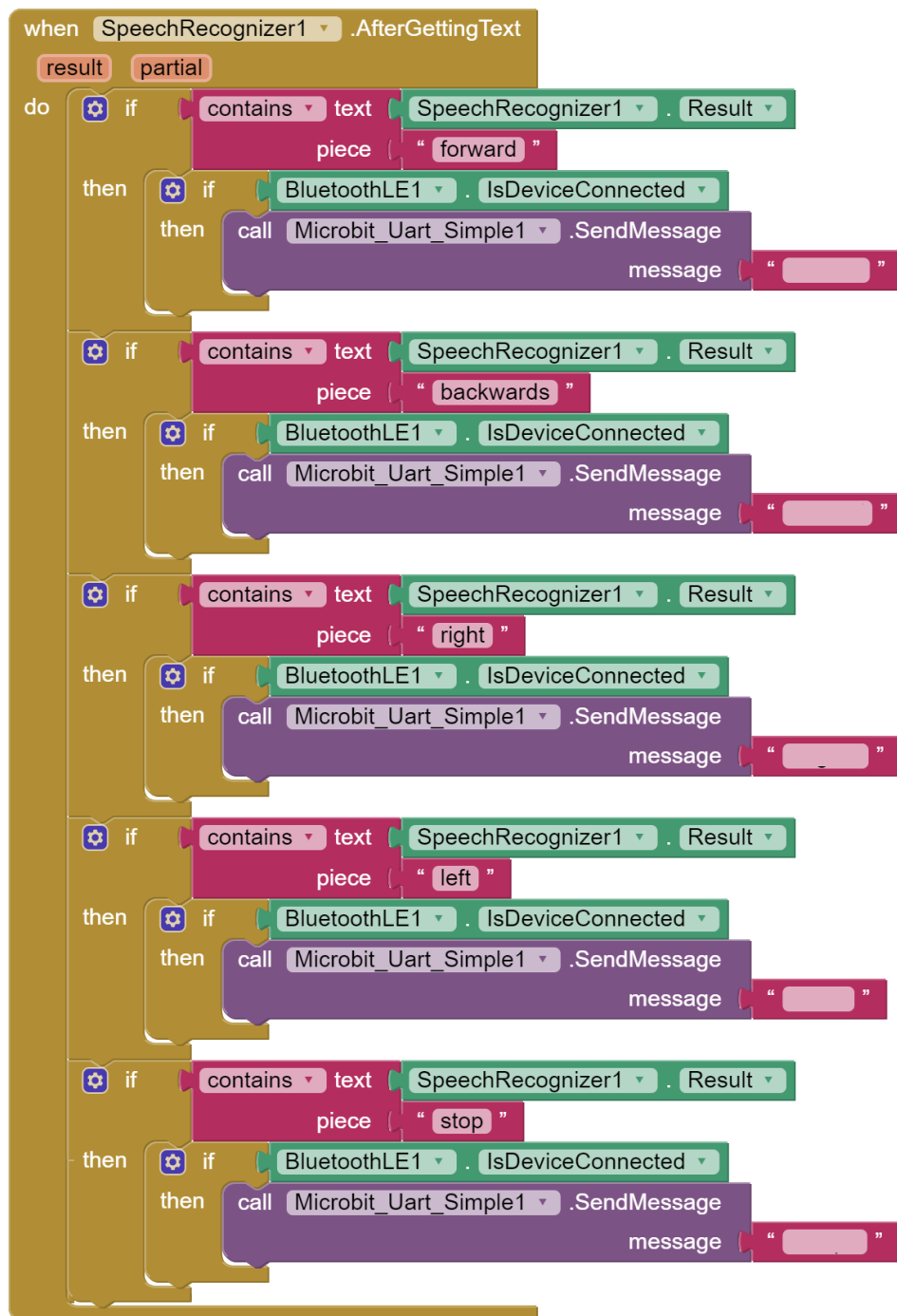
4) Filtragem do texto recebido

Depois de programar o que vai acontecer quando se clica no botão Falar (nomeadamente receber os comandos de voz como texto)? é altura de programar a nossa aplicação para reconhecer partes específicas do texto recebido, de modo a permitir que o nosso carro robótico execute os movimentos correspondentes.

O guião seguinte é semi-estruturado. Tente colocar os comandos na ordem correta para programar a aplicação para fazer avançar o carro robótico (enviando através do Bluetooth a mensagem correspondente), **se** a palavra "avançar" estiver contida no texto reconhecido.



Agora, preencha o script anterior com todos os comandos necessários para permitir que a sua aplicação mova o carro robótico em conformidade, se o comando de voz correspondente for reconhecido (ou seja, para a frente, para trás, para a direita, para a esquerda e parar). Para isso, tente preencher o seguinte guião semi-estruturado.



5) Testar a aplicação

Agora, está na altura de testar a sua aplicação. Vá ao menu Construir e gere a sua aplicação. Depois de instalar o ficheiro .apk no seu dispositivo inteligente, abra-o e teste se o seu carro robótico corresponde aos comandos de voz programados, quando clica no botão Press to speak.

Tenta utilizar diferentes comandos de voz e verifica se o teu carro robótico responde a eles ou não. Anote as suas observações na tabela seguinte (*por exemplo, o comando de voz "Avançar" é bem sucedido, o comando de voz "ir para a frente" falhou, etc.*).

Comando de voz	Sucesso	Falha

Vamos tornar a nossa aplicação menos "aborrecida" adicionando um sensor de relógio

Até agora, implementámos o seguinte cenário: Sempre que quisermos dar um comando de voz, temos de premir o botão "Premir para falar".

Esta solução é funcional, mas pode não ser muito conveniente, especialmente se projetarmos este cenário para a vida real e para as condições reais de condução (ou seja, conduzir um carro numa cidade).

Para criar uma solução ótima, vamos modificar a nossa aplicação da seguinte forma:

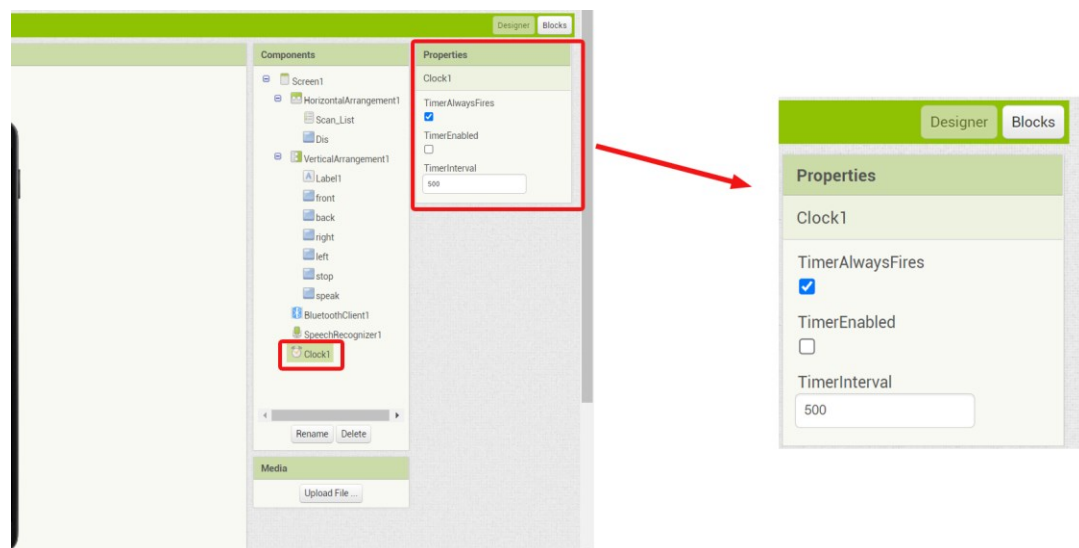
O utilizador terá de premir o botão "Premir para falar" apenas uma vez, e só no início do processo, para inicializar o reconhecimento de voz/fala. Quando o utilizador o fizer, a aplicação começará a procurar comandos de voz válidos. Após um intervalo de tempo específico (por exemplo, 2 segundos), a aplicação procura automaticamente um novo comando de voz.

Vamos ver como o podemos fazer:

1) Adicionar um sensor de relógio

Vá novamente ao menu Designer e arraste um sensor "Relógio" do separador Sensores para a área de desenho da aplicação.

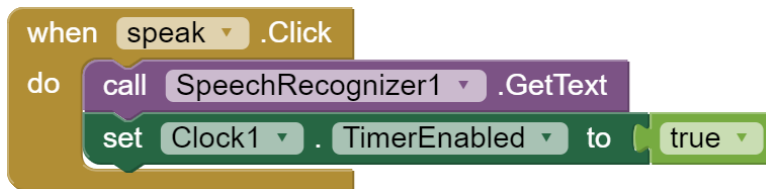
Nas Propriedades do Relógio, selecione a caixa TimeAlwaysFires e defina o TimeInterval para 2000 (milissegundos).



2) Modificar a aplicação

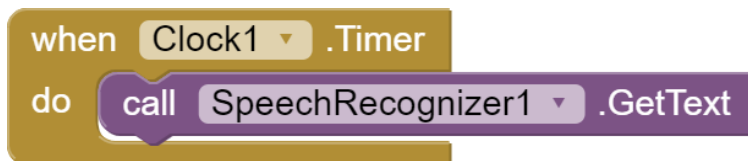
Agora, vá ao menu Blocks (Blocos) para modificar algumas partes do guião existente e adicionar mais alguns blocos de comandos.

Dentro do manipulador "when speak.Click", adicione o comando set Clock1 TimerEnabled e coloque um comando lógico true no lado direito do bloco. Isto permitirá que o nosso relógio seja ativado quando o botão "Prima para falar" for premido.



Agora que disparámos/ignorámos o temporizador pela primeira vez (depois de premir o botão "Falar"), o passo seguinte é reativar o Reconhecimento de voz sempre que a contagem decrescente termina. A duração do temporizador é renovada automaticamente através deste processo.

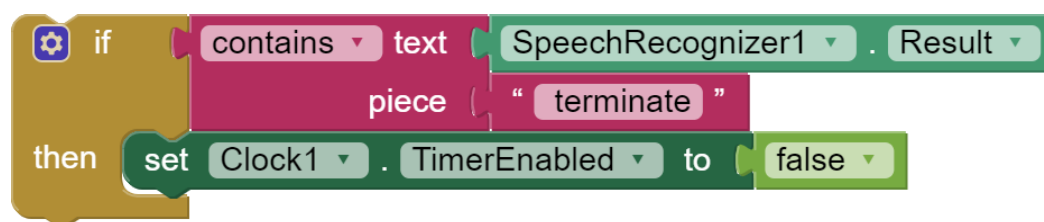
Para isso, precisamos de dar instruções à nossa aplicação para chamar o Reconhecedor de voz para obter o texto do que ouve, sempre que o Temporizador do Relógio dispara. Isto pode ser feito criando o seguinte bloco de comandos.



Vamos testar a aplicação. *Mais uma vez, aceda ao menu Build e gere a nova versão da aplicação.*

Funciona corretamente ou tem algum problema de funcionamento? Discuta com a sua equipa e escreva a sua resposta abaixo.

Acrescente o seguinte bloco de comandos ao texto "When SpeechRecognizer1.AfterGetting" e teste novamente a aplicação.



A avaria continua a existir?

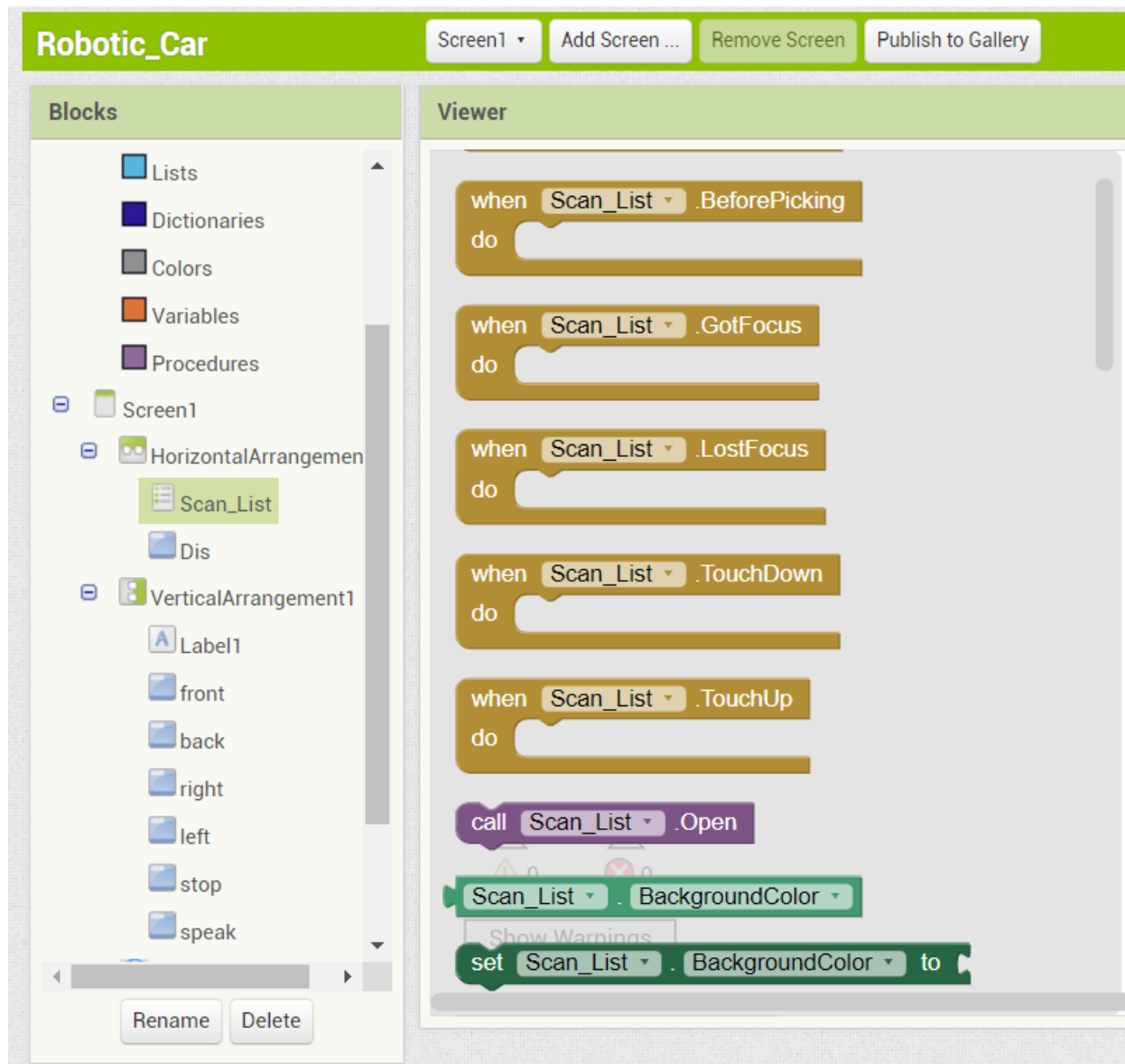
Faça mais experiências com a aplicação para otimizar a solução atual. Por exemplo, tente alterar a duração do intervalo de tempo e escreva as suas observações na tabela abaixo. Tenha em atenção que o tempo é contado em milissegundos (1 segundo = 1000 ms).

Temporizador Intervalo	Observação

Zona de dicas

Encontrar os comandos:

Para encontrar alguns dos comandos necessários, clique no item correspondente (ou seja, Scan_List no exemplo seguinte) e procure no menu flutuante que aparece.



Alguns comandos contêm mais do que uma opção. A imagem seguinte apresenta um desses comandos.

