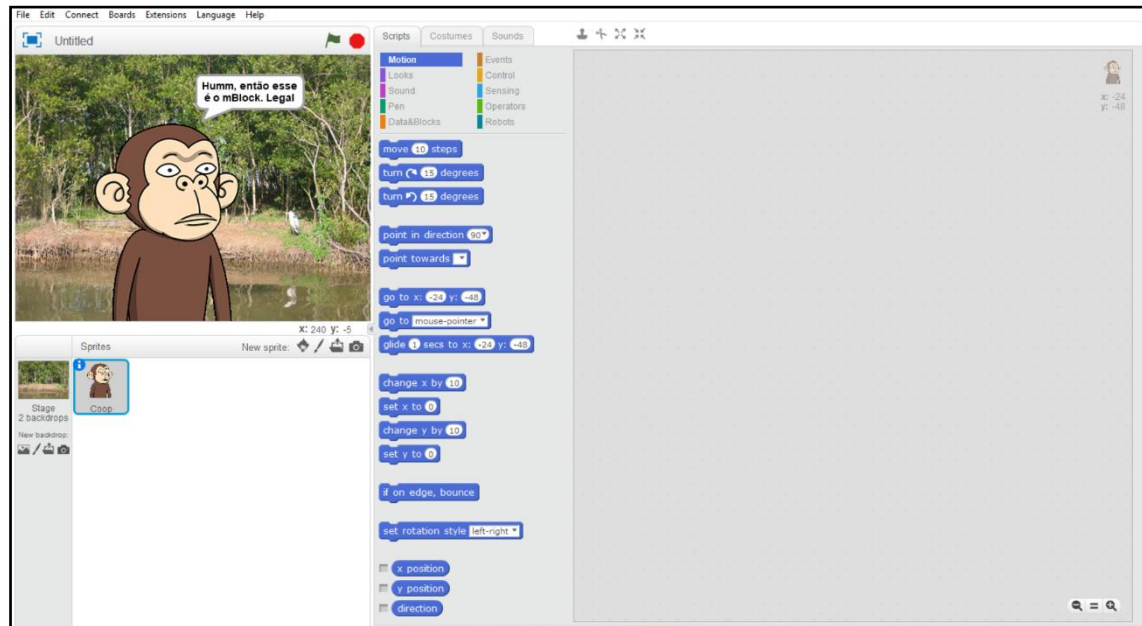
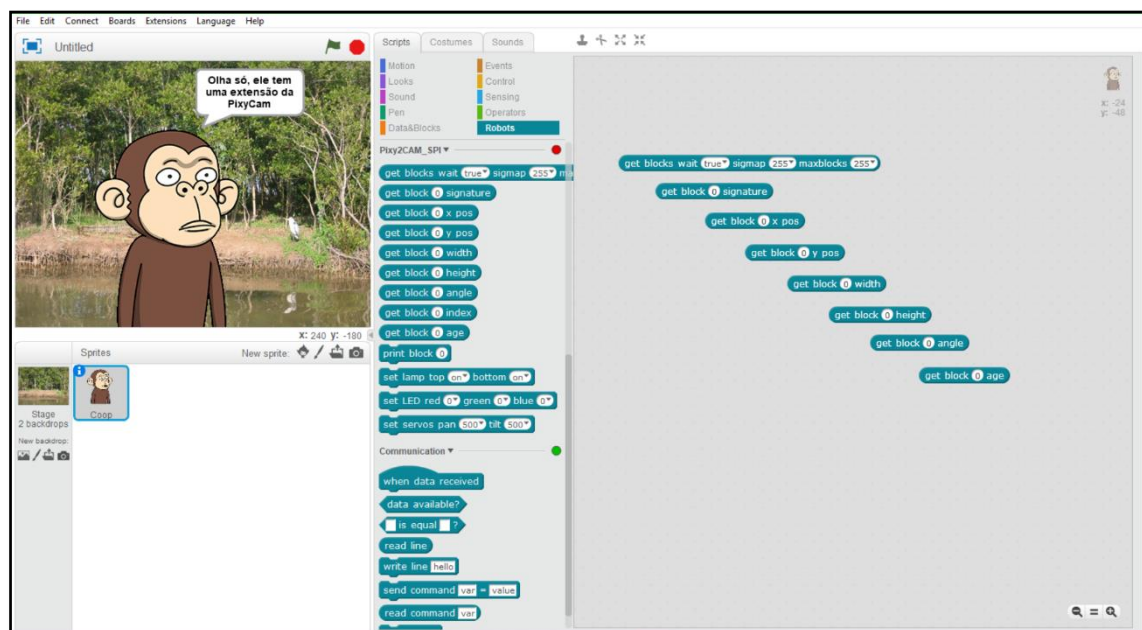


mBlock é uma ferramenta baseada em Scratch que utiliza programação em blocos para o desenvolvimento de códigos para jogos e gerenciamento de dispositivos. O mBlock possui suporte para diversos dispositivos diferentes e conta com uma vasta galeria de extensões que podem ser utilizadas para aumentar ainda mais o número de aparelhos que podem ser programados através da ferramenta.

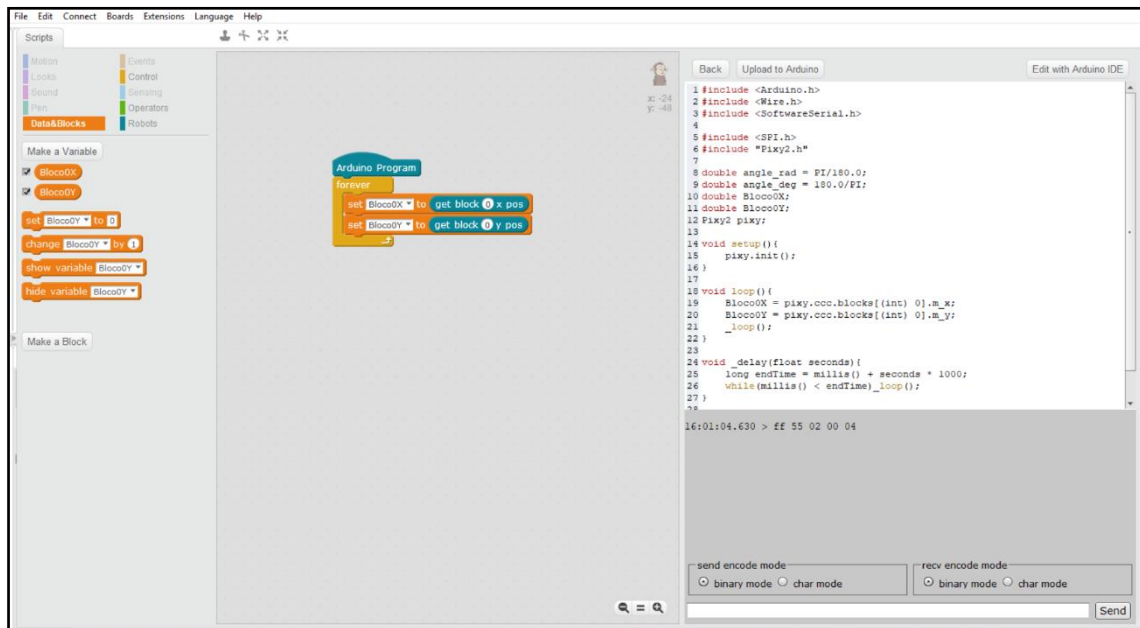


Sendo compatível com placas arduinos, controles, aparelhos bluetooth e diversos outros dispositivos, o mBlock se torna muito útil quando se quer programar, sem usar linhas de código, apesar do programa permitir essa possibilidade também, onde os códigos podem ser feitos através da linguagem Python. O mBlock fornece uma extensão para PixyCam2, criada por Grant Doher. Porém, essa extensão é utilizada para a programação na placa e não trabalha de maneira independente, ainda necessitando estar conectada em outro dispositivo.

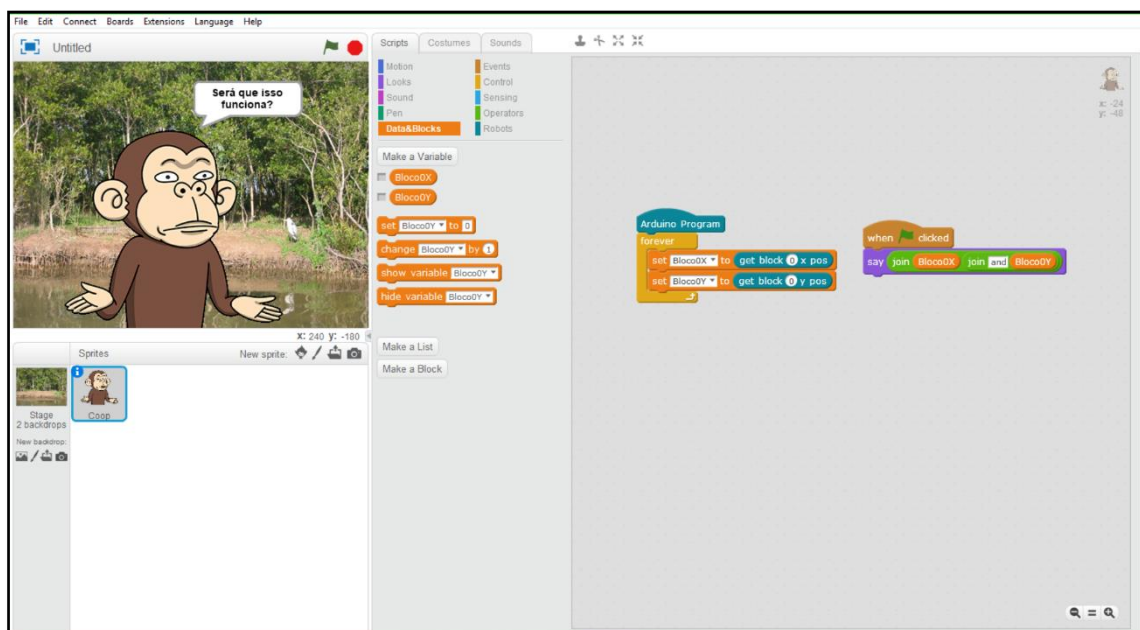


Como dito anteriormente, o mBlock possui suporte para diversos controladores Arduino, onde, similarmemente com outras ferramentas ao estilo Scratch, permite a criação do código da placa,

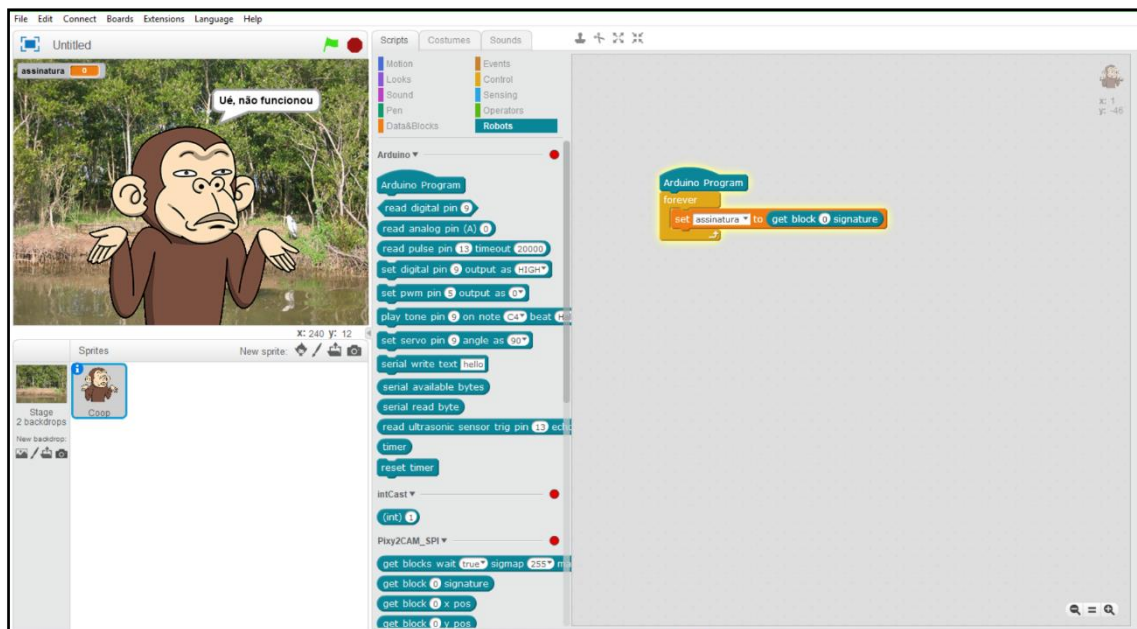
sem utilizar linhas de código. A ferramenta possui um modo voltado ao desenvolvimento do Arduino onde além dos blocos, é exibido também o código que está sendo gerado através dos blocos. É possível também fazer o *upload* do código para placa direto da ferramenta, sem a necessidade de abrir a IDE do Arduino.



Ao que aparenta, a programação em blocos utilizada na criação do código para Arduino difere e não interage com a desenvolvida sem a interação com a placa. É como se existissem duas realidades, a que interage com o jogo normal (criada com os blocos padrões do mBlock/Scratch) e a que interagem com o Arduino (que utiliza blocos específicos para placa). Entretanto, algumas coisas são visíveis dos dois lados, como variáveis, que podem ser utilizadas por ambos os modos, apesar de até o dado momento não ter sido visto como passar os dados do Arduino utilizando a variável presente nos dois modos.



Queremos utilizar dos dados dos objetos rastreados pela PixyCam. Para isso, precisamos elaborar o código em Arduino (podendo ser utilizada a extensão do Arduino e da PixyCam no mBlock) para que este receba os dados da câmera e então passe para o mBlock. Até o momento apenas a passagem por variável foi tentada, onde criamos uma variável (independente de em qual dos modos ela foi criada ela estará disponível para ambos) e então, através do Arduino mantemos um fluxo constante onde a variável em questão sempre é preenchida com os dados da câmera, e então utilizamos a variável criada no desenvolvimento do jogo em si.



No código acima, programamos a placa para definir a assinatura do objeto detectado em uma variável criada pelo programa. A leitura da assinatura é feita e a variável está sendo definida, podendo ser comprovado isso através do monitor serial do Arduino. Entretanto, os valores da variável do jogo (teoricamente a mesma variável do Arduino) não sofreu alterações.

Ao que tudo indica parece é que a transferência desses valores não é tão simples assim, e na verdade ambos os modos trabalham de maneira paralela, sem interagir um com o outro. Por outro lado, existem diversas extensões que podem ser utilizadas junto ao mBlock, e talvez com o uso de uma delas podemos realizar essa passagem de dados.

O mBlock, assim como o Scratch, são boas ferramentas para a criação de jogos simples, entretanto, apesar de suas inúmeras extensões e variedade de aplicações, a ferramenta é bem limitada e peca na criação de jogos mais complexos, já que este não é tecnicamente seu foco. Foi analisado então a utilização de outras ferramentas para um possível desenvolvimento futuro do Coopera. Entre elas, foi analisada a possibilidade da utilização do OpenCV.

O OpenCV é uma biblioteca multi-plataforma livre voltada para o desenvolvimento de aplicações que utilizem de visão computacional. Ela possui suporte para diversas linguagens e ferramentas e seu foco é através de sua tecnologia facilitar o desenvolvimento de aplicações que utilizem de visão computacional como reconhecimento de padrões, reconstrução 3D, rastreamento de objetos, edição de imagens, dentre outros sem a necessidade de recursos especiais.

Dentre as diversas funcionalidades da biblioteca esta possui recursos para o desenvolvimento de aplicações que façam o rastreamento de objetos. Existem diversos tutoriais no YouTube (e até mesmo dentro do próprio site do OpenCV) que demonstram como desenvolver tais aplicações. Também é possível o rastreamento de cores para diferentes formas, também possuindo tutoriais de como fazer a mesma.

Além do reconhecimento e rastreamento de objetos através do OpenCV (muito úteis no desenvolvimento do Coopera), também é possível realizar o reconhecimento de caracteres através de um software chamado Tesseract e integrar essa tecnologia no OpenCV (o reconhecimento de caracteres seria útil para o desenvolvimento de algumas fases no Coopera).

A biblioteca do OpenCV fornece diversos recursos que possibilitam o desenvolvimento de aplicações que utilizem de computação visual, porém, sua utilização no desenvolvimento é integralmente feita em código, o que a torna, tecnicamente, mais complicada de ser utilizada.

Diversos trabalhos dentro do laboratório do LARVA foram desenvolvidos utilizando o OpenCV juntamente com a linguagem Java. Alguns destes jogos foram Move4Math, Moviletrando e Dance2Rehab. Todos disponíveis no site do LARVA gratuitamente.