

Василиско

Команда

Орлов Артур
Конструктор-актёр
Личные достижения:
Робофинист Мини-сумо 2 место
PPO (региональный этап) 3 место.



Шашек Елена
Программист, инженер-конструктор.
Личные достижения:
3 место на Robocup 2022
1 место на RRO 2022
участие на РоботеПочтальоне 2022 и РобоStep 2022.

Курганова Вероника
Монтажёр- программист.
Личные достижения:
3 место во Всероссийском конкурсе по подводной робототехники.
Участвовал в конкурсе научно технического творчества ИМАКЕ.

Ивлев Артём
Программист инженер-конструктор.
Личные достижения:
Получил серебрянную медаль на международных соревнованиях детских изобретений, победитель PPO 2022, я занял 3 место на Robocup 2022, победитель финала проекта ИМАКЕ 2023г.

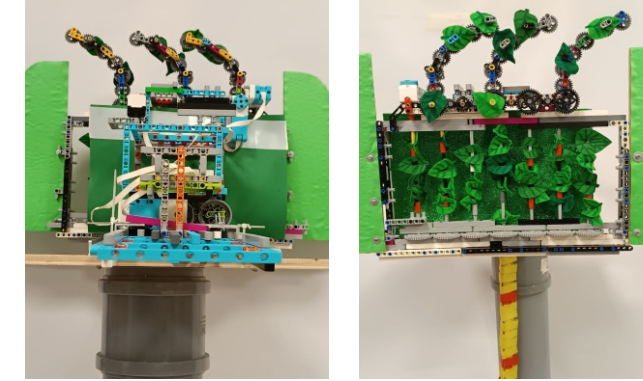
Кудрявцев Савелий
Инженер – программист.
Личные достижения:
Получил номинацию за проект «Светильник-сигнализация» в конкурсе технического творчества «Юные изобретатели спасателям» Участвовал в конкурсе научно технического творчества ИМАКЕ.

Ключевые особенности

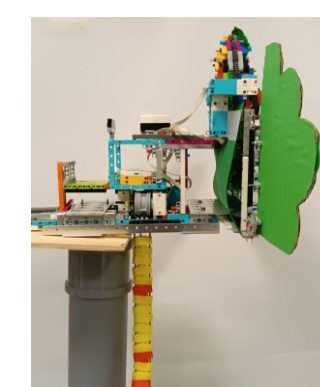
- 1) Использование пропорционального регулятора, для удержания змеи на определённом расстоянии. Ультразвуковой датчик расстояния замеряет текущее расстояние до руки. Далее мы вычисляем ошибку: разницу необходимого расстояния и текущего. Значение ошибки умножается на коэффициент и подставляется в итоговую скорость моторов.
- 2) Для видеораспознавания мы использовали камеру пикси кам, но перед этим мы её обучили в специальной программе видеть определённые объекты. После этого мы отфильтровали ненужные шумы и с помощью п-регулятора сделали так, чтобы она ехала за шариком.
- 3) Эффект бесконечного зеркала. В фильме был омут памяти в который можно было заглянуть и увидеть воспоминание. Для имитации такого эффекта мы использовали зеркало и полупрозрачную зеркальную плёнку. Свет многократно отражается и возникает эффект глубины.
- 4) Механизм змеи был взят с сайта открытого доступа, адаптирован и напечатан. В роботе выполнен шаровой механизм и за счёт системы подтяжки тросов и синхронности моторов мы добиваемся того, что змея может извиваться в разные стороны.

Роботы

Палочки
Были выполнены из двух кабель каналов, к каждой из них находятся шесть проводов, два из них напрямую идут от девяти вольтовой кроны к двум светодиодным лентам ещё четыре соединяют крону, кнопки и светодиодные ленты. Особенность наших палочек- отсутствие регулятора.



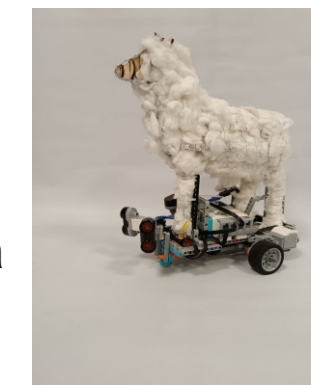
3D змея
Корпус сделан при помощи 3D печати. Три мотора обеспечивают натяг тросов и змея изгибается в различных плоскостях. К конструкции змеи мы присоединили механизмы кроны, которые посредством работы моторов и шестерёнок двигают прототипы веток и вращают листья. Всё запрограммировано на базе программы Spika.



Лего Змея.
механизм робота сделан на базе конструктора LEGO MINDSTORMS Education EV3 и Spika. Передний мотор выполняют функцию поворота и езды вперёд и задний мотор выполняет ту же функцию. Наша змея ездит свободно по сцене. В ней используется П-регулятор, который берёт показания с ультра-звукового датчика расстояния, вычисляет ошибку и умножает её на коэффициент.

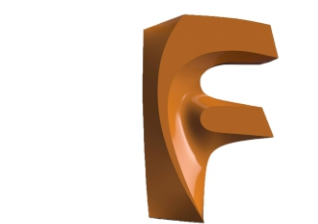
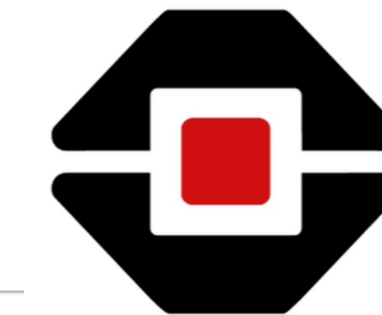


Мобильная платформа.
Патронус находится на мобильной платформе, которая запускается с помощью ИК датчика, после этого с помощью камеры Pixu Cam следует за красным шариком. Сначала данные фильтруются на наличие объекта с помощью пропорционального регулятора патронус выбирает направление движения за красным шариком.

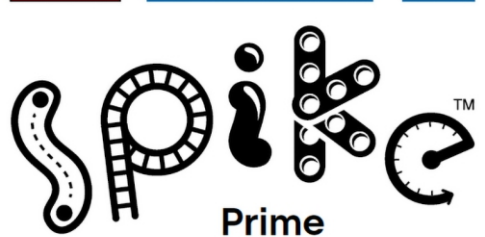


Омут памяти
Выполнен из зеркального пластика и полупрозрачной зеркальной плёнки. Адресные светодиодные ленты переключаются с помощью микроконтроллера Arduino и библиотеки NeoPixel

Программное обеспечение



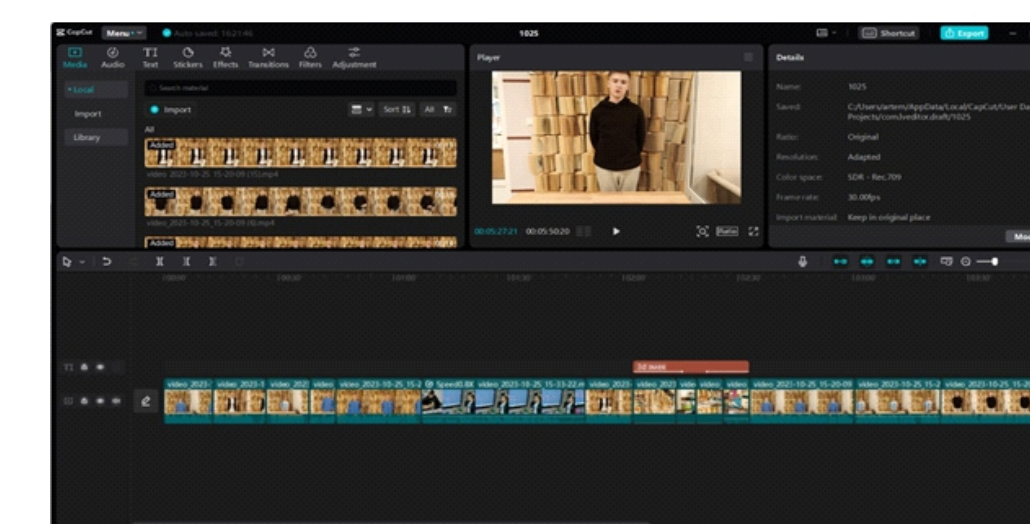
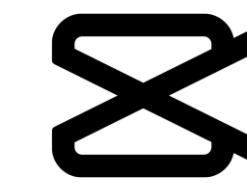
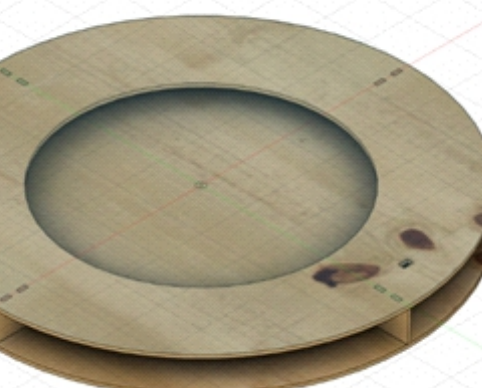
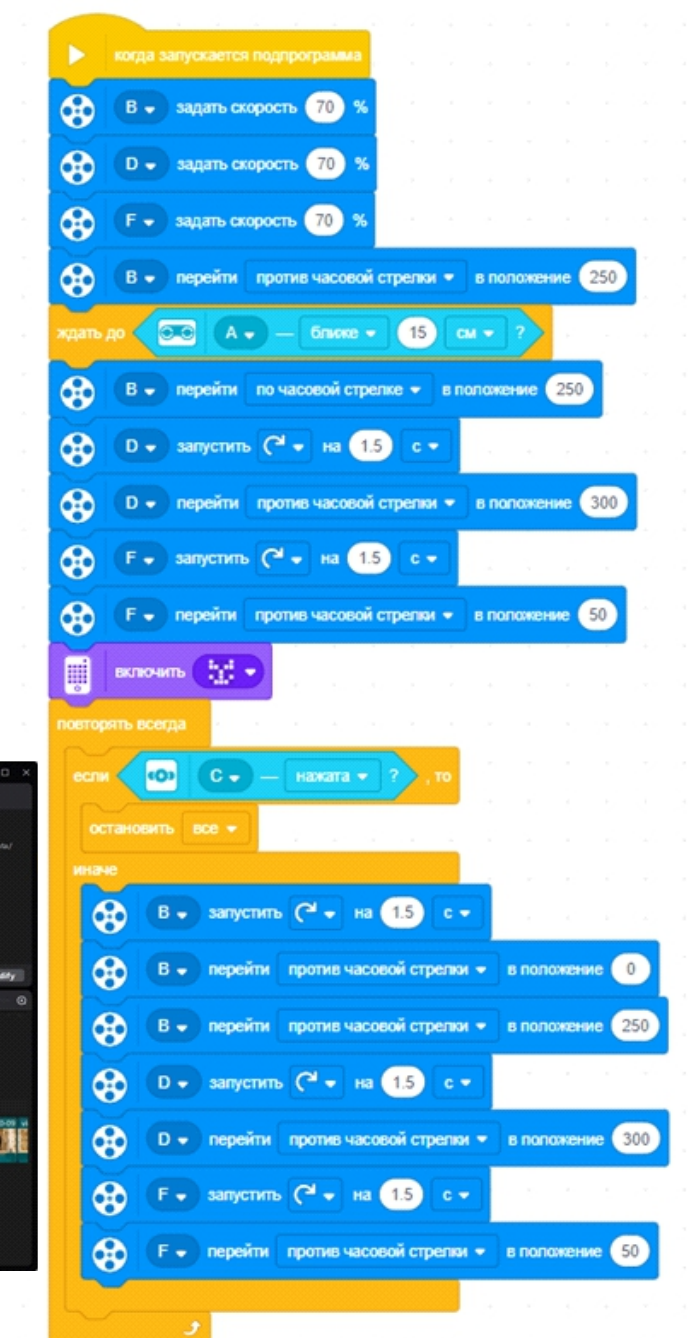
AUTODESK FUSION 360



```

1 #include <Adafruit_NeoPixel.h>
2 #define PIN 6
3 #define count_led 60 // количество светодиодов
4 Adafruit_NeoPixel strip = Adafruit_NeoPixel(count_led, PIN,
5 #define NEO_GRB // RGB
6 #define NEO_RGB // RGB
7 #define NEO_RGBW // RGBW
8 void setup()
9 {
10   strip.begin();
11   strip.show(); // Initialize all pixels to 'off'
12   Serial.begin(9600);
13 }
14 // прокрутка цикла выполняется снова и снова
15 void loop()
16 {
17   int sensorVal = analogRead(A0); //
18   //Serial.println("qqq");
19   digitalWrite(LED_BUILTIN, HIGH);
20   while(analogRead(A0)>500)
21   {
22     for ( int i=0; i< count_led; i++)
23     {
24       //Serial.println("vvv");
25       digitalWrite(i, HIGH); // случайная добавка цвета
26       blue_dob = random(0, 255);
27       strip.setPixelColor(i, 255+blue_dob, 0, blue_dob );
28       strip.show(); // отобразить на ленте
29       delay(10); // задержка
30     }
31   }

```



Решения и проблемы

- После печати деталей 3д змеи, они оказались слишком большими и тяжёлыми. Чтобы решить эту проблему, мы уменьшили макет в 2.5 раза.
- После печати уменьшенных моделей подложка на 3д принтере постоянно слетала. Мы увеличили температуру печати, и проблема пропала.
- В омуте памяти при наклеивании зеркальной плёнки образовывались воздушные карманы. Мы наклеили эту плёнку на орг.стекло и плёнка больше не провисала.
- Самая большая проблема, с которой нам пришлось столкнуться, это то, что ик-датчик, который был установлен на патронусе улавливал ИК-сигналы от камер видеонаблюдения и проекторов. Решение мы приняли быстро, сначала мы просто вставили в программу участие ик-пультов, что позволило немного контролировать робота. Позднее в промежутке между этапами соревнования мы заменили ик на видеораспознавание.

Взаимодействие

