Насоки за опростена версия на учебната програма

<u>1-2 Въведение</u>

Бърз поглед към основите и историята на роботиката. И как създаваме UML диаграма. Няма да има оценки по време на събитието.

<u> 3-4, 5-6 Изходни устройства - двигатели</u>

Фокус върху теоретичната част, а не упражненията. С тях ще се занимаем на по-късен етап от подготовката.

<u> 7-8 Изходни устройства – обратна връзка</u>

Съществуват лесни начини да разберем как работят различните устройства за обратна връзка (зрителни и слухови). Те ще бъдат включени в различни предизвикателства по време на събитието.

<u>9 – 14 Сензори</u>

Фокусираме се върху теоретичната част на показаните действия, за да разберем напълно начините, по които работят различните сензори.

<u> 15 – 16 Тимбилинг умения в 21 век</u>

Няма да ги тестваме, но е много важно да помогнем на екипите да се сближат и да подобрят основните ценности в екипа по време на процеса.

17-20 Задвижване

Фокус върху упражненията. Искаме теоретичните познания в глави 3-6 да се приложат в заниманията.

21-22 Сценарий "Критично мислене на летището"

Комбинираме всичко, което сме научили до сега като изпълняваме предизвикателствата.

23 – 28 Съобщаване на данни, вписване и обработка

Тези уроци не са включени в опростената версия, тъй като са по-сложни спрямо това, което се очаква от събитието.

29-30 Брейнсторминг изследване на креативността

Проектен анализ и проучване няма да се оценяват на събитието, така че тези уроци не са включени в опростената версия.

31-32 Презентация за решаване на проблеми от реалния свят

Презентационните умения са много важни, тъй като всеки екип трябва да направи презентация за своята държава.

<u>33-36 Програмиране</u>

Фокус върху съвременното програмиране с въвеждане на променливи и константи. Това дава възможност екипите да ползват различни начини за програмиране, за да изпълнят предизвикателството.

37-40 Създаване на състезателен робот

Въпреки че това не е задача, която ще бъде оценявана на събитието, сумо предизвикателството е много забавно занимание, което предизвиква екипите да построят силен и стабилен робот, за да се състезават с други екипи. По време на този процес екипите могат да комбинират и приложат всички придобити знания.



Изходящи устройства – двигатели

Урок 3.-4.



Необходими материали:

• Сглобена база за робот LEGO EV3 без сензори

EV3 Голям двигател

Има множество начини, с които да програмирате големият двигател да се задвижва:



С този блок можете да движите само един двигател едновременно. Порт номер **D** показва кой двигател ще се задвижи. Номер **50** показва мощността на двигателя и номер **1** продължителността на ротациите. Това означава, че двигателят извършва 1 ротация с мощност 50.

Фигура 3. Блок за голям двигател



Блокът за управление работи на принципа, че двигател В е "главен", а двигател С "второстепенен" и съгласно

С този блок можете да движите робот с два двигателя.

Фигура 4. Блок управление

стойността на управление, роботът се движи в различни посоки. "Второстепенният" се опитва да следва ротациите на "главния".









	U		(
Таблица 1. Различни	і стойности на	<u> управление с двигател</u>	с мошност 50 (г	и базов побот FV3
	enno anno enna ma	ynpachae e ooacanten	0 1110 00 10 0111 00 10	x 00000 p0000111 210

Стойност на	Посока на робота	Стойност на	Посока на робота
управление		управление	
0	Напред		
-25	Роботът за завърта наляво с	25	Роботът за завърта надясно с
	~45 градуса		~45 градуса
-50	Роботът за завърта наляво с	50	Роботът за завърта надясно с
	~90 градуса		~90 градуса
-100	Роботът за завърта наляво с	100	Роботът за завърта надясно с
	~180 градуса		~180 градуса



С този блок можете да зададете различна мощност и на двата двигателя по едно и също време. Когато стойностите на мощността са еднакво положителни и са над 0, то

Фигура 5. Блок за преместване

роботът се движи направо, когато са под 0, роботът се движи на заден ход направо, а с други комбинации роботът се

завърта съгласно зададеното. Опитайте сами!

Има много начини, по които да завъртите робота.



С този блок се движи само двигателят в част В. Това означава, че роботът се завърта напречно <u>около оста на двигател С</u> и радиусът на завъртане **r** е разстоянието между тези две колела. (Виж фигура 6)



Фигура 6. Завъртане на робот с един двигател (Автор: Размус Китс)



















С тази настройка роботът се завърта на място. Със стойност на задвижване 100 се завърта надясно, а с -100 наляво. На място, означава, че роботът се завърта около <u>оста си,</u> а не около оста на кормилото. Радиусът на завъртане **г** е

разстоянието между кормилото и оста на робота. (Виж фигура 7).



Фигура 7. Робот, завъртащ се около оста си. (Автор: Размус Китс)



С тази настройка роботът се завърта с двата двигателя и радиусът на завъртане и посоките зависи от стойността на Задвижване. Колкото по-малка е стойността, толкова поголям е радиусът. Положителните стойности завъртат

робота надясно, а отрицателните наляво. Стойност на задвижване 0 задвижва робота направо.



Блок за цикъла

Може да промените състоянието на Цикъла като натиснете символът за безкрайност(∞).

Може да избирате измежду следните:

Неограничено е зададената стойност по подразбиране за блока на Цикъла, което означава, че цикълът ще спре само когато Потребителят спре програмата (или батерията на робота се изтощи)

Времеви индикатор означава, че цикълът ще се повтори докато не измине определеното време. 5 секунди е зададената стойност по подразбиране. Можете също така да ползвате стойност с десетична запетая за по-точно определяне на времето.











inerciadigital



Брой показва колко пъти ще се повтори цикъла. Например, брой 5 означава, че цикълът ще се повтори 5 пъти.

Логическият маркер означава, че цикълът ще се повтаря докато стойността му не е Вярна (по предварително задаване) или Грешна, когато Потребителят я промени. Това означава, че вътре или извън цикъла трябва да се извърши задаване на Логически маркер и резултатът ще се приложи за условията на Цикъла.

Сензори (Започва с брик бутони и завършва със Съобщения). Можете да ползвате стойността от сензорите, за да прекратите Цикъла. Всички видове сензори могат да се ползват за тази цел. Имайте предвид, че стойността на сензорите трябва да е абсолютно същата както условието, когато програмата достигне до края на Цикъла, в противен случай условието на Цикъла няма да се изпълни и той няма да спре.



Блок за изчакване

Можете да го ползвате, за да изчакате определено условие да бъде вярно. Можете да изберете между тези условия:

Времевият индикатор е във фабрично зададена настройка. Това означава, че програмата изчаква определено време. По предварително задаване това е 1 секунда. Можете също така да ползвате десетична запетая, за по-конкретно зададено време.

Сензорите също така се ползват много често. Това означава, че роботът чака да се активира датчик (като сензор за докосване например) или се достигне определена стойност на датчика (например програмата чака изминаването на разстоянието на ултразвукови сензори до над 50 см и след това програмата продължава.

Резултати от обучението

Преценете заедно с партньора си, дали сте достигнали следните цели на обучението и как ползвате математиката, за да постигнете това.

Какво е значението на:

- Двигател
- Изчакване
- Блок на Цикъла









Изходящи устройства – обратна връзка



Урок 7.-8.

Необходими материали:

• Сглобена база за робот LEGO EV3

В този урок ще научим за различните изходящи устройства, които дават аудио-визуална обратна връзка на потребителя.

Упражнения

Упражнение № 1:

Вашият LEGO Mindstorms EV3 робот има LED светлини при бутоните. Обичайно има светеща червена светлина когато роботът стартира или се изключва, и зелена светлина когато роботът е ВКЛЮЧЕН и очаква потребителя. Зелената светлина примигва когато програмата вътре в робота е активна.

Вашата първа задача е да програмирате роботът си да показва червена светлина в продължение на 3 секунди, зелена светлина за 2 секунди и оранжева светлина 1 секунда. Можете да ползвате *Статус на светлина на брика и Блока за изчакване* (виж фигура 3).



Фигура 3. Статус на светлина на брика отляво и Блока за изчакване отдясно

















Упражнение № 2:

Следващата задача е да имитирате светлините на светофара. Попитайте приятели или Гугъл, за да видите цикъла на светофара. След това програмирайте робота си да действа като светофар.

Помислете:

Какво трябва да промените, за да може пешеходците, които искат да пресекат, да чуват светофара? Цикълът ще започне с ВКЛЮЧЕНА червена светлина.

Упражнение № 3:

След това ще ползваме екранът на LEGO Mindstorms EV3, за да изобразим различни неща. На първо място се опитайте да покажете усмихнато личице на екрана на EV3. Можете да ползвате блока на Дисплея (виж Фигура 4).



Фигура 4. Блок Дисплей в програма EV3

Програмата сработи ли като за първи път? Какво трябва да добавите, за да виждате усмихнатото личице за по-дълго?

Упражнение № 4:

Направете 30-секунден скрийнсейвър или анимация. Можете да ползвате различни предварително зададени LEGO картинки, можете да направите ваша собствена или да ползвате настройките на блок Дисплей, за да преместите картина върху екрана на EV3.

Упражнение № 5:

Вашият робот LEGO Mindstorms EV3 има вграден говорител. Програмирайте роботът да лае като куче. Можете да ползвате блока за *Звука* (виж фигура 5.









Фигура 5. Блок за Звука в програма EV3

Упражнение № 6:

Ползвайте *Изсвири нота* или *Изсвири тон* от вашия блок на Звука, за да програмирате вашият робот да изсвири известна песен (например "Мери има агънце" и др.) Ползвайте Гугъл, за да намерите *Homu* за песните.

Резултати от обучението:

- Учениците знаят кои са трите основни LED цвята, които могат да ползват с EV3 брика.
- Знаят как да ползват предварително въведени фигури на блока на Дисплея.
- Знаят кои файлове, тонове и ноти могат да ползват в блока на Звука.









Урок 9.-10.

Необходими материали:

• Сглобена база на робот LEGO EV3 с всички сензори

Сензорите са сетивата на робота

Хората няма да могат да съществуват на Земята без сетива. Представете си, че в един момент няма да можете да чувате, да чувствате допир, да виждате или да вкусвате. Би ли било възможно да получите информация или обратна връзка от света?

Роботите са като нас – ние искаме да навигират и да действат в същия свят, в който живеем. Затова им даваме сензори, за да засичат обекти, разстояния, звучи и други.

Можете ли да свържете какви сензори трябва да използват роботите, за да бъдат като хората?

Човешко сетиво	Кореспондиращ сензор
Слух	
Зрение	
Допир	
Вкус	
Мирис	

Успяхте ли да намерите отговор за сетива като вкус и мирис? Науката достигнала ли е до този етап?

Потърсете в Гугъл, ако още не сте го направили!

Как се чувстват роботите?

Работете по двойки с приятел и се опитайте да откриете следните:

- 1. Затворете очи и се опитайте да ходите из класната стая или където се намирате
- 2. Вашият приятел ще ви потупа по рамото, когато разстоянието между вас и някакви предмети и пречки пред вас е по-малко от половин метър.
- 3. Вижте дали можете да се придвижите към вратата
- 4. Сменете ролите си











Как се почувствахте когато очите ви бяха затворени? Независимо дали достигнахте или не успяхте да достигнете до вратата, ви е било по-лесно от на робот, тъй като знаете плана на стаята или с други думи – картата.

В това упражнение имитирайте робот EV3 със сензор за допир.

Аналогови сензори

Сензорите са разделени на две основни категории аналогови и цифрови. Аналоговите сензори са тези, които изпращат електрическо напрежение към контролера на робота. Стойността на напрежението е пряко в зависимост от физическата стойност, която сензорът измерва. Това може да е температура, сила, светлина и т.н. Аналоговият сензор не казва на контролера, че температурата е 30°С, а например извежда 2,4 V. Напрежението, отговарящо на температурата, зависи от сензора и неговите характеристики. Тъй като микроконтролерът е цифрово устройство, той не може да бъде свързан с аналогови



сензори без "Аналогов към Цифров Преобразувател (ADC)". ADC е устройство, което преобразува сигнала на аналоговия сензор в цифров сигнал, с други думи, последователност от битове. Обикновено микроконтролерите имат вграден ADC и не се нуждаят от нищо повече от аналогов сензор, който да се свърже с подходящ порт на микроконтролера, за да започне ползването.

Какъв е единственият аналогов сензор, който EV3 роботът име? Защо мислите така?





1. Накарайте роботът да се задвижи напред с натискане на сензора. Ползвайте следния пример:



Вторият оранжев блок е блокът за изчакване (фигурата по-горе), това означава, че роботът ще няма да прави нищо, освен да чака да се изпълни определено условие. Сега роботът просто чака някой да натисне сензора – след това продължава да изпълнява следващия блок. Само оранжевите блокове са такива за изчакване.









Вижте, че сензорът за допир има три етапа (фигурата по-горе):

- 0 реагира само когато бутонът не е натиснат
- 1 реагира само когато бутонът е натиснат
- 2 реагира само когато бутонът е натиснат и след това отпуснат
- 2. Програмирайте робот със сензор за допир да следва следния алгоритъм:



Резултати от обучението

Преценете заедно с вашия партньор дали достигнахте следните резултати. Какво е значението на:

- Сензори за роботи
- Аналогови сензори
- Аналогов към Цифров Преобразувател
- Блокове за изчакване
- Сензор за допир (състояния на сензора за допир)









Сензори

Урок 11.-12.

необходими материали:

• Сглобена база на робот LEGO EV3 с всички сензори

Дигитални сензори

Последният път научихте за това какво са аналогови сензори. Сега ще разгледаме дигиталните сензори.

Работете по двойки и опитайте следното:

- 1. Затворете очи и се опитайте да вървите в класната стая или където се намирате
- 2. Вашият приятел ще ви казва разстоянието между вас и предметите пред вас на всеки 5 секунди
- 3. Проверете дали можете да стигнете до вратата
- 4. Сменете ролите си

Основната разлика между това упражнение и "ходенето със сензор за допир" е че сега получавате реална информация от сензора, вместо само стойност.

В това упражнение имитирате робот EV3 с ултразвуков сензор за разстояние. Роботът получава информация за разстоянието от сензора повече от 15 пъти в секунда.









Работен принцип на дигитален сензор.

Елементарен дигитален сигнал сигнализира за присъствие или отсъствие на напрежение, с други думи, логическа единица или нула. Можем да ползваме опростената електрическа верига (фигурата по-горе) като пример, тъй като лампата е свързана с електрозахранването чрез превключвател. В този случай, можем само условно да кажем, че лампата е запалена при логическа единица и че лампата не е запалена при логическа нула.

Ултразвуков сензор

Опитайте примерът по-долу. Това ще накара роботът да чака с ултразвуков сензор докато препятствието не е по-близо от 50 см и след това ще се задвижи напред. Проверете дали ултразвуковият сензор е прикрепен към порт 4.



Можете ли да преправите програмата така че роботът да се задвижи напред докато не засече нещо по-близо от 50 см?

Жиро сензор

Опитайте примера по-долу. Това ще накара робота да изчака, за да бъде завъртян на място на поне 50° и след това да се задвижи напред. Роботът ползва жиро сензор за тази цел. В случай, че роботът не работи добре, опитайте се да рестартирате жирото като премахнете жицата на сензора от контролера за 2 секунди и след това отново внимателно я поставите.



Жиро сензорът също така може да засече скорост на промяна на ъгъла градус/секунда.









Светлинни сензори

Опитайте примера по-долу. Роботът няма да помръдне докато светлинният сензор не види нещо по-тъмно от 50%. Можете да опитате с различни повърхности и да видите коя повърхност кара робота да се задвижи.



1 🕒 🖲 **(#)** Brick Buttons ÷ 0 Colour Change Infrared Senso 🛛 🚓 Ambient Light Intensity Motor Rotation

Можете ли да преправите програмата така че роботът да се движи напред докато светлинният сензор не засече тъмна повърхност?

Светлинният сензор има свойството да засича цветове и околната светлина. Вижте на картинката отляво.



Аларма против кражба

Когато имате нещо хубаво винаги има някой, който иска да ви го отнеме. Вашата задача е да ползвате всички възможни сензори и методи, за да подсигурите роботът си. Изградете най-добрата роботна аларма против кражби.

- 1. Поставете робота на работната повърхност
- 2. Стартирайте програмата на алармата против кражба
- 3. Роботът ще издава звук когато някой се опита да го премести от работната повърхност.













Вижте примерната програма (от учителя) и я накарайте да работи както вие желаете. Опишете програмата в полето по-долу. Трябва да ползвате паралелна обработка в тази програма. Това означава, че има програми, които работят паралелно, отделно една от друга.

Когато сте готови, направете състезание между другите групи и им позволете да се опитат да "откраднат" роботът ви. Единственото правило е, че жиците от сензорите не могат да бъдат премахвани.

Резултати от обучението

Преценете заедно с приятелите си дали достигнахте следните цели на обучението. Какво е значението на:

- Дигитални сензори
- Паралелна обработка
- Дигитален сензор EV3
- Може ли сензор за допир да се нарече дигитален?
- Може ли бутон на EV3 контролер да се счита сензор?

















Дигитални сензори

Урок 13.-14.



Необходими материали:

- Сглобена база на робот LEGO EV3 със сензори
- Черно тиксо и осветена повърхност

Следване на права линия

Следването на права линия е един от основните начини да накарате роботът да отиде където е необходимо. Нека да научим основите със софтуера на LEGO Mindstorms:



Отворете дейност "Превключване" и следвайте инструкциите. Накарайте роботът да следва линия и вижте дали може да го накарате да върви още по-бързо!









Опишете алгоритъма си за следване на линия в полето по-долу.

Безопасни роботи

Сега след като знаете как да накарате роботът да следва линия – можете ли да построите робот за фабрика, който премества стоки от едно място до друго?

- Роботът ще взема стоката от едно място (обозначено с черна линия) и ще я доставя до друга черна линия. Можете да поставите стоката отгоре на робота и да я премахнете с ръка.
- 2. Какво ще стане, ако има човек на пътя на робота? Може ли вашият робот да засече човек и да спре и да изсвири алармата?



Тук можете да намерите програма-образец. Но тя не работи идеално добре. По-ниският блок се нарича прекъсвач на цикъла и спира по-горния цикъл. Можете ли да го накарате да работи?













Резултати от обучението

Това е краят на частта за сензорите. Преценете заедно с вашите приятели дали сте достигнали следните цели на обучението.

- Сензори за роботи
- Аналогови сензори
- Дигитални сензори
- Паралелна обработка
- EV3 дигитален сензор
- Може ли сензор за допир да се нарече дигитален?
- Бутоните на EV3 контролерът могат ли да се нарекат сензори?
- Ползване на променливи за програмите

- Аналогов към Цифров Преобразувател
- Прекъсване на цикъла

















Развитие на уменията по наука, технологии, инженерство и математика чрез роботиката

Урок 15-16: Тимбилдинг – алгоритмично мислене

Резултати от обучението:

- Умения, нужни в 21 век (в частност комуникация, сътрудничество и креативност)
- Познаване на термина алгоритъм и алгоритмично мислене.

Необходими материали:

Тиксо или тебешир

Хартия и молив

Кодове с примери

Алгоритми и алгоритмично мислене

Алгоритъмът е последователност от команди, която се изпълнява в даден ред. Може да се разглежда като подробна рецепта. Пример за алгоритъм може да бъде направата на чаша чай. Този алгоритъм би описал подробните стъпки на "завиране на вода, намиране на чашата си, добавяне на пакетче чай, изсипване на водата в чашата докато се напълни, запарване за х минути, отпиване на чай от чашата." Когато този алгоритъм е разработен, можете да го повтаряте всеки път, когато искате да изпиете чаша чай. Имате ли предложения за други алгоритми, описващи действията в ежедневието ви?

Алгоритмичното мислене включва разбивка на по-големи и комплексни задачи на помалки и по-управляеми подзадачи. Също така включва организиране и анализиране на данни по логически начин и създаване на алгоритми/рецепти за разрешаване на комплексни задачи.

Развитието на алгоритмично мислене изисква учениците да се научат да подхождат към задачите по систематичен начин. Също така изисква учениците да предлагат решения, за които компютрите могат да помогнат при правилно програмиране.

















Алгоритмичното мислене включва

- подробни стъпки, необходими за разрешаване на задачата
- ползване на умения по програмиране, за да накараме компютъра да реши задачата

Упражнение 1: Алгоритмично мислене – Курс по преминаване на препятствия Алфа

Приблизително време:	Въведение: 5 минути
	Изготвяне на алгоритъм: 10 минути

Изпълнение на задачата пред класа: 5 минути

Кодове с примери

Код	Как да ползвате кода	Описание
Нов играч	Нов играч Алекс	Алекс застава в квадрат А
Напред	Алекс напред Алекс се премества една стъпка	
Назад	Алекс назад	Алекс се премества една стъпка назад
Наляво	Алекс наляво	Алекс стои в същия квадрат като се завърта наляво
Надясно	Алекс надясно	Алекс стои в същия квадрат като се завърта надясно
Скок	Алекс скача	Алекс прескача квадрата пред него като се приземява два квадрата по- напред

















Стъпки

- Учителят рисува решетка 5 х 5 квадрата с тиксо или
 тебешир на пода или навън, решетката, представена тук
 е само примерна
- А: начална точка, Z: крайна точна, X: опасна зона
- Поставете А и Z в противоположни ъгли, след това маркирайте опасните зони с X
- Опишете решетката и кодовете по-горе на учениците
- Разделете класа на двойки
- Дайте на всяка двойка лист и химикали
- Накарайте двойките да запишат алгоритъма си за
 разрешаване ред по ред, като ползват кодовете и примерите
- След 10 минути учителят избира две дойки, които да изпълнят решенията си пред класа

Α		X	
	X		
X			X
	x		z









Упражнение 1: Алгоритмично мислене – Курс по преминаване на препятствия Браво

Приблизително време:	Въведение: 5 минути
	Изготвяне на алгоритъм: 10 минути
	Изпълнение на задачата пред класа: 5 минути
	Обсъждане: 5 минути

При тази задача добавяме още една буква, например S. Класът решава предварително каква команда представлява тази буква.

Например:

S = Махане с ръка, Пляскане и Усмивка

Когато учениците започнат работа с техния алгоритъм, за да преместят играча от А към Z, задачата сега е при всяко стъпване на квадрат с буква S играчът да "Маха с ръка, Пляска и да се Усмихва".

Учителят решава колко букви S да има в решетката.

A	S		x	
		x		S
	S			
x		S	S	х
S		x		Z



Групово обсъждане:

- Как ползвахте алгоритмично мислене, за да решите задачата?
- Колко уникални решения мислите, че има, за да може играчът да достигне крайната точка?
- Кое от тази задача мислите, че ще е важно когато по-късно програмирате по двойки?









Урок 16: Тимбилдинг – Изграждане на паралелепипед

Резултати от обучението:

- Работа с умения, нужни в 21 век с цел подобряване на уменията за комуникация и сътрудничество.
- Познаване на термините водач и навигатор, както се ползват в програмирането.

Необходими материали:

Комплект LEGO Mindstorms edu EV3 (учителят подготвя пакети с 34 части, нужни за изграждането на паралелепипеда предварително)

Пликове с цип или подобни (един на двойка)

Инструкции за построяване на паралелепипеда:

https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-cuboid-dc93b2e60bed2981e76b3bac9ea04558.pdf

Приблизително време Въведение: 10 минути Задача: 20-35 минути Обсъждане: 10 минути



Въведение в концепцията – Водач и навигатор

Работата по двойки при роботиката и програмирането има няколко предимства и често води до добри дискусии и по-добро програмиране.

Учениците ще имат една от двете роли *"водач"* и *"навигатор"* по време на програмиране. Тези две роли се въвеждат рано, за да улесним сътрудничеството и да разберем защо използваме тези понятия. И двете роли са важни за решаването на тази задача. В тази задача водачът ще направи построяването, а навигаторът ще извърши обяснението. При програмирането водачът е програмистът, докато внимателният навигатор извършва обзора и дава съвети и идеи на водача.









За да се минимизира евентуален смут, всяка група трябва да седи с гръб един към друг в голям кръг с разстояние помежду си.

Упражнение:

- Започнете като разделите класа на двойки. Всяка двойка ще седи с гръб един към друг в голям кръг като водачът е от външната страна.
- Учителят ще даде на навигатора компютър и инструкциите за построяване
- Водачът ще получи частите 34 LEGO Mindstorms EV3 от учителя
- Навигаторът обяснява на водача как да построи паралелепипеда
- Водачът не може да вижда инструкциите за построяване и има право да ползва само обясненията на навигатора

Съвет 1: Учениците от всяка двойка дават подходящи наименования на тухлите LEGO, за да улеснят обясненията и по-лесно да се разбират.

Съвет 2: Оставете паралелепипедът сглобен след това, тъй като ще ви е нужен за някои от уроците по роботика.

Групово обсъждане:

- Каква е най-трудната част от тази задача?
- Кои стратегии ползвахте при различните двойки, за да изпълните задачата?
- Какво от тази задача мислите че е важно при програмиране по двойки покъсно?









Резултати от обучението

Преценете и опишете заедно със съотборника си до колко добре усвоихте целите на това обучение? В кои точно моменти вашият екип се справи добре при разрешаване на задачата? Какво може да се подобри следващият път от процеса на решаване на задачата?

Бележки от обсъждането:









База за задвижване

Урок 17.-18.



Необходими материали:

- Комплект LEGO Mindstorms Edu EV3
- Видео: <u>https://education.lego.com/en-us/lessons/ev3-dep/make-it-move-with-wheels#Planitem0</u>

Проектиране, изграждане и програмиране

Упражнение №1:

Проектиране, изграждане и програмиране на робот, който може:

- Да се движи на разстояние до 1 м
- Да ползва поне един двигател
- Да ползва колела за придвижване
- Да показва разстоянието, което е изминал

Идеи за изграждане:

- Голям двигател и колело
- Скосени зъбни колела

Ако екипът има нужда от още помощ, ползвайте инструкциите на Задвижване от Лего обучението:

• База за задвижване

Изследвайте различните начини да контролирате на базата ад се задвижи в права линия.

- Програма 1: Ползвайте блокът за задвижване, за да програмирате роботът да се движи на разстояние до 1 м. Диаметърът на стандартното колело е 56 мм.
- Програма 2: Накарайте базата да направи точно същите движения, ползвайки блокът за задвижване на танка.











Какви разлики забелязвате? Опишете по-долу

• Програма 3: Накарайте го да покаже изминатото разстояние.

Упражнение №2:

Вашият робот трябва да се премести на разстояние 1 метър върху груба повърхност. Какви промени бихте направили при построяването на робота и защо? Идеи за построяване:

• Вериги

Резултати от обучението

Преценете заедно с партньорът си как достигнахте следните цели от обучението:

- описване на различните механизми на движение и техните характеристики.
- показване на изминатото от робота разстояние.
- избор на механизъм за движение за робота в зависимост от терена.

База за задвижване

Урок 19.-20.

Необходими материали:

• Комплект LEGO Mindstorms Edu EV3

















Програмиране

За следващите упражнения ползвайте роботите, направени от учениците в предишния уъркшоп.



Упражнение №1:

Програмирайте вашия робот :

• да извърви квадратен маршрут (1м х 1м) посредством

различни параметри (ротации, градуси, секунди) по програмните блокове

Опитайте се да познаете алгоритъма на програмата. Опишете програмата.

Упражнение №2:

Всеки екип е отговорен за програмиране на роботизирана кола, която ще се движи по специфичен маршрут по улиците на града. (Улиците на града се проектират от учителя. Предположете, че повърхността на улиците е грапава, така че децата да ползват веригите на робота.)

Опитайте се да познаете алгоритъма на програмата. Опишете стъпките на програмата.

















Критично мислене в сферата на роботиката – избор на мисия на база време и точки

Урок 21.-22.

Резултати от обучението:

- Умения, нужни в 21 век (в частност критично мислене)
- Създаване на алгоритми за решаване на задачи

Необходими материали:

Сглобен базов робот LEGO EV3 със сензори

Подложка (или маркирано място на пода)

Сглобен паралелепипед

 Приблизително време:
 Въведение: 10 минути

 Задача:
 55 минути

 Предизвикателство:
 прибл. 15 мин (3 минути на екип)

 Обсъждане:
 10 минути

Сценарий на летище

Летище "ХелиФорс" наскоро е закупило нов робот, който да решава рутинни задачи и да изпълнява някои опасни такива, които се случват на летището всеки ден. Преди да се поверят задачите на робота е важно той да се тества и да е сигурно, че може да ги изпълнява самостоятелно.



















Нива и точки

Има четири нива с различни видове задачи за изпълнение. Ниво 1 съдържа по-малко задачи от ниво 2. Всяко ниво дава различни точки (вижте таблицата), ако задачата е изпълнена правилно и както се очаква. Ако задачата не е изпълнена, не се дават точки.

Време

След частта с програмирането, на всеки отбор се предоставят максимум 3 минути, за да изпълнят всички задачи. Екипите трябва да направят план кои задачи техният робот ще реши първо и последно, а изразходеното време също ще дава определен брой точки. Ако екипът не се справи с програмирането и за четирите нива по време на частта за програмиране, екипът все още ще има шанс да тества своя робот и да получи точки за изпълнените от тях задачи.

Ни	ва	Точки
1a		200
1b		200
1c		200
2		300
3		500
4		700 + 100

Не е нужно да се изпълнят всички задачи от ниво 1, за да се получат максималния брой точки.

Ако екипът е решил всички задачи (време)		Ако екипът е решил само някои задачи (време)	
Под 3 минути	200	Под 3 минути	100
Под 2 минути	400	Под 2 минути	200
Под 1.5 минути	600	Под 1.5 минути	300



Илюстрация 1: Подложка при сценарий летище

















Ниво 1: Тест за прецизност А

Роботът стартира в СТАРТОВАТА ЗОНА.

- а) Тръгва напред по посочен маршрут в ЗОНА 1.
- b) Стоп
- с) Изсвирва сигнал СТОП
- d) Връща се на СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.

Ниво 1: Тест за прецизност В

Роботът стартира в СТАРТОВАТА ЗОНА.

- а) Тръгва напред по маркиран кръг в ЗОНА 1
- b) Спира
- с) Изсвирва сигнал ЗАВИЙ НАЛЯВО
- d) Завърта се на 90 градуса
- е) Спира
- f) Връща се на СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.

Ниво 1: Тест за прецизност С

Роботът стартира в СТАРТОВАТА ЗОНА.

- а) Тръгва напред към маркирано място С в ЗОНА 1
- b) Спира
- с) Изсвирва сигнал СТОП
- d) Връща се в СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.









Ниво 2: Тест на сензора за цвят

Роботът стартира в СТАРТОВАТА ЗОНА като сензорът за цвят сочи надолу.

- а) Тръгва напред докато не се появи червен цвят в зона 2
- b) Спира
- с) Издава звук за аларма
- d) Завърта се на 180 градуса
- е) Връща се в СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.

Ниво 3: Тест на сензора за допир

Роботът стартира в СТАРТОВАТА ЗОНА като сензорът за допир е насочен напред.

- а) Тръгва напред докато не достигне място, обозначено със СТОП в зона 3
- b) Завърта се на 90 градуса
- c) Задвижва се напред докато не достигне стабилен обект с височина 10 см (може да е тухла ЛЕГО, книга и др.) и се натиска сензора за допир
- d) Изсвирва сигнал "Изпълнено"
- е) Показва на дисплея: "Натиснат сензор за допир"
- f) Връща се в СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.

Ниво 4: Тест на средния двигател и ултразвуковия сензор

Роботът стартира в СТАРТОВАТА ЗОНА със ултразвуковия сензор, насочен напред.

- а) Тръгва напред до мястото за обратен завой в зона 4
- b) Завърта се на 90 градуса
- c) Ползва ултразвуковия сензор, за да спре при обект ПАРАЛЕЛЕПИПЕД наляво по пистата
- d) Ползва средния двигател и хващач, за да вземе ПАРАЛЕЛЕПИПЕДА
- е) Завърта се на 180 градуса
- f) Тръгва напред и поставя предмета в БЕЗОПАСНАТА ЗОНА, от ляво в средата на кръга дава допълнителни 100 точки
- g) Показва на дисплея: Всички тестове завършени
- h) Връща се в СТАРТОВАТА ЗОНА

За да се получат точки трябва да се изпълнят всички задачи.

















Критични точки преди стартиране на програмиране

Ето няколко въпроса, които да си зададете докато планирате какви мисии да изберете, когато започнете работа по предизвикателството "Летище":

- Възможно ли е да разпределите задачите измежду членовете на екипа, за да бъдат ефективно изпълнени?
- Коя задача дава най-много точки в сравнение с времето за изпълнение?
- Колко време можете да ползвате за програмиране на всяка задача предвид даденото време, ще бъде ли достатъчно?
- Трябва ли да програмирате вашия робот да изпълнява няколко мисии с едно излизане, или да намалите риска като прави едно или две неща на излизане?

Резултат от сценарий "Летище"

Екип:	Точки	Дадени точки	Време, прекарано за задачите от всички нива/точки	Време, прекарано за някои задачи/точки
Ниво 1а	200		Под 3 = 200	Под 3 = 100
Ниво 1b	200		Под 2 = 400	Под 2 = 200
Ниво 1с	200		Под 1,5 = 600	Под 1,5 = 300
Ниво 2	300			
Ниво 3	500			
Ниво 4	700 + 100			
ОБЩО:				

Резултати от обучението

Помислете за това колко добре изпълнявате целите на обучението и за какво си заслужава да говорите след тази задача?

Преценете какво да запазите и какво да игнорирате, както и какво да подобрите, ако ви се даде подобно предизвикателство?



















Създайте презентация

Споделете вашето решение

Урок 31.-32.

Резултати от обучението:

- Работа с уменията, нужни в 21 век (практическа комуникация)
- Презентационни умения (дигитална презентация по избор)

Необходими материали:

Зависи от презентационния формат, съгласуван в клас

Урок 31: Изгответе презентация и споделете решението

Ако сте иноватор, който желае да продаде или сподели идеите си, в определен етап ще трябва да се договорите с инвеститори и дружества, за да ги убедите да инвестират във вашата идея. Най-добрият начин да направите това е да изготвите добра презентация. Сега ще направите презентация със своята идея и решение! На добрата презентация ѝ е необходима ясна комуникация, иновативно мислене, добро планиране и много практика.

Въпроси, които да вземете предвид:

- Как може идеята да бъде обяснена толкова ясно, че всички да разберат проблема без предварителни познания?
- Как можете да убедите слушателите, че проблемът е важен за разрешаване?
- Как ще покажете, че това е реалистична идея, че може да се внедри в реалния живот на обществото?
- Как могат ползваните алгоритми да се представят по добър и разбираем начин?
- Как можем да убедим слушателите, че сме вдъхновени и дълбоко ангажирани с работата си, как можем да ги убедим в резултата и неговия потенциал?

Задача: Ползвайте източниците, които имате, за да създадете презентация. Помнете, че трябва да сте креативни, изчерпателни и експресивни, за да можете наистина да предадете съобщението си до слушателите. Добрата идея не винаги се нуждае от екстравагантна презентация, но екипът трябва да им възможност да предаде визията и решението си. Креативността не изисква задължително много оборудване, но вие решавате. Успех!









Урок 32: Време за презентация – Споделете решението си с класа

Време за презентация и получаване на обратна връзка

Сега след като сте подготвили презентацията си, трябва да я представите. Останалите от класа ще наблюдават като публика. Въпреки че само един екип ще презентира по едно и също време, слушателите също ще имат задача: Да дадат на презентиращата група сандвич тип "обратна връзка" за техните усилия; две (филийки) положителни коментари и между тях един конструктивен коментар за това какво може да се подобри.

Например:

Положителен 1: Личи си, че се вдъхновявате от идеята си, ентусиазмът ви ме кара да искам да я видя реализирана!

Конструктивен съвет: Връзката ви с публиката може да бъде още по-добра, ако се фокусирате въхру зрителния контакт и ако четете по-малко от бележките си.

Положителен 2: Имате чудесен алгоритъм за това как да ползвате роботиката за справяне с проблем от реалния живот, накара ме да повярвам, че наистина може да се случи и че може да е от огромно значение за тези, които се нуждаят от това!

Когато всички са споделили презентациите си, аплодирайте всеки от тях, за добрата работа и отдаденост! Браво, сега официално сте програмисти!

Възможни критерии за оценка:

Яснота относно проблема и решението Свободно разказване или ползване на бележки? Зрителен контакт с публиката? Ангажираност по проблема и презентацията Логическо проектиране на алгоритъма Иновативно ползване на роботиката при разрешаване на проблеми Ниво на въздействие въхру качеството на живота

Реалистичност

Екипте могат да дадат и други предположежния относно съотносими критерии за оценка!



Co-funded by the Erasmus+ Programme of the European Union





Знаете ли че?

Думата "критичен" има етимологичен произход от двата гръцки корена "kriticos" (което означава преценявам) и "kriterion" (което означава стандарти). Следвателно от етимологична гледна точна, думата означава разработване на "преценка на база стандартите")

Източник: criticalthinking.org



Резултати от обучението

Заедно преценете и обележете как постигнахте целите на обучението и различните критерии за оценка? Ако имате подобна задача в бъдеще, какво ще направите по същия начин и какво по различен?

Бележки:

















Програмиране

Урок 33.-34.



необходими материали:

- Базов робот LEGO Mindstorms Edu EV3 със сензори за допир и цвят
- Базов комплект LEGO за сглобяване

Променливи и константи

Най-основото нещо при една програма е да съхранявате информация, която може да се ползва повторно или да се оцени по-късно в програмната логика.

Какво е програмна променлива?

Променливата е място за съхранение на данни, на която даваме име в рамките на програмата. Името предоставя начин за обозначение на данните, така че да можем да разберем по-добре целта им в логиката на програмата. Мислете за това като за тефтер или кутия, която пази стойността за вас. Можете да назовете променливата както желаете.

Можете да дефинирате вида променлива:

- Нумерологична (съдържа цифра)
- Логична (съдържа стойност Вярно/Грешно)
- Текстова (съдържа редове от текст ... "Здравей, свят")
- Цифров масив (съдържа числа ... 1,2,3,10,55)
- Логически масив (съдържа набор от логики ... Вярно, Вярно, Грешно)

Могат да се ползват и като входни и като изходни данни, така че можете да....

- Пишете дадете стойност на променливата
- Четете възстановите последната писмено зададена стойност на променливата

За да съхранявате и оцените информацията на променливата, ползвайте блока за Променливата, който ще познаете по иконата куфар. Писането (входящите данни) са с извита дъга нагоре, а Четенето (изходящите данни) с извита дъга надолу.









Какво е програмна константа?

В компютърното **програмиране константа** е стойност, която не може да бъде променена от програмата по време на нормално изпълнение, т.е. стойността е **постоянна**. Това контрастира с променлива, която е идентификатор със стойност, която може да бъде променена по време на нормално изпълнение, т.е. стойността е променлива. **Константата** е посочено място в паметта, което временно съхранява данни, които остават същите по време на изпълнение на програмата.

Блокът на константата осигурява отправна точка за проводник за данни, чиято стойност можете да изберете ръчно. Вие избирате типа на проводника за данни, като изберете режим (Текстови, Цифров или Логичски) и въвеждате стойността му в полето Стойност. Блокът на константата е полезен, ако искате да конфигурирате настройките на няколко блока с една и съща стойност.

Упражнения

След всяка упражнение, учениците трябва да представят разработените си проекти до другите екипи за обсъждане и промени.

Упражнение № 1

Можете ли да накарате програмата да показва брой пъти, в които сте натиснали бутона?

Упражнение № 2

Можете ли да напишете програма, която да брои черните линии, които сте преминали?

Логика

Логическият блок извършва логически упражнения по входящите данни и показва резултата. На логическия блок се задават входящи данни, които са Верни или Грешни и той произвежда Верни или Грешни изходящи логически стойности, които могат да се ползват за входящи в цикъл и при условия на превключване. Намира се в таба на червената програмираща палета.

Различни режими в логическия блок

Modes	Inputs Used	Result
AB AND	А, В	True if both A and B are True, otherwise False
OR OR	А, В	True if either A or B (or both) is True, False if both A and B are False
XOR	А, В	True if exactly one of A and B is True, False if both A and B are True, False if both A and B are False
NOT	А	True if A is False, False if A is True









Режими	Ползвани входящи данни	Резултат	
		Вярно, ако и А и В са верни, в противен случай	
		Грешно	
		Вярно, ако А или В (или и двете) са верни,	
		грешно ако А и В са грешни	
	Вярно, ако точно едно от А и В е вярно,		
		Грешно, ако и А и В са верни,	
		Грешно, ако и А и В са грешни	
		Вярно, ако А е грешно,	
		Грешно, ако А е вярно	

Упражнение № 3

Направете така, че роботът ви да се придвижва напред докато ИЛИ сензорът за допир е натиснат или цветовият сензор засече черен цвят. Следвайте стъпките.

- Включете двигателите
- Добавете блокове Логически или Сензор
 - А. Ползвайте логическия блок в режим ИЛИ

• В. Добавете входящите данни: Вземете блоковете на цветовия сензор и сензора за допир и ги включете към логическия блок като входящи данни

- Добавете Цикъл и условия за изход от цикъла:
 - Поставете блоковете на сензорите и логическия блок в цикъла

• Като условие за изход от цикъла, изберете логическото. Включете резултата на логическия блок в условията за изход

• Ако резултатът от СТЪПКА 2 е верен, трябва да излезете от цикъла и да спрете робота

Упражнение № 4

Всеки екип трябва да изгради система, която ще разчете ниво на водата (сини тухли). Когато нивото на водата премине лимита, то двигателят ще задейства функцията за безопасност и водата ще продължи да тече. Отворете програми **Упражнение 4 Тръби** и **Упражнение 4 Преливане** за помощ.

Математика

Блокът за математиката извършва математическите изчисления по входящите данни и показва резултатът. Можете да направите проста математическа операция с една или две входящи данни или да впишете формула с до четири стойности. Изберете математическата операция, която искате да ползвате, като изберете режим с "Избор на режим". След като направите избора можете да изберете стойности за входящите данни. Наличните входящи данни ще се променят в зависимост от режима.











Упражнение № 5



Какво мислте, че ще прави горепосочената програма?

Резултат от обучението

Заедно с партньора си преценете доколко достигнахте следните цели:

Ползване на програмните концепции при алгоритми като

- о Константи
- о Променливи
- о Логически операции
- о Математика

















Програмиране

Урок 35.-36.



Необходими материали:

- Комплект LEGO Mindstorms Edu EV3
- Базов комплект LEGO за сглобяване

Сравнение

Блокът за сравнение сравнява две цифри, за да открие дали са еднакви или едната е по-голяма. Можете да изберете едно от шест различни сравнения. Резултатът от изходящите данни е Вярно или Грешно. Изберете вида сравнение, който искате, като ползвате Избиране на режим. Блокът ще изчисли резултата чрез сравняване на двете данни А и В, както е показано в таблицата по-долу.

Режим	Ползвани входящи данни	Резултат
Равно на	A, B	Вярно ако А = В, в противен случай Грешно
Р азлично от	A, B	Вярно ако А ≠ В, в противен случай Грешно
По-голямо от	A, B	Вярно ако A > B, в противен случай Грешно
По-малко от	A, B	Вярно ако A < B, в противен случай Грешно
По-голямо или равно на	A, B	Вярно ако А ≥ В, в противен случай Грешно
По-малко или равно на	A, B	Вярно ако А ≤ В, в противен случай Грешно









Тази блокова последователност тества дали стойността на променлива "мощност" е по-голяма от 100 и ако да, я задава да бъде 100. Блокът за сравнение сравнява стойността на променливата до 100 и логическият резултат се ползва от <u>Превключвателя</u> за да избере дали да промени стойността на променливата.

Паралелно и серийно програмиране

Серийно програмиране е когато всяка команда се изпълнява една след друга. Паралелно е когато повече от една команда се изпълнява по едно и също време.

Какво са паралелни лъчи?

Паралелните лъчи ви позволяват да стартирате два или повече блока едновременно. Какво става, ако имате една или повече прикрепени ръце, свързани с двигатели и искате да завъртите тези ръце, докато роботът се движи, за да завърши мисия.

За да създадете успореден лъч, щракнете и плъзнете бъмпа в десния център на който и да е блок и освободете, след като сте задържали курсора върху обърнатия бъмп от левия център на блока.



Забележка: Преди разделяне, блоковете ще се изпълняват един по един. След разделянето на блоковете на два "лъча" те ще стартират едновременно.

В реалния живот

Разпознавате ли къде се използват тези концепции около вас? Можете ли да посочите и да назовите машини / роботи до вас, където виждате тези програмни концепции, използвани в типа поведение? Като например как се създава логиката на асансьора?









Какво прави този робот?

След всяка упражнение учениците трябва да представят разработените си проекти на другите екипи за дискутиране и промени.

Упражнение № 1:

Стъпка 1: Изградете робот, който може да се движи и да вдига предмети. Инструкциите на образователния проект Lego EV3 ще ви помогнат. Ползвайте и ултразвуков сензор.

Стъпка 2: Можете ли да напишете програма, която ползва паралелни лъчи, които трябва да преместят и вдигнат предмет едновременно?

Упражнение № 2:

Стъпка 1: Използвайте предишната конструкция на робот с два двигателя и ултразвуков сензор. Този робот е кученце, а ултразвуковият сензор е очите на кученцето.

Стъпка 2: Програмирайте кученцето да следва вашата ръка. Когато ръката ви е между 10 - 15 см, тогава кученцето ще продължи напред, а когато ръката ви е между 0 - 15 см, кученцето ще отиде назад.

Стъпка 3: Използвате ли променливи в програмата си и защо?

Резултати от обучението

Заедно с партньора си преценете доколко достигнахте следните цели:

- Ученикът знае разликата между серийно и паралелно програмиране и разбира кое кога да ползва.
- Ползва програмни концепции в алгоритми като
 - о Сравнение







