Николина Николова • Елиза Стефанова • Мирослава Николова Диана Петрова • Олег Константинов • Станислав Иванов

СВИТЪК С УРОЦИ ОТ ТЕМАТА "КОМПЮТЪРНО МОДЕЛИРАНЕ"



КОМПЮТЪРНО МОДЕЛИРАНЕ и информационни технологии



Николина Николова • Елиза Стефанова • Мирослава Николова Диана Петрова • Олег Константинов • Станислав Иванов

СВИТЪК С УРОЦИ ОТ ТЕМАТА "компютърно моделиране" за **5. Клас**



СЪЗДАВАНЕ НА ГРАФИЧНИ ИЗОБРАЖЕНИЯ С ИЗУЧАВАН ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ

📩 ТУК ЩЕ НАУЧИТЕ КАК:

- да създавате собствен герой;
- да създавате код за чертане на изучавани равнинни фигури.

🗹 Мисия 1

В часовете по компютърно моделиране се научихте да създавате образователни игри, в които използвахте герои от тематични графични колекции. А какво ще направите, ако не откривате подходящ герой за вашия проект? Ще създадете свой собствен!

В следващите часове ваш първи помощник ще бъде Роботко – хайде, нарисувайте го във визуалната среда за програмиране.

Визуалната среда за програмиране Scratch

Scratch е интерактивна платформа за визуално програмиране, в която можете да създавате проекти, като анимации, истории, игри.

- Средата има два варианта за използване:
- онлайн на адрес https://scratch.mit.edu/
- програма, инсталирана на вашия компютър.



Трик: Стартиране на онлайн Scratch

Ako goceza не сте използвали Scratch:

- 1. Заредете в браузър адреса https://scratch.mit.edu/
- 2. Присъединете се към общността на Scratch любителите, като регистрирате собствен профил.
- 3. Влезте в профила си.
- Създавайте ваши собствени проекти, разглеждайте какво творят вашите приятели, или потърсете нови идеи!



Основни обекти в Scratch

- Герои (спрайтове) основният обект в Scratch. Героите са графични изображения и може да се добавят в проекта от тематичните колекции в Scratch, от външен файл или да се нарисуват. Всеки герой има характеристики: име, размери, костюми, местоположение, посока на движение, показване – видим или скрит и се управлява чрез команди.
- Сцена и фонове (декори) сцената е мястото, където се развива действието в проекта. Сцената може да има няколко декора – различни графични изображения.

🗹 Любопитно

Спрайт означава фея, gyx, митичен герой, но gymama означава още компютърен графичен oбеkm, koumo може ga се gвижи по сцената и ga се управлява чрез команди.

Спрайт н	Роботк	0	+	x 6	2 1 y	-84
Показване	0	ø	Размер	100	Посока	90
Роботко						



Работна площ на Scratch

Работната площ на Scratch позволява управление на героя чрез *блокови команди*. За целта първо от областта за герои (раздел за спрайтове) се избира героят, който ще се управлява.

Програмният код се сглобява, като се извлачват и сглобяват блокове, подобно на пъзел.

Блоковете са групирани в раздели според тяхното предназначение. Всеки раздел е означен със специфичен цвят, например блоковете за движение – в синьо, за рисуване – в зелено, и т.н.

Видове блокове в Scratch

Блоковете в Scratch се делят на няколко вида в зависимост от мястото, където може да бъдат добавяни в програмния код.



😫 ЕКСПЕРИМЕНТИРАЙТЕ...

- 1. Разгледайте последователно различните групи с блокове.
- 2. За kou om mях можете да предположите за kakBo служат?

Създаване на собствен герой

В Scratch може да създадете ваши авторски герои във вграден графичен редактор като използвате инструменти за изчертаване на геометричните фигури *правоъгълник, окръжност* и линия и рисуване с четка. С изрисуваните фигури можете да извършвате операциите преоразмеряване, копиране, завъртане. На геометричните фигури можете да задавате плътно или преливащо запълване, дебелина и цвят на контура. Можете да групирате няколко фигури в една за по-лесна работа, както и да разполагате фигурите една под друга.

🗹 Припомнете си ...

- Как се използват инструментите за рисуване със свободна ръка в графичен редактор?
- 2. Как се използват инструментите за изчертаване на геометрични фигури?

5.1.

🗹 Мисия – възможна 1!

Създайте Роботко!

- 1. Създайте нов проект в Scratch.
- 2. Изтрийте героя Котка.
- 3. От менюто Избор на спрайт изберете Нарисувай.
- Нарисувайте и форматирайте основните повтарящи се фигури и след това ги копирайте и преоразмерете, ако е необходимо.
- 5. Използвайте инструментите *Назад* или *Отзад*, за да разположите фигурите една зад друга и да се виждат само части от тях.
- Групирайте вече готовите части от героя чрез натиснат клавиш Shift и маркиране на елементите, след което натиснете бутона Групиране, за да можете по-лесно да ги премествате.
- 7. Запазете героя с име Роботко, като в полето *Спрайт* въведете името му.

Трикове за професионалисти

t

÷

.

Q

Нарисувай

- Скицирайте на хартия вашия герой и анализирайте от какви геометрични фигури е съставен.
- Приложете знанията и уменията си за рисуване в графичен редактор.

-1-

Отзад



Трик за професионалисти За да изчертаете окръжност, квадрат, хоризонтална или вертикална линия, задръжте натиснат клавиша Shift, gokamo чертаете.

о Ш

Растерен и векторен режим

Забелязахте ли в графичния редактор на Scratch бутона *Преобразуване в растерен режим*? Когато героят се преобразува в растерен режим, изображението се превръща в растерно и се виждат отделните пиксели, от които е изградено. Затова е препоръчително да създавате вашите герои във векторен режим.



Векторен режим Растерен режим

🗹 Мисия 2

Роботко вече е готов за своята първа задача – да изчертае ограда с формата на квадрат със страна 100 стъпки!

Блокове за рисуване

Героите в Scratch може да оставят следа (път) при движението си. За тази цел от долния ляв ъгъл на работното поле натиснете бутон Добавяне на разширение 2 и изберете Молив . В менюто с групите блокове най-отдолу се появява икона

Най-често използвани блокове са:

- Молив долу героят започва да изчертава своя път;
- Молив горе спира изрисуването на пътя;
- Изтрий всичко изтрива всичко нарисувано до момента;
- Направи цвета на молива задава цвят на следата;
- Направи дебелината на молива задава дебелина на следата.

🗹 Мисия 2 – възможна!

Напишете код, с който Роботко да изрисува оградата:

- 1. Работете в проекта, в който създадохте Роботко.
- 2. Променете размера на Роботко на 50%.
- 3. Добавете в кода на Роботко блока *Молив* долу от група *Молив* и блок за цикъл, който да изчертае квадрат със страна 100.
- 4. Задайте дебелина и цвят на молива по ваш избор.
- 5. Изпробвайте кода.
- 6. Запазете проекта:
- с име *Ограда*, ако работите в онлайн версията на Scratch;
- с име Ograda.sb2 в Scratch 2 или Ograda.sb3 в Scratch 3.

🗹 Костюми за Роботко

Създайте костюми за Роботко, които да се сменят при смяна на посоката на движение.

🗹 Любопитно

Един от стиловете в изобразителното изкуство, при който се използват различни геометрични фигури, се нарича кубизъм. Един от най-известните художници, творили в този стил, е Пабло Пикасо.

изтрий всичко

100

и молив долу ч

Трик за професионалисти Използвайте блока Изтрий всичко след блока, за да изчертавате на чисто при всяко стартиране на проекта.

Трик за професионалисти

Защо да намалите размера на героя? Уместно е да намалите размера на героя, с който рисувате, защото в противен случай той ще закрива рисунката.

🗹 Припомнете си ...

- 1. За какво се използва блок за цикъл?
- 2. Kakßu ca xapakmepucmukume на kßagpama?

Ограда правоъгълник Направете копие на проекта

Паправете копие на проекта *Ограда* и модифицирайте кода, maka че Роботко да изчертава правоъгълна ограда с размери 100 и 50 стъпки.

молив долу направи дебелината на молива направи цвета на молива повтори премести се с 100 стълки











СЪЗДАВАНЕ НА ГРАФИЧНИ ИЗОБРАЖЕНИЯ С ИЗУЧАВАН ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ

📩 ЩЕ УСЪВЪРШЕНСТВАТЕ УМЕНИЯТА СИ:

за създаване на собствен герой;

• за изчертаване на различни равнинни фигури.

🗹 Приятел на Роботко

Създайте нов герой с име Сферичка, която заедно с Роботко да се справя с нови предизвикателства.



Раница

6

спрайт

÷

спрай

✓ Припомнете си от 4. клас

За какво се използват променливите в език за блоково програмиране?

Променливи в Scratch

Променливите в Scratch се създават и променят от групата блокове *Променливи*. Всяка променлива има име и стойност. Препоръчително е името да е смислено и да подсказва какви данни се съхраняват в променливата.

🔮 ЕКСПЕРИМЕНТИРАЙТЕ...

- 1. Създайте нов проект и добавете Роботко.
- 2. Въведете кода, показан отдясно.
- Пуснете програмата и изберете едно число, на което да разделите 360. Най-уместно е числото да е между 3 и 10.
- Отговаряйте на въпросите с ga, gokamo не се получи затворена геометрична фигура.
- 5. Каква фигура получихте?
- 6. Колко стъпки трябваше да направите, за да получите фигурата?
- Има ли връзка числото, което въведохте в началото, с фигурата, която получихте?

Изчертаване на многоъгълник

За да изчертаете многоъгълник, на който всички страни са равни, трябва да разделите 360 градуса на броя на страните на многоъгълника и на получения ъгъл да завъртите героя след изчертаването на една страна.

Например, ако искате да получите петоъгълник, разделяте 360 на 5 и получавате ъгъл на завъртане 72 градуса.



🗹 Многократно използване на герой

В *Раницата* освен код можете да запазвате и герои. Провлачете героя в раницата. За да го използвате в друг проект, отворете раницата и го провлачете в областта за герои.

Име на новата Завъртане	променлива:
Завъртане	
спрайтове	Само за този спрайт
Облачна пром на	иенлива (съхранявана сървър)
	Отказ Добре
	□ Облачна пром на



🗹 Направи ли ви впечатление, че ...

Броят на повторенията в блока за цикъл е равен на:

- числото, което се съдържа в името на многоъгълника (*триъгълник, петоъгълник*);
- броя на страните на многоъгълника;
- числото, на което трябва да разделите, за да получите ъсъла на завъртане.



🗹 Промяна на многоъгълника

Ако увеличите стъпките, с които се премества героят, ще получите по-голям многоъгълник, а ако увеличите броя повторения и числото, с което делите 360, ще получите многоъгълник с повече страни.



70 gecemote 60

🗹 Обърнете внимание!

100

Посоката на завъртане има значение!



🗹 Запазване на местоположението

Ако искате дадена фигура да се изчертава винаги на едно и също място, можете да запазите точното място на героя, който я изчертава.

За целта преместете героя в позицията, която харесвате, и издърпайте блока *Отиди go x: ... y: ...*

Когато вземате блока *Отиди до*, в него се запазва точното място на героя върху работно поле.

🗹 Геометрията е изкуство!

Отворете файла **GeometryArt.sb3** в nanka Scratch om ресурсните файлове към учебника или проекта **GeometryArt** в студиото Компютърно моделиране 5. клас, изд. "Просвета" с agpec https://scratch.mit.edu/studios/30097637 и експериментирайте – променяйте броя на повторенията и ъгъла на завъртане в двата блока за цикъл по следното правило: броят на повторенията е равен на числото, с което делите 360.

🗹 Влагане на блокове за цикъл

Блоковете за цикъл може да се влагат един в друг. Вложеният блок изчертава геометричната фигура, а външният блок изчертава повторенията на фигурата.



Заменете блока за завъртане с блоковете *Молив горе* и *Премести се с 50 стъпки*. Как се промени фигурата?



📩 ТУК ЩЕ НАУЧИТЕ КАК:

- да откривате еднотипни действия в компютърна програма;
- да създавате код за анимиране на компютърните герои
- с подпрограми;
- да усъвършенствате съществуващи проекти чрез използване на собствени блокове или подпрограми.
- на сооствени олокове или подпрограми.

Понятие	Описание
Подпрограми (блокове, функции)	Именувана последователност от инструкции,
	реализираща даден алгоритъм, която може да се
	използва многократно в компютърна програма



🗹 Мисия 1

Роботко иска да се научи да танцува. Помогнете му това да стане бързо и лесно! Използвайте Роботко и костюмите му от файла **Robotko.sb3** или проекта **Robotko** в студиото *Компютърно моделиране 5. клас,* изд. "Просвета".

Планиране на движенията на компютърен герой

Героите в Scratch имат три вида движения:

- движения на части от героя без промяна на местоположението му – използват се различни костюми на героя
- движения с промяна на местоположението му.
- движения с промяна на части от героя и местоположението му.

При работа с героите може да групирате тематично движенията и да им измислите имена, например – скачай (скок), падай (падане), танцувай (танц). Може многократно да използвате тези движения, като ги оформите като **подпрограми**.

Подпрограма в Scratch

Подпрограмата в различните езици се нарича с различно име – собствен блок, функция или друго. В Scratch тя се реализира чрез собствен блок. Блокът се създава от групата блокове Моите блокове, има име и може да има едно или няколко полета за въвеждане на число, текст или условие, наречени **параметри**. След създаването му той се появява в групата Моите блокове и се използва както всички останали блокове.

💾 Трик за професионалисти

Студиото *Компютърно моделиране 5. клас, изд. "Просвета"* може лесно да откриете в сайта на Scratch, като напишете името му в полето за търсене и изберете *Студиа*.

Предимства на подпрограмите

- Позволяват разделянето на сложна задача на няколко по-лесни подзадачи.
- Създават се веднъж и лесно се редактират.
- Намаляват дублиращия се код в програма.
- Правят кода на програмата по-прегледен и с това се намалява допускането на грешки.
- Използват се многократно с различни стойности на параметрите.

🔮 Трик за професионалисти

Препоръчително е името на собствения блок и параметрите да подсказват какво е предназначението му.





🗹 Мисия 1 – възможна!

- 1. Отворете файла Robotko.sb3.
- 2. Отворете група *Моите блокове* и натиснете бутона *Създай блок*.
- 3. Задайте име на блока *танцувай* и натиснете бутона *Добре*.
- 4. Появява се блок (трябва да добавите кода на блока винаги през 1 секунда да се сменят двата костюма *ляво* и *дясно* на Роботко.
- 5. Изпробвайте кода.



🗹 Как да откривате еднотипни действия в компютърна програма?

Еднотипните действия в компютърна програма може да откриете, като потърсите в кода еднакви последователности от блокове, които може да се различават по стойностите в блоковете.

•••

🗹 Omkpuūme ...

Отворете файла GeometryDrawing.sb3 в nanka Scratch. Открийте общите и различните елементи. Какви собствени блокове може ga съзgageme, за ga намалите gължината на koga?

Дефинирайте ги и модифицирайте koga. Запазете проекта с име **GeometryDrawing1**.





🗹 Мисия 2

Роботко иска да изчертава с един блок различни геометрични фигури. Възможно ли е това?

Подпрограма с параметри

Блоковете *петоъгълник* и *шестоъгълник* се различават единствено по броя повторения в блока за цикъл и числото, с което се дели. Тези различни стойности може да се представят като параметър при дефинирането на подпрограмата, който да ги обобщава, и подпрограмата ще изчертава многоъгълници с различен брой страни.

🗹 Мисия 2 – възможна!

Дефиниране на собствен блок с име *многоъгълник*.

- 1. Отворете нов проект.
- 2. Отворете група *Моите блокове* и натиснете бутона *Създай блок*.
- 3. Задайте име на блока *многоъгълник*, добавете параметър поле за въвеждане на число или текст с име *брой страни* и натиснете бутона *Добре*.
- 4. Появява се блок който трябва да добавите кода на блока:
 - 4.1. цикъл *повтори*, като за стойност на повторенията провлачвате блока брой страни от блока *дефинирай*
 - 4.2. блок премести се със 100 стъпки
 - 4.3. блок завърти се надясно ... на градуса като делите 360 на блока брой страни.
- 5. В групата *Моите блокове* се появява блок и можете да го провлачите в областта за код и да напишете конкретно число за параметъра *брой страни*.
- 6. Изпробвайте кода.







СЪЗДАВАНЕ И ИЗПОЛЗВАНЕ НА СОБСТВЕНИ БЛОКОВЕ ИЛИ ПОДПРОГРАМИ

📩 ЩЕ УСЪВЪРШЕНСТВАТЕ УМЕНИЯТА СИ:

за създаване и използване на собствени блокове

за работа с параметри на блокове

•••

🗹 Блок многоъгълник с два параметъра 🛛

Запазете вашия блок многоъгълник в *Раницата* или във файл **Polygon.sb3**. В нов проект го редактирайте, така че потребителят да въвежда като параметър и дължина на страната.



🗹 Изчертайте окръжност

В нов проект експериментирайте в блока *многоъгълник* със стойностите на страните и дължините на страните, така че да изчертаете окръжност.



🗹 Редактиране на собствен блок

Можете да редактирате вече създаден собствен блок – от контекстното му меню изберете *Редактиране*.

🗹 Емблеми на марки

Сещате ли се за емблеми, в koumo участват геометрични фигури? Емблеми на kou марku автомобили можете да изчертаете в Scratch? Опитаите да създадете и ваши собствени емблеми.

🗹 Изчертайте красива розетка

Kamo използвате блока *многоъгълник*, изчерmaume ваша авторска розетка.



10 окръжности

12 шестоъгълника със страна 50

24 триъгълниka със страна 100

🔮 Трик за професионалисти

- Използвайте блок за цикъл, в който да вложите блока *многоъгълник*.
- Експериментирайте с комбинации от различни фигури.
- Задавайте големи стойности в блока направи дебелината на молива, за да изчертавате запълнени розетки.

😫 ЕКСПЕРИМЕНТИРАЙТЕ ...

Каква е ролята на етикетите при създаване на собствен блок?



УПРАЖНЕНИЕ

🗹 Снежинки

Създайте дизайн на хартия за опаковане с основен елемент *снежинка*, която да променя големината и цвета си и да се появява на случайно място на сцената.





🔮 Трик за професионалисти

- Като използвате блока многоъгълник, съзgaume собствен блок снежинка с параметър дължина
- Използвайте блока избери случайно от ... go ... за избор на стойност на параметъра дължина на снежинката (от 5 go 25), цвят и разположение.



🗹 Шевици

В клуба по български фолклор сте учили за шевиците и искате да нарисувате мотив в Scratch, с koūmo да участвате в училищна изложба на тема "Българските шевици".



🗹 Падащи снежинки

Създайте собствен блок, който да имитира nagaне на обекти (kanku, снежинки) от горната част на сцената от случайна позиция. Използвайте файла FallDown.sb3 в nanka Scratch или проекта FallDown в студиото Компютърно моделиране 5. клас, изд. "Просвета".



Клонинг

Клонингът е идентично копие на герой и се използва, когато има нужда героят многократно да се появи на сцената и да се изпълни кодът му. Създава се от групата блокове Контрол с блока създай клонинг и най-често се използва с блоковете при старт като клонинг и премахни този клонинг.





Трик за професионалисти

- Използвайте вложени блокове за цикъл, за да изчертаете мотив 1. Във вложения цикъл изчертавате броя на бодовете на един ред, а във външния цикъл – целия мотив, който е своеобразен квадрат
- Редактирайте блока бод, като му добавите параметър дължина, който да задава големината на бода в стъпки, за да изчертавате комбинации от бодове с различна големина.



СЪЗДАВАНЕ И ИЗПОЛЗВАНЕ НА СОБСТВЕНИ БЛОКОВЕ ИЛИ ПОДПРОГРАМИ

📩 ЩЕ УСЪВЪРШЕНСТВАТЕ УМЕНИЯТА СИ ЗА:

- създаване и използване на собствени блокове;
- създаване и използване на подпрограми, koumo предават стойност.

🗹 Подпрограмите идват на помощ

Ваш приятел има нужда от помощ – създава компютърна игра, в която, за да се отключат различни умения на героя, трябва да се въведе код, който представлява сбора на цифрите на случайно генерирано трицифрено число. А учителят по математика ви моли да създадете програма, чрез която лесно да проверявате дали едно трицифрено число се дели на 3. Какво е общото между двата проблема? Помислете как да помогнете за решаването им.

Подпрограми, които предават стойност

Подпрограмите може да предават стойност към останалата част от програмата – например да изчисляват резултат или да генерират текст.

В Scratch предаването на стойност се осъществява с помощта на променливи.

😫 Трик за професионалисти

За да предадете стойност от собствен блок към останалата част на програмата, създайте променлива, която да пази стойността. Можете да използвате променливата както в собствен блок, така и извън него.

🗹 Откъде да започнете?

- Определете входните данни, които подпрограмата трябва да обработи.
- Задайте подходящо име на подпрограмата и съответния/те параметър/параметри.
- Помислете как чрез деление с частно и остатък на числата 10 и 100 може да намерите цифрите на единиците, десетиците и стотиците на трицифрено число.
- Открийте кои блокове от група Оператори ще ви бъдат полезни.



- Операторът закръгли надолу/нагоре от ... връща цяло число.
- Използвайте променливи за запазване на стойностите на единици, десетици и стотици, както и временен резултат.



Анализ на задачата и математически модел

Нека въведеното число е **789** *Цифрата на единиците* = остатъка при деление на числото 789 с 10 *Цифрата на десетиците*: Първо се намира цялата част при деление на числото 789 със 10 (т. е. 78), а след това – остатъка при делението ѝ с 10 (т.е. 8) *Цифрата на стотиците* = цялата част при деление на числото 789 със 100.



🗹 Помислете..

Може ли подпрограмата да се оптимизира, kamo се сведе до серия повтарящи се действия?

🗹 Коледна украса

e una

За кой ден да пресметна броя на топките и чака

Роботко има играчка – топка за украса на коледната елха. По неин модел на следващия ден направил една топка, а всеки следващ ден започнал да прави толкова топки, колкото е направил за предишните два дни общо, като се брои и топката, която е модел. Хрумнало му, че може да напише подпрограма, с която да изчислява колко топки ще изработи за даден ден.

Анализ на задачата и математически модел

Последователно, чрез цикъл, пресмятате броя на monkume за 3-ти, 4-ти, и т.н. ден, докато стигнете до желания ден.

Ще ви бъдат нужни две променливи *Число 1* и *Число 2*, koumo да пазят информация за броя на monkume om предходните два дни.

Забелязвате, че ако ви интересува n-ти ден, броят на повторенията ще бъде n - 2, тъй като за първите два дни броят на monkume е известен.

🗹 Любопитно

Числата, отговарящи на броя на monkume през всеки следващ ден, са известни като **числа на Фибоначи**. Те се откриват навсякъде около нас – в природата, в изкуството, в архитектурата, дори в човешкото тяло!

🗹 За най-любознателните

Можете ли да пресметнете общия брой топku до даден ден включително?

🔮 Трик за професионалисти

Добавете още една променлива общо, която първоначално има стойност 2 (monkume за първи и втори ден) и след това променяйте в цикъла стойността ѝ с *резултата* за всеки ден.





дни ще изработя

Препоръчително е да извеждате разбираеми за потребителя съобщения чрез използването на вложени блокове *съедини*.

топки за 4 с



🗹 Коледни топки

Допълнете проекта Заключен космически град в студиото към учебника, за да се изчислява броят monku, koumo ще изработи Роботко, като използвате знанията си за редицата на Фибоначи. Използвайте собствения блок фибоначи, който създадохте, и допълнете пропуските в koga.



Код на Роботко Проект Коледни топки когато 🎮 е щракнато промени костюм на костюм 1 -🗹 За по-въздействащ проект отиди до х: 35 у: -140 Използвайте блоковете за клонинги, за да визуализирате броя monku, koumo Роботко ще кажи Аз мога да създавам коледни топки! за 2 сек създаде. кажи Първия ден ще създам 1 топка за 2 Kog на Ball кажи На втория ден вече ще има 2 топки за 2 На третия - 3 топки за 2 кажи ато 💌 е щракна На четвъртия - 5 топки за кажи 2 ки се иди до х: 🚺 у: 🗐 На петия - 8 топки и така! за кажи Тази последователност се нарича Числата на Фибоначи за 2 cei кажи питай След колко дни ще се върнеш, за да вземеш топките?) и чака кажи съедини Създадох съедини резултат топки! 3a 2 сек до х: 🛛 избери случайно от 🔁 -230 до 🔁 -130 у: 🗍 и от -170 до 100 повтори 5 дефинирай фибоначи ден промени костюм на ляво 🖛 изчакай 0.1 1 чиспо1 • промени костюм на дясно 🕶 изчакай 0.1 се 2

Собствен блок фибоначи

14 ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

🗹 Заключен космически град

В студиото към учебника в проекта Заключен космически град, за да отключите вратата за града, е необходимо да намерите трицифреното число, скрито в игралното поле. Съберете цифрите на това число и Роботко ще провери дали сте въвели правилния код. Допълнете кода на Роботко, за да може той да пресмята сбора на цифрите на кода. Довършете кода, като запълните пропуските с необходимите блокове.

🗹 От къде да започнете

- 1. Добавете в проекта собствения блок сбор цифри трицифрено от Раницата.
- 2. Извикайте блока *сбор цифри трицифрено* с параметър променливата *kog*.
- Във вложените блокове съедини допълнете koga, за да каже Роботко колко е сборът от цифрите на koga.
- При проверката дали е верен въведеният код, допълнете с променлива, в която сте записали сбора на цифрите на кода.

Намерих заключен космически град!

Проект Заключен космически град

Скрито число





смени декора с Space City 1 • кажи Бравоі Ние сме вътреі за 2 сек наче кажи Кодът е грешені за 2 сек кажи Не ни пуснаха! за 2 сек

Забелязвате ли, че в решението е използвана абсолютно същата подпрограма *Сбор цифри-трицифрено*, която създадохте в първия проект?

Програмистите обичат да пишат код веднъж, а да го ползват много пъти. Това явление се нарича **многократна използваемост на kog**.

Както вече знаете, в Scratch готов код се пренася чрез *Раницата*. Използвайте я често! Onumaйте се да onucвате кода си maka, че да можете да го ползвате многократно.



📩 ТУК ЩЕ НАУЧИТЕ КАК:

 ga създавате сценарий на образователен проект по тема от предоставен списък;

• ga gekomnosupame сценария на програмно изпълними geūности с изучаван блоков език за програмиране.

🗹 Мисия 1

Роботко и приятелят му Тоботко искат да започнат да учат български език. Решили са, че е добре първо да разпознават кои думи са съществителни имена и кои са гла-голи. Помогнете им ученето да е и полезно, и приятно, като създадете игра, в която те да се научат!

Какъв ще бъде сценарият на тяхното обучение?

Къде ще бъде то? Как ще се провежда?

Сценарий на проект

При създаване на проект е важно да измислите сценарий за него. Сценарият описва основната идея на проекта, участниците и мястото, основните дейности.

Когато създавате сценария, си отговорете на следните въпроси:

- 1. Какво искате да направите игра или анимация?
- 2. Кои герои ще участват в проекта ви?
- 3. Къде ще се случват събитията?
- Ако сте избрали игра, ще трябва да отговорите и на следните въпроси:
- 1. Колко играча ще има играта?
- 2. Какви ще бъдат правилата на играта?
- 3. Как ще се управляват героите?
- 4. Каква е механиката на играта?
- 5. Как ще оценявате представянето на играчите?
- 6. Какви допълнителни обекти ще участват в играта?
- Какво ще бъде условието за край на играта? Ако решите да правите анимация, трябва да помислите върху следните въпроси:
- 1. Кои ще бъдат ключовите моменти в анимацията?
- 2. Ще има ли интерактивен елемент? Ако да, какъв ще бъде той?
- 3. Колко различни сцени ще има и какъв ще бъде преходът между тях?

Механика на игра

Игрите се състоят от различни игрови механики. Игровата механика е съвкупност от правила, по които играчът взаимодейства с играта.

Вече познавате игрови механики за: точки при победа, състезание между играчите, решаване на пъзел, случаен шанс, таймер за време и други.

🗹 Мисия 1 – възможна!

Примерен сценарий на играта за разпознаване на съществителни имена и глаголи може да е следният:

- 1. Действието да бъде на необитаема планета.
- 2. Участват два героя Роботко и Тоботко.
- Играта се играе от двама играчи, като единият управлява Роботко със стрелки наляво и надясно, а другият – Тоботко с клавишите А и D.
- Игровото поле да е разделено на две части, като от горната част ще падат думи.,
- 5. При докосване на дума от робот:
- ако думата е глагол, той получава точка;
- ако думата е съществително име, точка получава противникът му.
- 6. Играта приключва 60 секунди след стартирането.



🗹 Сценарии на образователни проекти

Предложете сценарий за образователен проект по някоя от следните теми:

- 1. Екология и грижа за околната среда
- 2. Математически зависимости
- 3. Географски места в България
- Исторически събития в историята на България
- 5. Експерименти по човекът и природата
- 6. Интерактивно представяне на творба, изучавана по литература.

🗹 Мисия 2

Вече имате готов сценарий на играта. За да създадете по-лесно играта, трябва да разделите сценария на отделни дейности, които да реализирате като подпрограми.

Декомпозиране на сценарий

Сценарият представлява решаване на сложна задача, която се състои от няколко по-малки и лесни задачи (подзадачи). Декомпозиране на сценария означава да го разделите на отделни независими една от друга подзадачи.

💾 Трик за професионалисти

При създаването на проект с език за блоково програмиране можете да разделите дейностите и на такива, свързани с визуалното оформление – дизайн, които включват избор (създаване) на герои, сцени, текстови съобщения, и на такива, свързани с функционалността на проекта – създаване на кода, проверка дали работи правилно, оформяне на еднотипни действия като подпрограми.



🗹 Разделяй и владей

Проучете какво представлява техниката *Разделяй и владей* и как можете да я приложите при създаването на вашия проект.

🗹 Мисия 2 – възможна!

Играта, която искате да създадете, може да се раздели на следните дейности:

- 1. Движение на героите наляво и надясно
- 2. Проверка за излизане на героя от обособеното поле
- 3. Падане на думите от горната част на сцената
- 4. Докосване на дума от робот
- Таймер за изтичане на време Всяка от тези дейности може да представите като подпрограма в Scratch и да създадете вашата игра.



🗹 Декомпозирайте вашия сценарий

След като сте измислили сценарий по тема от предложения списък, разделете го на подзадачи, които да създадете като подпрограми с изучаван език за блоково програмиране.

🗹 Допълнителни елементи към сценария

С какво може да допълните измисления от вас сценарий?

Какви ще бъдат допълнителните подзадачи, които ще трябва да направите?

СЪЗДАВАНЕ НА ПРОЕКТ СЪС СРЕДСТВАТА НА ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ – І ЧАСТ – СЦЕНАРИЙ И ДЕКОМПОЗИРАНЕ

📩 ТУК ЩЕ НАУЧИТЕ КАК:

 създаване на сценарий на образователен проект по тема от предоставен списък;

 gekomnosupaнe на сценарий на програмно изпълними geйности с изучаван блоков език за програмиране.

0 0

🗹 Грижа за дърво

Ще участвате в състезание по природни науки и трябва да направите проект, който илюстрира важността на светлината и водата за растенията. Измислете сценарий и го декомпозирайте на подзадачи, които да решите чрез подпрограми.

🗹 Откъде да започнете?

Припомнете си kou са основните въпроси, на koumo трябва да отговорите при създаването на сценария на проекта:

- Игра или анимация игровият елемент в проекта ще го направи по-интересен и по-привлекателен – можете да давате точки при поливане и достатъчно светлина и да отнемате при недостатъчно количество вода и светлина.
- Какви са правилата на играта опишете ги точно и ясно – целта на играта е възможно най-дълго време играчът да се грижи за дървото, без да го преполее или без то да изсъхне, като го полива и му осигурява необходимата светлина.
- Какви герои и сцени са необходими изберете или създайте подходящи герои и сцени. Можете да използвате примерния проект Грижа за дърво в студиото към учебника.



У Колко са героите в примерния проект?

Cmapmupaùme npoekma u omkpuùme koлko герои има.

🗹 Начално състояние на героите

Определете какво е началното състояние на героите при стартиране на играта – например чашата е пълна, облакът не закрива слънцето, дървото е зелено.

🗹 Променливи

Основните параметри – вода, светлина, закриване на слънцето, време на живот на дървото, зависят от действията на потребителя. За проследяване на стойностите им ще са необходими променливи. Определете колко променливи са необходими и дали се отнасят за всички герои, или само за някои.

🗹 Точки за поливане, светлина, време

- Определете начални стойности на точките при стартиране на нова игра.
- Задайте при какви стойности на точките за поливане и светлина дървото ще бъде преполято или ще изсъхне, ще изгори или му липсва светлина и играта ще свърши:
- светлина: nog 0 точки липса на светлина, над 100 точки прегаряне
- вода: nog 0 moчku изсъхване, над 100 moчku – преполиване



😫 Трик за професионалисти

Опишете какво прави всеки герой по време на играта и как неговите действия се отразяват на хода на играта. Помислете с какви блокове ще реализирате сценария му и ако е необходимо, помислете за кои действия трябва да дефинирате собствени блокове.

🗹 Сценарий на героя Sun

По време на играта слънцето е статично и не извършва никакви действия.

🗹 Сценарий на героя Cloud

При стартиране на играта облакът се намира на определено място и се прави проверка дали не е върху слънцето. Ако не е върху слънцето и потребителят го докосне с мишката, той се премества върху слънцето и се намаляват точките за светлина. След това се връща на първоначалното си място.

🗹 Сценарий на героя Glass Water

При стартиране на играта чашата е пълна и се намира на определено място. При влачене с мишката и докосване на дървото се имитира поливане, като водата в нея се излива. Точките за вода се увеличават с 10, чашата се връща на началното си място и отново е пълна.

🗹 Сценарий на героя Tree

Ако дървото не е полято (точките за вода ca nog 0), казва "Пресъхнах". Ако е преполяmo (точките са над 100), казва "Удавих се". Ако няма достатъчно светлина (точките за светлина са nog 0), казва "Нямам светлина". Ако има прекалено много светлина (точките са над 100), казва "Изгорях". При всички тези случаи дървото си сменя костюма с подходящ и играта свършва.





🗹 Проверка за събитие

Много често в програмирането се използва допълнителна променлива, чрез която се предава информация дали някакво събитие е настъпило или не. Променливата има само две стойности – "Да" или 1, ако събитието се е случило и "Не" или 0, ако не се е случило. Например в играта е важно да знаете дали потребителят "държи" чашата с вода, дали облакът е върху слънцето. Ако събитието не е настъпило, например потребителят не е натиснал и задържал основния бутон на мишката, променливата Държим чаша има стойност "Не". Ако събитието е настъпило, променливата приема стойност "Да".





Част от кода на героя Тгее





📩 ТУК ЩЕ НАУЧИТЕ:

- kak8o представлява алгоритъмът за броене на елементи;
- kakßo представлява алгоритъмът за размяна на стойности на две променливи;

 kak ga peaлusupame алгоритмите в изучаван език за блоково програмиране.

🗹 Мисия 1

Роботко и Тоботко искат да подобрят играта за разпознаване на частите на речта в проекта от Урок 5.3., като добавят и прилагателни имена. При докосване на прилагателно име двата робота искат да си разменят точките.

Те мислят да наложат и ограничение за време 30 секунди, за да подобрят обучението си.

Помогнете им да реализират идеите си.

Алгоритъм за броене

Често се налага да преброяваме елементи – брой точки, брой съществителни имена, брой изминали секунди или брой оставащи секунди, брой костюми на даден герой.

За да преброите елементи в компютърна програма, е необходимо да създадете променлива, в която да запазвате броя им. При преброяването стойността на променливата се увеличава с 1 или намалява с 1.

Важно е в началото да зададете стойност 0 на променливата, ако броите точки например. Ако броите оставащо време, началната стойност ще са секундите, които отброявате.

🗹 Алгоритъм за броене – словесно описание

Алгоритъмът за броене може да се опише по следния начин:

- Задавате начална стойност на променливата, в която ще запазвате броя: 0 – ако при броенето елементите се увеличават; първоначален брой елементи, ако намаляват.
- Проверявате за нов обект или настъпило събитие и увеличавате или намалявате променливата с 1.

🗹 Описание на алгоритми

Може да описвате алгоритмите, като използвате прости и ясни изречения за този, който ще изпълнява алгоритъма.

Добра практика е първо да напишете на лист алгоритъма, а след това да го пресъздадете в Scratch.

🗹 Алгоритмите са навсякъде около нас

Посочете примери за алгоритми от вашето ежедневие.

🗹 Мисия 1 – възможна!

За да зададете ограничено време за играта на Роботко и Тоботко, ще ви трябва брояч на времето.

- 1. Създайте променлива Време и в началото на играта ѝ задайте стойност 30.
- 2. В цикъл *повтаряй докато* времето стане равно на 0, задайте изчакване 1 секунда чрез блока *изчакай* и намалете променливата *Време* с 1.



😫 Трик за професионалисти

След kamo onuweme алгоритъма за броене на секунди, може да го обособите в собствен блок и към него да създадете текстов герой *Брояч*-, koù-

то да запазите в *Раницата*, за да използвате брояча в бъдещи проекти.





🗹 Ах, че беля ...

Роботко пие космически сок от звезди в сива чаша, а Тоботко пие сок от астероиди в синя чаша, но Сферичка объркала чашите, докато наливала сока. Как можете да размените съдържанието в чашите, така че всеки робот да пие любимия си сок в любимата си чаша?

 Изсипвате сока от астероиди от сивата чаша в допълнителната чаша 3. Изсипвате сока от астероиди от допълнителната чаша в синята чаша



Алгоритъм за размяна на стойности на две променливи

Променливите могат да запазват само една стойност в даден момент. Съответно, ако присвоите на променлива нова стойност, тя заменя старата.

За да размените съдържанието в чашите, използвахте допълнителна чаша. По същия начин, ако искате да размените стойностите на две променливи – Променлива1 и Променлива2, ще ви е необходима трета помощна променлива – Допълнителна променлива.

🗹 Алгоритъм за размяна на стойности – словесно описание

Дадени са Променлива1 и Променлива2. За размяна на стойностите им:

- 1. Създайте допълнителна променлива Допълнителна променлива.
- 2. В Допълнителна променлива запазете стойността на Променлива1.
- 3. В Променлива1 запазете стойността на Променлива2.
- 4. В *Променлива2* konupaùme cmoùнocmma на *Допълнителна променлива*.

🗹 Графично представяне на алгоритми

Може да илюстрирате алгоритмите, като използвате графични изображения и схеми.



🗹 Мисия 2 — възможна!

За да реализирате идеята на Роботко и Тоботко – при докосване на прилагателни имена да си разменят точките, ще използвате алгоритъма за размяна на стойности. Създайте допълнителна променлива Допълнителна променлива.

В нея запазете точките на Роботко. Задайте като точки на Роботко тези на Тоботко, а като точки на Тоботко запишете стойността на Допълнителна променлива.

направи	Допълнителна про	оменл	тива 👻 на	Точ	ки Роботко
направи	Точки Роботко 💌		Точки Тобот	гко	a
направи	Точки Тоботко 🗢		Допълнител	на п	роменлива

🗹 Брояч+

Създайте брояч на секунди, чрез който да засичате кой от двама играчи е по-бърз. Стойност? Оформете кода като собствен блок.

🗹 Алгоритми за бъдеща употреба

Запазвайте в *Раницата* вече създадени важни алгоритми, като броячи и размяна на стойности, за да ги използвате в други проекти.

🗹 Алгоритми за проект

Помислете как може да допълните сценария на вашия проект, избран в Урок 5.3., за да използвате изучените алгоритми за броене и размяна на стойности.

СЪЗДАВАНЕ НА ПРОЕКТ СЪС СРЕДСТВАТА НА ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ – II ЧАСТ – АЛГОРИТМИ ЗА БРОЕНЕ НА ЕЛЕМЕНТИ И ЗА РАЗМЯНА НА СТОЙНОСТИ

📩 ТУК ЩЕ УСЪВЪРШЕНСТВАТЕ УМЕНИЯТА СИ ЗА:

прилагане на алгоритъма за броене на елементи;

 прилагане на алгоритъма за размяна на стойности на gße променливи.

🗹 Нови костюми

След посещението на Заключения космически град Роботко има нови костюми, но не знае колко точно са. Помогнете му, като го научите да брои колко костюма има. Отворете проект Костюми на Роботко и

отворете проект *костюми на Роботко* и gonълнете koga на Роботко, kamo създадете алгоритъм за броене на kocmlomume. Проверете колко kocmloma има Роботко и сравнете с броя, получен от алгоритъма. Получава ли се същото число?

😫 Трик за професионалисти

Помислете колко променливи ще са необходими, за да преброите костюмите.

Ще ви трябват ли помощни променливи, за да запазите информация?

Къде ще съхранявате новата информация? Какви начални стойности ще зададете на променливите – обърнете внимание, че спрайтът има поне един костюм.

Създайте променливите предварително, за да може да ги използвате при описанието на алгоритъма.

Анализ на задачата

Ще са ви необходими две променливи: Брой костюми;

Първи костюм.

В променливата *Брой костюми* ще запазвате броя на костюмите, които има Роботко, а в променливата *Първи костюм* ще запазвате името на костюма, от който сте започнали. За да запазите името на първия костюм, може да използвате блока костюм име от групата *Външност*.

За да преброите костюмите, ще е необходимо да преминете през всеки от тях и да увеличавате броя им, запазен в променливата Брой костюми.

🗹 Припомнете си..

По какъв начин подпрограмата може да предаде стойност към останалата част от koga?

дефинирай Броене на костюми
направи Първи костюм – на костюм име –
направи Брой костюми 🗸 на 🚹
следващ костюм
повтаряй докато костюм име 🔹 = Първи костюм
промени Брой костюми 🔻 с 🚹
следващ костюм
изчакай 0.5 сек

🗹 Подобрение на "Заключен космически град"

В предходните уроци допълнихте проекта Заключен космически град, който се намира в студиото към учебника.

Отворете този проект и го променете, така че спрайтът Код автоматично в началото на играта да преброи колко костюма има и да използва това число при избирането на случаен костюм.

🗹 Проект "Грижа за дърво"

Открийте в проекта Грижа за дърво от Урок 5.3. Създаване на сценарий и декомпозиране – упражнение, части от koga, koumo представляват преброяване на елементи.

Създайте собствена реализация на алгоритъма за броене на елементи.

УПРАЖНЕНИЕ

🗹 Роботко и Тоботко създават костюми

Роботко и Тоботко искат да се научат да създават нови костюми, като разменят части от своите костюми. Помогнете им, като допълните проекта *Комбинация на костюми* в студиото към учебника.

Анализ на задачата

За да разменяте частите на дадени обекти, е необходимо те да бъдат отделени в самостоятелни спрайтове. Лесно може да направите това, като копирате първоначалния спрайт и изтриете ненужните елементи. За да размените частите на двата робота, е необходимо да запазите първоначалните места на всяка от тях за всеки робот. За тази цел ще ви трябват няколко променливи:

ТялоРоботко, КракаРоботко, РъцеРоботко; ТялоТоботко, КракаТоботко, РъцеТоботко. Във всяка от променливите трябва да запазите първоначалната позиция на съответната част, а при натискане на някоя от частите на робота трябва да приложите алгоритъма за размяна на стойности за частите на Роботко и Тоботко.



😫 Трик за професионалисти

Може да запазите позицията на даден обект при стартиране на играта или текуща позиция при изпълнение на играта, като използвате блоковете *х позиция и у позиция от групата Движение*.

Необходимо е да запазите стойностите от тези блокове в променливи.

Създайте променливи с подходящи имена – например Начална Х позиция.

😫 Трик за професионалисти

Може да използвате наготово еднотипен код на спрайт за друг спрайт, като провлачите кода на първия спрайт върху втория спрайт в областта за спрайтове и след това редактирате кода, ако се налага.





📩 ТУК ЩЕ НАУЧИТЕ:

 ga прилагате алгоритъм за намиране на максимален елемент от три елемента;

 ga прилагате алгоритъм за намиране на минимален елемент от три елемента.

🗹 Мисия

Роботко и Тоботко искат да разберат кой най-добре се е научил да разпознава думи. Затова те искат да допълнят своя код така, че да запомнят най-добрия и най-слабия резултат до момента.

Помогнете им, като измислите алгоритъм за решаване на техния проблем!

Алгоритъм за намиране на максимален елемент от три елемента

За решаване на някои задачи е необходимо намирането на най-голям (максимален) елемент. Това може да е най-високата цена на даден продукт, най-големият балон, най-добрата оценка или най-високият ученик в класа.

Най-големият от три елемента – *A*, *Б* и *B*, може да се намери с помощта на допълнителна променлива – *Най-голям елемент*.

Предполагате, че най-големият елемент е първият – *A*, и запазвате стойността му в променливата *Най-голям елемент*. Последователно проверявате дали *Б* или *B* са по-големи от *Най-голям елемент*. Ако за някой от тях това е така, запазвате стойността му в *Най-голям елемент*.

Алгоритъмът за намиране на максимален елемент може да се опише по следния начин:

Имате три елемента А, Б и В.

- 1. Създайте помощна променлива Най-голям елемент
- 2. Запазете стойността на А в променливата Най-голям елемент
- 3. Ако *Б* е по-голям от *Най-голям елемент*, запазете стойността на *Б* в *Най-голям елемент*.
- 4. Ако *B* е по-голям от *Най-голям елемент*, запазете стойността на *B* в *Най-голям елемент*.

А > Б За най-голям елемент



За най-малък елемент



🗹 Алгоритъм за намиране на най-малък елемент

Ако в алгоритъма за най-голям елемент замените всички проверки за по-голямо (>) с проверка за по-малко (<), ще получите алгоритъма за намиране на най-малък елемент.



🗹 Мисия – възможна!

За да разберете кой е по-добрият – Роботко или Тоботко, е необходимо да използвате алгоритъма за намиране на максимален елемент.

Създайте собствен блок, който да намира най-голям елемент от три зададени.

Създайте променлива Най-добър резултат и намерете максималния елемент от Най-добър резултат, Точки Роботко и Точки Тоботко.

Може да запазите най-добрия резултат в облачна променлива, за да откриете кой е най-добрият от различните опити!



🗹 Грижа за дървото

Отворете проект Грижа за дървото от стуguomo към учебника. Допълнете кода така, че да намирате най-добрия и най-слабия резулmam.

🗹 Да бъдеш Scratcher

Kamo cu създадете акаунт в онлайн платформата на Scratch, първоначално сте нов Scratcher.

Ако потвърдите имейл адреса си, работите усърдно и споделяте проекти, може да станете пълноправен Scratcher.

Когато сте готови да станете Scratcher, ще получите съобщение в Scratch, за да станете такъв!

Ако cme Scratcher, получавате допълнителни възможности, като използване на облачни променливи, създаване на собствени студиа и комуникация с други като вас.

🗹 Облачни променливи

Понякога е необходимо да се предава информация между потребители. Например това може да е най-добрият резултат досега в играта или най-краткото време за стигане до края на играта. Променливите в Scratch запазват информацията, която искате, но данните в променливите се запазват само на вашия компютър. Ако друг потребител отвори проекта ви, той ще има друга стойност в променливите.

За да съхранявате информация, която всеки nompeбител на вашия проект може да вижда, е необходимо да използвате облачни променливи.

Облачните променливи се създават като обикновените, но е необходимо при създаването да изберете опцията *Облачна променлива (съхраняване на сървър)*. След което информацията от тази променлива ще се запази на сървър на Scratch и всеки, който отвори вашия проект, ще вижда записаните данни.

🗹 Описание на алгоритми

Опишете чрез средствата на Scratch алгоритмите за намиране на максимален и минимален от три зададени елемента. Оформете koga kamo собствен блок.



СЪЗДАВАНЕ НА ПРОЕКТ СЪС СРЕДСТВАТА НА ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ – III ЧАСТ – АЛГОРИТМИ ЗА НАМИРАНЕ НА МАКСИМАЛЕН И МИНИМАЛЕН ЕЛЕМЕНТ ОТ ТРИ ЕЛЕМЕНТА

📩 ЩЕ УСЪВЪРШЕНСТВАТЕ УМЕНИЯТА СИ ЗА:

- прилагане на алгоритъма за максимален елемент от три елемента.
- прилагане на алгоритъм за минимален елемент от три елемента.



🗹 Музикални роботи

Роботко има три помощника – Червен, Зелен и Лилав робот. Те могат да запомнят една нота. Роботко иска да се научи да намира кой от роботите е запаметил най-високия тон и кой е запаметил най-ниския тон.

Отворете проекта *Музикални роботи* от студиото към учебника и допълнете кода на Роботко така, че да намира най-малка стойност и най-голяма стойност.

Оформете алгоритмите като собствени блокове.

Допълнете koga на Роботко, kamo използвате създадените собствени блокове и ги приложете, за да намерите нотите с най-висок и най-нисък тон.





🗹 Блокове за музика

Освен блоковете *Молив* за рисуване, в Scratch може да намерите и блокове за *Музика*, които ви позволяват да създавате собствени мелоguu, kamo използвате ноти.

За да добавите блоковете от категория *Музика*, от долния ляв ъгъл на работното поле натиснете бутон *Добавяне на разширение* и изберете *Музика*. В менюто с групите блокове най-отдолу се появява икона Най-често използваните блокове от категорията Музика са:

- Изсвири барабан изпълнява тон с избран ударен инструмент
- Изсвири нота издава тон с избрана височина
- Направи инструмент на променя инструмента, с който се изпълняват нотите, при блока Изсвири нота
- Направи темпо задава скоростта на изсвирване на нотите
- Увеличи темпо променя скоростта на изсвирване на нотите

🗹 Зарадвайте роботите

Допълнете koga на роботите maka, че те ga реагират, ako нотата, koято е записана при тях, е най-ниският тон или най-високият тон.

И Други алгоритми за намиране на максимален и минимален елемент

В алгоритмите за намиране на максимален и минимален елемент измежду три дадени елемента използвахте допълнителна променлива.

Възможно ли е да намерите кой елемент е най-голям и кой е най-малък, без да използвате допълнителна променлива?

Помислете как може да реализирате такъв алгоритъм със средствата на Scratch.

Оформете кода като собствен блок.

Кога според вас е възможно да прилагате този алгоритъм?



🗹 Класация в игра

За да направите класация на най-добрите потребители на вашия проект, е необходимо да използвате облачни променливи.

Създайте толкова облачни променливи, колкото места от класирането искате да запазите, и променяйте техните стойности, когато играта приключи.

Внимавайте за количеството облачни променливи. В един проект не може да има повече от 10 облачни променливи.

🗹 Грижа за дървото

Отворете проекта *Грижа за дърво* от студиото към учебника.

Допълнете koga на спрайт Tree, maka, че ga се запазва резултатът на играча.

Създайте облачни променливи – Първо място и Второ място.

Създайте собствен блок, който да намира най-голям резултат от три дадени.

Като използвате създадения блок, намерете коя от променливите Първо място, Второ място и Време има най-голямата стойност. Добавете код, който да променя променливите Първо място и Второ място спрямо това какъв резултат е получил играчът (стойността на променливата Време).





📩 ТУК ЩЕ НАУЧИТЕ:

 ga прилагате алгоритъм за подреждане на три елемента по големина.



🗹 Мисия

При пътешествията си из космоса Роботко откри три планети и се замисли коя от планетите е най-голямата и коя е най-малката. Интересно – коя е средната по размер? Помогнете му, като опишете алгоритъм, който да подрежда планетите по големина.

Алгоритъм за подреждане на три елемента по големина

Понякога е необходимо да подреждате елементи по някакъв признак. Може да подреждате ученици по височина, продукти в магазина по цена, участници в състезания спрямо техните резултати или играчите спрямо точките, които са събрали.

За решаване на такива задачи ще използвате алгоритъм за подреждане на три елемента по големина.

Когато подреждате три елемента по големина, ще ви трябват променливи, които да отговарят за *най-малкия*, *средния* и *най-големия* елемент.

За да подредите променливите, е необходимо да изпълните следните стъпки:

1. Запазете стойностите на вашите елементи съответно в променливи Най-малък елемент, Среден елемент и Най-голям елемент.

2. Проверете дали *Най-малкият елемент* е по-голям от *Средния елемент*. Ако е така, разменете стойностите им.

3. Проверете дали *Средният елемент* е по-голям от *Най-големия елемент*. Ако е така, разменете стойностите им.

4. Отново проверете дали *Най-малкият* елемент е по-голям от *Средния елемент*. Ако е така, разменете стойностите им.

След като изпълните трите стъпки, елементите са подредени – стойността на всеки елемент е на правилното място.

🗹 Наблюдателни ли сте?

Забелязахте ли, че Стъпки 2. и 4. са еднакви? Разгледайте схемата и обсъдете:

- Защо се налага Стъпка 4. да повтори Стъпка 2.?
- В каква ситуация при Стъпка 4. не се прави размяна на стойностите?



🗹 Подреждане в нарастващ и намаляващ peg

Ако замените всички проверки за по-голямо (>) с проверка за по-малко (<), алгоритъмът ще подрежда елементите в намаляващ ред – в първата променлива ще е най-големият, а в последната – най-малкият от тях.



За нарастващ ред

За намаляващ ред

🗹 Мисия 1 – възможна!

Допълнете кода на Роботко, така че да подрежда планетите по големина.

- 1. Отворете проекта Роботко и планети от студиото към учебника.
- 2. Създайте нов блок Подреждане на елементи с три параметъра – Елемент1, Елемент2, Елемент3.
- 3. Създайте три променливи, които важат за всички спрайтове – Най-малък елемент, Среден елемент, Най-голям елемент.
- 4. Запазете в тях стойностите на параметрите.
- 5. Опишете чрез блоковете на Scratch алгоритъма за подреждане по големина за създадените от вас променливи.
- 6. Допълнете кода на Роботко така, че алгоритъмът да се изпълнява за променливите, в които са записани размерите на планетите – Планета1 размер, Планета2 размер, ПланетаЗ размер.

🗹 Подреждане на планетите

Може да използвате съобщения, за да поста-

ирай Подреждане на елементи Елемент1 Елемент2 Елемент3 аправи Най-малък елемент 💌 на Елемент1



🗹 Описание на алгоритми

Запишете в Раницата блока за подреждане, който създадохте.

🗳 ЕКСПЕРИМЕНТИРАЙТЕ .

Работи ли алгоритъмът за подреждане, ако елементите не са числа, а думи?



СЪЗДАВАНЕ НА ПРОЕКТ С ЕЗИК ЗА БЛОКОВО ПРОГРАМИРАНЕ – V ЧАСТ – ПРЕДСТАВЯНЕ НА ЦЯЛОСТНА ПРОГРАМА ЗА РЕАЛИЗИРАНЕ НА ОБРАЗОВАТЕЛЕН ПРОЕКТ

📩 ТУК ЩЕ НАУЧИТЕ:

 kak ga създавате собствен образователен проект;
 kak ga представяте цялостна програма за реализиране на образователен проект.



🗹 Мисия 1

Разгледахте процеса на създаване на различни елементи на един проект. Време е да създадете собствен образователен проект!

🗹 Идеи за проекти

Може да изберете да създадете един от следните проекти:

- 1. Грижа за дърво
- 2. Състезание Кой е най-бърз?
- 3. Килим от шевици
- 4. Викторина по математика
- 5. Пъзел географски места в България
- 6. Подреди историческите събития
- 7. Експерименти по човекът и природата
- 8. Интерактивно представяне на избрана от вас творба по литература.

или измислете собствена тема на проekm.

🗹 Дейности при създаване на проект

Припомнете си на какви видове дейности може да разделите процеса на създаване на проект.



🗹 Откъде да започнете?

Когато създавате сценарий на проект, отговорете на следните въпроси:

- Какво ще създавате игра или анимация?
- Kou ще са героите в проекта ви?
- Къде ще е действието на проекта ви?

🗹 При създаване на анимация

Когато създавате анимация, трябва да решите:

- Кои ще са ключовите моменти във вашата анимация?
- Ще добавяте ли интерактивен елемент?
 Ако да какъв ще бъде той?
- Колко различни сцени ще има и кога ще се преминава от една сцена към друга?

🗹 При създаване на игра

Когато създавате игра, трябва да решите:

- Колко играча ще играят?
- Какви ще бъдат правилата на играта?
- Как ще се управляват героите?
- Какви механики на играта ще включите?
- Как ще оценявате играчите?
- Ще имате ли допълнителни обекти, различни от героите?
- Какво ще е условието за край на играта?

🗹 Механики за проекта ви

Припомнете си какво е механика на игра. Изберете какви механики ще има вашата игра.

🗹 Избор на механика на игра

Едни от често срещаните механики в игрите ca:

- Получаване на точки
- Състезание между играчи
- Решаване на пъзели
- Случайни събития
- Таймер за време
- Игра в отбор
- Нива на сложност.

Помислете и за други механики, koumo ga добавите към играта, за да стане уникална и интересна.



🗹 Мисия — възможна! - Сценарий

Помогнете на Роботко да разбере коя от неговите нови ракети е най-бърза, като създадете проект *Състезание – Кой е най-бърз?*.

Отговорете на въпросите за сценария:

- 1. Създавате анимация.
- 2. Героите са Роботко, три ракети и финиш, който ще бъде планета.
- 3. Действието е в космоса.
- 4. Основните моменти са състезанието на ракетите и подреждането им в класация.
- 5. Интерактивността на анимацията се състои в това, че потребителят определя кога започва състезанието.
- При започване на състезанието всяка ракета се движи със случайно генерирана скорост.
- Анимацията се състои от две сцени състезание и класация. Преминаването от едната към другата ще бъде веднага след приключване на състезанието.

🗹 Мисия – възможна! - Герои

Вече имате готов сценарий на анимацията, но ви трябват героите. За всеки герой трябва да създадете неговия външен вид. Единият от героите е Роботко и може да използвате създадения спрайт от урок 5.1. *Създаване на графични изображения с изучаван език за блоково програмиране.* За другия герой – *Планетата*, до която ще е необходимо да стигнат космическите кораби на Роботко, може да използвате една от планетите от проекта *Роботко и планети* от студиото към учебника.

За космическите кораби може да използвате спрайта *Rocketship* от колекцията със спрайтове на Scratch.

Създайте три копия на спрайта Rocketship и направете всяко от тях да бъде с различен цвят.

🗹 Създаване на герои

Припомнете си как може да създавате собствени герои.

Обмислете kakви герои ще участват във вашия проект. Създайте ги.



🗹 Трикове за професионалисти

Ако създавате нови уникални герои за вашите проекти, направете скица на всеки герой на хартия и помислете:

- 1. Чрез kakвu геометрични фигури може ga се изчертае?
- 2. Как са разположени тези геометрични фигури?

Друг вариант е да използвате готовите геpou om konekцията или да ги редактирате. Ako pewume да редактирате героите от konekцията, прегледайте от kakвu елементи се състоят, и ги редактирайте, kamo променяте размера, формата, цвета или изтриете ненужни елементи.



🗹 Зачитайте авторските права

Важно е, когато използвате чужд код или спрайт във вашите проекти, да благодарите на автора за използваните от вас елементи. Това може да направите в секцията *Бележки и кредити* на страницата на проекта ви.

🗹 Декомпозиция на сценария

Припомнете си kakßo означава gekoмпозиция на сценарий.

След като сте измислили сценарий на избрания от вас проект, разделете сценария на подзадачи и помислете дали вече сте създавали някои от тях като подпрограми, които може да използвате.

🗹 Създаване на проекта

Като използвате средствата на блоковия език за програмиране Scratch, създайте функционалната част на проекта си. Помислете кои части от кода може да бъдат представени като собствени блокове.

🗹 Мисия – възможна! - Подзадачи

Вече имате готов сценарий на анимацията и сте избрали героите. За да създадете по-лесно анимацията, разделете сценария на отделни дейности, а повтарящите се части обособете като подпрограми. Анимацията може да се раздели на следните подзадачи:

- 1. Начални думи на Роботко
- 2. Задаване на скорост на космическите кораби на случаен принцип
- 3. Стартиране на състезанието чрез натискане върху Роботко
- 4. Движение на корабите до планетата
- 5. Запазване на времето за пристигане на всяка ракета
- Подреждане на ракетите от най-бавна към най-бърза.

🗹 Алгоритми в проекта

Помислете kou om изучаваните алгоритми може да използвате. Кой om тях в коя подзадача ще използвате?

🗹 Използвайте раницата

По време на създаване на проектите от предходните уроци сте добавяли различни спрайтове и скриптове в *Раницата*. Използвайте ги във вашия проект.

🗹 Променливи

Помислете kakвu променливи ще са ви необходими.

Каква информация ще искате да запазите? Към kou спрайтове ще се отнасят променливите?

Ще има ли променливи, които се отнасят за всички спрайтове, или облачни променливи?



🗹 Таймер за отчитане на време

Използвайте блок таймер, за да отчетете времето, за което корабите стигат до планетата.

Когато състезанието започва, запазвате текущото време, а след това, когато някой кораб стигне до планетата, неговото време е таймер (текущо време) – Начало (таймерът при започване).



🗹 Как да създадете "Пъзел – Географски места в България"?

Примерен сценарий за Пъзел – Географски места в България:

- 1. Проектът е игра.
- 2. Играчът е само един.
- На полето е представена карта на България с отбелязани точки за поставяне на различни географски обекти.
- 4. Около картата са разположени спрайтове, koumo nokaзват различните географски обекти в България.
- Когато курсорът на мишката се доближи до спрайт с обект, този спрайт започва да следва курсора на мишката.
- При доближаване на спрайт с географски обект до точка на картата се проверява дали е разположен правилно или грешно.
- 6.1. Ако географският обект е правилно разположен, той остава на това място.
- 6.2. Ако географският обект не е на правилното място, той се връща в начална позиция.
- Играта приключва, когато са поставени всички обекти.
- 8. Записва се времето, за което играчът е поставил обектите.
- 9. Записва се времето на най-бързия играч.



🗹 Реализация на примерните идеи за проекти

Примерни варианти на цялостни програми на предложените идеи за образователни проекти може да разгледате в студиото към учебника.

🗹 Kak ga създадете "Килим от шевици"?

Примерен сценарий за Килим от шевици:

- 1. Проектът е анимация.
- Анимацията се състои от gße сцени сцена за избор на шевица или мотив и сцена за изобразяване на шевицата или мотива.
- 3. На първата сцена са представени различни видове шевици.
- При натискане на шевица се преминава към сцена за изобразяване на килим от избраната шевица.
- 5. В сцената за изобразяване на килима той се създава с помощта на блокове от групата *Молив* и се разказва за особеностите на конкретната шевица.
- След като приключи представянето на шевицата, се преминава отново към сцената за избор на шевица.
- На първоначалната сцена има бутон, при чието натискане се създава килим от всички шевици.



🗹 Представяне на проект

Време е да покажете резултата от вашата работа! Представете проекта си пред вашите съученици:

- Разкажете им накратко каква е идеята му, кои от героите са ваши авторски, използвали ли сте собствени блокове, кои от изучените алгоритми сте приложили.
- 2. Демонстрирайте как работи вашият проект.

🗹 Взаимно оценяване

Оценете проектите на вашите съученици спрямо следните критерии:

- 1. Оригинална идея
- Дизайн създадени са собствени герои, подбрани са подходящи сцени
- Функционалност проектът работи вярно, използвани са собствени блокове, приложени са някои от изучените алгоритми.

EUROPEAN EDUCATIONAL PUBLISHERS GROUP

Издателска група

ПРОСВЕТА е член на Асоциацията на европейските издатели на учебници

veta.bg e

www.prosveta.bg www.e-uchebnik.bg www.e-prosveta.bg