

Компютърно моделиране

Методическо ръководство за учителя

Съдържание

Методически указания по теми и уроци	2
Урок 1. Дигитални устройства	2
Урок 2. Дигитална самоличност	5
Урок 3. Последователни действия	9
Урок 4. Блокове за движение Алгоритъм и код	13
Урок 5. Стъпка по стъпка. Подреди дума	17
Урок 6. Стъпка по стъпка. Открий път	18
Урок 7. Повтарящи се действия. Цикъл	19
Урок 8. Цикъл. Робот-художник	21
Урок 9. Знаем и мога. Последователни и повтарящи се действия	22
Урок 10. Знаем и мога. Диагностика	23
Урок 11. Среда за програмиране Scratch. Мисия „Пустиня“	23
Урок 12. Движение на героя	28
Урок 13. Завъртане и чертане. Да спасим принцесата	31
Урок 14. Позициониране и преместване. Скрый рибката	34
Урок 15. Цикли в Scratch. Космическо пътешествие	36
Урок 16. Позициониране, движение, чертане. Знаем и мога (обобщение)	38
Урок 17. Сцена и декори. Кой къде живее	38
Урок 18. Герои и костюми. Зимни игри	41
Урок 19. Герои и костюми. Да направим снежен човек	46
Урок 20. Озвучаване на проекта. Гласът на животните	47
Урок 21. Героят мисли. Въвеждане на текст. Делфинът риба ли е?	49
Урок 22. Героите говорят. Разговор за делфините	51
Урок 23. Диалог между повече герои. Още за делфините	52
Урок 24-25 Работа по проект. Разказ за цветята	54
Урок 26-27. Компютърна анимация. Изгубеното мече	55
Урок 28-29. Работа по проект. Пролетни игри	57
Урок 30-31. Знаем и мога	57
Урок 32. Преговор и обобщение	57

Методически указания по теми и уроци

Урок 1. Дигитални устройства

Урок за нови знания

Цел. Формиране на знания за дигитални устройства

Задачи:

1. Да се запознае с основни компоненти на дигитално устройство.
2. Да се формират знания за управление на дигитално устройство.
3. Да се формират знания за здравословно и безопасно използване на дигитални устройства

Очаквани резултати:

- Познава основните компоненти на дигиталните устройства – стационарни и мобилни.
- Знае как да въвежда информация в съответното дигитално устройство.
- Знае къде да следи за изходна информация на съответното дигитално устройство, както и на включени към него изходни устройства.
- Умее да започва и приключва работа с конкретно устройство.
- Знае, че устройството може да изпълнява различни задачи след задаване на точни и ясни команди.
- Знае, че чрез устройството може да се извършват определени дейности само ако е инсталирана съответна програма.
- Познава основните проблеми, свързани със здравословното състояние на човека, използващ дигитални устройства.
- Прилага правила за предотвратяване на физическа преумора и травми след продължителна работа в дигитална среда.
- Организира правилно компютърното си оборудване (добра светлина на екрана, добре поставени входно и изходно устройство, добро разположение на стол и бюро, безопасно ниво на звук).

Учебно съдържание:

Дигитално устройство е всяко устройство, което може да бъде програмирано да извършва обработка на информация, подадена в цифров (дигитален) вид. Основните компоненти на едно дигитално устройство са централният процесор (обобщено понятие, за аритметично-логическо, управляващо устройство и памет) и входно-изходните устройства.

Дигитални устройства са персоналният компютър в неговите най-разнообразни форми (стационарна компютърна конфигурация, преносим компютър – лаптоп, таблет, смарт-телефон и др.), както и различни вградени компютърни системи в различни автоматизирани устройства (перални, кухненски уреди, банкомати, детски играчки и др.)

За да работим с дигитално устройство, трябва да разполагаме с входно и изходно устройство. В основния случай входните устройства са клавиатура и мишка, а изходното – компютърният екран (монитор). При стационарен компютър тези компоненти са ясно обособени като отделни технически средства. При мобилните устройства те са интегрирани в общия дизайн на дигиталното устройство. При лаптопът клавиатурата е в

едно и също тяло с централния процесор, а екранът е вграден в капака на преносимия компютър. Мишката е реализирана чрез вграден до клавиатурата тъч-пад. При таблета и смарт-телефона липсва физическо устройство – клавиатура. Възможността за въвеждане на символи е реализирана програмно. Екранът е вграден в основното тяло на устройството. Мишката е заменена с тъч-технология – управление на програмата и въвеждане на текст става чрез пръст на ръката или чрез електронна писалка.

Централният процес при стационарната компютърна конфигурация е разположен в компютърната кутия. При преносимия компютър, таблетите, смарт-телефоните и техните конфигурации централният процесор е част от основния корпус на дигиталното устройство. Размерите на всеки от основните компоненти намалява с появата на нови дигитални устройства.

Вградените дигитални устройства могат да се видят в огромно разнообразие от уреди, които ни заобикалят в ежедневието. При всички тях има вид клавиатура (с клавиши или сензорни бутони), както и екран, на който се визуализира зададената команда и резултатът от нейното изпълнение.

Всяко дигитално устройство има процедура за включване и процедура за изключване. За работа с дигиталното устройство трябва да има инсталирана операционна система, която управлява връзката между потребителя и инсталираните програми (приложения). За да се изпълни някаква задача на дигиталното устройство, на него трябва да има инсталирана съответна компютърна програма, която да даде възможност задачата да бъде изпълнена.

Методически насоки за хода на урока

Чрез темата „Дигитални устройства“, като първи урок от годишното разпределение по предмета „Компютърно моделиране“, учителят въвежда учениците в предмета и ги мотивира за работа през учебната година.

Учениците живеят в общество, определено от развитието на дигиталните технологии и са представители на дигиталното поколение деца. За тях дигиталните устройства са предмети от ежедневието. На база натрупан до момента житейски опит, учениците разпознават различни дигитални устройства и имат знания и умения за използване на някои от тях.

Целта на урока е да се систематизират знания и да се формират понятия за дигитално устройство (компютър, таблет, смарт-телефон), за основните части (процесор, входно-изходни устройства), за програма, за видове информация (текст, графика, звук, видео и др.).

Урокът може да започне с беседа, в рамките на която учениците да представят своите знания по темата – какви устройства разпознават, какви са основните им части, как се включват и изключват и т.н.

Вниманието на учениците се насочва към машините и тяхното предназначение. Чрез машината човекът може да свърши неща, които сам не може да направи. Например, да премести огромни предмети, да вдигне тежачи тонове неща, да лети в небето и в космоса, да плува под водата и т.н. Някои от машините се управляват от човека. В ежедневието учениците виждат, например, колите и управляващите ги шофьори. Самолетът е машина, която се управлява от пилот. Корабът – от капитан и т.н. Прахосмукачката се управлява от човека.

Има и машини, които нямат нужда от управление. Те са създадени да правят едно и също нещо постоянно или при натискане на някакъв бутон. Например, светофарът работи автоматично, сменяйки последователно цветовете си. Чрез натискане на бутон пускаме

различни домакински уреди като кафе-машината, микровълновата фурна, пералнята, прахосмукачката-робот и др.

Дигитално устройство е всяка електронна изчислителна машина с автоматично управление, която може да обработва информация в цифров (дигитален) вид. Работата на дигиталното устройство се управлява от компютърна програма. Програмата е поредица от команди, написани на език за програмиране.

Дават се примери на дигитални устройства – настолен компютър, лаптоп, таблет, смарт-телефон. Уточняват се понятия за основните елементи на една настолна компютърна конфигурация – компютърна кутия, в която е разположен процесорът на компютъра, клавиатура, монитор, мишка. Компютърната конфигурация наричаме компютър. Компютърът е дигитално устройство.

Обръща се внимание на основните компоненти на дигиталното устройство – процесорът и входно-изходните устройства. Мозъкът на дигиталното устройство е процесорът. Това е изчислителното устройство на машината. При настолна конфигурация процесорът е разположен в компютърната кутия, докато при лаптопът и мобилните устройства, процесорът е вграден в корпусът на устройството. Размерите на дигиталните устройства стават все по-малки. Размерът на процесорите – също намалява, но с развитие на технологиите процесорите запазват и развиват мощността си независимо от намаляването на размерите им.

Вниманието се насочва към входно-изходните устройства. Чрез клавиатурата и екрана „общуваме“ с дигиталното устройство. Устройствата, чрез които получаваме информация от машината се наричат „изходни устройства“. Устройствата, чрез които ние даваме информация на машината са „входни устройства“. Клавиатурата е входно устройство. Екранът е изходно устройство. Назовават се разнообразни периферни устройства и всяко се определя като входно или изходно устройство. Входните устройства са мишка, клавиатура, микрофон, Изходни са – принтер, монитор, проектор, тонколони, слушалки.

Дигиталното устройство преработва информация, подадена в цифров вид, но за потребителя информацията е под формата на текст, графични изображения, звук, видео и др.

Основните функции на дигиталното устройство е да изчислява, анализира, намира, съхранява и забавлява. Чрез подходящи илюстрации се дискутира всяка една от функциите на машината.

Поставя се акцент на поведението в компютърния кабинет. От една страна следва да се спазват правила за безопасно и здравословно поведение. От друга – да се съхранява техниката и да не се нарушават лични права. Вниманието на учениците се насочва към проблеми като опасност за увреждане на зрението, предпоставка за гръбначни изкривявания, заседнал начин на живот, намаляване на живата комуникация между хората и др. Акцентира се върху нуждата от висока дигитална култура на поведени в съвременното ежедневие.

Урок 2. Дигитална самоличност

Урок за нови знания

Цел. Формиране на знания за дигитална идентичност

Задачи:

1. Да се формира знание за потребителски профил.
2. Да влиза с потребителско име и парола в среда за управление на учебното съдържание
3. Да разбира приликите и разликите между дигитална и физическа идентичност.

Очаквани резултати:

- Създава личен аватар в среда за управление на учебния процес.
- Работи във виртуална среда.
- Прави разлика между дигитална и физическа идентичност.
- Познава основните заплахи в дигитална среда.
- Прилага правила за реакция при заплахи в дигитална среда.
- Знае, че не трябва да предоставя лична информация при комуникация или работа във виртуална среда.

Учебно съдържание:

Дигиталната идентичност е представянето на индивида в различни софтуерни платформи, предимно онлайн, чрез потребителско име и парола. Дигиталното представяне може да включва и аватар – графично изображение, избрано от потребителя, с което той се представя в съответното компютърно приложение или интернет услуга или платформа.

Създаването и избора на аватар е възможност за безопасно и творческо присъствие на ученика в дигиталния свят, което му позволява спокойно да изразява себе си и да изучава себе си.

От друга страна дигиталната идентичност крие редица рискове. Чрез аватара природно срамежливи деца могат да добият по-голяма увереност и да разкрият своя потенциал онлайн и да продължават да бъдат срамежливи и несигурни в реалното общуване. При други деца могат да се отключат асоциални или разрушителни модели на поведение в мрежата или офлайн. Други има опасност да изберат идентичност, която да ги злепостави и да им донесе неприятни последици в реалния живот.

В рамките на беседа в този урок и в хода на цялостното обучение по предмета Компютърно моделиране, учителят трябва да работи активно, за да помогне на учениците да формират гражданска позиция по следните въпроси и проблеми:

- Да разбират приликите и разликите от начина си на представяне онлайн и офлайн.
- Да осъзнават възможността в интернет да бъдеш анонимен и че тази анонимност крие рискове за измама и заблуда, и как това би могло да формира различно поведение онлайн и офлайн.
- Да разбират последиците, до които може да се стигне, когато дигиталната идентичност се отличава в голяма степен от физическата идентичност.

При създаване на онлайн аватари учениците трябва да осъзнават важността на начина си на представяне, да проявяват уважение към другите в дигиталната среда, да мислят за репутацията в сегашно и бъдещо време.

За целите на обучението по Компютърно моделиране е особено подходящо използването на средата code.org.

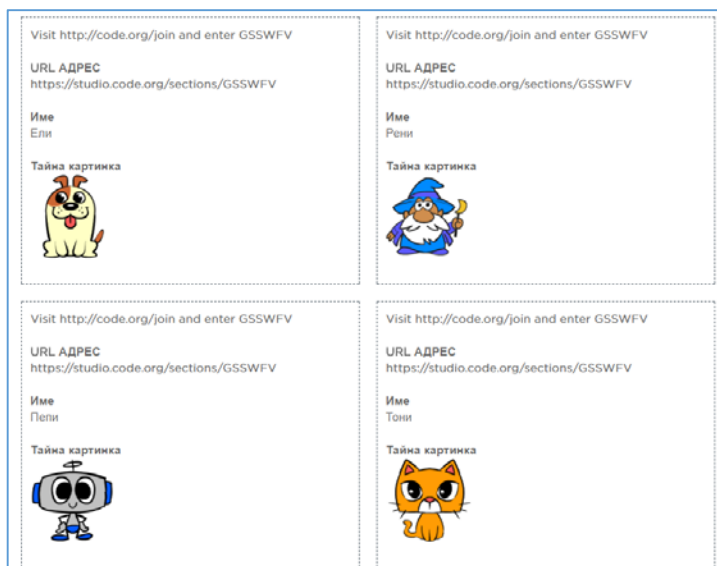
Code.org е нестопанска организация, чиято визия е, че всеки ученик във всяко училище трябва да има възможност да изучава компютърни науки, както изучаването на математика, биология, физика и др. Повече за организацията може да се прочете на адрес: <https://code.org/about>

В сайта на code.org учителят създава своя регистрация с роля „учител“. Регистрацията е без такси.

За по-ефективна работа в хода на обучението по Компютърно моделиране се препоръчва учителят да създаде свой клас в платформата (*Моето табло/Създаване на раздел или секция*). При щракване върху бутона „Създаване на раздел/секция“ се отваря екран, от който учителят трябва да избере по какъв начин учениците ще достъпват учебното съдържание. Препоръчваме работа с потребителско име и картина за парола (картина за вход). На следващата стъпка учителят въвежда име на раздела – например, „Компютърно моделиране 3А клас“; избира клас, например „3“; избира курс – „Курс 1“. Запазва промените.

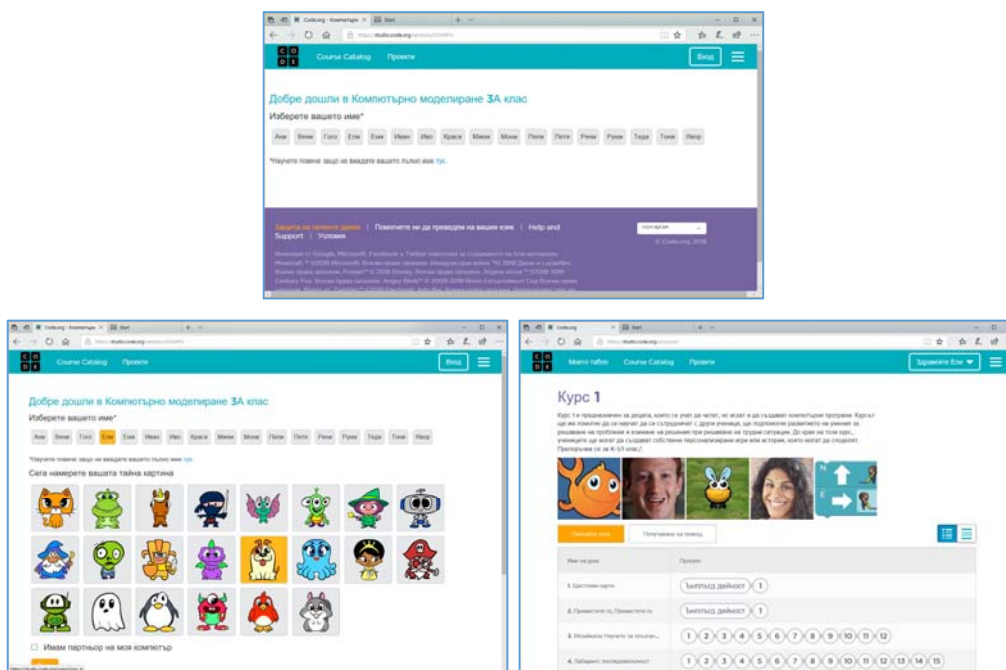
Следва добавяне на ученици към този раздел. Учителят може да въведе само първото име и инициал, ако е необходимо за всеки от учениците в класа. Това ще бъде тяхното потребителско име. Паролата е в картинен вид и се генерира автоматично от платформата.

След като учителят създаде своя клас и въведе потребителските имена за учениците си може да сподели връзка за достъп. Платформата предлага възможност за разпечатване на карти за достъп за всеки ученик. Тези карти се генерират автоматично. При щракване върху връзката „Отпечатайте карта с информация за влизане в сайта на вашите ученици“ се генерират съответните карти, съдържащи връзката, потребителското име и паролата (картинната парола) за всеки от учениците.



Така учениците получават предварително създадени акаунти за системата. Отделният ученик трябва да стартира програма-браузер и да въведе в адресното поле дадения URL адрес. За да влезе в своя профил, ученикът избира своето име (това е неговото потребителско име) и щраква върху своята картинна парола (отпечатаната на картата му

картинка). Така всеки ученик работи в своя профил, като следва своето темпо на работа. Учителят може да проследи индивидуалният напредък на всеки от учениците в класа.



Методически насоки

Урокът е подходящо да протече като беседа, в която основната информация да бъде предоставена от учениците. Целта на учителя е да предразположи децата да споделят своя опит до момента от общуване в дигитална среда – какви аватари са използвали и използват, в какви среди, има ли неписани правила, според които по аватарите се ориентират за физическата идентичност на реалния човек; има ли случки и казуси, които са им интересни, които са ги притеснили, които искат да споделят и т.н. В хода на тази дискуссия учителят има възможност да изведе положителни и отрицателни модели на поведение, които да бъдат анализирани.

Поставени в естествена среда на диалог, учениците могат сами да направят за себе си изводи и да коригират модели на поведение, които до този момент не са определяли като потенциално опасни или неприемливи от морална или етична гледна точка.

Вниманието на учениците се насочва към правилата за поведение в Интернет. Дискутират се опасности, култура на поведение, основни правила. В учебника са маркирани дадените твърдения като Вярно и Грешно. Електронният учебник съдържа интерактивни задачи за фронтална работа за формиране на знания за безопасно поведение в Интернет.

Учителят може да предложи на учениците да изиграят една игра, с която се цели на практика да се покаже как зад даден аватар може да стои съвсем различна личност и трябва да се спазват правилата за безопасност.

Игра „Кой съм аз“

Всеки ученик записва на лист кратко описание на себе си, което не е точно. Описва външния си вид, като променя един-два елемента – цвят на очи, коса, възраст, занимания, хоби. След като са готови, всички листчета се събират в кутия и учителят на случаен

принцип избира лист и чете. Този, който е писал описанието, няма право да се издава. Останалите ученици се опитват да познаят кой е автор на текста.

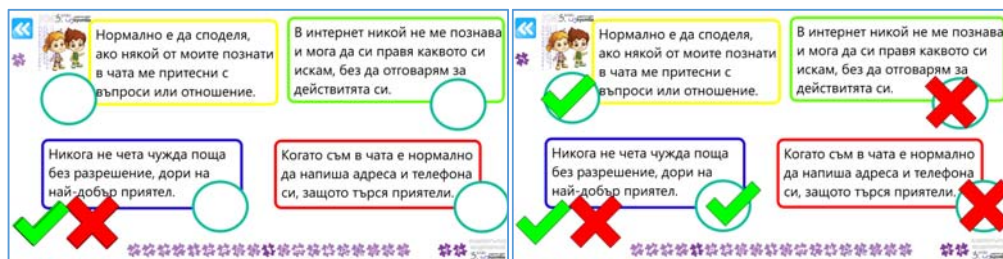
В края на играта ще се отчете, че познатите автори са много малко или няма такива и ще се направи извод, че за да използваме безопасно интернет е задължително да се спазват определените правила.

В края на часа, ако учителят е създал своя клас в платформата на code.org може да раздаде карти за достъп и да покаже на учениците начина на влизане с потребителско име и картинна парола в системата.

Друг вариант е, работата с платформата code.org да стартира следващия учебен час, а в този учениците да направя свои карти за вход, съдържащи име и аватар. Ако учителят постави такава задача, той/тя може да използва измислените от учениците потребителски имена при създаване на класа си в платформата на code.org. Така вместо да влизат с истинските си имена, учениците ще използват зададените от тях псевдоними.

Е-учебник

Е-учебникът съдържа ресурси към урока. Чрез инструмента „Избери“ при шракване върху лупите могат да се визуализират на цял екран избрани акценти от учебното съдържание. За затвърждаване на знанията за правилата на поведение в интернет са разработени интерактивни задачи, при които учениците трябва да маркират като правилни и неправилни модели на поведение по конкретни казуси.



Урок 3. Последователни действия

Урок за нови знания

Цел. Формиране на знания за конструиране на последователни действия чрез подреждане на блокове за оформяне на изображения.

Задачи:

1. Да се формират знания и умения за подреждане на части от хартиен пъзел по указана последователност.
2. Да се формират умения за извършване на действията плъзгане и спускане.
3. Да се формират знания за понятието „блок“

Очаквани резултати:

- Подрежда части от изображение с хартиен пъзел.
- Познава работното поле и се ориентира в конкретна визуална среда.
- Знае как да подрежда блокове в указана последователност във визуална среда, като извършва действията плъзгане и спускане.
- Подрежда части от цяло изображение във визуална среда.

Учебно съдържание:

За реализирането на даден проект в реалния или дигиталния свят е важно да се разработи точен алгоритъм. Целта е да се даде последователност от действия, след изпълнението на които да достигнем до желаната цел. В един алгоритъм има последователни действия, има повтарящи се действия, има и действия, които зависят от някакво условие. Пренесено в науката Информатика, говорим за програма и команди (оператори). Командите, които се изпълняват в една програма могат да бъдат прости команди, водещи до изпълнение на едно действие, команди за цикъл, чрез които се реализират повтарящи се действия и условни оператори – команди, които се изпълняват по различен начин, в зависимост от това дали дадено условие е изпълнено или не.

Първа стъпка във формирането на алгоритмични умения е използването на команди за последователни действия за изпълнението на просто-дефинирана задача.

За създаването на компютърни програми се използват различни езици за програмиране със сложен синтаксис и семантика. За формиране на начални знания и умения за програмиране в малката възрастова група – начално и основно училище – се използват така наречените блокови среди за програмиране, при които командите се „сглобяват“ с помощта на градивни „блокове“ – отделни команди. Работи се в специално разработени среди, където потребителят, респективно – ученикът, може да подреди своя код.

По този начин акцентът се поставя върху изграждането на умения за логическо и алгоритмично мислене, а не върху заучаването на конкретен език за програмиране с характерните за него синтаксис и семантика.

В урока са поставени два акцента:

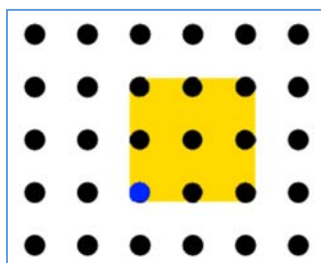
1. Формиране на знания и умения за конструиране (създаване) на краен продукт или постигане на поставена цел чрез използване на последователни действия.
2. Формиране на знания и умения за работа с блокове (плъзгане и спускане) във визуална среда.

За формиране на знания за последователни действия се използват задачи, при които с помощта на отделни елементи от хартия учениците сглобяват изображение (пъзел), следвайки дадени от учителя указания. За обезпечаване на учебния процес са подготвени

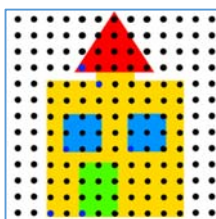
ресурси за изрязване (в електронен вид, както и в допълнителната папка за ученика). Учениците използват квадратна мрежа от точки и набор от геометрични фигури. Учителят дава указания, а учениците подреждат фигурите. Вниманието на учениците се насочва към важността на това да зададеш точна позиция на фигурите.

В конкретната задача се дефинира команда за разполагане на фигура в мрежата от точки.

Командата: „На ред 2, точка 3 постави фигурата“ означава „в указаната точка в мрежата постави синия връх на фигурата. Разположи фигурата вдясно и нагоре спрямо точката“. Редовете се броят от долу нагоре, а колоните (точките) – от ляво надясно.



След изпълнение на командите, дадени в учебника, учениците трябва да получат следната геометрична композиция:



По втория акцент в урока е предвидена работа в платформата Code.ORG. <https://code.org/>.

В блоковите среди за програмиране програмата се подрежда от блокове чрез използване на действията влачене и пускане (drag and drop)¹.

Блок наричаме картинното представяне на съответната команда от езика за програмиране. Блокът може да се избере, да се постави на работното поле и да се мести чрез влачене. Блоковете се подреждат един до друг. Обикновено визуалните среди за програмиране предлагат дизайн на блоковете, който подсказва дали един блок може да се съчетае с друг. Ако двата блока не могат да се „сглобят“ означава, че не е възможно логически да бъдат две последователни команди от съответната компютърна програма.

Първият урок по темата „Конструиране на последователни действия“ цели да се формират умения у децата да осъществяват действията щракване, влачене и пускане с компютърна мишка или с пръст, или писалка върху сензорен екран. Сглобяването на компютърния код в блоковите среди за програмиране наподобява работата с конструктор или подреждане на пъзел. Отделните команди на кода (програмата) се нареждат като картинки една под друга. Съществуват и блокове (за командите за цикъл), които обхващат няколко блока и ги обособяват в група (тялото на цикъла). За формиране на умения за работа с блокове може да се използват първите задачи от Курс 1 в Code.ORG.

¹ Като стил на изложение, в съответствие с приетата от МОН учебна програма по Компютърно моделиране за 3 клас, авторите приемат терминологията „плъзгане“ и „пускане“. Тези термини ще бъдат използвани по-нататък в настоящото методическо ръководство.

Методически насоки

В началото на урока се прави преговор на наученото в предходния час по Компютърно моделиране. Учителят може да провери формираните знания чрез задаване на въпроси от вида:

- Кои са основните компоненти на дигиталните устройства – стационарни и мобилни?
- Как се въвежда информация в съответното дигитално устройство?
- Къде се следи за изходна информация на съответното дигитално устройство, както и на включени към него изходни устройства?
- Как се започва и приключва работа с конкретно устройство?
- Какво е потребителско име, парола, аватар, дигитална самоличност?

Вниманието на учениците се насочва към диалога между героите от учебника – Боби и Ани. Поставя се темата на урока – създаване на алгоритъм за подреждане на дадено изображение чрез последователни действия.

Вниманието на учениците се насочва към важността на ясното и точно задаване на командите, които трябва да се изпълнят една след друга. Може да се разиграе следната игра: всеки ученик получава празен лист формат А4. Учителят дава инструкции за прегъване и откъсване на част от листа, които всеки ученик изпълнява, без да задава допълнителни въпроси. В края на играта всички ученици разгъват своите листове и се сравняват получените резултати.

Примерни указания: сгъни листа на 2; откъсни горния десен връх на листа; Прегъни листа на две. Откъсни долния ляв връх на листа. Прегъни листа на 2 части. Откъсни горния десен връх на листа. Разгъни листа.

След разгъване на листовите ще се установи, че има голямо разнообразие на крайните резултати. Различният краен резултат се дължи на неточно поставените указания – сгъването може да стане по ширина или по дължина; не е уточнено как трябва да се ориентира листа, за да се определят позициите горе-долу и ляво-дясно; не е уточнено колко голяма част от листа трябва да бъде откъсната.

Преминава се към работа с дидактичните материали, които трябва да са предварително изрязани и готови за използване по време на часа.

Уточнява се начина на изпълнение на командите, дадени от учителя. Положението на фигурата се определя от позицията на нейния връх, маркиран със синя точка.

Учителят диктува последователност от действия, а учениците подреждат фигурите върху своите мрежи.

Следващата задача на учениците е сами да предложат последователност от действия, за да получат като композиция, геометричните композиции, дадена в учебника. Дава се време на работа. След това един ученик диктува, а учителят изпълнява. При неточности се коментира допуснатата грешка.

За онагледяване на процеса на създаване на алгоритъм с последователни действия са разработени интерактивни задачи в електронния учебник.

Преминава се към работа с визуалната среда Code.org. Учителят прави демонстрация чрез упражненията от Курс 1. Въвежда понятието „блок“. Отделните елементи, от които подреждаме картинката се наричат блокове. Цялата картинка може да бъде един блок, но може да бъде „нарязана“ на няколко блока, които трябва да се „сглобят“, за да се получи цялото изображение.

Въвеждат се понятията „плъзгане“ и „спускане“. Основните действия с мишка, които учениците изпълняват са щракване, влачене и пускане. Добре е да се използва българска терминология вместо навлезлите чуждици „кликваме“, „драгваме“ и т.н.

Всеки ученик влиза в платформата, проследявайки връзката, дадена от учителя (*връзката към класа на учителят в code.org се съхранява като любима страница в използвания браузер, за да се спести време от въвеждане на URL адрес всеки час*). Всеки ученик влиза в своя профил и започва работа по задачите от Курс 1, Урок 3 Мозайка.

В края на урока се прави обобщение на наученото. Акцентира се на използваните термини и понятия – последователни действия, блок, плъзгане, спускане.

E-учебник

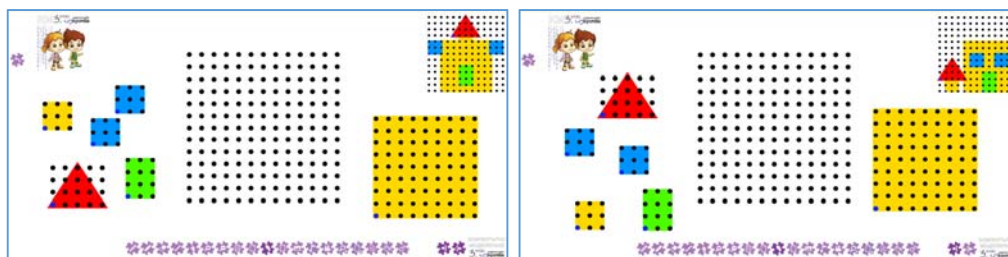
Електронното съдържание към урока позволява да се визуализират на цял екран избрани акценти – дефиниции, илюстрации, правила.



Задачите за подреждане на изображения от дадени геометрични фигури са обезпечени с интерактивни ресурси. По първата задача учителят или избран ученик може да подреди на интерактивната дъска изображението по дадените в учебника последователни команди.



Към задача 2 са предложени интерактивни ресурси, с елементи като елементите от хартия, с които учениците работят самостоятелно. Чрез последователното подреждане на елементите се описват и стъпките в последователния алгоритъм. Учениците могат да направят алгоритми, които се различават в поредността на стъпките. Чрез елементите могат да тестват верността на описаните алгоритми.



Урок 4. Блокове за движение Алгоритъм и код.

Урок за нови знания

Цел. Формиране на знания за конструиране на последователни действия чрез използване на блокове за движение на обект в квадратна мрежа.

Задачи:

1. Да се формират знания и умения за задаване на движение в квадратна мрежа чрез подреждане на блокове за движение в четирите посоки.
2. Да се формират знания и умения за описване на движение в мрежа чрез блокове за движение.

Очаквани резултати:

- Представа движението на обект в мрежа (по мозайка) чрез блокове.
- Определя вида на блоковете.
- Открива грешки в готов код.

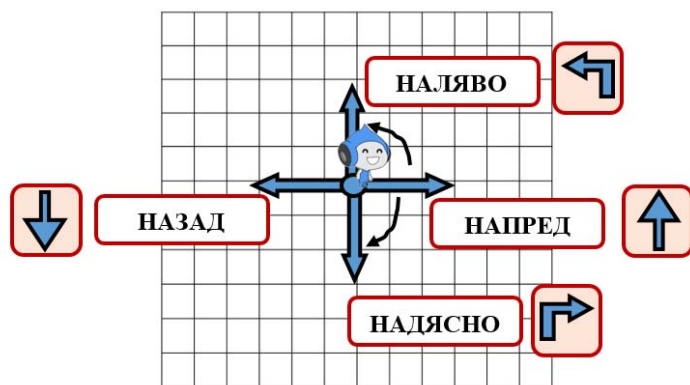
Учебно съдържание:

Затвърдяването на знания за последователни действия може да се реализира чрез задачи за движение в мрежа. Обектът се движи в четири основни посоки, минавайки през съседни квадратчета. Не може да се движи по диагонал.

За дефиниране на движение в мрежа се използват различни системи. Няма единна терминология, която да се използва винаги при движение в мрежа. Според средата и според задачата се дефинират командите, чрез които става движението в квадратна мрежа.

В настоящият учебник авторите въвеждат три системи за движение в квадратна мрежа, които условно ще наречем система „Напред-назад“, „На изток – на запад“ и „Надясно – наляво“ (според командите за движение по хоризонтална ос).

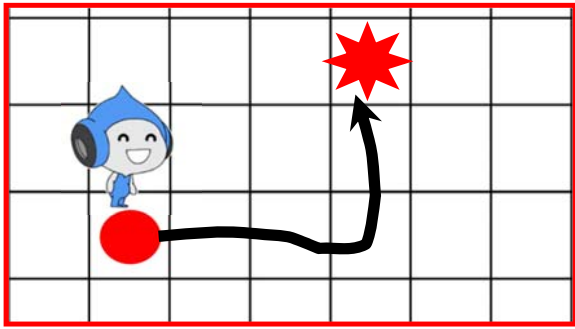
Системата „Напред-Назад“



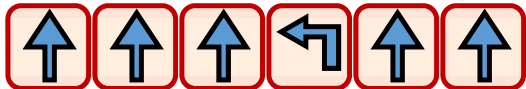
Командите за движение Напред и Назад предизвикват изместване на обекта в пространството с една стъпка. Отправната точка е изнесена върху героя, поставен в точката на движение.

Завъртането на обекта е на място. То не променя позицията на героя в разстояние – само в посока.

Пример 1: Пико трябва да стигне от точката до звездата по очертания маршрут.

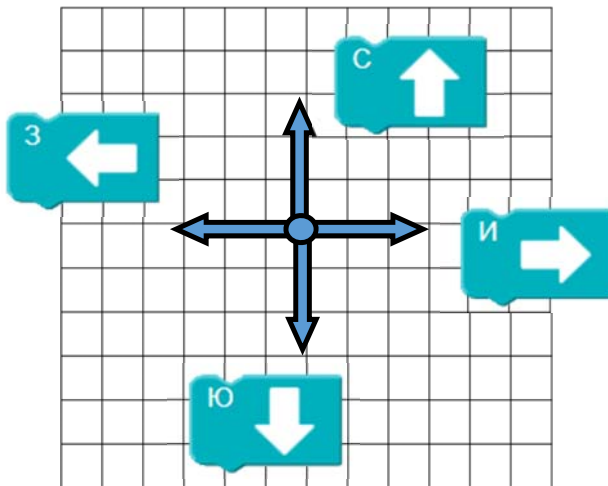


Командите, които трябва да изпълни според системата „Напред-Назад“ са:



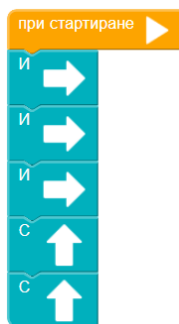
Система „Изток – запад“

В тази система се използват географските посоки – изток, запад, север, юг. Отправната точка е у човека, който гледа фронтално квадратната мрежа.

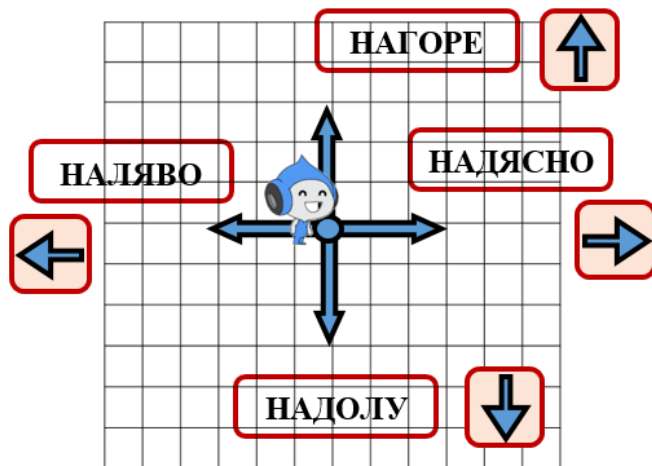


Тази система на движение се използва в платформата Code.org. При задаване на всяка от командите се прави по една стъпка в съответната посока. Тук завъртането и движението са интегрирани в една команда.

Решението на задачата от пример 1 е:

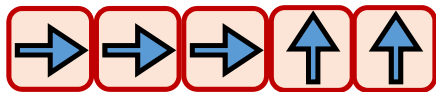


Система „Надясно – наляво“



Тази система се използва в средата Scratch, с която учениците ще се запознаят в хода на обучение по Компютърно моделиране. Отправната точка е изнесена върху героя, поставен в точката на движение.

Решението на задачата от пример 1 е:



Методически насоки

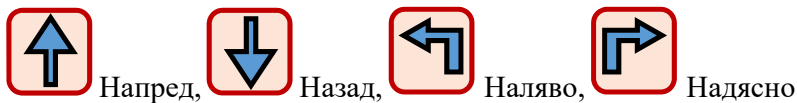
В урокът продължава темата за последователните действия. Нов акцент е формирането на знания за движение в мрежа в четирите посоки.

Въвежда се системата „Напред – Назад“. Използват се командите за преместване – „Напред“ и „Назад“ и командите за завъртане – „Наляво“ и „Надясно“.

Чрез директна демонстрация и повторение учителят затвърдява значението на командите за учениците. Разглежда се конкретен пример – изписването на буква В в мрежа.

Въвежда се понятието „код“. Програмен код е записването на алгоритъма със средствата на конкретен език за програмиране. В случая учениците записват своя алгоритъм, използвайки блокове от команди.

На учениците се предлага работа по двойки. Учениците се разделят по двойки. Единият влиза в ролята на програмист, а другият – в ролята на робот. Двойката разполага с маркер, хартия на квадратчета и картончета с команди



Всяка двойка избира една буква. Записва я в мрежа. След това програмистът трябва да програмира изписването на буквата в мрежата, като използва дадените команди за движение. След като програмата е записана, роботът изпълнява последователно

Урок 5. Стъпка по стъпка. Подреди дума

Упражнение

Цел. Затвърждаване на знанията за конструиране на последователни действия чрез подреждане на блокове за движение на обект в квадратна мрежа.

Задачи:

1. Да се затвърдят знания и умения за описване на движение в мрежа чрез блокове за движение.
2. Да се развият уменията за създаване на последователен алгоритъм чрез задачи за описване на дума в мрежа от букви.

Очаквани резултати:

- Самостоятелно създава последователен алгоритъм за описване на дума.
- Спазва правописни правила при работа с текст.
- Посочва правилно подреждане на блокове за постигане на формулирана цел.

Методически насоки

Урокът продължава темата за конструиране на последователни алгоритми. Знанията и уменията на учениците се надграждат със задачи за откриване на дума в мрежа от букви. Задачите развиват комбинативно мислене, както и езиковите умения.

В хода на фронтална работа учителят и учениците заедно откриват дума в лабиринта и разглеждат алгоритъм, описан с блокове за движение, който описва нейното изписване. Въведен е герой – пчела – който при движението си изписва думата, подреждайки една до друга буквите, през които минава.

На учениците се поставя самостоятелна индивидуална работа да открият сами дума в дадената в учебника им таблица и да конструират алгоритъм за нейното изписване. За целта използват комплекта от командни блокове за движение (*командните блокове за подреждане на алгоритъма трябва да бъдат предварително изрязани и подготвени за работа*).

За затвърдяване на знанията на учениците за създаване на последователен алгоритъм може да се постави екипна задача.

Екипна задача (Игра-състезание): Класът се разделя на екипи (примерно, 4 екипа). Всеки екип получава дума. Учениците трябва да съставят алгоритъм за нейното изписване и да го опишат чрез програмен код от блокови команди. Когато са готови, екипът избира един робот, чрез който да тества верността на алгоритъма. Роботът на групата започва да изпълнява командите, като чертае с маркер върху хартия с квадратчета подадените му команди. Екипът, който пръв изпише правилно думата печели състезанието. Примерна дума: „robot“ или „РОБОТ“.

За проверка на знанията на учениците са предвидени две тестови задачи за индивидуално изпълнение.

В първата задача е дадена мрежа от букви и код. От учениците се иска да открият пътя на движение и да намерят думата в мрежата, разположена по този път. Те трябва да определят и началната позиция на пчелата. Описаната дума е „програмист“. В мрежата има много думи, което позволява да се поставят допълнителни задачи от учителя, както

и задачата учениците сами да съставят свои задачи и да ги поставят за решаване един на друг.

Предвидена е работа в средата *code.org* за затвърждаване на знанията за последователни действия и движение в квадратна мрежа. Подходящи задачи за изпълнение са упражнения 14 и 15 от Урок 4. По преценка на учителят всеки ученик може да продължи работата си по упражнения в *Code.org*, които са започнати в предходните часове.

В края на урока се прави обобщение на свършената работа. Активизира се употребата на въведените понятия – последователни действия, алгоритъм, блок, код.

Урок 6. Стъпка по стъпка. Открий път

Упражнение

Цел. Затвърждаване на знания за комбинирано използване на командни блокове за движение.

Задачи:

1. Да се затвърдят знания и умения за описване на движение в мрежа чрез блокове за движение.
2. Да се развиват уменията за избор на команди за движение според поставената задача.
3. Да се коригира грешка в команден код.

Очаквани резултати:

- Познава различните блокове за движение
- Умее да подрежда блокове в указана последователност
- Придвижва обект до крайната цел

Учебно съдържание

Урокът предвижда затвърждаване и преговор на вече въведените командни бутони за движение. Акцентира се върху възможността за използване на различни команди за движение в мрежа.

Ако обектът, който трябва а придвижваме е ориентиран (примерно, пчела или герой с лице и гръб, разположени с лице в една от посоките) използваме системата от команди за движение „Напред – Назад“. В този случай командата за завъртане не измества обекта от неговата позиция, а само променя посоката му.

Ако обектът, който трябва да придвижваме в мрежа не е ориентиран (например, пешка) използваме системата „Изток – запад“. Всяка от командите измества обекта с една стъпка в указаната посока. Отправната точка на движение е у играча, а не в героя.

Така се развиват уменията на учениците да могат да избират различни системи за движение в квадратна мрежа според задачата, която им е поставена.

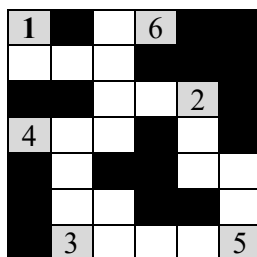
Методически насоки

В началото на урока в хода на фронтална работа учителят обобщава използваните до момента команди за движение – системите „Напред – назад“ и „Изток – запад“. Коментират се основните разлики между тях и се прави извод кога коя система е подходяща за използване.

Предлага се игра на зарчета.

Задача (игра) Вземете зарче и пионка. Хвърлете зарчето веднъж и поставете пионката на полето на падналото се число. Хвърлете зара втори път – това е числото, до което пешката трябва да се достигне, минавайки през лабиринта. Изберете команди и опишете алгоритъма за достигане на целта. (Тъй като пешката не е ориентиран обект – няма лице и гръб – трябва да се използва системата „Изток – запад“).

Същата игра може да се изиграе с герой Пчела. В този случай пътят на пчелата се програмира с командите от системата „Напред – назад“. Тук могат да се използват и двете системи с блокове за движение.



Предвидена е работа във визуалната среда – Курс 1, Уроци 4 и 5. Учителят избира конкретни упражнения за изпълнение, за да се затвърдят знанията и уменията за работа във визуална среда, както и знанията и уменията за откриване и коригиране на грешка в даден код.

В края на урока може да се реализира игра – Робот – златотърсач.

Игра. Изберете един ученик за „робот“. Помолете го да излезе от стаята или да е занимава с нещо друго. Скрийте предмет някъде в класната стая и заедно с останалите ученици създайте алгоритъм за достигане от вратата до търсения предмет. Извикайте „Робота“ и нека той изпълнява точно създадения алгоритъм. Например: „Две крачки напред, завий наляво, една крачка напред, завий надясно и т.н. Ако „роботът“ не достигне желаната цел, коментирайте грешките в алгоритъма.

В края на урока се прави обобщение на свършената работа. Дискутират се въпроси от вида:

- Какво представляват последователните действия?
- Какво е значението на алгоритъма за достигането на целта?
- Какво ще се случи, ако се промени последователността от действията?
- Как откриваме и отстраняваме грешки в алгоритъма?

Урок 7. Повтарящи се действия. Цикъл

Урок за нови знания

Цел. Формиране на знания за повтарящи се действия. Пропедевтика на понятието „цикъл“.

Задачи:

1. Да се формира понятие за „цикъл“.
2. Да се формират знания и умения за изпълнение на цикличен алгоритъм по зададена схема.
3. Да се формират знания за създаване на цикличен алгоритъм по дадено задание.

4. Да се формират знания и умения за използване на блок за повторение във визуална среда.

Очаквани резултати:

- Дава примери за повтарящи се действия от ежедневието.
- Изпълнява в мрежа зададен чрез схема цикличен алгоритъм.
- Създава цикличен алгоритъм по дадено графично задание.
- Определя колко пъти трябва да се повтори дадена последователност
- Определя последователността, която трябва да се повтори, за да се моделира действие във визуална среда.
- Във визуална среда сглобява крайна последователност от блокове, реализиращи цикличен алгоритъм.

Учебно съдържание:

Цикълът представлява многократно повторение на поредица от команди (операции, действия). За всеки цикъл трябва да има начална позиция, от която стартира, както и условие за край. Цикълът може да спре след точно зададен брой повторения или след като е изпълнено някакво условие.

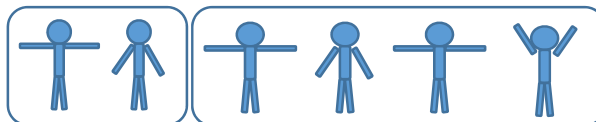
В курса по Компютърно моделиране в трети клас се работи с блокове за повторение, без да се използват условни оператори за цикъл. Целта е да се реализира пропедевтика на понятието цикъл, без излишно обременяване на учениците с понятия и характеристики.

За изясняване на понятието „цикъл“ се стъпва на познати за детето явления от заобикалящия го свят – смяната на деня и нощта, работата с календар – месеци, сезони и др.

Методически насоки

Урокът може да започне с фронтална работа по предварително подготвени нагледни материали, илюстриращи споменатите по-горе циклични явления в природата, познати на учениците – смяна на деня и нощта, работа с календар, фазите на деня и др.

С помощта на схеми от вида:



могат да се реализират циклично повтарящи се действия с целия клас. Учителят показва схема, учениците я изпълняват прави от местата си.

Всеки ученик рисува свой комплекс за утринна гимнастика в учебника, използвайки основните графични елементи. По двойки учениците изиграват своите комплекси, като повтарят направената редица 4 пъти. Това е цикъл с 4 повторения.

Чрез беседа с учениците учителят въвежда понятието ЦИКЪЛ. Повтарянето на едни и същи действия се нарича цикъл. Цикълът може да се повтаря постоянно или да се повтаря даден брой пъти. Примери за постоянно повтарящи се действия в природата има много – циклично се сменят:

- Деня и нощта (при обикаляне на Земята и около оста ѝ)
- Сезоните (при обикаляне на Земята около слънцето)
- Фазите на луната (при обикаляне на луната и около земята) и др.

Изиграните от децата комплекси в началото на часа са пример за цикъл с краен брой повторения. Те са повторили по 4 пъти движенията от дадената схема, след което са спрели.

Следващ акцент в урока е кодът, в който се използва блок за цикъл.

Учителят и учениците проследяват резултата при изпълнение на алгоритмите, зададени чрез КОД 3 и КОД 4 – и двата кода водят до изчертаване на квадрат. Изводът е, че един и същи алгоритъм може да се реализира с различни кодове. Въвежда се блокът за цикъл. В него са важни два параметъра – да се посочи колко пъти трябва да се повтори действието и кои команди трябва да се повторят. Прави се извод, че чрез блок за цикъл алгоритъмът става по-кратък и по-разбираем.

Вниманието на учениците се насочва към задача, в която по очертан път трябва да създадат цикличен алгоритъм. Чрез диалога между героите в учебника се акцентира върху технологията на създаване на цикличен алгоритъм. Първо трябва да се открие повтарящия се елемент и второ изходната позиция на „героя“ трябва да бъде една и същата в началото на всяко повторение.

Предвидена е работа в платформата на Code.org. Учениците влизат в акаунтите си и работят по Курс 1, Урок 13 – Лабиринт. Цикли.

В края на урока се прави обобщение на свършената работа и на основните акценти от учебното съдържание.

Урок 8. Цикъл. Робот-художник

Упражнение

Цел. Затвърдяване и надграждане на знания за повтарящи се действия.

Задачи:

1. Да се развият знания и умения за откриване на повтарящи се елементи в очертан път в квадратна мрежа.
2. Да се затвърдят знанията и уменията за използване на команден блок за цикъл.
3. Да се развият знания за създаване на цикличен алгоритъм по дадено задание.
4. Да се затвърдят знания и умения за използване на блок за повторение във визуална среда.

Очаквани резултати:

- Създава цикличен алгоритъм по дадено графично задание.
- Определя колко пъти трябва да се повтори дадена последователност от движения в мрежа
- Определя последователността, която трябва да се повтори, за да се моделира действие във визуална среда.

Методически насоки

В началото на урокът се прави преговор на знанията за последователни действия и по-конкретно се припомнят задачи от урок 4 – Блокове за движение, където учениците програмират „Робот“, който изчертава букви от азбуката в квадратна мрежа.

Разглежда се даденият в учебника код. В него има допуснатата умишлено грешка – целта е учениците да проследят действията и да дадат възможни решения за оправяне на кода. (Например, двете команди Надясно-Надясно могат да се заменят с команда Наляво).

Разглежда се изчертаването на буква В в мрежа и последователният алгоритъм се записва отново, като отделни части от последователния код се заменят с код, в който се използва команден блок за цикъл. Прави се извода, че в един код могат да се съчетават различни видове команди – команди за последователни действия и команди за повтарящи се действия. Прави се и втори извод, че кодът с използване на команден блок за цикъл е значително по-кратък.

За затвърдяване на знанията учениците описват по двойки избрана буква от азбуката. Първо я изписват в квадратна мрежа, а след това записват код с използване на команден блок за цикъл (чрез подреждане на картони с командите от портфолиото на ученика или чрез изписването му на лист чрез знаците, въведени за означаване на командите напред, назад, наляво и надясно).

За очертаване на буквите в мрежа се използва системата от команди „Напред – назад“. Линията по мрежата се очертава от робот-художник, който има лице и гръб и съответно може да бъде обръщан наляво и надясно.

Във втората част на урока е предвидена работа в платформата на Code.org. Учениците работят по Урок 18 – Художник-робот и довършват работата си по Урок 13 – Лабиринт.

В края на урока се прави обобщение на свършената работа и на основните акценти от учебното съдържание.

Урок 9. Знам и мога. Последователни и повтарящи се действия.

Урок за преговор и обобщение

Цел. Обобщение на знания за последователни и повтарящи се действия.

Задачи:

1. Открива повтарящи се части от последователен код и ги записва с блок за цикъл.
2. Създаване на цикличен алгоритъм по дадено задание.
3. Използва блок за повторение във визуална среда.

Очаквани резултати:

- Определя последователността, която трябва да се повтори, за да се конструира алгоритъм с повтарящи се действия.
- Във визуална среда сглобява крайна последователност от блокове, реализиращи цикличен алгоритъм.

Методически насоки

Използва се вече решавана от учениците задача – построяването на алгоритъм за изписване на буквата В в мрежа чрез алгоритъм с последователни действия. Там учениците се сблъскват с проблема за голямата дължина на алгоритъма. Построенията алгоритъм с последователни пресмятания е код с 36 реда.

След беседа, учениците откриват повтарящи се участъци в кода, които могат да бъдат записани с блок за цикъл. В даденият в учебника пример, кодът намалява повече от два пъти като дължина. Кодът може да бъде оптимизиран още повече, но за целите на обучението в трети клас не се предвижда въвеждането на вложени един в друг цикли.

Уменията на учениците да разчитат цикличен алгоритъм се проверяват с поставената им индивидуална задача. При правилно изпълнение, на квадратната схема трябва да бъде

нарисувана стълба с 3 стъпала. Подобни задачи са разработени и за фронтална работа с използване на електронния учебник.

Във втората част на урока е предвидена работа в платформата на Code.org. Учениците работят по различни упражнения от уроците за последователни и повтарящи се действия.

Урок 10. Знам и мога. Диагностика

Урок за диагностика на знанията

Цел: Оценяване на знанията и уменията на учениците.

Задачи:

1. Да се оценят знанията на учениците за дигитални устройства.
2. Да се оценят знанията на учениците за дигитална идентичност.
3. Да се оценят знанията и уменията на учениците за последователни и повтарящи се действия.

Диагностиката се осъществява чрез качествена оценка съобразно постигнатите очаквани резултати на ниво учебна програма по темите „Дигитални устройства“, „Дигитална идентичност“, „Конструиране на последователни действия“ и „Конструиране на повтарящи се действия“. Отчитат се индивидуалните постижения на учениците, чрез попълване на тест, включващ въпроси и задачи.

Тестовите въпроси и задачи са разделени в три части – една част за дигитални устройства и дигитална идентичности по една част за последователни и за повтарящи се действия. Задачите включват избор на верен отговор, свързване с вярното решение и/или описание на алгоритъм.

Методически насоки

Учителят запознава учениците с:

- типа на теста;
- критериите за оценяване;
- скалата за оценка.

Стимулира се самооценяването на учениците.

Урок 11. Среда за програмиране Scratch. Мисия „Пустиня“

Цел. Запознаване с интерфейса и възможностите на визуалната среда за програмиране Scratch.

Задачи:

1. Да се формират знания и умения за ориентиране в интерфейса на визуалната среда Scratch
2. Да се формират знания и умения за работа с менюта и бутони в Scratch
3. Да се формират знания и умения за работа с обекти на средата – герои, декори

Очаквани резултати:

- Познава работното поле и се ориентира във визуалната среда Scratch.
- Знае как да добави герой.
- Знае как да избере декор.
- Знае къде се намират отделните полета за промяна на характеристики на героя.

- Различава менюта и бутони.

Учебно съдържание:

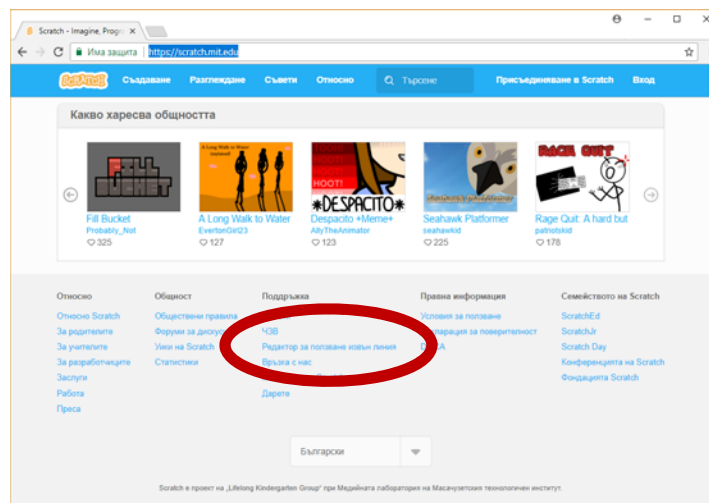
Как да накараме дигиталното устройство да изпълнява нашите инструкции? Какво е това визуална среда за програмиране? Какво е Scratch и какво мога да направя с негова помощ?

За да изпълнява нашите инструкции дадено дигитално устройство, ние трябва да му ги изкажем (опишем) на език който то разбира. Тези инструкции трябва да се напишат в някаква среда за програмиране. Ние вече сме работили с блокчета, знаем какво е код и команда. Scratch е блок-базирана визуална среда за програмиране.

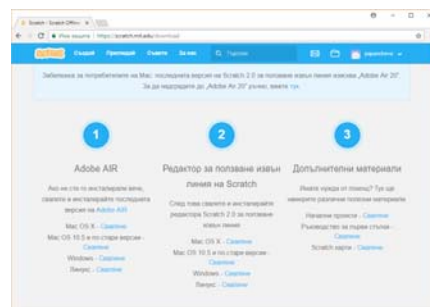
Със Scratch се създават компютърни програми, които ще наричаме „сценарии“ или „компютърни истории“. Всяка история е един програмен проект. Историята се разиграва върху сцена с различни декори. В историята участват герои, които могат да сменят външния си вид. Героите могат да се движат, да говорят или мислят (както в комикс). Историите могат да бъдат озвучени с помощта на звуци и мелодии.

Със средата за визуално програмиране Scratch може да се работи онлайн или офлайн.

За работа офлайн трябва да се инсталира програмата Scratch. Инсталацията става от сайта на средата – <https://scratch.mit.edu/>. В най-долната част на началната страница има поле с връзки, едната от които е „Редактор за ползване извън линия“



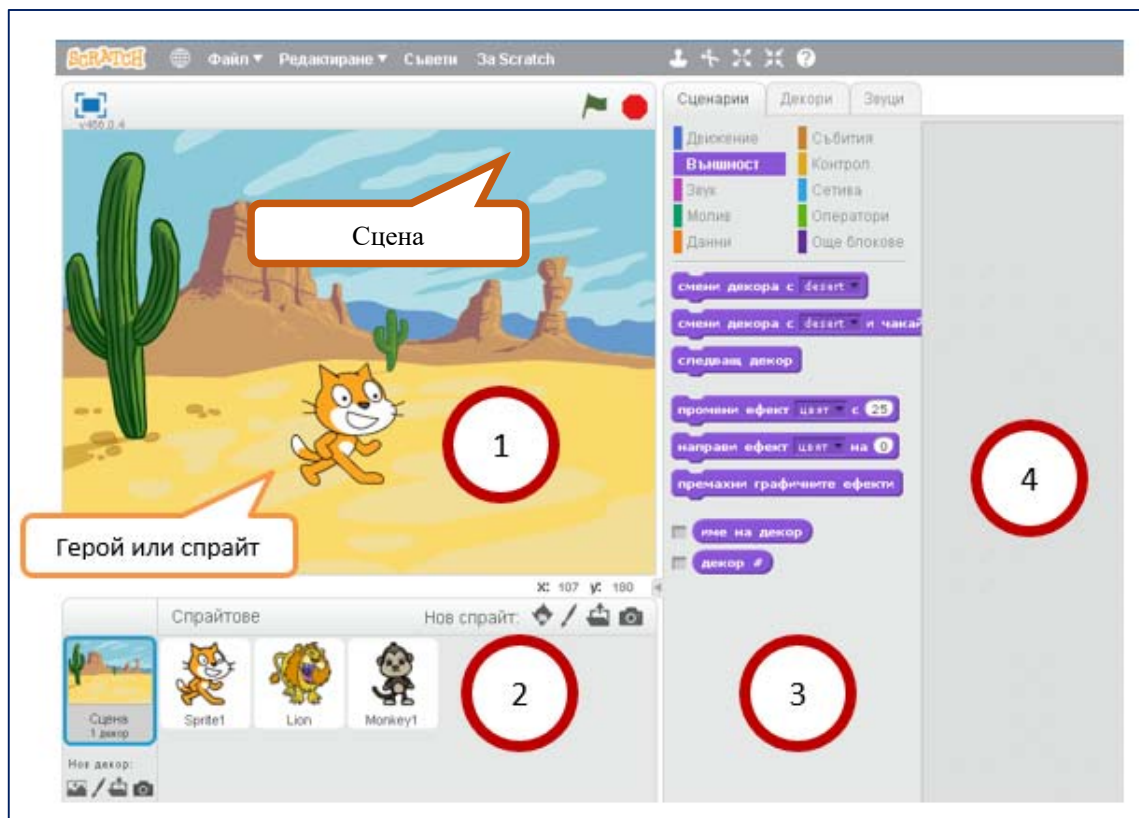
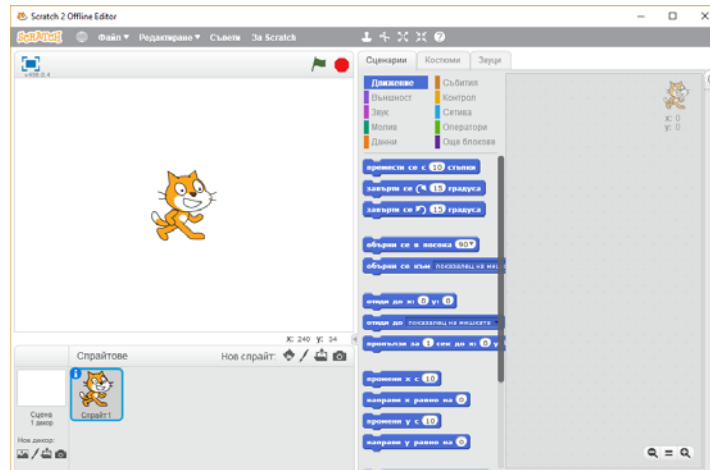
След проследяване на връзката се зарежда страница, която показва стъпките на инсталация. Първо трябва да се изтегли и инсталира програмата Adobe AIR. След това трябва да се изтегли и инсталира програмата Scratch. Третата стъпка е да се изтеглят допълнителни помощни материали под формата на PDF файлове.



След приключване на инсталацията на работния екран се появява икона на програмата Scratch 2



При стартиране на програмата се отваря нейния прозорец и стартов екран:



Работното поле на програмата Scratch се състои от четири основни части:

1. **Екран за преглед на проекта.** Сцена, на която героите (спрайтовете) се движат, мислят или говорят. Всяко действие, което зададем чрез програмен код, можем да видим как ще се реализира след изпълнение на програмата или на част от нея. В горния ляв ъгъл на екрана за преглед се намира бутон за преглед на цял екран. В горния десен ъгъл на екрана за преглед има два бутона – зелено знаме и червена точка. Чрез щракване върху знамето може да бъде стартиран целия проект, а при щракване върху червената точка всички стартирани действия се спират.
2. **Зона за обектите – герои и сцена.** В лявата част на зоната е визуализирана сцената с избрания в даден момент декор. В дясната част на зоната са визуализирани героите, участващи в проекта. Над частта за героите са разположени 4 бутона, с помощта на които могат да бъдат добавяни още герои.
3. **Зона за управление на проекта.** Това поле има три секции – Сценарии, Костюми/Декори и Звуци. Всеки обект, участващ в историята може да извършва определени действия и да има определен външен вид. Поведението на героя или на сцената се задава чрез програмен код, подреден от програмни блокове. Командните блокове се намират в секция „Сценарии“. Командните блокове са групирани в различни цветове според ролята, която имат. Външният вид на героите или на сцената се задават чрез костюм или декори и се намират в секция „Костюми/Декори“. Към всеки обект можем да зададем и звук. Наличните звукови файлове са в секция „Звуци“.
4. **Работно поле.** Когато е избрана секцията Сценарии, в работното поле се подреждат командните блокове, за да се състави съответния програмен код. Блоковете се поставят на работното поле чрез плъзгане и спускане от секцията „Сценарии“ на зоната за управление на проекта. Програмен блок може да бъде премахнат с обратно плъзване и спускане към секцията „Сценарии“.

Герои могат да бъдат добавяни по 4 начина, чрез щракване върху един от бутоните от горния десен ъгъл на зоната за обекти, както следва:



– избор от колекция с готови герои;



– рисуване на героя във вградения графичен редактор;








– избор на герой от файл от компютъра;



– снимане на героя с камерата на компютъра.

По аналогичен начин могат да се добавят декори за сцената.

В горната част на прозореца на средата Scratch се намира главното меню.

От  избираме езика (например Български). От  записваме проекта или отваряме други проекти. От  настройваме изгледа на сцената. От  получаваме помощ и отговори на въпросите си, а от  получаваме информация за последните новости, свързани с развитието на Scratch.

Вдясно от главното меню има бърза лента с инструменти – това са най-често използвани инструменти, разположени там за по-голямо удобство на потребителя. Чрез тези

инструменти могат да се увеличават или намаляват размерите на героите, да се копират обекти и кодове и да се изтрива обект.

Методически насоки:

В началото на урока, учителят провежда беседа с учениците за това как да накараме дигиталното устройство да изпълнява нашите инструкции и защо ни е нужна визуална среда за програмиране.

В хода на фронтална работа и използване на директна демонстрация и електронния учебник учителят запознава учениците със средата за визуално програмиране Scratch.


Разглежда се основния прозорец на средата за визуално програмиране Scratch и се коментират основните компоненти на тази среда. Въвеждат се понятия за интерфейса – зони, менюта, бутони. Изяснява се понятието герой (спрайт или духче). Показва се само един начин за добавяне на герой – чрез избор от вградената в програмата галерия. Показва се и аналогичния начин за избор на декор на сцената от вградената в програмата галерия на декори.

За затвърждаване на знанието на учениците се поставят няколко задачи за самостоятелна работа, обединени в създаване на кратък компютърен проект – Мисия „Пустиня“.

Задача 1. Стартирайте програмата Scratch ². Изберете декор пустиня от вградената в програмата колекцията декори.




Алгоритъм за изпълнение на задачата

1. Стартирай програмата Scratch.
2. Щракни върху бутона „Сцена“ в лявата част на зоната за обекти.
3. Избери секция „Декори“ от зоната за управление на програмата.
4. Щракни върху бутона  за достъп до вградената галерия от декори.
5. Открий декор с изображение на пустиня. Маркирай (щракни върху) декора. Щракни върху бутона ОК.

Задача 2. Добави нови герои от колекцията с герои.



Алгоритъм за изпълнение на задачата

1. Щракни върху бутона  за достъп до вградената галерия от декори.
2. Избери герой (например, маймуна). Маркирай (щракни върху) героя. Щракни върху бутона ОК.
3. Избери втори герой (например, лъв).
4. Чрез влачене и пускане с мишката премести тримата герои на различни места върху сцената.

Задача 3. Задай команда на котето да отиде до маймуната.

1. Щракни върху героя коте в зоната за обекти – герои и сцена.
2. От секция „Сценарий“ в зоната за управление на програмата избери синята група командни бутони.



² Работата може да бъде реализирана онлайн или с инсталирания офлайн редактор на програмата Scratch. При работа онлайн учителят трябва предварително да е проверил наличието на интернет достъп. За по-добра организация е желателно предварително инсталиране на програмата Scratch 2 за офлайн работа.

3. Открий бутона „Отиди до“ и с плъзгане и пускане го постави в зоната за създаване на програмата. Бутонът има падащо меню, от което се избира докъде да отиде котето.
4. Щракни върху командния бутон, за да стартираш действието. Котето се премества до маймуната.

За да стартират отново командата, децата трябва да преместят (чрез влачене и пускане) обратно котето по-далече от маймуната.

Задача 4. Запиши (съхрани) своя първи проект, създаден с програмата Scratch.

За съхраняване на проекта от меню Файл избираме „Запиши като“ (или само Запиши). Задаваме име и място за съхраняване на файла.

По преценка на учителят може да се постави самостоятелна задача, в която учениците да повторят алгоритмите, създавайки свой втори проект със същата сложност – избор на декор, избор на герои, използване на командния блок „Отиди до“, стартиране на действието и съхранение на проекта.

В края на часа се прави кратко обобщение с акцент върху въведената терминология. За нейното трайно усвояване трябва да се работи системно в следващите часове.

Урок 12. Движение на героя

Цел. Запознаване с интерфейса и възможностите на визуалната среда за програмиране Scratch.

Задачи:

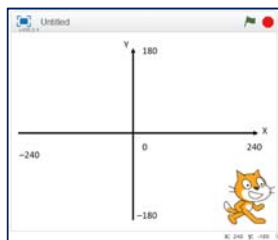
1. Да се формират знания и умения за задаване и промяна на позицията на героя върху сцената.
2. Да се формират умения за промяна на началната позиция на героя.
3. Да се формират знания и умения за описване на основни геометрични фигури.

Очаквани резултати:

- Задава позиция на героя върху сцената.
- Знае как да поставя героя на различни начални позиции и как да го премества.
- Познава и умее да описва познати геометрични фигури.
- Наблюдава промените при движение на героя и може да прави изводи.

Учебно съдържание:

Сцената, върху която се развиват компютърните истории има следните размери: ширина 480 и височина 360 точки. Позицията на всеки обект върху сцената се задава чрез координати. Центърът на координатната система е в центъра на сцената. В долният десен ъгъл се изписват текущите координати на показалеца на мишката.



Чрез обучението по Компютърно моделиране в трети клас се прави пропедевтика на работа с координатна система и с отрицателни числа, без да се въвеждат съответните понятия.

Пропедевтика на понятието „отрицателно число“

Първа стъпка при формиране на умения у учениците да движат героя е задаването на движение надясно и наляво с определен брой стъпки.

В групата командни бутони, управляващи движението на героя (групата е означена със син цвят) първата команда е „Преместите се с 10 стъпки“, като стойността на броя стъпки може да се променя. Чрез директна демонстрация учителят избира герой, примерно котето, и поставя командата „Премести се с 10 стъпки“ на работното поле. При щракване върху командния бутон котето се премества надясно.

Учениците могат да експериментират с различни стойности 10, 20, 30. Прави се извода, че колкото по-голямо е числото, толкова по-голямо е преместването на героя надясно върху сцената. (Ако героят им „избяга“, могат да го преместят в центъра чрез влачене и пускане с мишката).

Учителят поставя пред учениците проблем – как да преместят котето наляво. Показва им решението – записва знак минус пред числото, показващо стъпките. Учениците експериментират с различни стойности. Прави се извода, че колкото по-голямо е числото, показващо броя стъпки, толкова на по-голямо разстояние наляво се премества героят, ако се постави знак минус пред числото.

Формулира се извода, че движението надясно става с указания брой стъпки, а за да покажем, че искаме да се движим наляво, пред броя стъпки записваме знак минус.

Вниманието на децата се насочва върху факта, че с използването на знак минус, героят се движи наляво, но не се обръща обратно.

По този начин учениците ще приемат и с по-голяма яснота разположението на отрицателните числа върху числовата ос в курса на обучение по математика в по-високите степени на образователната ни система. Ще разберат, че колкото по-голямо по модул е числото, толкова по-вяво се намира то върху числовата ос.

Методически насоки:

Урокът започва с преговор на изученото предния час. Припомня се програмата Scratch и нейната работна среда. Активно се използват въведените понятия – за отделните части от прозореца на програмата Scratch, за елементите от интерфейса – бутони и менюта, за обектите – герои и сцена.

В проекта „Мисия „Пустиня““ учениците са изпълнили задачата да придвижат котето до маймунката, за да се пазят заедно от лъва. За изпълнение на тази задача е използвана командата „Отиди до ... маймунката“. При щракване върху командния бутон котето веднага се премества върху или зад маймунката, а не отива „плавно“ до нея. Учителят поставя пред учениците проблема и предлага да намерят решение как котето да отиде по-плавно до маймунката.

Към урокът е подготвен работен файл за задача 1, в който върху сцената на пустинята са разположени героите коте, маймунка и лъв.

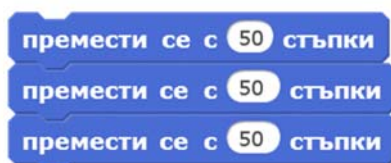
Първата задача на учениците е да преместят котето до маймунката.

Чрез фронтална работа учителят обяснява начина на използване на командата „Премести се ... стъпки“. Обяснява ролята на знака минус. Учениците експериментират заедно с

учителя, задавайки различна стъпка на изместване на котето. Героите в работния файл са подредени така, че ако се зададат 50 стъпки на преместване, след три щраквания върху командния бутон, котето ще отиде близо до маймунката.

Учителят поставя проблем – как котето само да направи три поредни премествания по 50 стъпки, за да отиде до маймунката, без да се налага ние да щракваме три пъти върху блока.

Прилагайки формираните вече знания за последователни действия, учениците могат да предложат да се подредят една под друга три команди за преместване по 50 стъпки. Командните блокове се управляват по същия начин, както тези, с които децата са работили в платформата на Code.org.



След тестване на кода (щраквайки върху залепените заедно командни блокове) учениците виждат, че котето директно се премества 150 стъпки напред и преходът отново не е плавен.

Учителят насочва вниманието им към групата командни блокове, оцветени в оранжево – командите от група „Контрол“. Оттам избират командата „Изчакай 1 сек.“ и поставят по един такъв блок между отделните премествания на котето. При тестване се установява, че вече котето се премества по-плавно до маймунката. Тук учениците могат да експериментират като намалят стъпката и увеличат броя блокове, като променят времето и т.н. (Намаляването на времето под 1 секунда би следвало да се избягва, защото за по-малка стойност от 1, трябва да се работи с десетични дроби, които учениците не познават).



Учителят показва на учениците как проектът може да бъде показан на цял екран чрез щракване върху бутона за смяна на изгледа, който се намира в горния ляв ъгъл на прозореца на програмата. Програмата има два изгледа – изглед на редактиране (в който се разработва компютърния проект) и изглед на представяне (изглед на цял екран).

Пред учениците възниква следния проблем – в изглед на цял екран не се виждат командните блокове и съответно няма как да бъде предизвикано изпълнение на командите за движение на котето към маймунката.

Вниманието на учениците се насочва към командните бутони в тъмно-кафяв цвят – групата командни блокове. Обръща се внимание на две от командите – „Когато е щракнато зеленото знаме“ и „Когато този спрайт е щракнат“.

Когато най-отгоре на кода се постави първата команда, при щракване върху зеленото знаме става изпълнение на командите под него.

Ако най-отгоре на кода се постави втората команда, кодът се изпълнява при щракване върху съответния герой.

Учениците експериментират с двата варианта. След тестване трябва ръчно да върнат котето на изходна позиция, за да могат да видят новото действие. По-нататък ще формират знание как да задават началната позиция на героя.

На учениците се поставя индивидуална работа по проект. Те трябва да създадат самостоятелно свой проект, при който котето и маймунката са едно срещу друго и ритат топка. Топката отива плавно от котето до маймунката и след това се връща пак плавно до котето.

С работата по задачата се затвърдяват знанията за средата – как се избира фон, как се избира герой, както и за преместването на героя с определен брой стъпки в двете посоки – надясно и на ляво. В случая програмират движението на героя „Топка“. Учителят разполага с готово примерен проект в е-ресурсите към учебника.

Урок 13. Завъртане и чертане. Да спасим принцесата

Цел. Формиране и развиване на знанията за движение на героя в Scratch. Формиране на знания и умения за работа с инструмент за чертане.

Задачи:

1. Да се формират знания и умения за движение на героя, свързани със завъртане и обръщане.
2. Да се формират умения за използване на блоковете за движение и завъртане при решаване на конкретна задача.
3. Да се формират знания и умения за изчертаване на основни геометрични фигури при движение на героя.

Очаквани резултати:

- Познава начини за усложняване на движения.
- Работи с предоставените от средата бутони и блокове за решаване на конкретна задача.
- Познава и умее да описва познати геометрични фигури.
- Умее да движи героя по проста траектория.

Учебно съдържание:



Чрез изучената вече команда за движение „Премести се с ... стъпки“ учениците прилагат на практика знанията си за конструиране на алгоритъм с последователни действия при работа с визуална среда за програмиране, реализирайки командите „Напред“ и „Назад“. Предстои да научат как да реализират използваните в началото на учебната година

команди „Наляво“ и „Надясно“, с които са конструирали различни алгоритми на идейно ниво.

Припомня се задачата за изчертаване на квадрат и се поставя проблемът как подобна задача може да бъде реализирана с програмата Scratch.

За целта учениците трябва да усвоят следните знания:

- да могат да завъртат героя, за да реализират движение в различна посока;
- да могат да чертаят линия по екрана, за да очертаят геометрична фигура или произволна начупена линия.

Всяка среда за програмиране дефинира посоки на движение. Посоките в Scratch са определени по следния начин:

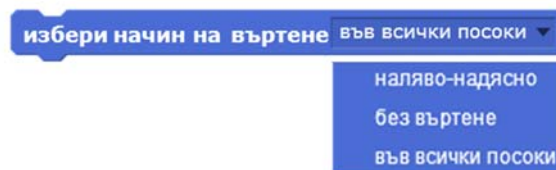


Учениците не са изучавали мерната единица за ъгъл – градус. За избор на посока използват командата „Обърни се в посока ...“ и от падащото меню избират една от опциите – надясно, наляво, нагоре, надолу. Така се запознават с още един начин на ориентиране в мрежа – спомената по-рано .

По-рано са работили с две системи – „Напред-назад“ и „На изток – на запад“. Предстои им да се запознаят и с новата система „Надясно – наляво“, която се използва в средата Scratch.

Посоката определя движението на героя. Ако се зададе команда „Премести се ... стъпки“, героят се премества в посоката, която му е зададена.

Друга команда, която управлява начина, по който изглежда героя при движение в различна посока е командата, определяща начина на въртене – „Избери начин на въртене“.



Програмата Scratch предлага три опции за избор. Ако се избере „без въртене“, героят остава само с една позиция върху екрана, независимо от посоката, в която се движи. Ако се избере „наляво-надясно“, героят се обръща огледално, когато посоката на движение е наляво, но не променя вида си при движение нагоре и надолу. Ако се избере „във всички посоки“

посоки“, героят има 4 пози – по една за движение във всяка посока. В този случай при движение наляво, героят е обърнат „с главата надолу“, сякаш ходи по тавана.



При движението си героят може да оставя следа по екрана. Така чрез движението на избран герой можем да чертаем фигури и произволни начупени линии. Командите за чертане се намират в зелената група командни блокове – „Молив“.

Когато е зададена команда „Молив долу“ при всяко движение на героя след него по сцената се очертава линия. За спре да чертае след себе си, героят трябва да изпълни команда „Молив горе“. При поставянето на нов герой на сцената автоматично се задава команда „Молив горе“.

За линията, която чертае героят, можем да изберем цвят и дебелина. Цветът може да се избере с една от командите „Избери цвят на молива ...“



В първият случай трябва да се запише число. Цветовете в компютъра се задават в цифров (дигитален) вид. За избор на цвят може да се въведе число от 0 до 200. Например, 0 е червено, 50 – зелено, 100 – синьо. Могат да се изберат 200 цвята. Във втория случай цветът се избира, чрез докосване на цвят по избор от произволна област от прозореца на програмата Scratch (подобно на инструментът „Пипетка“ в програмата MS Paint).

Дебелината на линията се задава с командата „Избери дебелина на молива“. По-голямо число отваря на по-дебела линия.

Изтриването на всичко, начертано по сцената става с командата „Изчисти“. Командата изтрива всичко от сцената, независимо от това кой герой е чертал и кой герой изпълнява командата „Изчисти“.

Методически насоки

В хода на фронтална работа с директна демонстрация и използване на електронния учебник учителят запознава учениците с посоките в програмата Scratch и начина на завъртане на героите. Заедно експериментират чрез работа с подготвения към урока демонстрационен файл.

Учителят и учениците разработват примерен сценарий на история на тема „Да спасим принцесата“ по следната идея – баба Яга е отвлякла принцесата и я е затворила във висока кула. Докато баба Яга лети над покривите със своята метла, описвайки огнени квадрати, извънземен принц ще построи стълба, за да спаси красивата принцеса.

Така се поставя първият проблем – как да накарат баба Яга да лети, очертавайки след себе си огнени квадрати.

Учителят запознава учениците с командните блокове за чертане и заедно експериментират за очертаване на линия с избран цвят и дебелина.

Преминва се към изпълнение на задачите. Първата задача е да се създаде алгоритъм, при изпълнението, на който баба Яга да очертае квадрат в небето. Коментира се логиката на програмата и учениците преминават към нейното изпълнение. Работи се с предварително подготвен файл за изпълнение на задачата. Учениците трябва да поставят героя и да създадат кода.

За втората задача учениците работят по предварително подготвен файл. Принцът трябва да се изкачи до прозорецът на кулата, изграждайки (очертавайки) стълба.

Към електронния учебник са дадени допълнителни задачи за изпълнение в продължение на темата за спасяване на принцесата. Като изпитание пред принца се поставят задачи за по-сложен път до принцесата (по-сложни начупени линии, които трябва да очертае героят).

Урок 14. Позициониране и преместване. Скрый рибката

Цел. Формиране и развиване на знанията за движение на героя в Scratch. Формиране на знания и умения за позициониране на героя върху сцената.

Задачи:

1. Да се формират знания и умения за позициониране на героя на сцената.
2. Да се формират умения за използване на блоковете за движение и завъртане при решаване на конкретна задача.

Очаквани резултати:

- Задава позиция на героя върху сцената.
- Знае как да поставя героя на различни начални позиции и как да го премества.
- Наблюдава промените при движение на героя и може да прави изводи.
- Познава начини за усложняване на движения.

Учебно съдържание

Пропедевтика на понятията „координатна система“ и „координати“

Позицията на всеки обект на сцената се определя от неговите координати според въведената в програмата координатна система. Чрез обучението по компютърно моделиране се прави пропедевтика на знанията по темата за координатна система, без въвеждане на точните математически понятия за координатна система, координати и координатни оси.

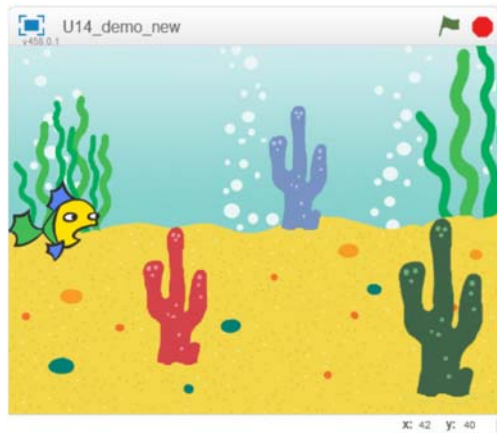
В хода на работа по точно формулирани задачи се цели да се формира разбиране у учениците за това, че мястото на всеки герой може да се опише с помощта на две числа, означени с X и Y. Във всеки момент в долният десен ъгъл под екрана за преглед се показват координатите на показалеца на мишката. Поради тази причина, за да видим координатите на героя, трябва да щракнем върху него или да задържим показалеца на мишката върху него – виждаме приблизителните му координати. Върху бутона на активния герой в зоната за обекти се появява малка буква i – знак за допълнителна информация. При натискане върху бутончето i се отваря прозорец, в който са записани точните стойности на числата X и Y за героя: -80 и -60. Тази информация не е предвидено да се дава на учениците.

Методически насоки

Учителят припомня реализираните до момента задачи, свързани с движението на героите по сцената. Котето се преместваше до маймунката наляво и надясно на определен брой стъпки, топката между котето и маймунката се движи чрез изместване наляво-надясно, баба Яга се движи в четирите посоки на определен брой стъпки, принцът се движи само в две посоки с определено изместване.

Учителят стартира демонстрационното упражнение и поставя задачата – рибката трябва да плува на зиг-заг между водораслите. Първо да плува до червеното водорасло, след това до синьото и до зеленото.

Със знанията които имат към момента учениците могат да придвижат рибката вдясно, а след това надолу, докато уцелят мястото на червеното водорасло, но това не решава точно поставената задача.



Въвежда се понятието „позиция“. Мястото, на което се намира героят се нарича негова позиция. Позицията на всеки герой се определя от две числа X и Y. Учителят показва двете команди – Отиди до ... и Пропълзи за ...сек. до ...

Когато преместим героят до дадена позиция върху екрана, числата в тези две команди се променят. Учителят мести рибката на различни места и показва, как числата в бутоните на командите се сменят. Програмата Scratch сама изчислява позицията на героя в даден момент и попълва стойностите на двете числа X, Y в двете команди.

Ако искаме рибката да плува от изходната позиция до червеното водорасло изпълняваме следния алгоритъм:

1. Преместваем рибката до червеното водорасло. Стойностите на числата, определящи позицията ѝ там се записват автоматично в командата Пропълзи ...
2. Взимаме командата Пропълзи ... и я поставяме в кода на рибката. Не променяме числата, тъй като те вече са изчислени от програмата Scratch.
3. Връщаме рибката в началната ѝ позиция.

При щракване върху командния блок, който сме подредили виждаме как рибката пропълзва от изходната позиция до червеното водорасло.

Така може да се подреди целия код на движение на рибката от водорасло до водорасло. При задаване на повече секунди в командата Пропълзи ..., движението става по-бавно.

Вниманието на учениците се насочва към още един детайл. Рибката не плува правилно – насочена е надясно, а плува надолу. Нужно е рибката да се обърне към водораслото. Учителят показва наличието на такъв команден блок в синята група и начина за работа с

него. Командата е: „Обърни се към ...“. От падащото меню може да се избере между показалецът на мишката и всички включени в проекта герои. Учителят задава на рибката да се обърне към червеното водорасло.

На учениците се поставя задачата да довършат проекта „Скрий рибката“. Подготвен е работен файл, в който са поставени героите, а учениците трябва да запишат само кода за движение.

При изпълнение на първата задача, проектът може да се надгради с добавяна на акула, която да се движи наляво-надясно. Рибката трябва да плува от водорасло до водорасло, криейки се от акулата.

Учениците работят самостоятелно. Напомня им се текущо да съхраняват работата си.

Допълнителна бележка: в подготвеният работен файл героите са преименувани с имена, записани на кирилица – червено водорасло, растение 1 и т.н. В удобен момент учителят може да покаже на учениците как могат да променят имената на своите герои, за да се ориентират по-лесно в логиката на програмата. (преименуването става при щракване върху символа *i*, който се появява при избор на героя).

Урок 15. Цикли в Scratch. Космическо пътешествие

Цел. Формиране на знания и умения за управление движението на героя чрез цикличен алгоритъм

Задачи:

1. Да се формират знания и умения за използване на блоковете за реализиране на цикличен алгоритъм при решаване на конкретна задача за движение и чертане.
2. Да се формират умения за разчитане на чужд код, реализиращ цикъл.
3. Да се приложат знания и умения за комбинирано използване на блоковете за чертане, движение.

Очаквани резултати:

- Използва блокове от програмната среда за реализиране на цикличен алгоритъм
- Умее да разчита чужд код, реализиращ цикъл

Учебно съдържание:

В програмата Scratch командните блокове, с които се реализират циклични алгоритми се намират в групата команди „Контрол“. В трети клас се използват само два вида цикъл – за непрекъснато повторение и за повторение посочен брой пъти.



В първия случай действията, включени в цикъла се повтарят постоянно. Спирането на програмата става само то червения бутон за Стоп, намиращ се над екрана за преглед на проекта. Във втория случай действията, включени в цикъла се повтарят точно дадения

брой пъти и след това програмата продължава нататък, ако има други команди след цикъла. Може да се забележи, че блокът на цикъл „Винаги“ в долната си част няма издаден край и не може да се закачи за последващ го блок.

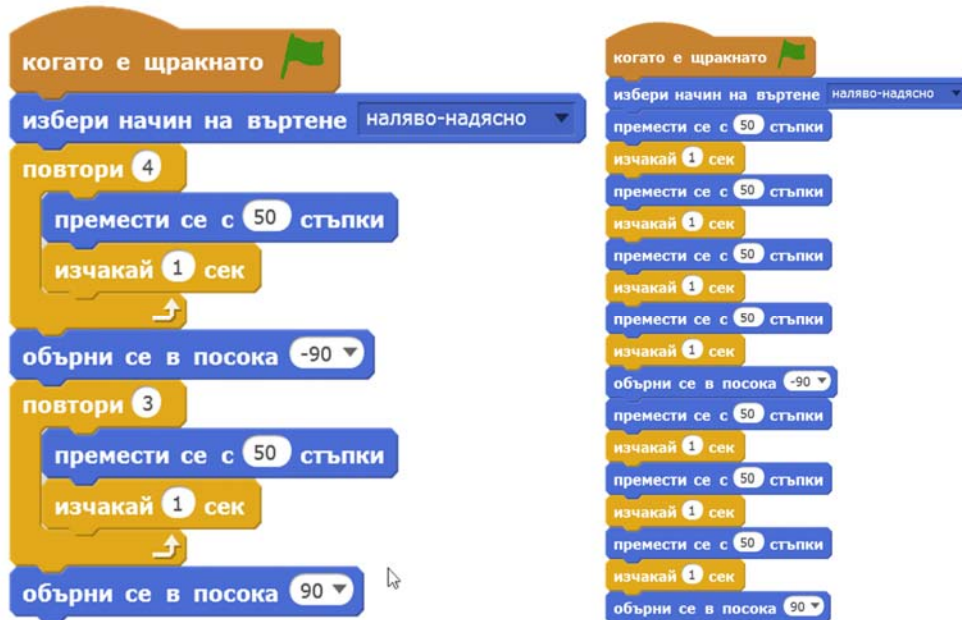
Операторите за цикъл се подреждат както останалите команди в кода на програмата. Не е възможно да се пресичат. Може в един цикъл да влезе друг. В трети клас не се изучават вложени цикли.

Учениците имат знания за понятието цикъл от уводната част на курса по Компютърно моделиране. Съставя ли са циклични алгоритми на идейно ниво и са работили с циклични алгоритми в средата на Code.org.

Методически насоки

Учителят избира някои от вече решените задачи, в които може да се използва цикъл, за да стане кодът по-кратък за изписване.

Разглеждам се движението на акулата от проекта „Скрий рибката“. Нека акулата прави 4 премествания вдясно, след това се връща 3 пъти наляво и спира. Кодът е дълъг и включва много повтарящи се редове, които могат да бъдат програмирани чрез 2 цикъла.



Учителят припомня на учениците задачата за принца и стълбата. Там също има повтарящи се действия, които могат да бъдат заменени с един цикъл.

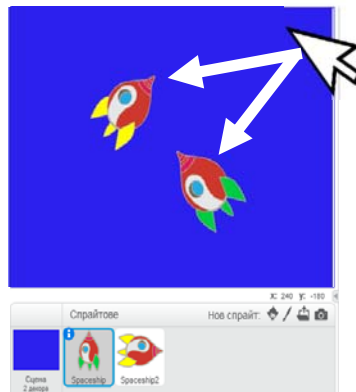
След редактиране на един от двата алгоритъма по преценка на учителя се преминава към поставяне на нов проект – Космическо пътешествие.

Учителят използва разработеният демо-проект към урока и поставя задачата на учениците – да се създаде част от компютърна игра. Ракетата трябва да се движи постоянно в посоката, в която се движи показалецът на мишката. Така накъдето се движи мишката, натам ще се движи и ракетата.

След дискусия с класа формулират извода, че за тази задача трябва да се използва цикъл „Винаги“. Действието на програмата ще бъде спирано от бутона Стоп, над екрана за преглед.

Обясняване се новото използване на командния бутон „Обърни се към ...“, който учениците са използвали в проекта „Скрий рибката“. При изпълнение на тази команда, героят се обръща в посока надясно и се завърта с тази си страна винаги към показалеца на мишката.

Разглеждат се два примера. В единият ракетата (когато е в посока надясно) е насочена нагоре, а в другия – върхът ѝ сочи вдясно. Учителят трябва да отдели повече внимание на този казус, защото се смесват посока вдясно от гледна точка на програмата и посока вдясно от гледна точка на ученика, базиран на житейския му опит и естественото ориентиране на предметите, според тяхното предназначение и форма, а не според зададената координатна система.



Учениците работят по задачата да съставят програма за постоянно преместване на ракетата в посоката на мишката.

Дадени са допълнителни задачи за очертаване на траектория на ракетата. В траекторията има циклично повторение, което учениците трябва да открият и програмират самостоятелно.

Урок 16. Позициониране, движение, чертане. Знаем и мога (обобщение)

Цел. Обобщаване и затвърждаване на знанията и уменията за движение на герой

Задачи:

1. Да се систематизират знанията на учениците върху движение на герой
2. Да се формират умения за използване на блоковете за движение, чертане и реализиране на цикличен алгоритъм при решаване на конкретни задачи

Очаквани резултати:

- Използва блокове от програмната среда Scratch за решаване на практически задачи с движение на герой

Методически насоки

Обобщителният урок включва систематизиране на изучените понятия и команди, както и предложения за работа по задачи, с които се затвърждават и надграждат знанията на учениците. Предложени са разнообразни задачи, обезпечени с необходимите работни файлове и нагледни материали в електронния учебник. По подобие на дадените задачи учителят и учениците могат да доразвият и свои проектни идеи.

Урок 17. Сцена и декори. Кой къде живее

Цел. Формиране на знания и умения за промяна на обкръжението на героите при създаване на компютърен проект със Scratch.

Задачи:

1. Да се формират знания и умения за избор на декори за историите от наличните в средата Scratch.

2. Да се формират знания и умения за създаване на собствени сцени според конкретен компютърен проект.
3. Да се формират знания и умения за управление на декорите.

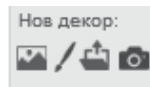
Очаквани резултати:





- Преглежда, избира и задава сцени от наличните в средата.
- Създава собствени сцени.
- Планира и реализира идея със смяна на сцени.

Учебно съдържание:

Сцената, върху която се реализира компютърният проект е обект, за който можем да изберем фон и да го управляваме. Всеки фон в средата Scratch наричаме „декор“. Сцената може да има няколко декора.

Нов декор може да бъде добавен по четири основни начина, като се използват разположените в долния ляв ъгъл на прозореца на програмата бутони.



1. Декорите могат да бъдат избрани от галерия от изображения, включена в програмата Scratch. Достъпът до галерията се осъществява чрез първия бутон ().
2. Програмата предоставя възможност потребителят сам да нарисова декор. За тази цел се използва вградения в средата Scratch графичен редактор. За създаване на декор чрез рисуване се използва втория бутон ().
3. Декор може да се зареди от файл, съхранен на компютъра () – файл в графичен формат (png, jpeg или gif). Размерите на сцената са 480x360 пиксела. По-големи изображения се скалират пропорционално.
4. Декор може да бъде заснет директно от камерата на компютъра ().

Вграденият графичен редактор на програмата Scratch предлага стандартния набор от инструменти за рисуване и чертане. Четката и линията имат цвят и дебелина. Могат да се чертаят фигури – правоъгълник или кръг. Фигурата може да бъде без запълване или запълнена с плътен цвят. Контурът има цвят и дебелина. Цветът на запълването на фигурата съвпада с цвета на контура на фигурата.

Графичният редактор поддържа работа с растрена и с векторна графика. Векторната графика е по-качествена и се предвижда основно работа с нея. Така, ако учениците имат някакъв предварителен опит от работа с редактора MS Paint могат да видят разликата между създаването на графични изображения – растрени и векторни.

Чрез инструментът за текст (при работа с растрена графика) може да се въвежда текст върху изображението за фон. Не се поддържа въвеждане текст на кирилица. Ако на фонът трябва да има текст на кирилица, то той трябва да бъде въведен в друга програма и вмъкнат като картинка. Гумата има различна големина.

Всеки декор на сцената може да бъде именуван с име, което помага на потребителя да следва определена логика на подредба, хронология и развитие на събития.

Чрез бутоните в горната част на екрана могат да се правят по-сложни редакции на фона. От бутона „Добави“ може да се зареди изображение от галерията. А от бутона „Импорт“ към фона може да се включи произволно изображение (png, jpeg, gif).

Методически насоки

За формиране на знания и умения за работа със сцена и декори в средата Scratch е избрана темата „Кой къде живее“. Чрез фронтална работа по първата задача учителят показва как към сцената могат да се добавят декори и как тези декори могат да се управляват. Към урокът е разработен готов демо-файл, в който е избран герой – момичето Аби, която ни „развежда“ из своята къща. Чрез смяна на декорите се зареждат различни стаи от дома на Аби.

Учителят стартира вторият работен файл към урока, в който е избран само героят Аби. Припомня как се избира декор на сцената. Това учениците вече са правили в първия урок за работа със средата Scratch – Мисия „Пустиня“.

Новото тук е, че се добавят няколко декора, които учениците ще се научат как да управляват. Учителят поставя 3 подходящи декора (примерно, спалня, кухня и дневна стая от дома на Аби) от вградената в програмата галерия.

Обръща внимание на учениците, че декорите се номерират в реда на тяхното поставяне. Декорите могат да се разместват с влачене. Декори могат да се изтриват. Учителят изтрива белия декор, който се появява със стартирането на всеки нов проект. Изтриването става с помощта на инструмента за изтриване от бързата лента с инструменти или от знака X който се появява при щракване върху декор в раздел *Декори*.

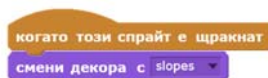
В средата Scratch сцената може да бъде управлявана както и героите. Сцената не може да се движи, но може да променя външния си вид – могат да се сменят декорите или да се променя цвета на избрания декор. Вниманието на учениците се насочва към групата команди „Външност“. Учителят обяснява действието на основните команди:

1. Следващ декор – сменя декора с декора със следващ пореден номер.
2. Смени декора с ... – сменя декора с точно определен декор, който се избира от списъка с наличните декори.

По преценка на учителят може да се обясни действието на командите Промени ефект ... и Направи ефект ... Тази задача може да се остави и за самостоятелна откривателска работа на учениците. Чрез опити те сами могат да открият как действат двойката команди.

Втората задача на учениците е да създадат свой проект, при който чрез смяна на декора да покажат къде живеят съответните животни. За целта към урока е подготвен работен файл. В него са поставени три герои – жаба, пеперуда и пингвин. При щракване върху герой трябва да се зареди декор, който показва къде живее съответното животно.

Деца сами избират подходящи декори, за да покажат къде живеят животните. Избират схема за управление на проекта. При щракване върху даден герой да се зареди съответния декор. Чрез задачата учениците научават, че декорите могат да се управляват и чрез кода на героя.



Учителят обръща внимание на учениците върху това, че в началото проектът започва от бял фон, на който са разположени героите. За да започва винаги така проектът при

стартиране, това трябва да се зададе в кода. За целта се добавя част към кода на някой от героите – при щракване върху зеленото знаме, декорът да се смени с декор номер 1 (декорът с бял фон). Така се прави пропеедвтика на необходимостта от задаване на начално състояние (входни данни) за програмата, която създава програмистът – къде и как да са разположени героите, коя да е сцената и т.н.

Последният акцент в урока е създаването на собствен декор чрез вградения в средата графичен редактор. Третата задача на учениците е да нарисуват сами декор. Работят по третия за урока работен файл, в който е включено четвърто животно – риба. При щракване върху рибата трябва да се зарежда декор, показващ средата, в която живее рибата – например, река, море, аквариум или др.

С помощта на вградения редактор учениците могат да нарисуват море, използвайки инструментите Четка и Кофа. Четката има цвят и дебелина. При избор на Кофа могат да изберат цвят, както и стил за запълване. Важно е очертаващата линия да ограничава точно областите на запълване, за да не се излива цвета върху цялата страница. С помощта на инструментите за чертане на фигури (кръг, отсечка) може да се добавят слънце и/или облаци и други елементи към изображението за новия декор.



В края на урока учителят прави обобщение на наученото с акцент върху очакваните резултати, формулирани при разработването на урока.

Урок 18. Герои и костюми. Зимни игри

Цел. Формиране на знания и умения за промяна във външния вид на героя – избор на костюм и създаване на нов костюм на героя. Надграждане на знанията за създаване на нов герой със средствата на вградения графичен редактор.

Задачи:

1. Да използва герой с различни костюми от галерията с герои.
2. Да редактира костюмите на герой от вградената галерия.
3. Да програмира промяната на външния вид на героите в свой проект.

Очаквани резултати:

- Използва различни костюми, вградени в средата.
- Променя външния вид на герой.
- Планира и реализира проект с промяна на външния вид на героя.

Учебно съдържание:

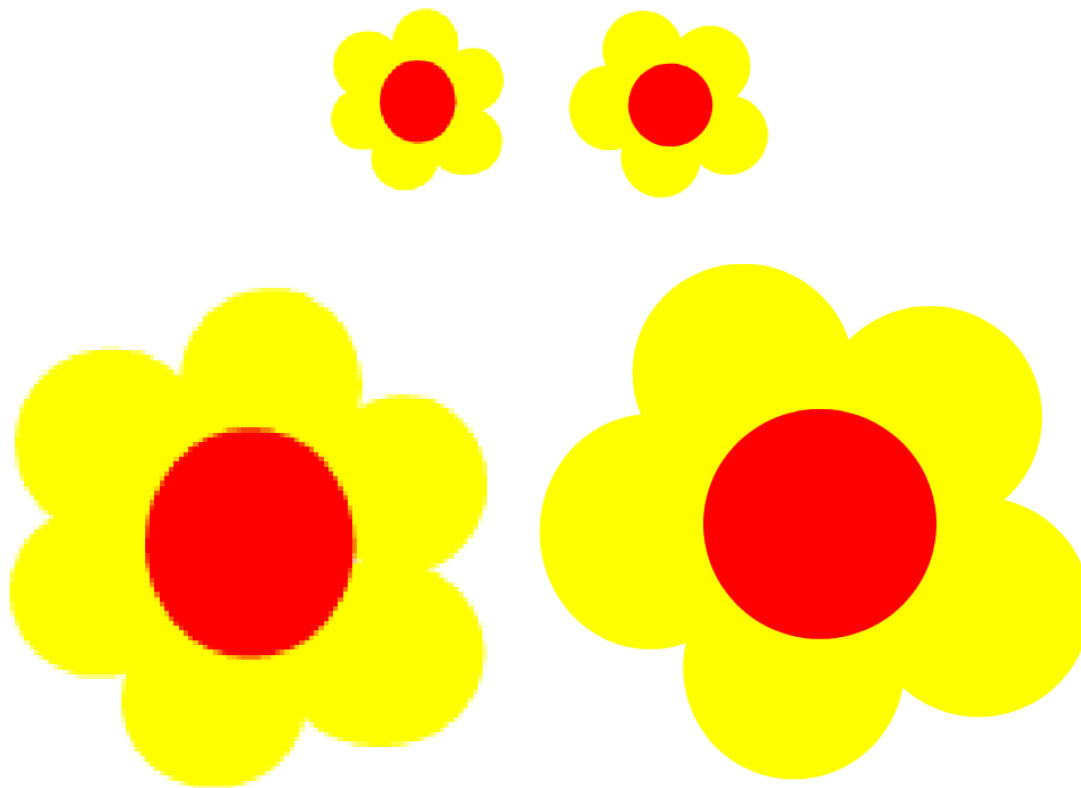
В този урок се усвояват знания и умения за промяна на външния вид на героя/.

Героите, както и сцената са обекти, на които можем да управляваме външния вид – начина, по който изглеждат. Както сцената има декори, които могат да се променят, така и героят има костюми, които могат да се сменят. Основните команди, с които се управлява външния вид на героя са: „Следващ костюм“ и „Облечи костюм ...“.

Чрез смяна на костюма, съчетана с използване на команди за цикъл и изчакване се получава анимирано изображение. /Така учениците се подготвят за въвеждане в темата за анимация, анимирано изображение, кадър, анимиран проект.

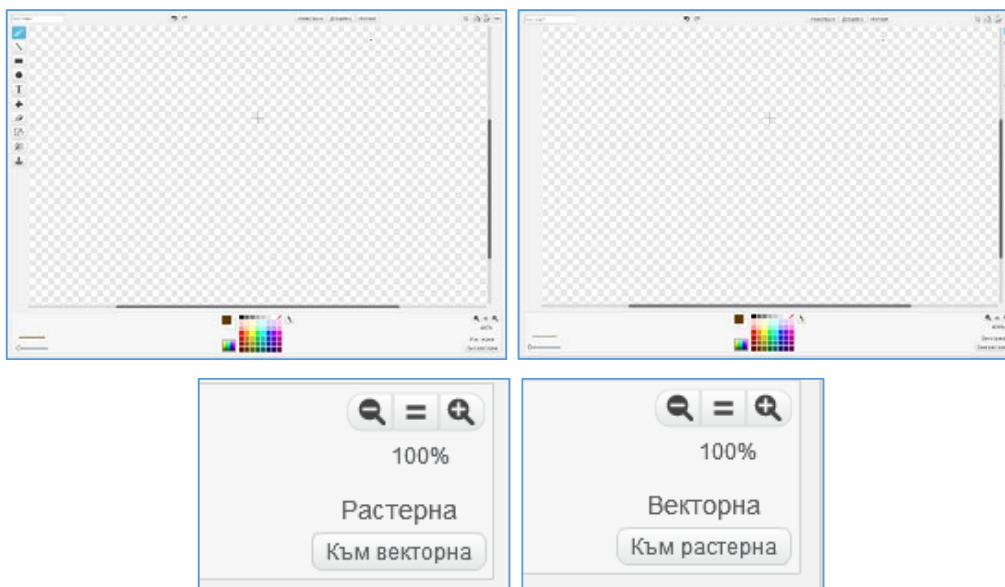
Героите могат да се създават от потребителя чрез използване на вградения в средата графичен редактор. Графичният редактор на средата Scratch поддържа работа с растерни и с векторни изображения. Тези понятия не се въвеждат при работа с учениците, но се обяснява разликата за работа в двата формата. При векторна графика изображението се представя с помощта на математически формули. Това изображение запазва качеството си при преоразмеряване, защото при всяка смяна на размера, с помощта на формулите се преизчислява линията на изображението, така че да се запази качеството ѝ. Растерното изображение е подобно на гоблен – изображението се състои от отделни пиксели – малки квадратчета, оцветени със съответния цвят. При увеличаване на размера на растерно изображение неговото качество се разваля.

Още при създаване, растерното изображение има по-лошо качество, в сравнение с векторното. Разликата в качеството проличава още повече с увеличаване на размера.



При растерно изображение, след като един елемент е създаден, той не може лесно да бъде променян. Докато при векторно изображение могат да се правят детайлни промени на линията във всеки елемент от изображението. Инструментите за работа с векторно и с растерно изображение са различни. Във вградения редактор в програмата Scratch в

долната част на прозореца има възможност за превключване между инструментите за работа с векторна или с растерна графика.



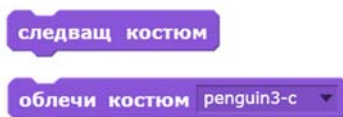
Предвидено е запознаване на учениците с инструментите за векторна графика и редактиране и създаване на векторни герои.

Методически насоки

Работата е планирана в рамките на два учебни часа.

Чрез директна демонстрация по подготовеният към урока първи демо-проект, учителят запознава учениците с понятието „Костюм“ на героя. Прави се аналогия между сцената и декора и героя и неговите костюми. Една сцена може да има няколко декора. Един герой може да има няколко костюма. При избор на герой от галерията с герои се зареждат и неговите костюми. Има герои с по един костюм, но има и герои с два, три и повече на брой костюми. В подготовеният демо-проект е избран героят Penguin3, който има три костюма, означени като penguin3-a, penguin3-b, penguin3-c.

Както при работа със сцената се използва команда за смяна на декора, при работа с герой може да се използва команда за смяна на костюми. Учителят насочва вниманието на учениците към командите за управление на външния вид на героя от групата „Външност“. Обяснява действието на двете основни команди:



Както декорите на сцената, така и костюмите на героя са номерирани и при команда „Следващ костюм“ героят приема външен вид, според следващия по ред костюм. С помощта на командата „Облечи костюм ...“ можем да изберем точно определен от костюмите на героя от вграденото в блока на командата падащо меню.

Чрез последователна смяна на костюмите можем да накараме пингвинът да „танцува“ с кънките върху снега. По този начин се прави подготовка на учениците за въвежда на темата за анимирани изображения и компютърна анимация в следващите уроци.

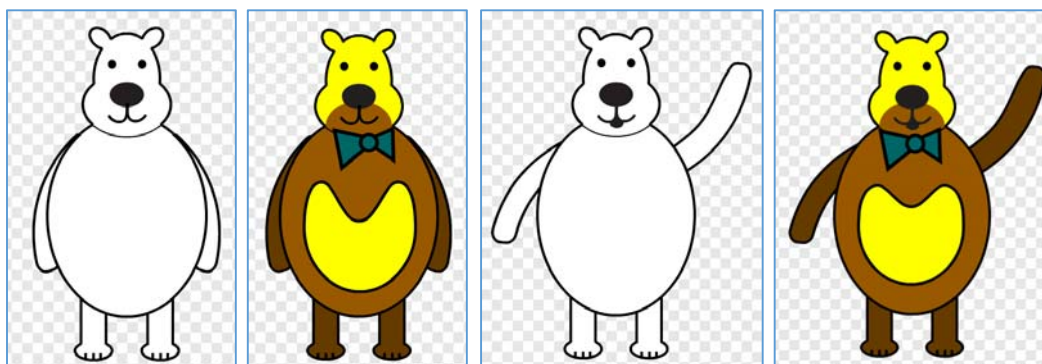
Движението на героя може да се реализира с използване на различни команди. Ако се използва командата „Следващ костюм“, съчетана с цикъл за постоянно повтаряне и команда за изчакване, героят сменя последователно трите си костюма и има три пози, през които преминава движението му. Може да се направи и нов танц, при който да се използват само два от костюмите.






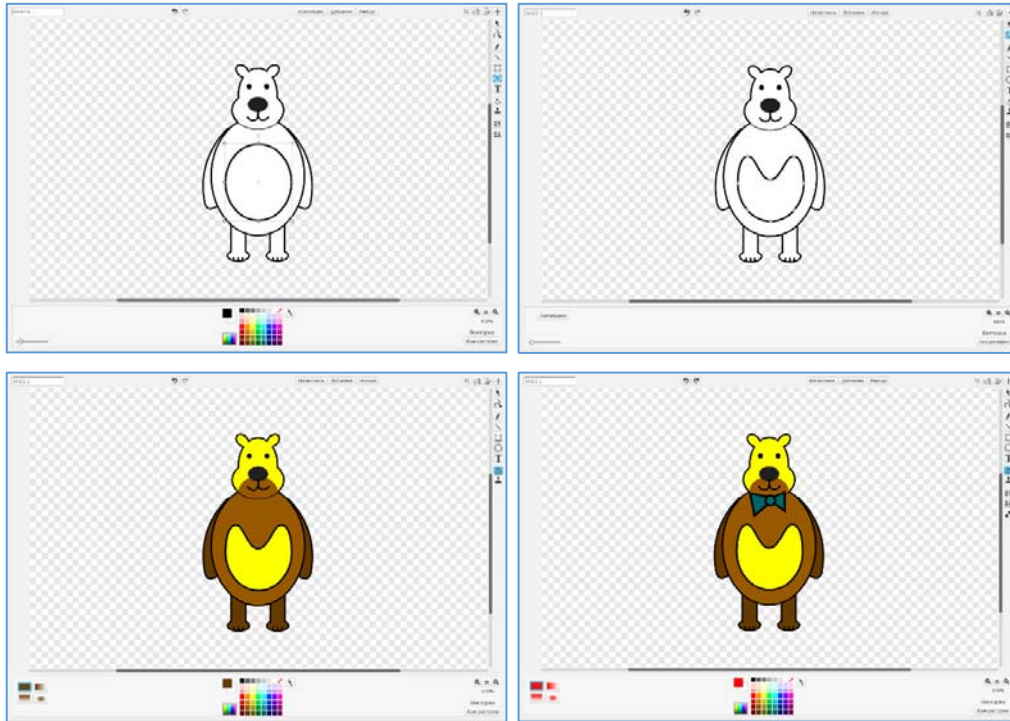
Следващ акцент в урока е създаване на собствен герой с използване на вградения в средата Scratch графичен редактор. Учителят показва подготовения към урока втори демо-проект.

Целта е учениците да се научат да работят с инструментите на вградения графичен редактор. Като първа стъпка те редактират герой от галерията – героят bear2. Изображението е векторно и чрез промяната му, учениците се запознават с инструментите за работа с векторна графика.

Постепенно учителят показва как използва различни инструменти на графичния редактор и стъпка по стъпка създава новият герой.



1. С помощта на инструмента Елипса () се очертава коремчето на героя.
2. С помощта на инструмента Оформяне () се оформя вдлъбнатината на коремчето.
3. С помощта на инструмент Оцветяване на форма () се оцветява мечокът.
4. С помощта на командата Добави от галерията с изображения се добавя панделката на героя.



По същия начин, като се повтарят стъпките на описания алгоритъм, се оформя и вторият костюм на героя.

На учениците се поставя задача да доразвият проекта Зимни игри, като към пингвина от първата задача, добавят новия герой и го оформят по показания начин. За да го раздвижат, използват същия код, както при пингвина – постоянна смяна на двете пози с малко изчакване между тях.



На по-бързо работещите ученици може да се постави задачата да създадат с копиране още декори и да оцветят лампичките от декора winter-lights в различни цветове. С въвеждане на код, аналогичен на кодовете на героите, но с команда „следващ декор“ ще се анимира фонът на проекта, като лампичките ще светят в различни цветове.



Урок 19. Герои и костюми. Да направим снежен човек

Цел. Формиране на знания и умения за създаване на нов герой със средствата на вградения графичен редактор.

Задачи:

1. Да създаде свой герой като използва инструментите на вградения редактор.
2. Да създаде различни костюми на създадения самостоятелно герой.
3. Да програмира промяната на външния вид на героя.

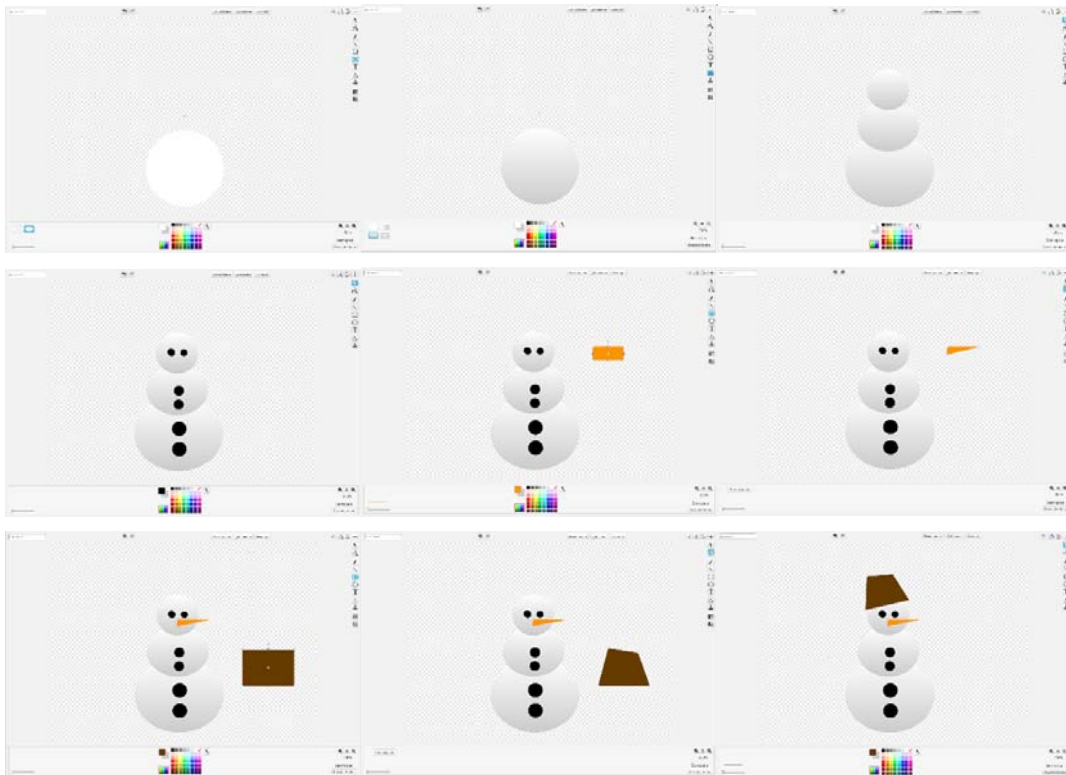
Очаквани резултати:

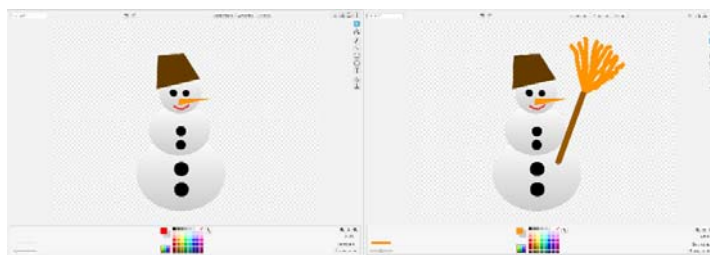
- Създава нов герой.
- Създава нови костюми на новия герой.
- Планира и реализира проект с промяна на външния вид на героя.

Методически указания

Урокът продължава темата „Зимни игри“. Задачата на учениците е добавят към вече създадения проект нов герой – снежен човек. Новият герой трябва да бъде самостоятелно разработен с използване на инструментите на вградения графичен редактор. Работи се с инструментите за създаване на векторно изображение.

Съпките за създаване на снежния човек са визуализирани със следните изображения:





Без да прави директна демонстрация, учителят само обяснява последователността на действията за изработване на новия герой. Акцентира на отделните инструменти. Показва само по-трудните от стъпките – например, как се създава носа и как се групират елементите на метлата. Учителят показва как се съхранява създадения герой и как вече съхранен герой може да се използва в друг проект на Scratch, като се зареди от файл.

Оставя се достатъчно време за самостоятелна работа на учениците.

Урок 20. Озвучаване на проекта. Гласът на животните.

Цел. Формиране на знания за вмъкване на звук в компютърен проект на Scratch и възпроизвеждането му при действие на потребителя.

Задачи

1. Да се формират знания за звуковата информация, която може да бъде използвана в проекти, създадени със средата Scratch.
2. Да се формират знания за командните блокове за работа със звук – избор на звук, управление на нивото на звука, на темпото и др.

Очаквани резултати

- Познава средствата на програмната среда за възпроизвеждане на звук
- Планира и реализира проект с озвучаване на героите.


Учебно съдържание:

В средата за визуално програмиране Scratch може да се използва два вида озвучаване – чрез звуци и мелодии или чрез ноти.

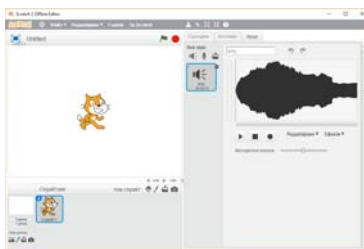
В първия случай може да се използва звуков файл от вградената в програмната среда галерия, може да се използва звуков файл от компютъра на потребителя или да се направи аудио-запис чрез използване на вграден или включен към компютъра микрофон. Средата за визуално програмиране Scratch позволява използване на аудио файлове във формат MP3 и WAV. Възпроизвежда се само моно-звук.

Във втория случай се композира мелодия, като се използват специализирани блокове, за възпроизвеждане на музикалните ноти. Всяка нота е означена чрез число – 60 е C, 61 е C#, 62 е D и т.н.

В трети клас предвиждаме запознаване на учениците само с първия начин на озвучаване на създаден от тях проект, а именно – чрез използване на готов звуков файл.

Някои от героите в Scratch имат зададен звук. При вмъкване на героя можем да проверим дали има зададен към него звук. Избираме героя и отваряме раздел „Звуци“. Ако има звук, можем да го чуем чрез щракване върху бутона „свири“ .

Например, основният герой – котката на Scratch – има звук по подразбиране „Мяу“. Можем да го чуем като изберем котката, отворим раздел „Звуци“ и щракнем върху бутона „Свири“.



Задача. Нека целта ни е при щракване върху герой да се възпроизвежда неговия звук. Трябва да напишем код, който да изпълнява тази задача.

Отваряме раздел „Сценарии“. Блоковете, които използваме за работа със звуци са оцветени в лилаво. Можем да използваме командните блокове „Пусни звук ...“ или „Пусни звук ... докато свърши“. Да изберем първия вариант. Поставяме командния блок на работното поле. От падащото меню на командата избираме звук „Мяу“. При двойно щракване върху командата се чува прикачения звук.

За да се стартира звука при щракване върху героя трябва да се добави команда за настъпило събитие. От раздела команди „Събития“ избираме командата „Когато този спрайт е щракнат“.



Така всеки път при щракване върху героя се чува избрания звук.

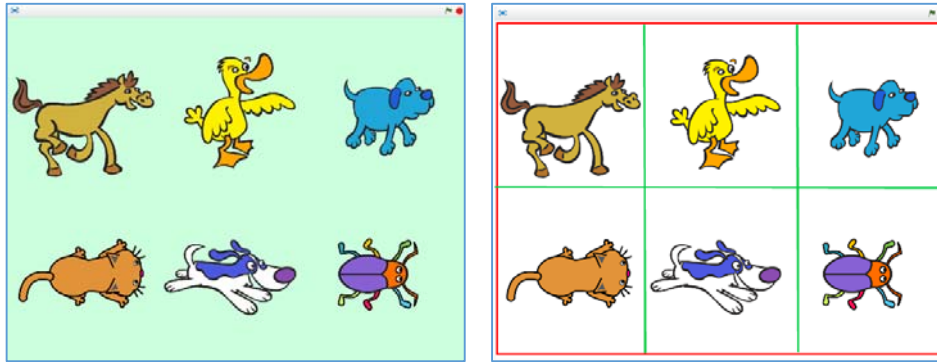
Методически насоки

В хода на фронтална работа чрез използване на електронния учебник и директна демонстрация учителят запознава учениците с възможностите за възпроизвеждане на звук в средата Scratch. Създава се код, при който при щракване върху спрайта се възпроизвежда указан звук.

За затвърдяване на знанията следва самостоятелна работа на ученици по проект „Гласът на животните“. Учителят показва примерна реализация на проекта, като коментира всяка от стъпките, които трябва да се изпълнят за неговото реализиране.

Стъпка 1. Създаване на подходящ фон.

Животните могат да бъдат свободно разположени върху екрана или да бъдат подредени в таблица. С помощта на вградения редактор учениците трябва да оцветят фона или да го оформят като таблица.



Стъпка 2. Вмъкване на герои-животни.

От галерията на програмата Scratch се вмъкват животни по избор.

Разполагат се върху фона. При нужда се използва инструмента за намаляване на размера, за по-добро разположение на героите върху екрана.

Стъпка 3. Задаване на звук за всеки герой.

Героите може да имат присвоен звук по подразбиране.

Добавят се допълнителни звуци – за кучето, за щурца и др. като се използва вградената в Scratch галерия.

Стъпка 4. Създаване на код за всеки герой.

За всеки герой се създава код – при щракване върху героя да се възпроизвежда избран звук.

Стъпка 5. Тестване и съхраняване на компютърния проект

След създаване на всички кодове, проектът се тества. При щракване върху всяко от животните трябва да се чува съответния звук. Готовият проект се съхранява. (Файл/Записване).

Урок 21. Героят мисли. Въвеждане на текст. Делфинът риба ли е?

Цел. Формиране на знания за вмъкване на текст в компютърен проект на Scratch под формата на комикс – текст в текстова кутия, показващ съобщение или „мисъл“ на героя.

Задачи

1. Да се формират знания за текстова информация, която може да бъде използвана в проекти, създадени със Scratch.
2. Да се формират знания и умения за въвеждане на текст от клавиатурата на компютъра – смяна на езика, въвеждане на главни и малки букви, преминаване на нов ред, въвеждане на препинателни знаци.
3. Да се формират знания за използване на блок за визуализиране на въведен текст под формата на съобщение или „мисъл“ на героя.

Очаквани резултати

- Познава средствата на програмната среда за въвеждане на текст.
- Въвежда текст от клавиатурата.
- Въвежда текст в компютърен проект на Scratch под формата на мисъл на героя като използва команден блок.

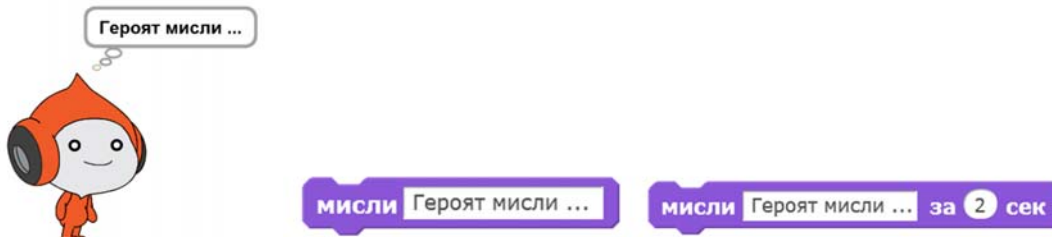
Учебно съдържание:

Въвеждането на текст е важно умения за времето, в което живеем. Правилното въвеждане на текст е свързано както с правилното изписване на думи и изрази, така и спазване на правила за въвеждане на текст. Основни правила, с които учениците трябва да се запознаят са:

1. Между думите се оставя само по един интервал.
2. Препинателните знаци точка, запетая, удивителен и въпросителен знак се въвеждат веднага след предхождащата ги думи (без да се оставя интервал) и след тях се оставя един интервал, преди да се въведе следваща дума.

Актуализират се понятията „Клавиатура“ и „Клавиш“. Припомня се, че клавиатурата е входно устройство за компютърната система и за всяко друго дигитално устройство. За въвеждане на текст учениците трябва да могат да сменят езика, да въвеждат главна буква, да въвеждат препинателни знаци.

В средата за визуално програмиране Scratch може да се въвежда и използва текстова информация. Текстът се въвежда в кутии, визуализиращи мисълта на героя (с балон за мисли) или визуализираща речта на героя (с текстова кутия, съдържаща думите на героя)



Командните блокове за записване на мислите на героя са два – „Мисли ...“ и „Мисли ... за ...сек.“ При изпълнение на първата команда, записаната мисъл се показва на екрана и стои до задаване на ново съобщение. Ако няма нов текст за мисъл или говор, надписът остава постоянно на екрана. При изпълнение на втората команда записаният текст се визуализира само за зададеното време, след което изчезва.

При поредица от мисли, първият команден блок може да се съчетае с използване на командата „Изчакай ... сек.“ Например, на кодът показан по-долу при щракване върху зеленото знаме над героя се показва балон, в който е записано „Мечката ражда малките си.“. Изчакава се 4 секунди. Появява се нов надпис над главата на героя – „Мечката е бозайник.“ и този надпис остава на екрана.



Методически насоки:

В хода на фронтална работа се актуализират и надграждат знанията на учениците за работа с текстова информация.

Обръща се внимание на работата с компютърната клавиатура. Въвеждат се понятията „клавиатура“ и „клавиш“. Формират се знания за избор на език. Формират се знания за въвеждане на текст – главна и малка буква, преминаване на нов ред. Когато целият текст

трябва да бъде с главни букви се използва клавишът Caps lock. Ако само една буква трябва да бъде главна, то при нейното въвеждане се задържа натиснат клавиша Shift.

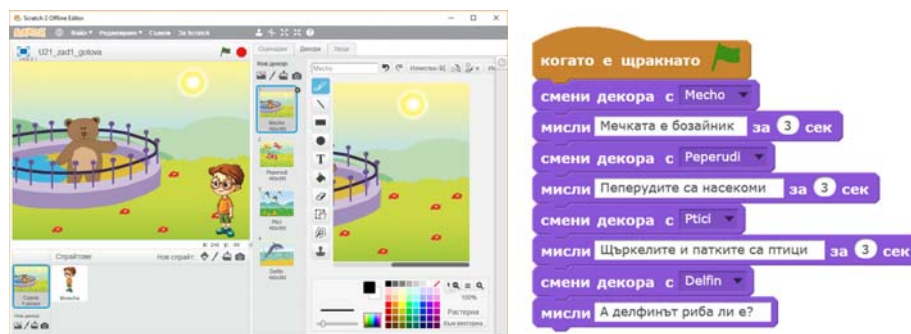
Alt+Shift – сменяш езика от латиница и кирилица и обратно.
Caps Lock – въвеждаш само главни букви
Shift + буква = главна буква
Enter – преминаваш на нов ред.

С помощта на електронния учебник и директна демонстрация учителят запознава учениците с командните блокове за изразяване на мислите на героя.

За затвърдяване на знанията следва самостоятелна работа по проект „Делфинът риба ли е ...?“, като учениците използват предварително подготвен работен файл. Работният файл се отваря с двойно щракване от работната папка на ученика.

В работният файл сцената има 4 декора. Сценарият се състои в следното – героят мисли ... Мечката е бозайник. Щъркелите са птици. Пеперудите са насекоми. А делфинът ... дали е риба?

За всяка от мислите е подготвен отделен декор. Целта е да се надградят знанията на учениците за смяна на декора. Кодът на героя може да бъде следният:



Урок 22. Героите говорят. Разговор за делфините.

Цел. Формиране на знания за вмъкване на текст в компютърен проект на Scratch под формата на комикс – текст в текстова кутия, показващ „реч“ или „мисъл“ на героя.

Задачи

1. Да се формират знания за използване на команден блок за визуализиране на въведен текст под формата на съобщение или „реч“ на героя.
2. Да се формират знания за комбинирано използване на командни блокове за говор и изчакване, за да се получи диалог между герои.

Очаквани резултати

- Познава средствата на програмната среда за въвеждане на текст.
- Въвежда текст от клавиатурата.
- Въвежда текст в компютърен проект на Scratch под формата на говор на героя като използва команден блок.
- Съчетава командни блокове за говор и за изчакване и създава диалог между герои.

Учебно съдържание

Реализирането на диалог между герои може да се направи по два начина в средата Scratch:

- Чрез изчисляване на времето на „говорене“ и времето на „чакане“ между събеседниците.
- Чрез изпращане на съобщения между събеседниците.

В курса по Компютърно моделиране в трети клас е предвидено да се формират знания и умения за създаване на диалог между героите само по първия начин – чрез изчисляване на времената на „говорене“ и „изчакване“ между героите.

Следва се сленият алгоритъм – докато единият герой „казва“ даден текст за зададено в командата време, то другият герой изпълнява командата „изчакай“ за същото време. След това ролите се разменят. При повече от двама герои времето на изчакване трябва да се изчислява по-прецизно, за да не се застъпват репликите на героите.

Методически насоки:

Учителят въвежда учениците в темата, като надгражда вече формираните знания за работа с командите „Мисли ...“. Разликата в случая е, че при командите „Кажих“ има двама или повече герои, които трябва да реализират диалог помежду им.

Към урокът е подготвен работен файл, в който са записани по два кода. При щракване върху клавиша интервал двамата герои говорят едновременно. При щракване върху зеленото знаме се стартира правилния код, при който героите се изчакват. Така след поставяне на проблемна ситуация, учителят излага алгоритъма за създаване на диалог между герои.

Преминава се към самостоятелна работа на учениците по дадения в учебника диалог. Учениците трябва да изберат двама герои от галерията на програмата Scratch и между тях да разиграят описания диалог.

На по-бързо работещите учениците учителят може да постави аналогична задача, но по диалог, който ученикът трябва сам да измисли, запише и програмира. Примерна задача: *Избери животно, намери интересни факти за него и ги представи чрез диалог между герои.*

Урок 23. Диалог между повече герои. Още за делфините.

Цел. Затвърждаване на знанията за работа с текст в средата Scratch.

Задачи

1. Да се затвърдят знанията за работа с текст чрез командите „Мисли“ и „Кажих“.
2. Да се надграждат знания и умения за въвеждане на текст от клавиатурата на компютъра – смяна на езика, въвеждане на главни и малки букви, преминаване на нов ред и др.
3. Да се надградят знанията за създаване на диалог между повече герои в средата Scratch.

Очаквани резултати

- Познава средствата на програмната среда за въвеждане на текст.
- Въвежда текст от клавиатурата.
- Програмира диалог между повече събеседници.

Методически насоки

За затвърждаване на знанията на учениците за създаване на диалог между герои в средата Scratch е предвидена работата по следния диалог между трима герои (героите са Тера, Гобо и Пико от средата Scratch):

Пико: Здравейте, Тера и Гобо! Какво правите?

Гобо: Здравей Пико! Говорим си за делфините.

Пико: Наистина! Аз зная много за тях.

Тера: Здравей Пико! Кажете ми нещо интересно.

Пико: Делфинът има около 100 зъба. Той ги използва, за да улови плячката, но не дъвче с тях. Гълта храната цяла.

Гобо: Колко интересно! Кажете още!

Пико: Делфините живеят на стада и се грижат се за болни, стари или ранени делфини от стадото.

Тера: И ние се грижим за приятелите си! Довиждане Пико и Гобо!

Пико и Гобо: Довиждане, Тера!

По време на работата учителят обръща вниманието на учениците върху важността за правилно въвеждане на текста – без правописни грешки и със спазване на правилата за въвеждане на компютърен текст.

При разговор между повече герои времената за говорене и чакане се изчисляват по-трудно. Когато първия герой говори, примерно 2 минути и след него говори втория герой още 3 минути, то третия трябва да часа общо 5 секунди. Може да се записват сумарно времената на изчакване, но за да се проследяват по-лесно е добре да се записват в две отделни команди вместо в една обобщена. Например, реализирането на дадения диалог може да стане със следните команди между тримата герои – Пико, Тера и Гобо:

The image shows three sets of Scratch code blocks for characters Piko, Gobo, and Tera. Each set starts with a 'when green flag clicked' block, followed by a 'say' block for the character's dialogue, and then three 'wait' blocks (2, 3, and 2 seconds respectively) to simulate the character's speaking time. The code blocks are color-coded: orange for 'when green flag clicked', purple for 'say', and yellow for 'wait'.

Пико:

- когато е щракнато
- кажи "Здравейте Тера и Гобо! Какво правите?" за 2 сек.
- почакай 2 сек.
- кажи "Наистина! Аз зная много за тях." за 2 сек.
- почакай 2 сек.
- кажи "Делфинът има около 100 зъба. Той ги използва, за да улови плячката, но не дъвче с тях. Гълта храната цяла." за 2 сек.
- почакай 2 сек.
- кажи "Делфините живеят на стада и се грижат се за болни, стари или ранени делфини от стадото." за 2 сек.
- почакай 2 сек.
- кажи "Довиждане, Тера!" за 2 сек.

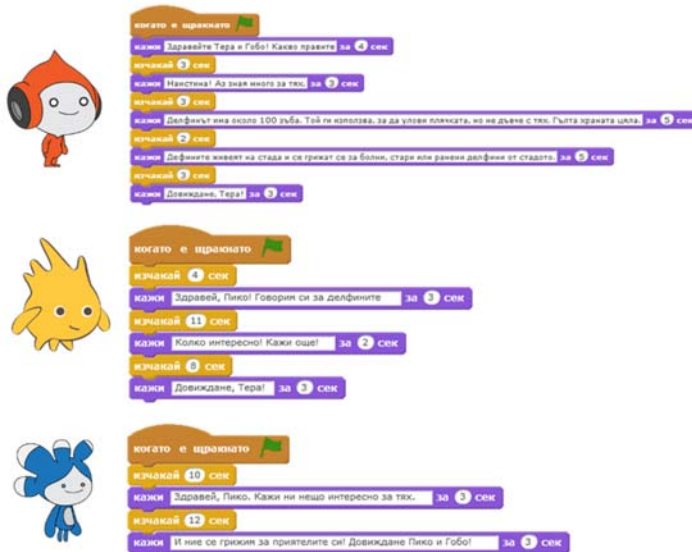
Гобо:

- когато е щракнато
- кажи "Здравей, Пико! Говорим си за делфините." за 2 сек.
- почакай 2 сек.
- кажи "Колко интересно! Кажете още!" за 2 сек.
- почакай 2 сек.
- кажи "Довиждане, Тера!" за 2 сек.

Тера:

- когато е щракнато
- кажи "Здравей, Пико. Кажете ми нещо интересно за тях." за 2 сек.
- почакай 2 сек.
- кажи "И ние се грижим за приятелите си! Довиждане Пико и Гобо!" за 2 сек.

Така поредността на командите се вижда добре и може да се проследи съответствието във времената между героите. След като се тества, кодът може да се оптимизира и тогава би изглеждал така:



Урок 24-25 Работа по проект. Разказ за цветята.

Цел. Затвърдяване и надграждане на знанията за работа със сцени и герои и създаване на диалог между героите в Scratch.

Задачи

1. Да се затвърдят знанията за работа със средата Scratch.
2. Да се развива творческо и критично мислене у учениците.

Методически насоки:

Идеята на проектната работа състои в следното: Гобо е от друга планета. Там няма цветя. В проекта, учениците избират герой, който да разкаже на Гобо за цветята на Земята.

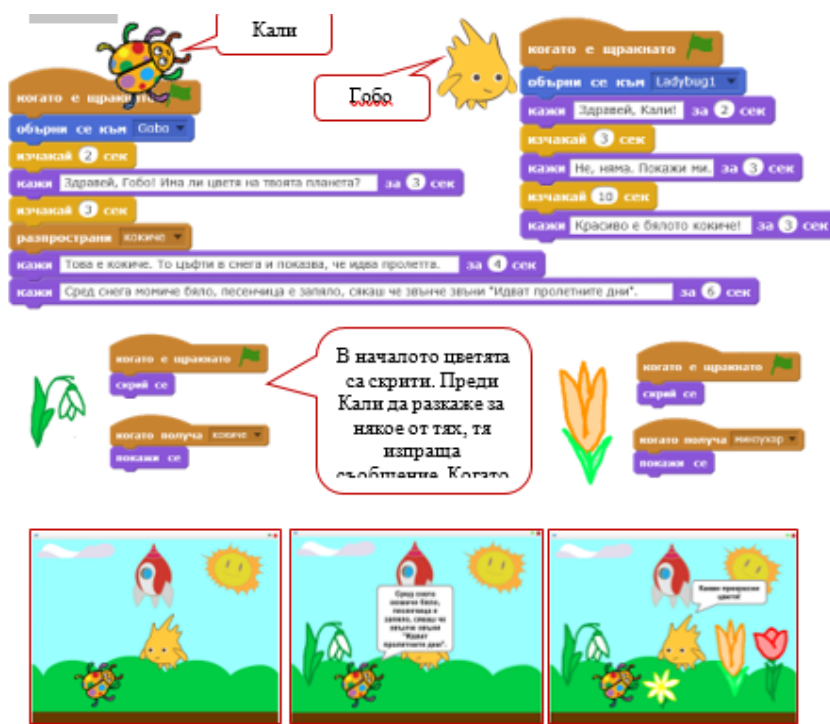
Предвижда се целият проект да се изработи самостоятелно от учениците, като те използват готови изображения от галерията на програмата, както и създават всичките им необходими герои.

За онагледяване на крайния резултат от проекта е разработен предварително проект, като ресурсите от готовия проект са предоставени като работни файлове. Така по преценка на учителят проектът може да бъде направен първо самостоятелно от учениците, а след това и с използване на предложените ресурси – герои и илюстрации.

Стъпка 1. Създай свои герои – цветя, като ги нарисуваш в редактора на Scratch.



Стъпка 2. Избери герой-разказвач (примерно, калинката Кали). Създай диалог между Кали и Гобо, в който Кали му разказва и показва цветята. Използвай стихове и гатанки за цветята



Във втората част на проектната работа се прави връзка с математиката. Учениците сами създават сцени, по които формулират кратки математически задачи. Отговорът на задачата трябва да бъде на екрана като нов герой. При щракване върху правилен отговор да има звукова индикация.

Урок 26-27. Компютърна анимация. Изгубеното мече

Цел. Формиране на знания за анимиран образ, кадър, компютърна анимация.

Задачи

1. Да се формира знание за кадър, анимиран образ, компютърна анимация.
2. Да се създаде анимиран проект във визуалната среда Scratch.

Очаквани резултати

- Създава анимирани образи в Scratch.
- Създава анимиран проект в Scratch.

Учебно съдържание

Анимацията е процес на бърза смяна на поредица от картини (изображения) с цел да се създаде илюзия за движение. Всяка от картините се нарича „кадър“. Разликата между два кадъра може да бъде в цвят, промяна в елемент от изображението или изместване на изображението.

Някои от героите в Scratch имат по няколко костюма. При бързата им смяна се получава анимирано изображение – котето сякаш се движи, хора танцуват и др. Може да се направи извода, че всеки от костюмите на едни герой е отделен кадър за създаването на анимиран образ.

При непрекъснатата смяна на кадрите се получава анимиран образ или анимирано изображение. За да се постигне илюзията за движение отделните кадри не трябва да се различават много, а само по определени детайли – това може да са цвят или промяна в отделен елемент от изображението и др.

Основните методи за създаване на анимирано изображение, с които учениците се запознават са промяната на цвета и промяната на един елемент в цялостно изображение.

Следващо понятие, с което учениците се запознават е компютърната анимация. Под компютърна анимация разбираме кратка история, разказана чрез движение, звук и кратък текст. За създаване на цялостен анимиран проект е важно да се създаде сценарий.

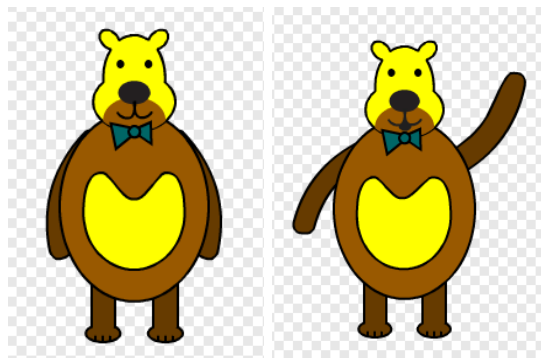
Методически насоки

В хода на фронтална работа учителят трябва да въведе новите понятия – кадър, анимиран образ, анимация.

До този момент учениците са работили с герои в средата Scratch, които имат по няколко костюма. Могат да се припомнят някои от реализираните проекти, например, пингвинът от проекта „Зимни игри“



Чрез смяна на костюмите и малко изчакване между тях, се получава анимирано изображение – танцуващ пингвин. Отделните костюми наричаме „кадър“. Крайният резултат от смяната на кадрите се нарича „анимирано изображение“.



Анимираното изображение на танцуващият пингвин се състои от 3 кадъра. А анимираното изображение на махащото с ръка мече се състои от 2 кадъра. Колкото повече са кадрите, толкова по-качествено е анимираното изображение.

Да разгледаме следната примерна история: *Мечо се е загубил в гората. Среща го Пико, който знае къде живее Мецана. Той води изгубеното мече при майка му, която е много щастлива.*

Работата по проекта цели да се формират знания и умения за разработване на сценарий, планиране на работата и реализиране на проекта. Работи се два учебни часа. В учебника са разписани сценарият и отделните стъпки за реализация на проекта.

Показани са техники, чрез които с инструментите на вградения редактор се създават нови герои – дърветата в гората, малкото мече и Мецана, декорът на проекта.

Вниманието на учениците се акцентира върху командите за разпространяване и получаване на съобщение. Чрез тези команди се надграждат знанията на учениците за управление на кода и се подготвят за учебното съдържание в четвърти клас. Когато един герой изпрати съобщение – всички герои го получават. Но само героят, който в кода си има команда „Когато получа съобщението ...“ реагира като изпълнява дадените му под този команден блок команди.

Урок 28-29. Работа по проект. Пролетни игри

Цел. Надграждане на знания за работа по проект – създаване на анимирана история

Задачи

1. Да се създаде анимиран проект, чрез използване на галерията на програмата, както и на изображения от файл.

Очаквани резултати

- Създава анимирани образи в Scratch.
- Създава анимиран проект в Scratch.

Проектната работа е обезпечена с ресурси, които учениците могат да използват, за да работят по дадения примерен модел или да им се даде свобода да разработят самостоятелно анимиран проект по свой собствен сценарий.

Урок 30-31. Знаем и мога

Урок за диагностика на знанията

Цел: Оценяване на знанията и уменията на учениците.

Отчитат се индивидуалните постижения на учениците, чрез попълване на тест, включващ въпроси и задачи.

Урок 32. Преговор и обобщение

Целта на урока е да се направи преговор и обобщение на наученото. Предвижда се провеждане на открит урок с презентация и демонстрация на избрани проекти и работни задачи от обучението през годината.