

Компютърно моделиране – 4 клас

Методическо ръководство за учителя

Румяна Папанчева

Тодорка Глушкова

Съдържание

Методически указания по теми и уроци	3
Урок 1. Видове информация	3
Урок 2-3. Футболисти-програмисти. Среда за програмиране Scratch. Последователни и повтарящи се действия (преговор). Входяща диагностика.....	5
Входяща диагностика	9
Урок 4. Информацията и дигиталните устройства	11
Урок 5. Информацията в съвременното общество	14
Урок 6. Условия за безопасност в дигитална среда.....	16
Урок 7. Блок за разклонение	19
Урок 8. Разклонен алгоритъм.....	21
Урок 9. Знаем и мога.....	23
Урок 10. Блокове за аритметични действия. Случайни числа.....	24
Урок 11. Блокове за аритметични отношения и логически оператори.....	28
Урок 12-13. На училище – Аритметични и логически операции	36
Урок 14. Управление на движението на героите. Изчакване, забавяне, ускоряване...	37
Урок 15. Синхронизиране на действията на героите.....	39
Урок 16. Компютърна анимация. Промяна на външния вид на героите	42
Урок 17. Работа с променливи	43
Урок 18. Знаем и мога.....	45
Урок 19. Работа с програмируемо устройство	46
Урок 20-21 Среда за визуално програмиране Blue Bot. Командни блокове за движение. Роботът се движи.....	49
Урок 22. Моят робот	54
Урок 23. Програмирам робот. Знаем и мога	56
Урок 24. Знаем и мога.....	57
Урок 25-28. Компютърна игра „На училище“	57
Урок 29-32. Работа по проект	59
Урок 33-34. Представи проект. Обобщение и самооценка	62
Изходяща диагностика.....	62
Критерии за оценяване на проект – Таблица за оценителя / учителя	63
Карта за самооценяване на проект	65
Критерии за оценяване на образователна игра – Таблица за оценителя / учителя..	66
Карта за самооценяване на игра.....	67

Методически указания по теми и уроци

Урок 1. Видове информация

Урок за нови знания

Цел. Формиране на знания за информация, видове информация и начините на получаването ѝ.

Задачи:

1. Да се формира понятие за информация.
2. Да се формира знание, че информацията съществува в различни форми (има различни видове информация)
3. Да се формира знание, че информацията може да се получи от различни източници.

Очаквани резултати:

- Познава формите на представяне на информацията (текстова, числова, звукова, графична).
- Познава начините за получаване на информация (включително с възприятията – слух, зрение, рецептори).

Учебно съдържание:

Информацията може да се разбира като знание за предметите и явленията. Всеки обект носи информация за себе си – цвят, размер, свойства, особености. Информация е знанието в колко часа излита или каца даден самолет на летището, информация е цената, записана под стоката на щанда за хранителни стоки, информация е списъка на учениците в класа – такъв списък може да носи информация за имената на учениците, за адресите им, за работата на родителите на учениците и т.н.

Информацията може да бъде различен вид (текст, звук, изображение, видео, диаграма и др.), както и да се получава от различни източници (чрез задаване на въпроси, чрез новините от радиото, телевизията, вестниците, интернет, чрез книгите, чрез разговори с други хора и т.н.). Това са двата основни акцента в урока.

Информацията може да бъде в звуков вид. Например, сирената на линейка означава, че някъде има болен пациент, сирената на пожарната кола означава, че някъде има пожар, който трябва да бъде угасен. Интервю по радиото носи информация по темата, която се обсъжда. Не е необходимо да виждаме говорещия, достатъчно е да чуваме неговите думи. Звуковата информация се възприема чрез човешкото сетиво за слух – ухото.

Информацията може да бъде получена от дадено изображение – картина, снимка, рисунка. Например, гледайки снимка на архитектурния план на апартамент можем да разберем колко са стаите, има ли тераса, къде се намират кухнята, спалнята, дневната, банята, тоалетната и т.н. От снимките (картини, изображения по стените) в класната стая може да се получи най-различна информация – за известни личности, за родния край, за правила по математика или български език и т.н.).

Частен случай на изображенията са символите и знаците. За по-бързо получаване на информация хората използват различни символи като указания. Има пътни знаци, чрез които получаваме информация и указания как да се движим безопасно. Знакът „Пешеходна пътека“ носи информация, че може да преминат пешеходци през платното,

зеленият знак на светофара носи информация, че ще ни бъде осигурено безопасно пресичане на кръстовището и т.н. Информацията, представена чрез картини, схеми, снимки и др. казваме, че е представена в графичен вид. За възприемане на графична информация използваме зрението си.

Информацията може да бъде представена под формата на текст. Чрез написаното в книги, вестници, списания, брошури ние научаваме много факти, знания за предмети и явления, интересни истории. Често в текста важната информация се откроява с по-голям размер на буквите или с различен шрифт и цвят. Текстовата информация отново възприемаме чрез зрението си.

Информацията може да бъде в комбиниран вид – мултимедийна информация. Тя може да съдържа едновременно текст, изображения, звук, анимация и т.н. Може да се обърне внимание на детските книги и на книгите за възрастни. В детските книжки изобилстват картинките, а текстът е малко, докато при възрастните е точно обратното.

Информация получаваме през сетивата си. Освен чрез слуха и зрението, можем да получим информация чрез сетивата си за вкус и за допир. Например, ако ядем ябълка, круша или друг познат плод със завързани очи, ще го разпознаем само по вкуса му, без да е нужно да го виждаме. Чрез ръцете и кожата си можем да получим информация за времето (топло, студено, дъждовно, снежно, слънчево). Ако хванем с ръцете си даден предмет получаваме информация за него – дали е мек или твърд, дали е направен от плат или от кожа и т.н.

В компютъра информацията се съхранява в различен вид – имаме звукова (музикален файл), текстова (доклад), графична (изображение), видео (видеоклип) информацията. А когато тези видове информация се съчетаят, казваме, че информацията е в мултимедиен формат (мултимедийна презентация, комбиниран документ, сайт).

Методически насоки

Урокът започва с беседа по основните акценти, описани по-горе – изяснява се същността на информацията. На база на илюстрациите се въвеждат различни видове информация. Дискутира се върху начина, по който получаваме информацията – чрез използване на различни сетива.

Поставя се проектна задача за седмицата:

Проектна задача за седмицата. Разделете учениците в екипи – единият екип да събере информация в звуков вид под формата на интервю. Задайте тема за интервюто. Например, учениците могат да интервюират възрастни хора за любимите им игри, когато са били малки или за това кои технологии са били нови, когато възрастните са били деца или младежи и т.н. Екипът трябва да предаде на учителя направените интервюта като отделни аудио-записи (чрез запис със смарт-телефон, таблет или диктофон). Вторият екип да събере информация в картинен вид. Например, да съберат информация за видовете сгради, които се намират близо до училището – обществени сгради (училище, детска градина, болница и др.), къщи, блокове, магазини и т.н. Екипът трябва да предаде на учителя галерия от снимки (направени със смарт-телефон, таблет или фотоапарат). Третият екип да събере информация в символен вид. Например, да открие и заснеме различни знаци в близост до училището – пътни знаци, табели на сгради и врати, символи от ежедневието, знаци на магазини и фирми, марки на коли и др. Екипът трябва да предаде галерия от снимки, съдържащи отделните знаци и символи. Четвъртият екип трябва да подготви информация в текстов вид по зададена тема. Екипът да направи проучване и да изготви кратък доклад по дадената тема. Докладът да бъде предаден под формата на текст (ако могат, учениците да го въведат в текстов редактор и да го

форматират, или да го напишат на ръка и да акцентират на важните неща в текста чрез подчертаване или използване на различни цветове). След като екипите предадат своите информационни пакети, учителят може да подготви една презентация и да представи събраната информация в мултимедиен формат.

Урок 2-3. Футболисти-програмисти. Среда за програмиране Scratch.
Последователни и повтарящи се действия (преговор).
Входяща диагностика

Уроци за упражнение и диагностика.

Цел. Преговор и затвърждаване на знанията за работа в среда за програмиране Scratch, за последователни и повтарящи се действия. Определяне на входяща диагностика на знанията на учениците.

Задачи:

1. Да се припомнят и обобщят знанията за визуалната среда за програмиран Scratch.
2. Да се затвърдят знанията и уменията за съставяне на алгоритми с последователни и повтарящи се действия.
3. Да се диагностицират знанията на учениците в началото на четвърти клас по теми, изучавани по Компютърно моделиране в трети клас.

Очаквани резултати:

- Познава работното поле и се ориентира във визуалната среда за програмиране Scratch.
- Знае как да променя характеристиките на герои.
- Познава и умее да работи с менюта и бутони.
- Може да състави алгоритми с последователни и с повтарящи се действия.

Учебно съдържание и методически насоки:

За да изпълнява нашите инструкции дадено дигитално устройство, ние трябва да му ги дадем (изкажем, опишем) на език който то разбира. Тези инструкции трябва да се напишат в някаква среда за програмиране. Вече сме работили с блокове, знаем какво е код и команда и сме запознати с работата във блоковата визуалната среда за програмиране Scratch.

От януари 2019 година Scratch 3.0 е новата версията на визуалната среда за програмиране. Версия 3.0 поддържа работа с файлове, създадени с версия 2.0. Така всички работни файлове, подготвени за ученика могат да се използват с двете версии на програмата при работа онлайн и офлайн. Проектите, създадени онлайн със Scratch 3.0 при запазване на компютъра не могат да се отворят със Scratch 2.0. За офлайн работа трябва да се инсталира Scratch 3.0.

Scratch 2.0 продължава да се поддържа. С версия 2.0 може да се работи само офлайн, т.е. програмата трябва да се инсталира на компютъра. Инсталирането може да стане от тази връзка: <https://scratch.mit.edu/download/scratch2>

Scratch 3.0 може да се използва онлайн и офлайн. Инсталацията на програмата може да се направи от следната връзка: <https://scratch.mit.edu/download>

Няма принципна разлика в начина на работа между двете версии. От гледна точка на формиране на алгоритмични умения, няма значение с коя от двете версии ще се работи. Оставяме на учителя да направи своя избор. Постоянното възникване на нови по-добри версии е естествен процес за софтуерната индустрия. Преминаването към по-нова версия за работа е качество, което трябва да се развива у съвременния човек, за да бъде постоянно в крак с новото и актуално съдържание, информация, технология, процедури.

Авторският колектив е разработил всички ресурси в електронния учебник и за двете версии. Така даваме възможност на учителя да направи избор, като осигуряваме работата му с помощни и нагледни материали. Новата версия предлага повече и по-добра графика. Има допълнителни приставки за работа с външни програмируеми устройства. Версията е особено подходяща за допълнителна работа в клубове по програмиране, роботика и проектна работа по интердисциплинарни теми.

В книгата за учителя всички обяснения са дадени паралелно за двете версии, за да сме максимално полезни на педагога. Така учителят има възможност да направи осмислен и съзнателен избор за версията, с която да работи – има три възможности – само 2.0, само 3.0 или и двете версии паралелно. И при трите избора ще бъдат постигнати целите на обучение по предмета Компютърно моделиране.

Интерфейс на средата

Прозорецът на средата Scratch (Scratch 2.0 и Scratch 3.0) се състои от четири основни части:

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.

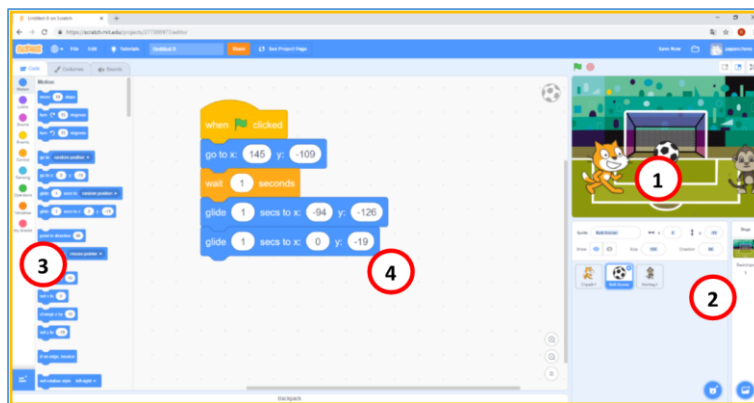
Зона за обектите – герои и сцена. В лявата част на зоната е визуализирана сцената с изборния в даден момент декор. В дясната част на зоната са визуализирани героите, участващи в проекта. В тази зона са разположени и бутони, с помощта на които могат да бъдат добавяни още герои.

Зона за управление на проекта. Това поле има три секции – *Сценарии/Код*, *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 секцията *Сценарии/Код*. 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 секцията *Костюми/Декори*. Към всеки обект можем да зададем и звук. Наличните звукови файлове са в секцията *Звуци*.

Работно поле. Когато е избрана секцията *Сценарии/Код*, 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/Код* на зоната за управление на проекта. Програмен блок може да бъде премахнат с обратно плъзване и спускане към секцията *Сценарии/Код*.



Прозорец на програмата Scratch 2.0



Прозорец на програмата Scratch 3.0

Декорите на сцената и избора на герои става по три основни начина:

- чрез избор от вградената в програмата колекция с декори/герои;
- чрез рисуване на декора/героя във вградения графичен редактор;
- чрез избор на декор/герой от файл на нашия компютър.

Във версията на програмата Scratch 2.0 декор или герой могат да се заснемат директно с камерата на компютъра и да се използват в проекта. Тази функционалност липсва във версия 3.0. Там е включена допълнителна възможност за избор на герой/декор – бутон

„Изненада“.



При щракване върху бутона *Изненада* се зарежда случаен декор или случаен герой.

След като изберем герой, може да променим размера му.

Във версия 2.0 увеличаването или намаляването на размера на героя става чрез съответните бутони в горната част на прозореца на програмата (бутонът със стрелки, обърнати навън увеличава размера на героя, а бутонът със стрелки навътре го намалява).

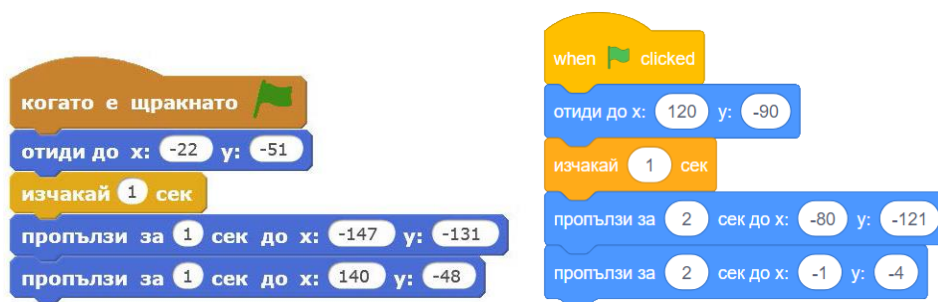
Във версия 3.0 увеличаването или намаляването на размера става чрез задаване на число в полето Size. Числото отговаря на размера в проценти. 100 – задава оригиналния размер на героя. 50 – два пъти по-малък размер. 200 – 2 пъти по-голям размер.

Учениците не са изучавали понятието процент и трябва да разберат начина на използване на инструмента на практическо ниво – чрез изпробване на различни варианти. Колкото по-малко от 100 е числото, което въвеждат, толкова размерът става по-малък. Колкото по-голямо от 100 е числото, толкова размерът на героя става по-голям.

Алгоритъм за изпълнение на задачата:

1. Избираме декор. От галерията се избира декор – футболно игрище с футболна врата.
2. Избираме герои. Избираме три героя (спрайта) – маймунка, коте и футболна топка. Разполагаме ги върху сцената. Уточняваме размера на героите.
3. Програмираме движението на топката¹.
 - Избираме начална позиция за топката – близо до маймуната.
 - Топката трябва да пропълзи за указано време до котето. Използва се командата „Пропълзи за... до ...“. Поставяме топката близо до котето и взимаме командния блок *Пропълзи ...* и го поставяме в кода на топката. Въвеждаме брой секунди за времетраене на движението.
 - Топката трябва да пропълзи от котето до вратата. Поставяме топката във вратата и взимаме командния блок „Пропълзи ...“ и го поставяме в кода на топката. Въвеждаме брой секунди за времетраене на движението.
4. Съхраняваме проекта. (Файл/Запиши като – при работа с инсталирана програма Scratch или Файл/Сваляне на вашия компютър – при работа онлайн).

Задава се стартиране на проекта при щракване върху зеленото знаме. Следователно, трябва да се зададе начална позиция на топката, за да може проектът всеки път да стартира по един и същ начин, а да не се налага ръчно да местим топката близо до маймунката. За целта се използва командата „Отиди до ...“. Следва се същия начин на работа, както с командата „Пропълзи ...“ – героят (топката) се поставя до маймунката (това е началната позиция на топката). Програмата Scratch попълва сама позицията на топката (двете числа в командата „Отиди до ...“). Взимаме командата „Отиди до ...“ и с влачене я поставяме в началото на кода на топката.



Код за задача 2 в Scratch 2.0 и в Scratch 3.0

Учениците изпълняват самостоятелно задача 1. За по-бързо работещите и/или за самостоятелна работа у дома учителят поставя и втора задача – героите да си подадат топката три пъти и след това единия да вкара гол.

¹ Учениците не работят с координатна система и координати, определящи позицията на героя. Използва се функцията на програмата Scratch автоматично да попълва в блоковете за движение текущата позиция на героя. При работа с командите „Отиди до ...“ и „Пропълзи ...“ героят се поставя на позицията, в която трябва да бъде, програмата автоматично е попълнила координатите на позицията му в командните блокове за движение, тези блокове се взимат и се подреждат в кода на героя.

Втори акцент в уроците за преговор е актуализирането на знанията на учениците за последователни и за повтарящи се действия.

Игра по двойки: играе се със зар и пионка върху игрално поле. Първият ученик хвърля зара и поставя пионката върху падналото му се число. Хвърля зара още веднъж – пада се друго число – това е числото, до което трябва да премести пионката, като премине през полетата на лабиринта. Подрежда се алгоритъм, описващ движението на пионката до целта. Използват се картончетата с команди от тетрадката на ученика.

За играта са осигурени картони в тетрадката на ученика – картон с игрално поле и картон с команди за движение (за изрязване).

Актуализира се понятието „цикъл“. Подрежда се цикличен алгоритъм с помощта на картончетата с команди.

Преминва се към работа в средата Scratch. Учениците трябва да изпълнят задача 2 – да се начертае стълбичка за награждаване на спортистите. Открива се повтарящото действие при чертане (нагоре и надясно) и броят на повторения (3).

Припомнят се командите от група Молив. Учителят прави кратка демонстрация чрез подготвените към урока ресурси и се преминава към самостоятелна работа на учениците по задачата.



Код за задача 3 в Scratch 2.0 и в Scratch 3.0

Допълнителна задача. Да се измисли емблема на футболен клуб, която да се чертае като се използва цикъл.

Входяща диагностика

Предвижда се в рамките на 20 минути (втория от двата часа) да се проведе входяща диагностика на знанията на учениците по Компютърно моделиране от трети клас.

Тестът е предоставен в тетрадката на ученика. Критериите и показателите за оценяване са както следва:

Тип на теста - Тестът съдържа 5 задачи с избираем отговор и 3 задача със свободен отговор.

Задача 1 е с избираем отговор, изискващ ученика да познава инструментите в конкретна визуална среда

Задача 2 е със свободен отговор, изискващ от ученика да разбере кода и да опише изпълнението му.

Задача 3 е с избираем отговор, изискващ от ученика да избере правилния код.

Задача 4 е със свободен отговор, изискващ ученика да познава инструментите в конкретна визуална среда.

Задача 5 е с избираем отговор, изискващ ученика да познава инструментите в конкретна визуална среда.

Задача 6 е със свободен отговор, изискващ от ученика да познава бутоните в конкретна визуална среда.

Задача 7 е с избираем отговор, изискващ от ученика познания за работното поле на конкретна визуална среда, къде се намират отделните полета за промяна на характеристики на герои.

Задача 8 е с избираем отговор, изискващ от ученика да познава предоставените от средата бутони и блокове.

Критерии за оценяване

Задача	Компетентности и очаквани резултати	Максимален брой точки
Задача 1	<ul style="list-style-type: none">• Познава работното поле на конкретна визуална среда• Знае къде се намират отделните полета за промяна на характеристики на герои• Ориентира се в цялостния прозорец на визуалната среда• Различава менюта, инструменти и бутони	1
Задача 2	<ul style="list-style-type: none">• Умее да определя вида на блоковете• Умее да разчита готов код	3
Задача 3	<ul style="list-style-type: none">• Умее да реализира промяна на костюмите• Умее да организира цикъл• Различава различни циклични алгоритми	2
Задача 4	<ul style="list-style-type: none">• Познава визуалната среда за програмиране• Познава възможностите на героя да „говори“	2
Задача 5	<ul style="list-style-type: none">• Използва блокове от програмната среда за реализиране на цикличен алгоритъм• Различава различни циклични алгоритми• Разчита чужд код, реализиращ цикъл	2
Задача 6	<ul style="list-style-type: none">• Познава работното поле на конкретна визуална среда• Ориентира се в цялостния прозорец на визуалната среда• Различава менюта и бутони	2
Задача 7	<ul style="list-style-type: none">• Познава работното поле на конкретна визуална среда• Знае къде се намират отделните полета за промяна на характеристики на герои	6

	<ul style="list-style-type: none"> • Различава менюта и бутони • Познава предоставените от средата бутони и блокове 	
Задача 8	<ul style="list-style-type: none"> • Познава работното поле на конкретна визуална среда и се ориентира в нея • Знае къде се намират отделните полета за промяна на характеристики на герои • Различава менюта и бутони 	2

Скала за оценка: Оценка=2+Брой точки/ 5;

Среден (3) – от 5 – 7 точки;

Добър (4) – от 8 до 12 точки;

Много добър (5) – от 13 до 17 точки;

Отличен (6) – от 18 до 20 точки.

Урок 4. Информацията и дигиталните устройства

Урок за нови знания

Цел. Формиране на знания за данни и тяхната обработка и съхранение чрез дигитални устройства.

Задачи:

1. Да се формират знания за предназначението на дигиталните устройства
2. Да се формират знания за данни и начина на съхраняването им в дигиталните устройства.
3. Да се формират знания и умения за разпознаване и сравняване на мерни единици за големина на файлове.
4. Да се затвърдят знанията за безопасна и здравословна работа с дигитални устройства.

Очаквани резултати:

- Разбира основното предназначение на дигиталните устройства.
- Знае как се съхранява информацията в дигиталните устройства.
- Знае как се обработва информацията в дигиталните устройства.
- Разпознава и сравнява мерни единици за големината на файловете, които използва.

Учебно съдържание:

Основното предназначение на дигиталните устройства е да изчисляват, да съхраняват, да намират бързо съхранена информация, да ни помогнат да анализиране информацията, да ни забавляват.

Илюстрациите, предложени в учебника илюстрират основните функции на дигиталните устройства.

- Изчислява – Чрез компютрите хората изчисляват за секунди задачи, които на ръка биха пресмятали дни наред. На илюстрацията е изобразено дете, което смята в

тетрадката си, и дигитално устройство, което пресмята за секунди големи числови изрази.

- Съхранява – Дигиталното устройство съхранява голямо количество информация. Всичките книги на една библиотека могат да се съхранят в цифров вид в паметта на един компютър. На илюстрацията момиченцето се чуди къде ще прибере всички тези книги. Написаното в тях може да се съхрани в паметта на един компютър в цифров вид.
- Търси и намира – Информация, запазена в компютъра може да бъде намерена много бързо. На илюстрацията детето се опитва да намери точно определена информация сред изписаните листове. Рови сред тях и не може да намери търсеното. В компютъра съхранената информация може да се открие по име, по дата на създаване, по мястото на което е запазена и т.н.
- Анализира – Чрез компютърните технологии, хората анализират бързо информацията, съобразяват се с много условия, ограничения и възможности, и успяват да вземат най-правилното решение за своите действия. На илюстрацията момиченцето излиза да играе. Облечено е с летни дрехи, но майката вижда прогнозата за времето и му подава якето, защото се очаква да стане облачно и може да завали и дъжд.
- Забавлява – Компютърът е добър приятел. С него можем да играем, да слушаме музика, да гледаме филми, да четем за интересни случки и събития. На илюстрацията са показани деца, които играят любима компютърна игра и се забавляват заедно.

Информацията в дигиталните устройства се събира, съхранява, обработва и разпространява.

В компютъра информацията се обработва в числов вид, поради което наричаме компютърните устройства – дигитални.

Информацията, която съхраняваме и обработваме с дигиталните устройства наричаме **данни**. Данните се съхраняват във файлове.

Акцент в урока е начина на съхраняване на информация в дигиталната техника. Дискутира се върху организацията на класовете в училище. Учениците в училище са разделени по класове – 1 клас, 2 клас, 3 клас, 4 клас и т.н. Във всеки клас учениците са разделени в паралелки – 4А, 4Б, 4В и т. н. Ако учениците от четвъртите класове са направили различни компютърни проекти, то учителят им ги съхранява по следния начин: проектите на отделен ученик, например на Ели, са събрани в една папка. Тази папка е поставена в общата папка на 4А клас, а папката на 4А клас е поставена в общата папка на всички четвърти класове в училището. Тази организация може да се илюстрира като дърво – стъблото са всички четвърти класове. Всеки от големите клонове е папката на отделния клас – 4А, 4Б, 4В. По-малките клонове са отделните ученици в класа – Ели, Вили, Емо, Крис, Жоро ... от 4А, Митко, Весо, Ники, Гери от 4Б и т.н. А още по-малките клони са проектите, които всеки от учениците е направил – те може да са рисунки, презентации, проекти със Scratch и др.

Такава организация на информацията се нарича йерархична или дървовидна.

Дава се втори пример с дървовидна организация – родословното дърво. То е представено като обърнато наопаки дърво – стъблото и корените са отгоре, а клоните и листата – надолу. В стъблото на дървото е бабата (баба Ани). От стъблото израстват клоните – нейните деца (например, леля Мими и чичо Асен). От децата на баба Ани израстват нови клони – техните деца (Ели и Ники – децата на леля Мими, и Вики – детето на чичо Асен).

По подобен начин се съхранява информацията в дигиталните устройства. Въвеждат се понятията файл и папка. Данните се съхраняват във файлове. Файловата система е дървовидна, подобно на родословното дърво или на организацията в училище. Файловете се подреждат в папки. В една папка може да има файлове, както и други папки.

Следващ акцент в урока е запознаването с мерни единици за информация. Въвежда се единицата байт. Основната мерна единица за информация е Байт (означаваме с 1В). Всеки символ се съхранява в 1 Байт. По-големите единици за съхранение на информация са килобайт (1 KB), мегабайт (1MB), гигабайт (1GB). Дадени са точните връзки между мерните единици, като не се изисква учениците да ги заучават наизуст. $1\text{ KB}=1024\text{ B}$, $1\text{ MB}=1024\text{ KB}$, $1\text{ GB}=1024\text{ MB}$. Обръща се внимание на това, че всяка следваща мерна единица е около 1000 пъти по-голяма от предишната, но връзката не е точно 1000, а 1024. По това мерните единици за измерване на информация се различават от мерните единици за маса или разстояние – килограм-грам; километър-метър и др.

Методически насоки

Урокът започва с диалог и актуализиране на знанията на учениците за дигиталните устройства. Припомнят се знания от предходната година – основни компоненти на дигиталното устройство, входни и изходни устройства.

Преминава се към новото учебно съдържание – изяснява се предназначението на дигиталните устройства. Въвеждат се понятията данни, файл, папка. Въвежда се нова мерна единица – байт. Въвеждат се по-големи мерни единици за информация – килобайт, мегабайт и гигабайт. В тетрадката на ученика са включени задачи за изчисление, при които трябва да се направи преминаване от голяма към по-малка мерна единица за обем информация.

Последният акцент в урока е върху актуализиране и разширяване на знанията на учениците за здравословна работа в богата на дигитални технологии среда.

Знаем, че работата с дигитални устройства не е безопасно. Някои правила за правилно и безопасно използване:

1. На всеки 30 минути работа с дигиталното устройство са необходими поне 10 минути почивка.
2. Работното място трябва да е удобно – мониторът да е на нивото на очите на една ръка разстояние; гърбът да е изправен, краката да са удобно поставени на пода; ръцете да са удобно поставени на плота.
3. Пряката слънчева светлина срещу монитора или очите трябва да се избягва.
4. От време на време поглеждайте навън или си направете малко гимнастика на очите.
5. Редувайте игра навън с работа пред компютъра.

В учебната тетрадка е включена задача, в която по дадени казуси учениците трябва да маркират дали моделът на поведение е правилен или не.

Учителят може да постави проектна задача за седмицата – да се създаде табло с правила за здравословна работа с дигитални технологии. Учениците да изработят картончета с написани правила, които да се закача на общо табло – постоянно достъпно в класната стая. Правилата за здравословна работа с технологиите трябва да се дискутират постоянно в хода на учебната година.

Урок 5. Информацията в съвременното общество

Урок за нови знания

Цел. Усвояване на знанията за значението и ролята на информацията в съвременното общество.

Задачи:

1. Да се придобият знания за ролята на информацията в съвременното общество.
2. Да се усвоят знания и да се придобият умения за правото да се използва информацията.
3. Да се придобият знания за това, че не всяка информация във виртуалното пространство е достоверна.

Очаквани резултати:

- Разбира, че дигиталните ресурси може да не са свободни за използване, копиране и разпространение
- Разбира, че не всяка информация във виртуалното пространство е достоверна.

Учебно съдържание:

За да се ориентираме в околния свят и да можем да определяме действията си ние непрекъснато се нуждаем от информация – за времето, програмата на телевизията, седмичната програма в училище и др. Информацията може да бъде представена под различна форма – върху камък, папирус, хартия, касета, дискета, диск, флаш-памет и др.

Днес информационните дейности се използват в много професии, а информационните услуги и продукти стават все по-търсени и ценени. Общество, в което широко се използват информационни и комуникационни технологии се нарича информационно общество. Основната задача на информационното общество е съхраняването на информацията за културното, историческото и научното наследство на човечеството.

Неотменими права на всеки член на информационното общество са:

- правото за достъп до информация;
- неприкосновеността на личната информация.

Много важни са свойствата на информацията. Някои от най-важните са:

- Достоверност – точно и вярно представяне на първообраза;
- Значимост – важност за описвания обект;
- Пълнота – включване на всички съществени характеристики на обекта;
- Съдържателност – липса на несъществени сведения;
- Актуалност – навременност при получаването;
- Достъпност – възможност за получаване от всички заинтересовани.

Следващ акцент в урока е безопасността в Интернет. Често в мрежата се попада на информация, която е противоречива, провокативна или грешна. С развитието на Интернет и социалните мрежи се появяват и източници, които имат за цел да злоупотребяват с доверието на интернет-потребителя. Ето защо преди да се сподели или да се използват някаква информация от уеб пространството, първо трябва да се провери дали тя е достоверна.

Обикновено причините за разпространяване на фалшива информация е печалбата или политически манипулации. Един от начините за печалба в Интернет са рекламите, върху които потребителите трябва да щракнат. Възникват така наречените „кликбайт“ сайтове. Те привличат посетители по всякакъв начин, защото всяко щракване върху страницата на сайта им носи печалба от реклами. В такива сайтове има фалшиви новини или подвеждаща информация. Обикновено се използват заглавия или изображения, които са атрактивни и подлъгват хората да щракнат върху тях.

Има много анонимни сайтове – без информация за авторите, за организацията, за историята ѝ, за мисията ѝ. Не можем да вярваме на информация от такъв източник или трябва задължително да я проверим. Обикновено в такива сайтове не се посочват източници. Използват се фрази от вида „много учени твърдят ...“, „общоизвестно е, че ...“. Липсват конкретни факти, базирани на конкретни автори и източници.

За да се предпазим от фалшиви новини или измами трябва да разсъждаваме критично и да проверяваме една новина от няколко независими източници. Добре е да си задаваме въпроси от вида: *Ако новината-сензация е истина, не трябва ли да я има и в най-големите новинарски сайтове? Ако цената на някой продукт е прекалено ниска, дали това не е измама?*

Методически насоки

Урокът започва с диалог по темата за предназначението на дигиталните устройства, за съхраняването и обработването на информация. Припомнят се вече усвоени в прения урок знания.

Вниманието на учениците се насочва към нуждата от информация в съвременното общество, общество, което наричаме информационно. Обръща се внимание на правата на всеки член на информационното общество за достъп до информация и за неприкосновеност на личната информация.

Учениците се запознават със свойства на информацията – достоверност, пълнота, значимост, достъпност, актуалност, съдържателност.

Провежда се беседа с учениците, в хода на която те споделят личен опит и мнение за ролята на технологии в живота – в семейството, в училището, извън тях.

Преминава се към темата за информацията в мрежата, като акцентът се поставя върху това дали тази информация е винаги безусловно истинска. Учениците се запознават с термини като фалшиви новини, интернет-измами, кликбайт сайтове, анонимни сайтове и др. Обръща се внимание на нуждата от критично мислене, на важноста да не се проявява наивност, на изграждането на навици информацията да се проверява от два или повече независими източници в Интернет.

Актуализират се знания за безопасно поведение в мрежата, касаещо личната безопасност на учениците. Припомнят се правила от вида:

- В Интернет не давам лична информация като адрес, телефонен номер, имена на родители, името на училището, без разрешение на родителите ми.
- Веднага казвам на родителите ми, ако попадна на информация, която ме смущава. Мога да подам сигнал и на адрес: www.safenet.bg
- Никога не приемам да се срещам с някого, когото „познавам“ от Интернет.
- Никога не изпращам мои и чужди снимки, както и информация без разрешение.
- Не отговарям на съобщения, които са обидни или по някакъв начин ме смущават. Ако получа такива съобщения информирам веднага родителите си.

- Не давам паролите си на никого, освен на родителите ми.
- Искам съгласието на родителите ми преди да сваля или инсталирам нови програми, защото от тях може да се разкрият данни за мен или семейството ми.

Последният акцент в урока е спазването на авторски права. Софтуерните продукти, с които обработваме информация, книгите от които получаваме информация, музиката, филмите – всички те изискват спазване на авторски права.

Софтуерът е продукт на интелектуално творчество и се закриля се от закон за авторското право. Основният обект на правна защита е възпроизвеждането на копия на компютърните програми и публичното им представяне. За да се придобие законно право да се използва дадена програма, трябва потребителят да притежава лиценз.

Видовете софтуер, според правото за неговото използване е:

- Свободни програми, които могат да се инсталират и използват безплатно.
- Програми за временно безплатно използване. След изтичане на този период е необходимо да се закупи лиценз за правото на използване.
- Демо-програми, които са предназначени да демонстрират възможностите на определена програма преди да бъде закупена и използвана.
- Версии за тестване, които се предоставят безплатно на потребителите да ги тестват.
- Платени програми, които могат да се инсталират и използват само след закупуване.

На учениците се поставя задачата да проучат и да определят какви по вид на право за ползване са програми и среди, като Code.org, Scratch, Windows.

Урок 6. Условия за безопасност в дигитална среда

Урок за нови знания

Цел. Формиране на знанията за безопасна работа в дигитална среда.

Задачи:

1. Да се актуализират и надградят знанията за дигитална самоличност.
2. Да се формират знания и умения за безопасна работа в дигитална среда.

Очаквани резултати:

- Не предоставя лични данни в дигитална среда.
- Познава по-известни заплахи при работа в дигитална среда.
- Знае как да получи помощ при необходимост.
- Познава етичните норми при работа в онлайн среда.
- Познава ползите и вредите от използването на дигитални устройства за околната среда.

Учебно съдържание:

В интернет има много полезна информация, забавни игри и много приятели. Но, в Интернет може да бъде и опасно. Някои хора правят фалшиви профили и ги използват, за да говорят и действат от името на друг човек, като по този начин искат да му навредят. Други лъжат кои са. Представят се за по-млади, за по-успели, подвеждат човека, с който

общуват. В Интернет има много сайтове с неподходящо съдържание за деца. В някои има неприлични сцени, грозни снимки, грозни и обидни думи – неща, които могат да ни уплашат или да ни накарат да се срамуваме. За всичко, което ни притеснява в мрежата трябва да споделяме с приятели и с възрастни.

Илюстрациите към урока показват често срещани заплахи, за които учителят и учениците трябва да проведат беседа, за да обсъдят опасностите и да търсят начин за тяхното преодоляване.

В Интернет не е сигурно кой стои срещу нас. Много хора създават неистинско мнение за себе си – представят се като по-млади или по-стари, по-успели, по-известни и др.

Друга опасност са фалшивите профили. Измамникът си създава профил с името и снимката на друг човек и от негово име пуска информация или води разговори с хора в Интернет. Така вреди на истинския човек, чийто профил е откраднал.

Третата често срещана опасност е чуването или виждането на нецензурирана информация със сексуално съдържание или съдържание показващо жестокост и бруталност. Тези ситуации могат да нанесат сериозни психологически вреди на Интернет потребителите.

Методически насоки

Урокът започва с актуализиране на знанията на децата за дигитална идентичност. Припомнят се понятията псевдоним, аватар, потребителско име, парола.

Беседата с учениците се насочва към темата за опасностите в интернет. Обсъждат се някои често срещани заплахи – лъжливите профили, фалшиви профили и сайтове с неподходящо за деца съдържание.

Обсъждат се модели на поведение в мрежата. Първият модел на поведение е свързан с реакция в случай, че някой ни притеснява в Интернет. В този случай жертвата на тормоз трябва да направи едно или няколко от следните действия:

- Да подаде онлайн сигнал през сайта <https://www.safenet.bg>.
- Да блокира този, който го тормози.
- Да сподели с възрастен.
- Да се обади на телефон 124 123.

Вторият пример задава етични норми на поведение в мрежата, разглеждайки случая, в който публикуваме информация в Интернет. Следва да постъпим по следния начин:

- Ако на снимката сме с приятели, трябва да ги попитаме дали можем да я качим в Интернет.
- Качваме информация, която е интересна и/или полезна.
- Когато публикуваме в Интернет, трябва да използваме учтив език, както, когато говорим с родители, учители и приятели.

Припомнят се някои основни правила за безопасност в Интернет:

- Никога не давам лична информация като име, адрес, телефон, училището, в което уча, месторабота или телефон на родителите си, без тяхно разрешение.
- Никога не изпращам свои снимки или снимки на твои близки, без тяхно разрешение.
- Не си уреждам срещи с непознати в интернет и не ходя на такива срещи без знанието на родителите ми.

- Винаги внимавам, когато разговарям с други в интернет. Знам, че в дигиталното пространство хората се представят за такива, каквито не са.
- Винаги съм вежлив и уважавам правата на другите.

Вниманието на учениците се насочва към възможностите да се подаде сигнал за опасност в интернет чрез подаване на информация през сайта www.safenet.bg или чрез позвъняване на национален телефон 124 123.

Последният акцент в урока е поставен върху влиянието на технологиите върху околната среда и живота на хората. Желанието ни да имаме най-новите технологии струва скъпо на природата. От една страна се произвеждат огромно количество дигитални устройства, за което се използват ценни ресурси, някои от които са ограничени като запаси в природата. От друга страна старите дигитални устройства се превръщат в огромно количество опасни битови отпадъци. Когато електронните устройства не се рециклират правилно, изхвърлянето им на боклука води до попадането на токсични отпадъци в почвите, водите и въздуха. Оттам отровата отива в храната ни. Затова е изключително важно дигиталните устройства да се рециклират внимателно и отговорно.

Освен, че влияят негативно върху природата, дигиталните устройства имат опасно влияние и в социалната сфера – за здравето на хората и за взаимоотношенията между тях. Все по-голям е броят на децата със затлъстяване. Макар дигиталните устройства да не са единствена причина, определено имат своя принос за по-малко движение, липсва на физическа активност и др. Намалява живота общуване между хората. Когато останат сами хората не търсят приятел, събеседник, компания – а намират такива в мрежата или с любима игра, или разглеждат социални мрежи и др.

В хода на урока учителят може да организира разиграването на две игри.

Игра 1. Кой съм аз?

Всеки ученик описва на лист хартия свой аватар – външен вид, занимания, интереси. При описанието може да се постави задача едно или две от твърденията да не са верни.

Всички описания се събират и последователно се теглят едно след друго. Учениците се опитват да разпознаят своя ученик по описания аватар. Целта да установят че е много трудно да разкрият истинската самоличност на стоящия зад аватара човек.

Игра 2. Вярно-Невярно

На картончета са записани начини на поведение в Интернет. Избира се случайно картонче и се определя дали написаното е вярно или невярно. Може да се въведе действие за правилно или неправилно поведение. Примерни текстове за картончетата:

- Когато съм в интернет винаги се представям с истинското си име
- Никога не давам личните си данни на непознати в интернет.
- Нормално е да споделя, ако някой от моите познати в чата ме притесни с въпроси или отношение
- В интернет не ме познават и мога да се държа както си искам
- Винаги съм вежлив и уважавам правата на другите.
- Не си уреждам срещи с непознати в интернет и не ходя на такива срещи без знанието на родителите ми.
- Никога не изпращам свои снимки или снимки на твои близки, без тяхно разрешение
- Няма нужда да притеснявам възрастните с моите проблеми в чата.
- Мога да споделя своята парола с приятелите си.

- Не е лошо, ако прочета пощата на приятелите си.
- Никога не отварям чужда поща и не пиша от чуждо име.

Урок 7. Блок за разклонение

Урок за нови знания

Цел. Формиране на знания за логически оператори.

Задачи:

1. Да се формира понятие за условни действия.
2. Да се реализира пропедевтика на работа с логически оператори Ако-Тогава (if-then) и Ако-Тогава-Иначе (if-then-else)
3. Да се реализира последователност от действия, в зависимост от верността на поставено условие.

Очаквани резултати:

- Определя дали дадено твърдение е истина или лъжа (true/false).
- Управлява действия в зависимост от верността на дадено условие.

Учебно съдържание:

В хода на обучение по Компютърно моделиране се формират основни знания от информатиката. След усвояването на последователни и на повтарящи се действия, следва запознаване с условните действия. Условие може да се включи както в алгоритъм с последователни действия (блок за разклонение), така и в алгоритъм с повтарящи се действия (цикъл с условие).

В обучението в четвърти клас учениците се запознават с използването на блок за разклонение.

Блоковете за разклонение са два вида:

1. Блок „Ако – Тогава“ (If – Then)
2. Блок „Ако – Тогава – Иначе“ (If – Then – Else)

Блокът „Ако – Тогава“ има следния формат:

Ако Условие, Тогава Действие.

Условието може да бъде изпълнено или не. Условието се задава чрез твърдение, което може да бъде вярно или невярно. Ако условието е изпълнено, казваме, че е вярно (на английски език – True). Ако условието не е изпълнено, казваме, че условието е Лъжа (на английски език False).

Когато условието е Истина се изпълнява действието, записано в командата. Ако условието не е изпълнено, т.е. е Лъжа, то записаното действие не се изпълнява.

Блокът „Ако – То – Иначе“ има следния формат:

Ако Условие, То Действие 1, Иначе Действие 2

Когато условието е Истина се изпълнява записаното в командата Действие 1. Ако условието не е изпълнено, т.е. е Лъжа, то Действие 1 не се изпълнява, но се изпълнява Действие 2.

И в двата случая, ако блокът за разклонение е част от програмен код, след като се изпълни условната команда се изпълнява следващата команда в кода.

Методически указания

Урокът може да започне с условна команда, която издава учителят. Например, учителят е приготвил изненада, която ще даде, ако учениците изпълнят някакво условие:

Ако запазите тишина за 20 секунди, **тогава** ще Ви дам карти и ще играем с тях през часа.

Учителят съобщава за изненадата и следи реакцията на класа. Ако учениците не запазят тишина, не дава карти. Изчаква и отново задава командата. Повтаря докато постигне желаната тишина за 20 секунди. След това изважда комплект карти и го поставя пред учениците.

Учителят обяснява действията си. Той/Тя е задала условно действие. Условието е тишината за 20 секунди. Ако има тишина, условието е изпълнено и картите са на масата. Ако няма тишина, няма карти. За по-добра нагледност, учителят използва две флагчета – червено и зелено. Зеленият флаг се вдига, когато условието е изпълнено, т.е. е Истина. Червеният флаг се вдига, когато условието не е изпълнено, т.е. е Лъжа. Повтаря отново командата. Ако класът запази тишина, вдига зеленото знаме. Ако не – вдига червеното знаме. На този етап при вдигнато зелено знаме има следващо действие – учителят дава картите. Ако е вдигнат червения флаг – няма действие.

Премайна се към втори пример. Учителят хвърля зар и дава следната команда: *Ако се падне четен брой точки, **тогава** момичетата да станат прави.* Учителят хвърля зара. Нека се паднат 4 точки. 4 е четно число. Следователно условието на командата е изпълнено. Тогава трябва да се изпълни действието – момичетата трябва да станат прави. Ако се паднат 3 точки – условието не е изпълнено – всички ученици остават седнали по местата си.

Премайна се към игра с картите. Учителят осигурява по едно тесте карти за двама ученици. Играе се с картите от 2 до 10 в четирите бои. Примерна игра: Картите се разбъркват. Слагат се на купчина по средата. Първият играч обръща най-горната карта. Ако е червена, тогава той/тя печели една точка. Играе следващия играч. Правят се по 10 хода и играта спира. Печели играчът, който има повече точки.

В тетрадката на ученика е предвидено поле, в което учениците да записват своите точки в хода на играта.

Учителят стартира отново играта със зара, но този път въвежда нови правила. Хвърля зара. Ако се падне четно число, тогава стават момичетата, иначе – стават момчетата.

Премайна се към изпълнение на командата. Учителят хвърля зара – нека се падне 3. 3 не е четно число. Следователно условието е Лъжа и учителят вдига червения флаг. В този случай трябва да станат момчетата. Обяснява се отново действието на командата „Ако – Тогава – Иначе“. Хвърля се отново зара няколко пъти, докато всички ученици разберат смисъла на условната команда от този вид.

Премайна се към игра с картите, като този път се въвежда по-сложно условие. Например: Първият играч тегли най-горната карта. Ако е червена, тогава играчът печели една точка, иначе – точка печели другият играч.

В тетрадката на ученика е предвидено място, където да се записва резултата от играта в новия ѝ вариант. Учениците играят по двойки, а учителят проследява играта и коригира отчитането на точките при нужда.

В края на часа се прави обобщение на начина на изпълнение на командите с условие в двата им вида: „Ако – Тогава“ и „Ако – Тогава – Иначе“.

Урок 8. Разклонен алгоритъм

Урок за нови знания

Цел. Формиране на знания и умения за създаване на разклонен алгоритъм по дадено задание.

Задачи:

1. Да се формират знания и умения за работа с условен команден блок (Ако-Тогава).
2. Да се състави разклонен алгоритъм, реализиращ различни последователности от действия в зависимост от верността на зададено условие.

Очаквани резултати:

- Сглобява крайна последователност от блокове, реализираща разклонен алгоритъм.

Учебно съдържание:

В процеса на формиране на алгоритмично мислене важен момент е умението за работа с блок-схеми. Блок-схемата се състои от блокове, които графично визуализират логиката на алгоритъма. След като се направи блок-схема на алгоритъма, той може да се превърне в код на произволен език за програмиране. Блок-схемата е и универсален език между хора, които или не знаят езици за програмиране, или знаят различни езици за програмиране.

Въвеждат се блокът за елементарно действие с форма на правоъгълник и блок за логическо условие с форма на ромб. От блока за логическо условие излизат два изхода – ДА и НЕ, които определят хода на действията в алгоритъма според това дали условието е изпълнено (т.е. е Истина) или не е изпълнено (т.е. е Лъжа).

Методически насоки

Урокът започва с беседа по темата „Пътешествие“. Учениците споделят дали и къде са пътували извън страната ни, споменават известни забележителности. Учителят задава същата команда, както в предходния урок: Ако запазите тишина за 20 секунди ще играем игри и ще пътуваме по света. Използва червения и зеления флаг. Ако учениците запазят тишина, вдига зеления флаг. Ако не – червения. Повтаря командата докато постигне тишината за 20 секунди, за да вдигне зеления флаг.

Учителят предлага на учениците следната игра. Той/Тя задава въпрос. Всеки ученик отговаря на въпроса за себе си и ако отговорът е ДА, тогава става прав, ако отговорът е НЕ, тогава остава седнал.

Поставя въпроса: Знаете ли кои са съседните държави на България? – стават учениците, които знаят. Учителят дава думата на избран ученик, за да назове държавите (Македония, Сърбия, Турция, Гърция, Румъния). Всички ученици сядат по местата си.

Учителят поставя нов въпрос: Знаете ли кои са столиците на съседните ни държави? – стават учениците, които знаят столиците. Учителят посочва един от правите ученици, да назове столиците на петте държави.

Преминава се към работа в учебника. Вниманието на учениците се насочва към първата задача – да се запишат съседните държави на България и техните столици. Дискутира се върху алгоритъма за решаване на задачата. За да може един човек да реши задачата, той първо трябва да си отговори на въпроса дали знае кои са държавите. Ако знае държавите, трябва да си отговори на въпроса дали знае техните столици. Ако и на двата въпроса отговори положително, тогава може да пристъпи към изпълнение на задачата – да запише държавите и столиците им. Но ако на някой от въпросите отговори отрицателно, тогава ще трябва да направи допълнителни действия – да намери информация, за да попълни знанията си.

Алгоритъмът за решаване на поставената задача е представен в учебника чрез схема. Учителят запознава учениците с понятието „разклонен алгоритъм“. Разклоненият алгоритъм е като кръстопът. Според определено условие, избираме един или друг път, за да продължим напред. Задаваме въпрос. Ако отговорът е ДА изпълняваме едни действия, в противен случай – други действия.

Учителят обяснява схемата. Въвежда понятието „блок-схема“. Блок-схемата описва логиката на алгоритъма и начина на неговото изпълнение. Запознава учениците с основните блокове в схемата. Алгоритъма започва с блок „Начало“ и завършва с блок „Край“. Другите основни блокове са с форма на правоъгълник и с форма на диамант (ромб). В правоъгълника записваме елементарни действия. В блока с форма на ромб записваме условие. Условието е под формата на въпрос, който има отговор Да или Не.

Записан с думи, алгоритъмът от схемата е следният:

1. Начало
2. Ако знаеш кои са съседните държави на България, тогава премини към точка 2. Иначе, прочети и ги научи. Премини към точка 1.
3. Ако знаеш столиците на съседните държави, тогава ги запиши в тетрадката. Иначе, прочети и ги научи. Премини към точка 2.
4. Край

Всеки от учениците изпълнява самостоятелно алгоритъма и записва в тетрадката държавите и техните столици.

В учебника населението на държавите е записано приблизително в милиони или хиляди души население. Информацията за население на държавите е взета от последните преброявания, както следва:

<http://www.turkstat.gov.tr/Start.do> Турция през 2018 г. 82 003 882 население

<http://www.statistics.gr/el/2011-census-pop-hous> Гърция през 2011 г. 10 816 286 население

<http://popis2011.stat.rs/> Сърбия през 2011 г. 7 186 862 население

https://bg.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%A0%D1%83%D0%BC%D1%8A%D0%BD%D0%B8%D1%8F Румъния през 2011 19 042 936

https://bg.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%9A%D0%B8%D0%BF%D1%8A%D1%80 Кипър през 2011 г 840 407 население

https://nso.gov.mt/en/publicatons/Publications_by_Unit/Documents/C5_Population%20and%20Migration%20Statistics/Demographic_Review_2014.pdf /стр.8/ Малта през 2014 г. 429 344 население

<https://statistiques.public.lu/fr/population-emploi/rp2011/caracteristiques-personnelles/index.html> Люксембург през 2011 г. 512 353

<http://www.stat.gov.mk/> Северна Македония през 2017 г. 2 075 301 население

В учебника числата са дадени като порядък в хиляди или милиони. Задачата е подходяща за поставяне на учениците, за да намерят самостоятелно информация за броя на населението в разглежданите държави.

Следващата задача се състои в следното – по даден алгоритъм, описан чрез блок-схема, да се разбере неговото действие. В резултат да се попълни таблица в тетрадката на ученика. Правилното попълване на таблицата зависи от правилното разбиране и изпълнение на алгоритъма, описан чрез блок-схема.

Задачата е да се подреди пъзел 3x3. На всеки ред на пъзела се подреждат по три парчета. В първото е записана държава, във второто – нейната столица, а в третото – броят на населението в държавата. Подреждат се трите най-големи по население държави в европейския съюз (Великобритания е изключена, поради брекзит) – Германия, Франция и Италия.

За задачата-игра се използва Картон 3 от тетрадката на ученика. Предварително трябва да се изрежат парчетата от пъзела. Отделят се парчетата с имена на държави – Германия, Франция и Италия. Останалите парчета се разбъркват. Взима се случайно парче с име на държава. Проследява се алгоритъма, описан с блок схемата. Нека е избрана държава Франция. Парчето се подрежда първо за реда. Следваме алгоритъма. Проверяваме дали държавата е Германия. Отговорът е НЕ. Преминаваме по разклонението за НЕ. Проверяваме дали държавата е Франция. Отговорът е ДА. Преминаваме по разклонението за ДА. Изпълняваме действието – намираме парчетата от пъзела, на които пише Париж и 67 мил. Подреждаме парчетата на реда до парчето Франция.

Алгоритъмът се повтаря 3 пъти, за да се нареди целия пъзел. След като пъзелът е нареден учениците попълват таблицата в тетрадката. Проследявайки пъзела по редове, попълват държавата, нейната столица и броя на населението ѝ. Учениците получават допълнителна информация за трите най-големи по население държави в ЕС. Подреждат ги по големина.

За самостоятелна работа се поставя задача 3. В учебника е даден подреден пъзел. От всеки ред получаваме информация за трите най-малки по население държави в ЕС – това са държавите с по-малко от 1 милион население – Кипър, Люксембург и Малта.

В картон 3 са дадени блокове, с които ученикът да подреди самостоятелно блок-схема, подобна на тази от задача 2, чрез която да опише подреждането на пъзела от задача 3. Във всеки блок от пъзела трябва да запише съответните команди и условия.

В края на часа се прави обобщение на свършената работа.

Урок 9. Знам и мога

Урок за упражнения

Цел. Затвърдяване на знанията за информация, дигитална идентичност и разклонен алгоритъм.

Задачи:

1. Да се затвърдят знанията за информацията и дигиталните устройства.
2. Да се затвърдят знанията за дигитална идентичност и правилата за поведение в Интернет.
3. Да се затвърдят знанията за построяване на разклонен алгоритъм.

Очаквани резултати:

- Познава различни видове информация.
- Познава входящи и изходящи устройства.
- Разбира и тества разклонен алгоритъм.
- Знае правила за безопасна работа в Интернет.

Учебно съдържание:

В урока се обобщават научените в първата част знания и умения. Тематично урокът разглежда пътешествията из България. В първата задача трябва да се открият различни видове информация.

В първата задача учениците трябва да различат между звукова, графична и текстова информация. Чрез звуковата информация се развиват знанията за народна музика. Чрез изображения са представени манастирите Троянски, Рилски, Бачковски и Дряновски. Чрез текст е дадена информация за реките Дунав и Марица.

Във втората задача се затвърждават знанията за входящи и изходящи устройства. С входящите устройства подаваме информация към компютъра, а чрез изходящите получаваме информация от компютъра. Учениците трябва да определят кои от изобразените устройства са входящи и кои – изходящи. Могат да се посочат примерни и извън тези от учебника.

С третата задача се развиват уменията за работа с разклонен алгоритъм. Задачата е да се открият две грешки. В третия блок за условие са разменени изходите за лъжа и истина. Третата команда е погрешна – Ани трябва да отиде на английски, а не на волейбол.

В последната задача са дадени правила и модели за поведение, които не са маркирани правилно като верни и неверни. Учениците трябва да коригират допуснатите грешки.

Урок 10. Блокове за аритметични действия. Случайни числа

Урок за нови знания

Цел. Формиране на знания за логически оператори.

Задачи:

1. Да се формира понятие за аритметични оператори в среда за програмиране.
2. Да се формира понятие за избор на случайни числа в среда за програмиране.
3. Да се формират умения за работа с блокове за аритметични операции и избор на случайни числа във визуална среда за програмиране.
4. Да се формират знания и умения за влагане на блок в блок за получаване на комплексни команди.
5. Да се формират знания и умения за операцията свързване (concatenate) в среда за програмиране

Очаквани резултати:

- Познава аритметични оператори и блокове, които ги представят.

- Използва блокове за избор на случайни (random) числа в интервал.
- Влага блок в блок.
- Познава допълнителни операции.

Учебно съдържание:

Едно от основните предназначения на дигиталните устройства е да изчислява. Всяка среда за програмиране има команди, с помощта на които да можем да съставим произволен числов израз и да пресметнем неговата стойност.

В средата за програмиране Scratch има отделна група в секция Сценарий/Код – групата *Оператори*, в зелен цвят. В тази група се намират блоковете за аритметични операции (събиране, изваждане, умножение и деление), блоковете за логически операции (и, или, не), както и допълнителни блокове – за избор на случайни числа, за съединяване и др. Символите, с които се означават аритметичните операции в езиците за програмиране са: (+) – събиране; (–) – изваждане; (*) – умножение; (/) – деление. Съответно блоковете за аритметични действия са:



Във всеки от кръговете може да се запише число или да се вложи друг блок. Например, числовите изрази $20+10 =$ и $20 + 2.5 =$ могат да се запишат чрез аритметични блокове по следния начин:



Методически насоки:

Вниманието на учениците се насочва към аритметичните изрази – коментират се действията и знаците на аритметичните операции. Припомня се функцията на дигиталните устройства да изчисляват многократно по-бързо от човека.

Учителят запознава учениците с новата група командни блокове – групата *Оператори*. Тази група е в зелен цвят. В нея се намират блоковете за аритметични операции. Въвеждат се символите, с които се означават аритметичните операции в средата за програмиране Scratch.

Преминава се към работа по задача 1. Подготвен е работен файл по компютърен проект на тема „Горското училище“. По него учителят поставя въпроси и оставя учениците да поработят самостоятелно в рамките на 10 минути, за да достигнат сами до отговорите им. Целта е чрез отговора на поставените въпроси да се разбере логиката на проекта и да могат след това учениците да довършат кода, според поставеното в задача 1 условие.

Примерни въпроси:

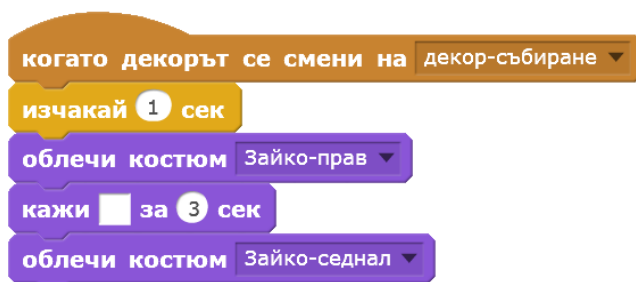
- Колко декора има в проекта? Кои са те? (Декорите са 5 – един за начало и по един за задача от събиране, изваждане, умножение и деление).
- Колко герои (спрайтове) има в проекта? (Героите са 5 – 4 бутона – по един за всяка от задачите и Зайо).
- Колко костюма има Зайо? (Костюмите на Зайо са два – седнал и прав)
- Как се сменят декорите? (При щракване върху някой от червените бутони, декорът се сменя с декора, на който на черната дъска е записана същата задача, която е записана и върху бутона)

- Какво се случва при щракване на зеленото знаме? (При щракване на зеленото знаме се зарежда начален декор – при него няма нищо записано на черната дъска. При щракване на зеленото знаме Зайчо е седнал, т.е. облича костюм Зайчо-седнал).
- Какво е програмирано да прави Зайчо при смяна на декора? Например, какво прави Зайчо, когато декорът се смени с декор за събиране? (При зареждане на съответния декор, Зайчо изчаква 1 сек., става прав и след това сядат).

В хода на самостоятелната работа учениците тестват кода и всеки самостоятелно трябва да достигне до отговор на колкото може повече от поставените въпроси.

След определеното време учителят дискутира по логиката на проекта „В горското училище“ с учениците до пълен отговор на поставените въпроси.

Обръща се внимание на кода на Зайчо, който в работния файл изглежда така:

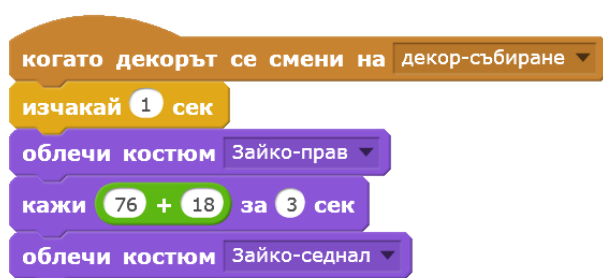


Учениците трябва да довършат кода, така че според задачата Зайчо да я прочете, да я пресметне и да каже отговора ѝ.

За пресмятане на задачата $76+18$ се използва блока за събиране от групата Оператори

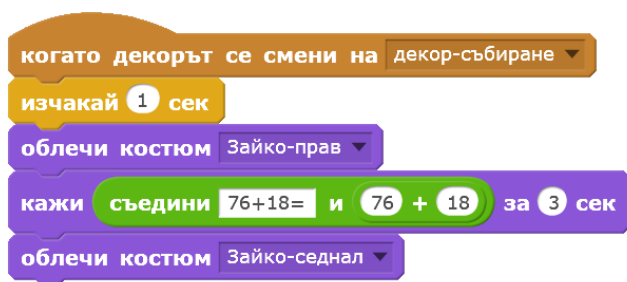
$$76 + 18$$

Този блок се поставя в командата „Кажич“:



Тогава Зайчо става, казва 94 и сядат.

Вниманието на учениците се насочва към командата „Съедини“, която се намира в същата група – *Оператори* и е зелена на цвят. Учителят показва как с помощта на тази команда може да се съединят в едно изречение текст, който въвеждаме от клавиатурата с число, което компютърът изчислява с помощта на блоковете за аритметични действия. Ако в първото поле на командния блок „Съедини“ въведем запис на задачата от действие събиране, а във второто поле поставим блок за аритметични действия (събиране) на двете числа, ще се получи следният резултат:



След демонстрацията на учителя, учениците програмират поведението на Зайко само за декора събиране в работния файл към проекта. Довършването на проекта ще се направи в следващите часове за упражнение за затвърждаване на знанията за работа с блокове за аритметични операции.

Втори акцент в урока е формирането на знания за избора на случайно число в средата за визуално програмиране Scratch.

В групата **Оператори** има команден блок „Избери случайно от ... до ...“. С помощта на този оператор се избира случайно цяло число в посочения интервал. Например, с командния блок



се избира случайно цяло число от 1 до 10 включително.

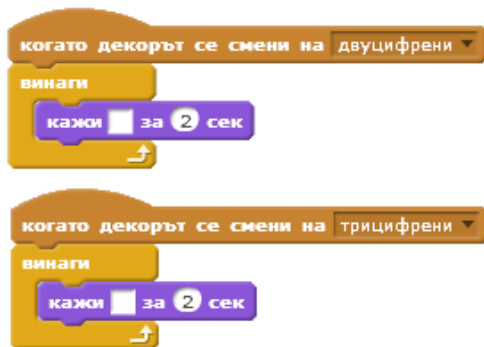
Преминава се към работа по задача 2. Учителят прави анализ на логиката на проекта заедно с учениците. По предварително разработения проект, формулира въпроси, чрез отговорите на които се изяснява как работи програмата. Така учениците ще могат самостоятелно да довършат програмния код по подготовения за тях работен файл към проекта „Училище сред ледовете“.

Проектът за задача 2 е на тема „Училище сред ледовете“. Учителят пингвин изпитва малките пингвини. Едната им задача е да дадат примери за двуцифрени числа, а другата задача – да дадат примери за трицифрени числа.

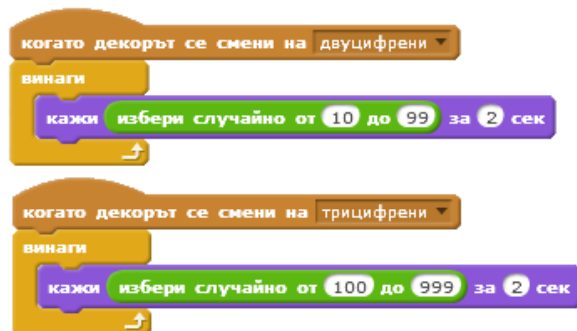
Въпросите, с помощта на които учителят и учениците разбират логиката на проекта са:

- Колко декора има в проекта? Кои са те? (Декорите са 3 – един за начало (декор „начало“) и по един за избор на двуцифрени (декор „двуцифрени“) и на трицифрени числа (декор „трицифрени“)).
- Колко герои (спрайтове) има в проекта? (Героите са 5 – 2 бутона – по един за избор на двуцифрени или трицифрени числа и 3 пингвина – един учител и двама ученици).
- Как се сменят декорите? (При щракване върху бутона $9 < ? < 100$ се зарежда декор за избор на двуцифрени числа – декор „двуцифрени“. При щракване върху бутона $99 < ? < 1000$ се зарежда декор за избор на трицифрени числа – декор „трицифрени“.)
- Какво се случва при щракване върху зеленото знаме? (При щракване върху зеленото знаме се зарежда начален декор – при него няма нищо записано на черната дъска.).
- Какво са програмирани да правят малките пингвини? Например, какво правят при декор „двуцифрени числа“? (При зареждане на декор „двуцифрени“, при щракване върху пингвинче, то „казва“ случайно двуцифрено число.).

Вниманието на учениците се насочва към кода на пингвините-ученици. В работния файл на учениците подготвения код е следния:



Учителят прави демонстрация на работата с командния блок за избор на случайно число. Учениците работят самостоятелно. Задачата им е да допишат програмния код до вида:



Кодовете се тестват. Проектите се съхраняват. Прави се обобщение на наученото в урока.

Урок 11. Блокове за аритметични отношения и логически оператори

Урок за нови знания

Цел. Формиране на знания за логически оператори.

Задачи:

1. Да се формират знания за оператори за сравняване – по-голямо, по-малко, равно.
2. Да се формират знания за логически оператори – и, или, не.
3. Да се формират умения за работа с блокове за сравняване и логически операции във визуална среда за програмиране.
4. Да се формират знания и умения за влагане на блок в блок за получаване на комплексни команди.
5. Да се формират знания и умения за въвеждане от команден ред в среда за програмиране Scratch (командата „Питай ... и чакай“).

Очаквани резултати:

- Използва блокове за сравняване на числа.
- Използва блокове за логически оператори.
- Влага блок в блок.

– Познава допълнителни операции.

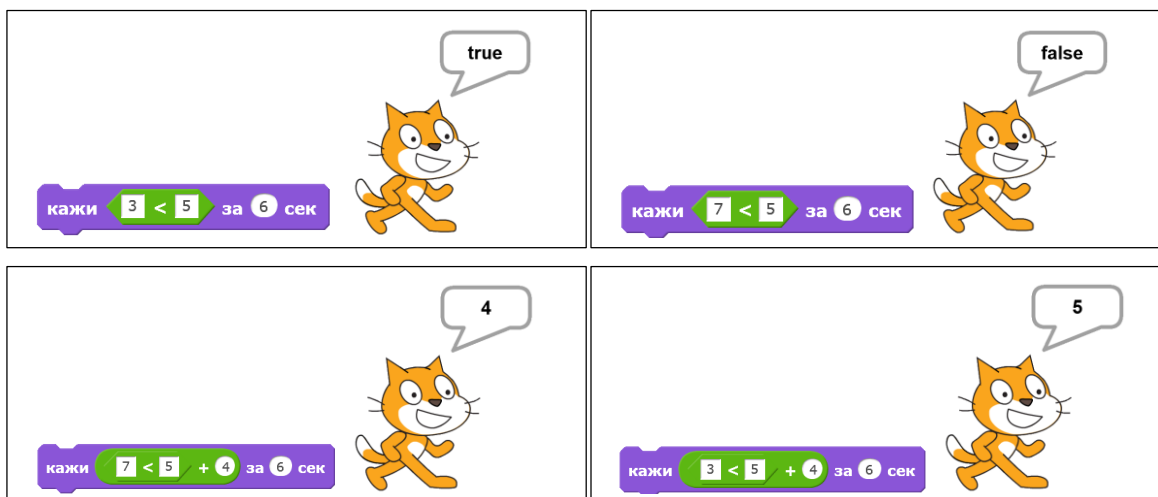
Учебно съдържание:

Работата с числови данни е тясно свързана с тяхното сравняване. В групата с командни блокове **Оператори** има блокове за аритметични отношения или блокове за сравняване на числа – блоковете „по-малко“, „по-голямо“ и „равно“. При изпълнение на командите за сравнение се получава резултат „Истина“ или „Лъжа“. Ако равенството или неравенството е вярно, то стойността на операцията е „Истина“ или „True“. Ако е невярно числово равенство или неравенство, тогава стойността на операцията е „Лъжа“ или „False“.



Блокът за сравняване може да се вложи във всеки друг блок. Ако се вложи в команди като Кажи ..., Мисли ..., тогава резултатът е текст – True или False. Ако се вложи на мястото на число, тогава *Истина* се пресмята като 1, а *Лъжа* се пресмята като 0, т.е. *Истина* = 1, *Лъжа* = 0. Например при събиране на *Истина* с 4, резултатът ще е 5. При събиране на *Лъжа* с 4, резултатът ще е 4.

Примери:



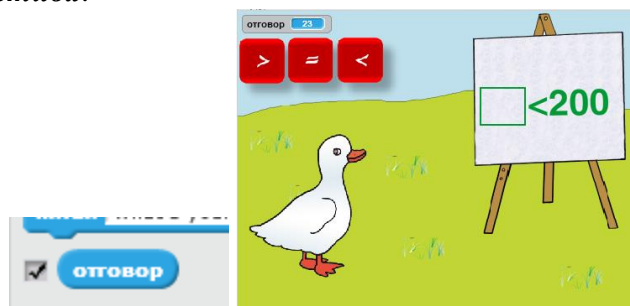
Най-често блоковете за сравняване се влагат в блоковете за разклонение. Те задават условието на командния блок за разклонение. Според това дали условието е истина или лъжа се изпълнява едно или друго действие. В блоковете за разклонение могат да се влагат само команди, чийто резултат е *Истина* или *Лъжа* (формата на тези блокове е със заострени краища). В условието на блок за разклонение не може да се вложи, например, блок за аритметично действие.



В групата с командни блокове **Сетива** има команден блок за въвеждане на текст от клавиатурата – командният блок „*питай и чакай*“.

При стартиране на командата „*Питай ... и чакай*“ на екрана се появява ред, в който от

клавиатурата се въвежда отговор на поставения въпрос или дадената задача. След въвеждане на отговора се натиска клавиша Enter. Въведеното от клавиатурата се запомня в бутона „Отговори“. Ако искаме да виждам какво е запомнено в *Отговор*, поставяме отметка в групата *Сетива*:



Така в горния ляв ъгъл на сцената се появява подвижен информационен бутон, в който виждаме последно въведено от клавиатурата число или текст.

Друга важна част от блоковете в групата *Оператори* са блоковете за логически действия. Следващ акцент в урока е запознаването с блоковете на логическите операции. Често условията, от които зависят действията ни не са свързани само с едно ограничение, а са комбинация от отделни условия. Например, условието може да изисква да се случват две неща едновременно – числото да е по-малко от 10 и да е четно. (В този случай имаме логическата операция Конюнкция или логическото И). Може да имаме условие, което да е вярно при две възможности – например, когато числото е 4 или когато числото е 5. (В този случай имаме логическа операция Дизюнкция или логическо ИЛИ). Или пък искаме числото да не е 4 (В този случай имаме логическа операция Отрицание, или логическо НЕ). Във всички тези случаи използваме логическите операции – И, ИЛИ, НЕ.

Резултатът от блоковете за логически операции е също Истина или Лъжа, както резултатът от блоковете за аритметични отношения. И всичко, казано за блоковете за аритметични отношения важи и за блоковете за логически операции – те имат същата форма и могат да се влагат във всички блокове (в правоъгълни, в кръгли и в ромбоидни места за попълване на командни блокове).

Логическото И е истина, ако и двете условия в операцията са истина. Например (*Вярно е, че 7 е по-голямо от 2, и че 4 е равно на 4.*):

$$7 > 2 \text{ и } 4 = 4$$

Логическото ИЛИ е истина, ако поне едно от условията е истина. Например, ИЛИ е истина в случаите (*Вярно е, че 3 е по-малко от 5 или 4 е по-голямо от 6.*):

$$3 < 5 \text{ или } 4 > 6 \quad 3 < 5 \text{ или } 4 = 4$$

Логическото НЕ е истина, когато условието е лъжа. Например, НЕ е истина в случая (*Вярно е, че 4 не е по-голямо от 6.*):

$$\text{не } 4 > 6$$

Методически насоки

В началото на урока се припомня последната изучена група команди – групата *Оператори* и командните блокове за аритметични действия, и за избор на случайно число.

Вниманието на учениците се насочва към блоковете за аритметични отношения или блоковете за сравняване – по-голямо, по-малко, равно. Обръща се внимание на тяхната форма – докато блоковете за аритметични операции са закръглени, то тези има остри краища.

Вниманието на учениците се насочва към различния резултат, който получаваме от блоковете за операции и от блоковете за сравнения. При операциите получаваме число, което е съответно сбор, разлика, произведение или частно. При сравняване се получава резултат „Истина“ (True) или „Лъжа“ (False). Ако имаме вярно числово равенство или неравенство, тогава резултатът от сравняването е Истина. Например:



Ако имаме невярно числово равенство или неравенство, тогава резултатът от сравнението е Лъжа. Например:

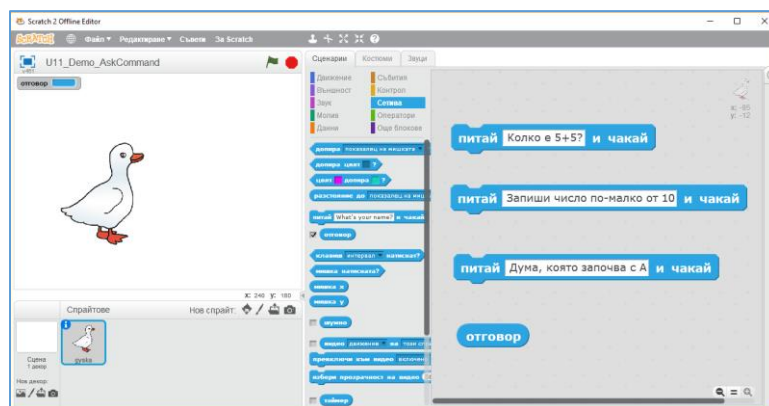


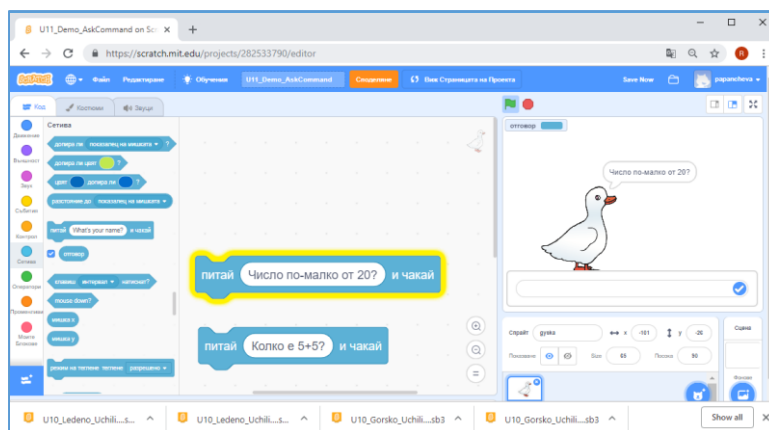
Блокът за сравняване може да се вложи във всеки друг блок – на правоъгълно, на кръгло и на ромбоидно място. Най-често блокът за сравняване го влягаме на мястото на условието в блок за разклонение. Ако резултатът от сравнението е истина (ако аритметичното отношение е вярно), изпълняваме едни действия, ако не – други действия.

Преди да се премине към задача 1 учителят изяснява командата „Питай ... и чакай“. Този блок се намира в групата Сетива. При стартиране на командата „Питай ... и чакай“ на екрана се появява команден ред, където от клавиатурата трябва да се въведе число или текст – според въпроса, който е зададен или задачата, която е поставена в командата. За да се завърши въвеждането, трябва да се натисне клавиша Enter или да се щракне с показалеца на мишката върху тикчето в края на командния ред.



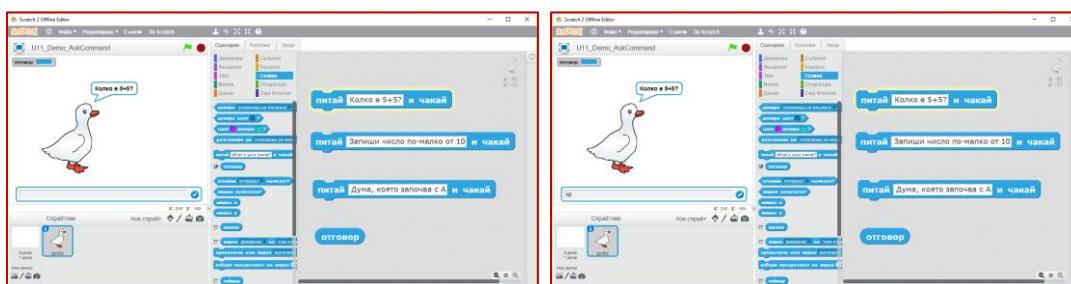
За демонстрация на действието на командата учителят използва подготвения към ресурсите демо-файл.



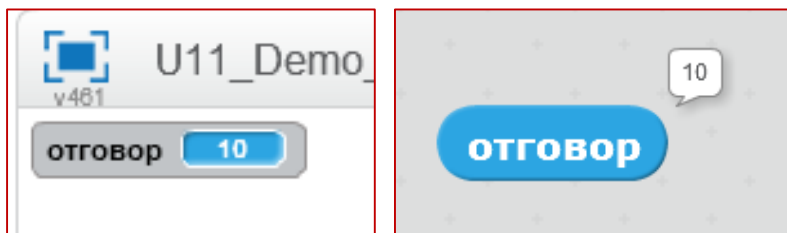


Щраква върху команда и обяснява какво се случва, какво трябва да се направи, какъв е резултатът. Например, щракваме върху първия команден блок – „Колко е 5+5?“.

Гъската „казва“: „Колко е 5+5?“. Появява се команден ред и програмата чака да въведем отговор.



Въвеждаме 10 от клавиатурата. За да покажем, че това е нашия отговор, или трябва да натиснем клавиша Enter, или да щракнем върху синьото бутонче в десния край на реда. След като въведем числото 10, то се запомня в бутона **Отговор**. Предварително пред **Отговор** е сложена отметка, за да можем да го виждаме върху сцената. Ако щракнем върху бутона **Отговор** в работното поле също можем да видим какво е запомнено в него в този момент.



Преминава се към работа по задача 1. Подготвени са работни файлове за учителя и за ученика. Учителят има завършен проект на тема „Училище във фермата“, а учениците имат работен файл, по който трябва да довършат проекта, като допишат кода.

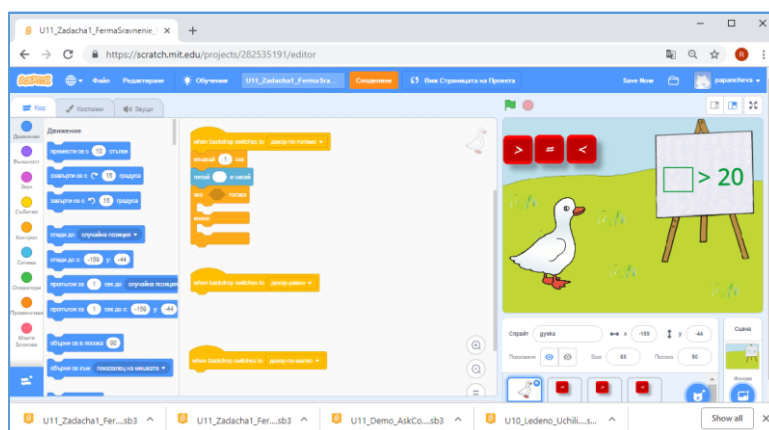
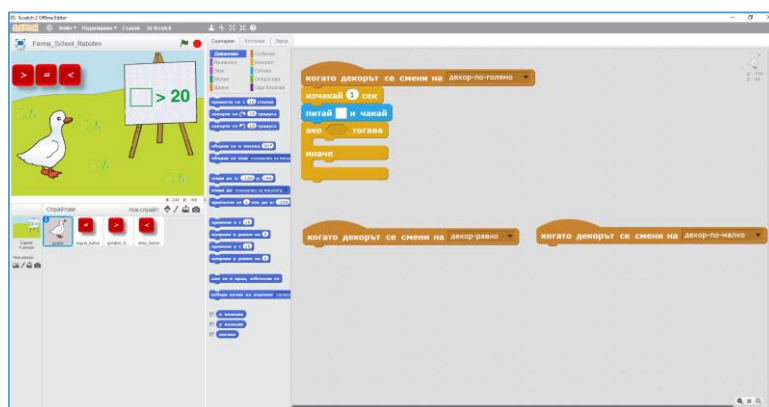
Учителят дискутира с учениците по готовия проект. Поставя списък от въпроси, чрез отговорите на които се изяснява логиката на проекта и след това се поставя самостоятелната задача на учениците.

По преценка на учителя работата по задачата може да се остави за края на часа. В урокът има много нови неща, а след този урок са предвидени два часа за упражнение, в които учениците ще довършват започнатите тук проекти.

Въпросите са:

- Колко декора има в проекта? Кои са те? (Декорите са 4 – един за начало и по един за аритметичните отношения – декор по-малко, декор по-голямо и декор равно).
- Колко герои (спрайтове) има в проекта? (Героите са 4 – 3 бутона – по един за всяко от аритметичните отношения (<, >, =) и един герой – гъската-учител, която ни поставя задачи).
- Как се сменят декорите? (При щракване върху някой от червените бутони, декорът се сменя с декор за съответното аритметично отношение)
- Какво се случва при щракване на зеленото знаме? (При щракване върху зеленото знаме се зарежда начален декор – при него няма нищо записано на дъската.)
- Какво е програмирана да прави Гъската при смяна на декора? Например, какво прави Гъската, когато декорът се смени с декор по-малко? (При зареждане на декор по-малко, Гъската ни поставя задача – „Запиши число по-малко от 200“ и чака да въведем число от клавиатурата. Въвеждаме число. Прави се проверка, ако нашето число е по-малко от 200, Гъската казва „Правилно!“. Ако въведеното от нас число не е по-малко от 200, тогава Гъската казва „Грешка“).
- Как се стартира нова задача? (За нова задача се щраква върху някой от червените бутони).

След като се изясни логиката на програмата учителят стартира работния файл към задачата и обяснява задачата на учениците. Те трябва да довършат кода само за първия декор – декор „по-голямо“. Целият проект учениците ще завършат в следващите уроци за упражнение. Следва самостоятелна работа на учениците по поставената задача за време.

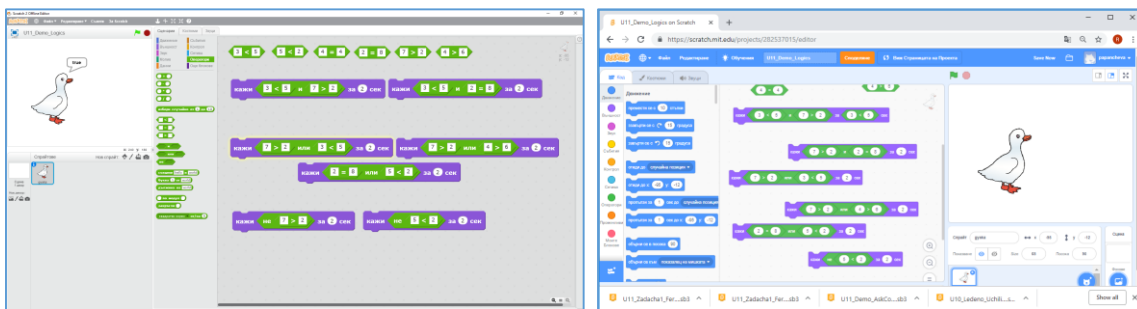


Следващ акцент в урока е запознаването с блоковете на логически операции.

В повечето случаи условията, от които зависят действията ни не са свързани само с едно изискване, а са по-комплексни. Например, искаме да се случват две неща едновременно (например, числото да е по-малко от 10 и да е четно). Или пък искаме да имаме две възможности (например, числото да е 4 или числото да е 5). Или пък искаме числото да не е 4. Във всички тези случаи използваме логическите операции – И, ИЛИ, НЕ.

Вниманието на учениците се насочва отново към групата с командни блокове – *Оператори*. В нея има група логически оператори – И, ИЛИ, НЕ. Обръща се внимание на формата на логическите блокове. Тя е същата, както и формата на блоковете за аритметични отношения (<, >, =). В резултат от логическите операции се получава Истина или Лъжа, както при блоковете за сравнение.

Към урокът е подготвен демо файл, с помощта на който учителят може да направи демонстрация с няколко избрани примера, за да обясни логическите операции. При щракване върху някоя от командите „Кажете ...“ се получава резултат Истина или Лъжа. Обяснява се защо резултатът е такъв.

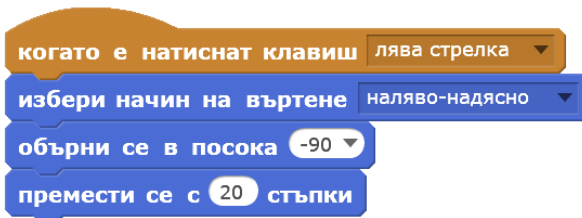
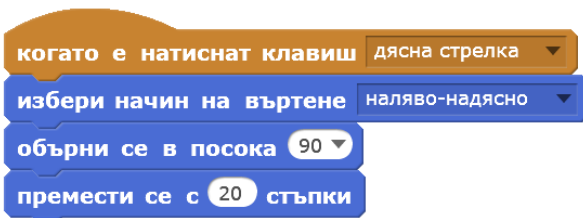


Премахва се към работа по задача 2. Учителят провежда фронтална беседа по готовия проект към задачата, за да обясни нейната логика. Въпросите, на които отговарят са следните:

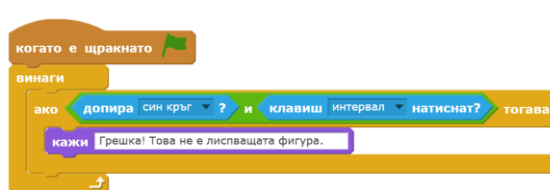
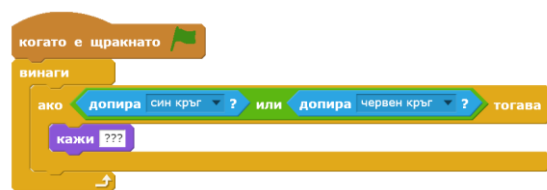
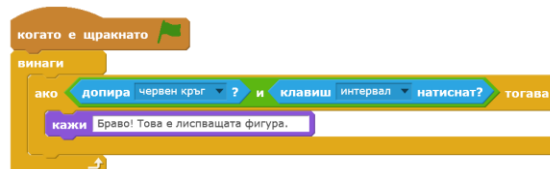
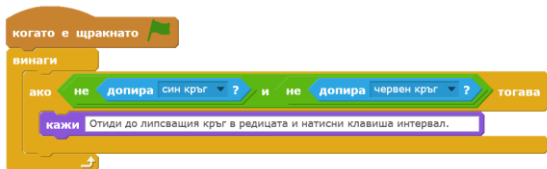
- Колко декора има в проекта? (Декорът е само един. На дъска е нарисувана редица от кръгове. Последният кръг липсва. Задачата е да се посочи кой от двата кръга долу – червен и син – е липсващия в редицата.).
- Колко герои (спрайтове) има в проекта? (Героите са 5 – 2 бутона със стрелки в горния ляв край на сцената; 2 кръга – син и червен – в долната част на сцената; Гъската – която трябва да избере верния кръг.).
- Какво е програмирана да прави Гъската? (1. Гъската се движи наляво-надясно при натискане на клавишите стрелки. 2. Ако не се допира до никой от кръговете казва „Отиди до липсващия кръг в редицата и натисни клавиш интервал“. 3. Ако се допира до един от двата кръга, казва „??“. Ако се допира до червения кръг и е натиснат клавиша Интервал, казва „Браво! Това е липсващата фигура“. 4. Ако се допира до синия кръг и е натиснат клавиш Интервал, казва „Грешка! Това не е липсващата фигура“).
- Какво са програмирани да правят стрелките в горната част на сцената? (При щракване върху стрелка се появява информация за движението на Гъската: „Движи гъската с клавишите стрелки“).

Разглежда се готовия код. Обръща се внимание на блоковете „Допира ли?“ и „Натиснат ли е клавиш?“. Тези блокове се намират в групата с команди *Сетива*. Резултатът от действието на командите е *Истина* или *Лъжа*, съответно, ако е изпълнено условието. Ако е натиснат клавиш Интервал, резултатът от операцията е *Истина*, иначе е *Лъжа*.

Не е предвидено време за обясняване на движението. Това са знания от трети клас. Кодът, определят движението на гъската при натискане на лява и дясна стрелка е следният:

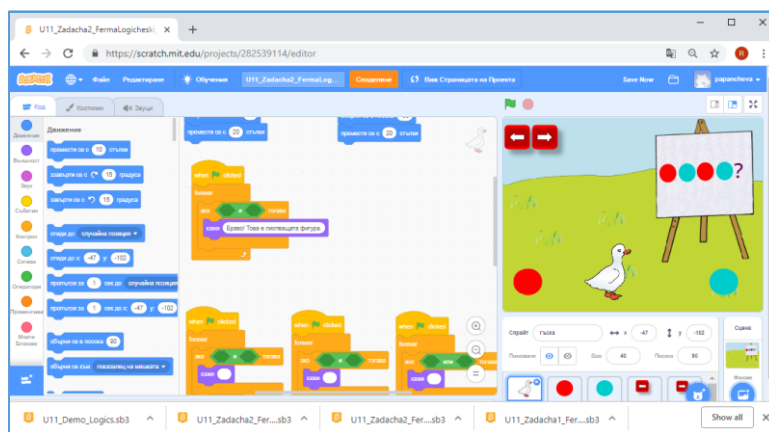
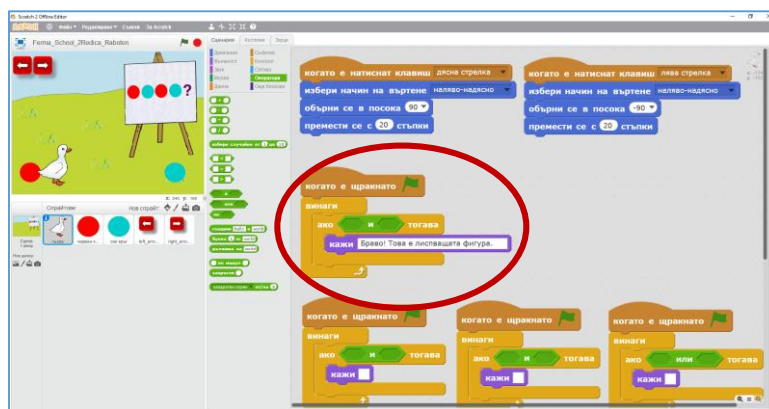


Учителят обяснява всеки от блоковете за разклонение.



Следва самостоятелна работа на учениците по работен файл. Трябва да програмират само случая, в който гъската допира червения кръг и при натискане на клавиша Интервал казва „Браво! Това е липсващата фигура“.

Останалата работа по довършване на проекта ще се реализира в следващите уроци за упражнение.



Урок 12-13. На училище – Аритметични и логически операции

Уроци за упражнение

Цел. Затвърждаване на знания за аритметични и логически оператори и уменията за работа с тях.

Задачи:

1. Да съставя програмен код с използване на блокове за аритметични действия.
2. Да съставя програмен код с използване на блокове за логически операции.
3. Да съставя програмен код с използване на блокове за аритметични отношения.

Очаквани резултати:

- Познава блоковете за аритметични и логически операции и може да ги използва при създаване на програмен код.
- Влага блок в блок.
- Познава допълнителни операции.
- Използва блокове за сравняване на числа.
- Познава допълнителни операции.

Методически насоки

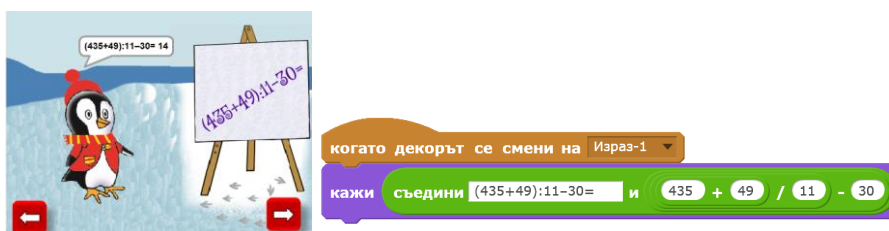
В рамките на два часа за упражнение учениците довършват проектите, които са разгледани в предните два урока – „Горско училище“, „Училище сред ледовете“ и „Училище във фермата – Мога да сравнявам“ и „Училище във фермата – Коя е липсващата фигура?“.

Учителят провежда беседа с учениците по вече разгледаните проекти. Преговарят се новите знания. Поставят се задачите. Учениците довършват проектите, като всеки работи със собствено темпо. Проектите се тестват и съхраняват.

Разработени са и допълнителни работни проекти, подобни на дадените, чрез които се работи за формулираните към урока цели и задачи.

Подготвен е нов работен файл по проект „Училище сред ледовете – числови изрази“. Проектът се състои в следното: има 3 декора, при които на дъската е записан по един числов израз. При зареждане на съответния декор пингвинът трябва да каже израза и неговия отговор. С помощта на стрелките в долната част на сцената става преход към следващ или към предишен декор. Учениците трябва да допишат кода, като конструират командата *Съедини* – да запишат дадения числов израз и чрез влагане на блокове да изчислят стойността на изразите.

Например, за показания декор, трябва да се запише кода на пингвина:



В учебника е дадена блок-схема, показваща логиката на кода на пингвина. Така се затвърдяват и знанията за конструиране на разклонен алгоритъм.

Урок 14. Управление на движението на героите. Изчакване, забавяне, ускоряване

Урок за нови знания

Цел. Формиране на знания за управление на движението на героите.

Задачи:

1. Да се формират знания за управление на движението на героите със забързване, забавяне, спиране.
2. Да се формират знания за управление на посоката на движение.
3. Да се формират знания за появяване или скриване на героя при настъпване на събитие.

Очаквани резултати:

- Умее да управлява движението на героите (изчакване, забавяне или ускоряване).
- Използва цикъл с условия за създаване на разклонен алгоритъм.

Учебно съдържание:

За създаване на компютърни проекти е важно да се познават модели на управление на движението на героите. Характеристики на движението са скоростта на движението и посоката на движение.

Скоростта на движение може да се управлява чрез размера на стъпката или чрез размера на времето на изчакване между две стъпки. В четвърти клас учениците не са запознати с десетични дробни, поради което не е предвидено да се използва изчакване с по-малко от 1 секунда време.

При една и съща стъпка на придвижване по-бързо се движи героят, който има по-малко време на изчакване между стъпките. При едно и също време на изчакване (или липсва на изчакване) между стъпките, по-бързо се движи героят, който прави по-големи стъпки.

В урока се въвежда условен цикъл – повтаряй докато. Командите от тялото на цикъла се изпълняват, само ако условието не е изпълнено. Пример, движи се напред докато стигнеш целта. Така движението се прави докато целта не е достигната. Когато целта се достигне, движението спира.

Методически указания

Урокът започва с актуализиране на знанията на учениците за командите за движение, изучени в трети клас. Обръща се внимание на командите „Премести се с ... стъпки“ и „Отиди до ...“. Преместването на определен брой стъпки става в посоката, в която е обърнат героя, поради което първо трябва да зададем посоката на героя. Командата „Отиди до ...“ се използва като героят се позиционира ръчно в желаната точка и тогава се взема командата „Отиди до ...“ и се подрежда в кода (програмата Scratch автоматично попълва стойностите за координатите x и y – не се предвижда въвеждането им от

учениците, тъй като не са формирани знания за координатна система и координати на точка.)

Учителят насочва беседа по темата за бързината на движение. Бързината на движение зависи от времето, за което се движим, както и от стъпката, с която се движим. При едно и също време по-бързо се движи този, който прави по-големи крачки. А при едни и същи като размер крачки, по-бързо се движи този, който прави крачките през по-малък интервал от време.

Аналогично може да се управлява движението на героите в средата Scratch. Учителят насочва вниманието на учениците към първите два кода, дадени в учебника и заедно определят кой от героите ще се движи по-бързо. Учителят прави демонстрация с подготвени урок демо-файл. След това сравняват вторите два кода и определят по-бързия герой. Проверяват, като учителят стартира втория демо-файл към урока.

Вниманието на учениците се насочва към съчетаването на командите „Премести се с ... стъпки“ и „Следващ костюм“. Ако героят има подходящи костюми, които представят отделни пози по време на движение, този герой може да се анимира така, че да се получи компютърна анимация на ходещ герой. Това става чрез смяна на костюмите. В учебника са показани четирите костюма на героя Пико от библиотеката на програмата Scratch (Pico walking).

Преминва се към работа по задача 1. По разработен към урока работен файл, учениците трябва да програмират Пико (Pico walking) да отиде до снежния човек горе (Snowman Red Hat – снежен човек с червена шапка). Да остане за малко при него. След това да слезе надолу до снежния човек със синя шапка (Snowman Blue Hat). Поставяме допълнително изискване нагоре движението да е бавно, а надолу – бързо. При движението, героят да ходи – анимацията на ходещ герой се получава чрез смяна на костюмите.

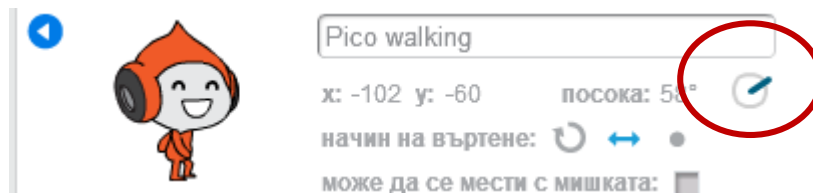
Припомня се как можем да реализираме ходене – чрез комбиниране на командите „Премести се с ... стъпки“ и „Следващ костюм“. Припомня се как едното движение да е бързо, а другото бавно. В случая ще се използва един и същ размер на стъпката на движение, като нагоре между стъпките на героя се изчаква по 1 секунда, а надолу такова изчакване няма.

Учителят и учениците коментират алгоритъма за изпълнение на задачата. Обръща се внимание на избора на начални условия. Чрез думите на момчето се обръща внимание на детайли, за които трябва да се замисля всеки програмист. Проектът трябва да започва винаги по еднакъв начин. Следователно в началото героите трябва да имат точно определени позиции. Използва се командата „Отиди до“. Учителят прави демонстрация – поставя Пико в позицията, от която иска да стартира движението му. Програмата Scratch веднага попълва числата x , y в командата „Отиди“ с позицията на героя. Учителят взема този команден блок и го подрежда в кода на Пико.

Следващия въпрос, който трябва да си зададем при движение е, ако героят се обръща как да се избере начина на завъртане. Ако изберем „Без завъртане“ – като слиза надолу Пико ще върви с гърба си напред. Това не е правилния избор за тази задача. Ако изберем „Във всички посоки“ – Пико, след като се обърне към горния снежен човек, ще ходи наклонен (перпендикулярно) на земята, а при слизване надолу ще се обърне с главата надолу. Следователно и това не е правилния избор на завъртане в конкретната задача. Избира се начин на въртене „Наляво-надясно“.

Последния въпрос на който трябва да си отговорим е в каква посока трябва да стартира движението. Командата „Премести се“ задава преместване на определения брой стъпки в посоката на движение. С командата „Обърни се към Snowman Red Hat“ Пико визуално

не се обръща към снежния човек горе, тъй като сме избрали начин на въртене само наляво и надясно, но посоката на движение се задава към снежния човек горе. Това може да се види, ако се щракне върху бутона **i** до героя Пико: Пико гледа надясно, но посоката му за движение е нагоре вдясно – към снежния човек с червена шапка.



Момичето поставя проблемната ситуация – как да придвижим Пико точно до снежния човек горе. Учителят запознава учениците с командата за условен цикъл – „*Повтаряй докато ...*“. С помощта на този цикъл се реализира разклонен алгоритъм с повтарящи се действия. Действието се повтаря докато се изпълни даденото условие. След това се продължава с изпълнението на командите след цикъла. Учителят обяснява по-подробно начина на изпълнение на такъв цикъл – до този момент учениците знаят, че има цикъл с краен брой повторения или с постоянно повторение. При новия цикъл логиката на изпълнение е следната: първо се проверява дали е изпълнено условието. Ако не е – се изпълняват действията, които са вътре в цикъла. Пак се прави проверка. Ако не е изпълнено условието – действията отново се изпълняват. Пак се прави проверка и т.н. Когато условието се изпълни, тогава действията в цикъла вече не се изпълняват и се преминава към следващата команда след цикъла.

На говорим език условния цикъл означава: повтаряй действията докато се изпълни зададеното условие.

Така движението на Пико следва логиката – движи се напред, докато стигнеш до снежния човек, т.е. Пико се премества определен брой стъпки. Ако не е стигнал до снежния човек, пак се придвижва напред. Когато стигне до снежния човек спира да се движи напред. Изпълнява се следващата команда в програмния код.

„Стигането“ до снежния човек се регистрира чрез командния блок от групата „*Сензори*“ „Допира ли?“. Ако Пико допира снежния човек, резултатът от оператора за допир е Истина. Тогава цикълът на движение спира.

Коментират се отделните блокове в кода на движение на Пико.

Преминава се към самостоятелна работа на учениците по проекта. Те трябва сами да програмират движението на Пико.

Като допълнителна работа е поставена втора задача – продължение на първата. Да се програмира снежния човек горе. При щракване върху него да слиза при снежния човек долу. Целта е да се използва командата „Пропълзи до ...“. Така се затвърждават знания за движение от трети клас.

Урок 15. Синхронизиране на действията на героите

Урок за нови знания

Цел. Формиране на знания за синхронизиране на движението на героите.

Задачи:

1. Да се формират знания за работа със съобщения.

2. Да се формират знания за появяване или скриване на героя при настъпване на събитие.

Очаквани резултати:

- Познава начини за поява на герой след настъпване на събитие.
- Умее да синхронизира движението на героите след настъпване на събитие (щракване върху зеленото знаме, смяна на декор, щракване върху герой и др.).
- Умее да синхронизира движението на героите чрез използване на система от съобщения.

Учебно съдържание:

В урока учениците се запознават с начините на управление на движението и синхронизиране на героите при настъпване на някакво събитие – смяна на декор, изпращане на съобщение и др.

Методически насоки

Към урока е разработен проект, продължение на работата от предходния час. Добавен е нов декор – Пико е пред дома си. Казва „Днес ще отида на пързалката“. След което декорът се сменя с вече познатия декор на зимната пързалка.

Урокът преминава като работа по общ сценарий с отделни сцени. Има два декора – „У дома“ и „На пързалката“. Героите са: Пико, двамата снежни човеци и пет снежинки.

Сцена 1. (Участва Пико и двамата снежни човеци) Пико е пред къщата – декор „У дома“. Казва „Днес ще отида на пързалката!“. Тръгва към края на сцената и се скрива. Декорът се сменя с декор „На пързалката“.

В началото на проекта трябва да синхронизираме действията на всички герои. Това става като се използва събитието – щракване върху зеленото знаме за старт на проекта. В началото героите снежни човеци не трябва да се виждат. Задаваме им команда „Скрий се“ при стартиране на проекта, т.е. при щракване върху зеленото знаме.

Логиката на проекта в началото се състои в следното:

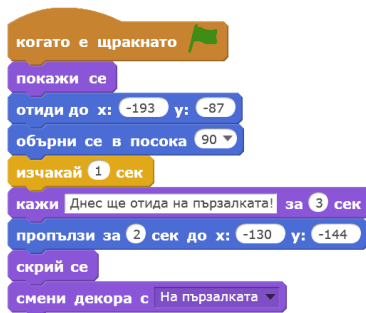
При щракване върху зеленото знаме:

1. Декорът се сменя с декор „У дома“ (зимен декор с къща).
2. Героите снежни човеци изпълняват командата „Скрий се“
3. Героят Пико застава пред къщата, изчаква малко и казва: „Днес ще отида на пързалката“. Пропълзва към края на сцената и се скрива.
4. Декорът се сменя с декор „На пързалката“.
5. Снежинките са разположени статично.

Учителят коментира логиката на проекта. Разглеждат се кодовете на всеки от героите, както и кода на сцената. Снежинките са разположени статично. Те няма да се променят в първите две сцени.



Кодът на Пико е следният:



При командите „*Отиди до ...*“ и „*Пропълзи ...*“, не се въвеждат ръчно координати, а героят се премества там, където искаме да отиде, взима се командата за движение, в която координатите са попълнени автоматично от програмата Scratch и този блок за движение се използва в кода.

Прави се извода, че в сцена 1, синхронизирахме героите чрез програмиране на действията им при щракване върху зеленото знаме за старт на проекта.

Преминава се към сцена 2. Пико отива на пързалката. Там са снежните човечи – със синята шапка е долу, а този с червената е горе на хълма. Пико застава до снежния човек долу. Обръща се към другия и бавно се качва до него. Когато е горе, спира.

Задачата е да програмираме движението на Пико до снежния човек с червената шапка, горе на хълма. Това е позната задача от предни уроци. Това, което трябва да се промени е че вместо блока за събитие „Когато е щракнато знамето“ да се използва блока за събитие: „Когато декорът се смени на На пързалката“. Още трябва да се съобрази, че за всички скрити герои трябва да се даде команда „Покажи се“ – за Пико и за двамата снежни човечи.

Преминава се към сцена 3. Пико е при снежния човек горе и казва: „Как ми се иска да завалят пухкави снежинки!“. Тогава снежинки започват да валият. Падат от небето до земята, отново и отново.

Първият проблем, който среща момичето е как да накара снежинките да започнат да се движат. Учителят запознава учениците с командите за разпространяване и изпращане на съобщения. Те са използвани и в трети клас, но на пропедевтично ниво. Изяснява се алгоритъма и начина на използването на съобщение в конкретната задача.

Следващият проблем при писането на кода е как да се постигне ефекта на снеговалеж. Снежните човечи трябва постоянно да падат от небето. Задачата може да се изпълни по различни начини. Тук е избран алгоритъм, при който се използва условен цикъл. Условието за край на повторението е докосване до определен цвят. Предварително в долната част на сцената към декора е допълнен елемент в светлосив цвят. При допир до сивия цвят снежинките се скриват.

За движението на снежинките се използва вече познатият модел на учениците. Избира се посока на движение – надолу. След това се използва цикъл с придвижване. Избира се малка стъпка – 3, за да се получи плавно и бавно падане на снежинката. След като се допре до сивия цвят снежинката получава команда *Скрий се*. Докато е невидима, снежинката изпълнява команда *Отиди до ...*, която я връща в началната ѝ позиция в горната част на сцената. Тогава снежинката изпълнява командата *Покажи се* и отново се изпълнява цикъла за движение надолу.

В сцена 3 действията на героите – на снежинките – се синхронизират чрез изпращане и получаване на съобщения.

Така в урока се разглеждат три много често срещани начина за синхронизиране на действия и движения – чрез стартиране на проекта от зеления флаг, чрез смяна на декор и чрез изпращане и получаване на съобщения.

Може да се постави допълнителна задачата на учениците – да доразвият проекта по свой избор и в избрана от тях посока. Към ресурсите на учителя е дадено примерно развитие на проекта – добавена и музика по време на падането на снежинките. Използва се и команда за спиране на всички процеси в края на кода.

Урок 16. Компютърна анимация. Промяна на външния вид на героите

Урок за упражнения

Цел. Затвърждаване и разширяване на знания за анимиране на героите, чрез промяна на външния им вид.

Задачи:

1. Да се разширят знанията и уменията за работа с инструментите за векторна графика.
2. Да се надградят знанията и уменията за създаване на анимирани образи, чрез смяна на костюмите.

Очаквани резултати:

- Умее да манипулира външния вид на героя с цел постигане на реалистични анимации.
- Работи с инструменти за векторна графика в графичен редактор.
- Комбинира различни кадри за създаване на анимирано изображение.

Учебно съдържание:

Външният вид на героите в Scratch може да се манипулира чрез вградения в средата графичен редактор. Редакторът поддържа инструменти за обработка както на растерна, така и на векторна графика. Визията на проектите е много по-добра при използване на векторна графика, тъй като при работа с растерна графика размерът на изображенията, ограничени в 480 на 360 точки (според размера на сцената) са с ниско качество при преглед на цял екран.

Във вградените библиотеки на декори и герои има голямо разнообразие от векторни изображения, с помощта на които могат да се създават интересни проекти. Манипулирането на външния вид на героите води и до развиване на творческо мислене и умения у учениците.

От трети клас учениците познават инструментите на вградения графичен редактор. Урокът е за упражнения, като знанията се затвърждават и надграждат.

Със всяка от задачите се дават и модели на работа, които учениците могат да прилагат след това самостоятелно, използвайки и други герои в Scratch.

Методически насоки:

В урока са включени три задачи, при които на базата на герой с един костюм учениците формират знания и умения как да създадат още костюми, за да могат да направят анимация с избрания герой.

Първата задача се състои в това да се създаде анимирана зимна картичка. Снежният човек се качва на хълма, сяда и се спуска надолу. След това става и отново се качва. Така действията се повтарят постоянно. Като допълнителна задача може да се зададе да се включи музика или да се добавят допълнителни елементи като падащи снежинки или включване на други герои.

Разместваме краката на снежния човек. Чрез инструмента за маркиране, можем да завъртаме и да местим отделните елементи. Редактираме изображението докато получим образ на седнал снежен човек.

Добър стил на работа е да се преименуват костюми и/или декори, така че да името им да показва как ги използваме в логиката на програмния код. В случая можем да преименуваме двата костюма като „Снежко прав“ и „Снежко седнал“.

Преминава се към програмиране. Учителят показва готовия код. Демонстрира действието му и оставя учениците да работят самостоятелно по проекта си. Към урока е подготвени и втори работен файл – за по-бързо работещите ученици или за самостоятелна работа.

От героя Еднорог, трябва да се създадат 4 костюма, чрез които да се анимира препускащия еднорог. По аналогичен начин може да се зададе и трета задача – да се анимира отново снежния човек, но като се създадат 4 костюма, чрез които да се създаде ходещ снежен човек. Обръща се внимание на взаимното положение на краката и на ръцете в отделните кадри (костюми).

В ресурсите на учителя са предоставени готови демонстрационни файлове.

Урок 17. Работа с променливи

Урок за нови знания

Цел. Формиране на знания за променливи и приложението им в среда за програмиране.

Задачи:

1. Да се формират знания за променливи и константи.
2. Да се формират знания и умения за създаване на променливи в Scratch.
3. Да се формират знания и умения за присвояване на стойност на променливи и използването им в числови изрази.

Очаквани резултати:

- Разбира същността на променливите.
- Създава и именува променливи.
- Присвоява стойност и използва числови променливи.

Учебно съдържание:

Информацията, която получаваме у дома, в училище, от медиите, от Интернет и др. съдържа имена, числа, действия и др. Някои от имената и числата са постоянни, а други

се променят. Например, нека Мими живее в Бургас, на улица Оборище 215. Номерът на къщата, в която живее Мими не се променя – той е постоянна величина. Мими слуша новините и следи температурата навън – сутринта температурата е 12 градуса, на обед е 19, вечерта – 11, през нощта – 5. Температурата навън се променя, има различни стойности.

Величините, които променят своите стойности наричаме променливи. Променливата можем да си представяме като кутия, в която ще съхраняваме нещо. Нека имаме 2 кутии – това са нашите 2 променливи. Това, което е в кутията, наричаме стойност на променливата. На променливата даваме име. Преди да използваме променливата, ѝ задаваме начална стойност.

В Scratch можем да работим с променливи. Чрез тях създаваме по-сложни проекти с изчисляване на изрази, отчитане на точки в компютърни игри, отмерване на време и др.

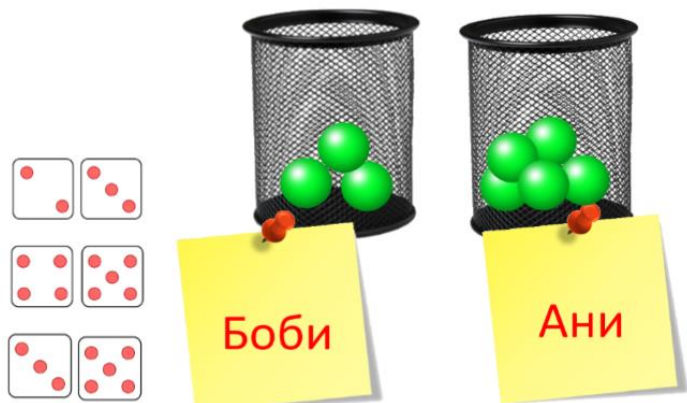
За работа с променливи има специална група с командни бутони – групата *Данни/Променливи*. Първо трябва да се създаде променлива. Щраква се върху бутона „Създаване на променлива“. Отваря се нов прозорец, в който има поле за въвеждане на името на променливата. Можем да отбележим дали променливата да е за всички герои (спрайтове) в проекта или само за текущо избрания. След въвеждане на името се щраква върху бутона „Добре“. Тогава в групата Данни се появяват командни бутони. Има две основни команди за работа с променливи – за задаване (присвояване) на стойност и за променяне на стойността (чрез добавяне или изваждане на число).

Методически насоки

Учителят провежда беседа с учениците, в хода на която дава примери за неща, които се променят и неща, които са постоянни. Въвежда се понятието „променлива“. Прави се аналогия между променливата и кутия. Това, което, поставим в кутията е стойността на променливата. Променливата има име – затова надписваме кутията, с която онагледяваме понятието „променлива“. Преди да започнем да работим с променливата, трябва да ѝ дадем начална стойност. Ако изпразним кутията, тогава началната стойност на променливата е нула.

Учителят разиграва следната примерна игра. Поставя две празни кутии (от метална мрежа, или други прозрачни кутии, за да се вижда какво има вътре в тях). Надписва ги. Нека едната кутия е надписана „Боби“, а другата „Ани“. Две деца хвърлят последователно зар – едното за кутията „Боби“, а другото за кутията „Ани“. Колкото точки се паднат на зара, толкова топчета поставя в съответната кутия съответното дете. Нека играта да се е развила по следния начин:

	Боби	Ани
Хвърляне 1	2	3
Хвърляне 2	4	5
Хвърляне 3	3	5



Стойностите на променливите се променят. В края на играта стойността на променливата „Боби“ е 3 топчета, а стойността на променливата „Ани“ е 5 топчета. Можем да променяме името на променливата. Вместо „Боби“ и „Ани“, можем да кръстим променливите така: „Точки на Боби“ и „Точки на Ани“.

Преминава се към начина за работа с променливи в средата Scratch. Учителят демонстрира как се създава променлива в Scratch. Задава първата задача – работи се по предварително подготвен работен файл с герой – балон. Трябва да се напише код, при който при щракване върху балона, той да променя цвета си. Освен това, при всяко щракване върху балона да се натрупват точки.

За решаване на задачата първо се създава променлива с име „Точки“. След това трябва да се зададе начална стойност на променливата – в началото на променливата „Точки“ се задава стойност 0. Учителят показва готовия програмен код и заедно с учениците изясняват всеки ред от него. Освен командите за работа с променливи, в кода е използвана и командата „Промени ефект ...“. В примера се променя ефекта „цвят“. Цветът в програмирането се задава чрез число. Като се промени числото, се променя и цвета. Учениците работят самостоятелно по първата задача.

Преминава се към работа по задача 2. Алгоритъма се усложнява, като се въвежда движение на балона. Балонът трябва да пада надолу. Отново при щракване върху него той променя цвета си и се натрупва по една точка. Движението надолу може да се реализира подобно на движението на снежинките от проекта „На пързалката“. В случая е даден нов алгоритъм. Учителят и учениците го разглеждат заедно по работния файл от ресурсите на учителя.

Алгоритъмът се състои в следното: балонът се движи надолу и нагоре. Докато се движи надолу го виждаме. Когато се движи нагоре е скрит. Движението се контролира чрез изпращане на съобщения. При всяко щракване върху балона той сменя цвета си и се скрива.

За допълнителна работа могат да се поставят задачи от вида: да се добави втори балон, който да се движи от друга позиция подобно на първия; при всяко щракване върху балона, той да променя костюма си и да се чува звуков ефект; играта да свърши, щом броят точки стане 10 и др.

В края на урока се прави обобщение на наученото – какво са променливите, как се задава стойност, как се променя стойността на променливата.

Урок 18. Знам и мога

Урок за упражнения

Цел. Затвърждаване на знания и умения за работа с аритметични и логически операции, управление на героите, работа с променливи.

Задачи:

1. Да се затвърдят знанията за работа с аритметични оператори.
2. Да се затвърдят знанията за управление на движенията на героите, да се съчетае движение с чертане.
3. Да се затвърдят знанията за работа с променливи.

Очаквани резултати:

- Познава и може да използва аритметични и логически оператори
- Разбира същността на променливите и може да работи с тях.
- Използва инструменти за чертане в средата Scratch

Учебно съдържание:

В урокът се припомнят знания от трети клас, свързани с използване на блокове за чертане. Работи се с блоковете за избор на цвят и дебелина на линията, изчистване, чертане.

Методически насоки

Урокът започва с актуализиране на знанията на учениците за командите за чертане – Изчисти, Молив долу, Молив горе, Избери цвят ... Избери дебелина ...

Поставя се задачата за урока – да се създаде компютърен проект, чрез който да се онагледят действие събиране. По случаен начин се избират две числа. Пресмята се техният сбор. Начертават се три отсечки – две от отсечките имат дължина, равна на събираемите, а третата отсечка има дължина, равна на сбора на двете числа.

За реализиране на поставената задача в урока са очертани стъпки, които учениците трябва да следват в своята работа. Използват работен файл, в който има единствено зареден декор.

Като втора задача е предвидено да се разработи подобен проект, но с действие изваждане. В случая трябва да се съобрази, че умаляемото трябва да е по-голямо от умалителя. Използват се оператори за сравняване, за да се осигури положителна разлика преди самото действие.

Учителят може да постави допълнителни задачи, при които вече реализираните проекти да се направят по нов начин като се използват променливи. В учебника е дадена задача използване на командата „Питай и чакай...“, като целта е да се използват и променливи.

Урок 19. Работа с програмируемо устройство

Урок за нови знания

Цел. Формиране на знания за възможности за включване на програмируеми устройства към дигитално устройство и управлението им чрез визуални среди за програмиране.

Задачи:

1. Да се формират знания за програмируеми устройства.
2. Да се формират знания за начина на свързване на програмируемо устройство с дигитално устройство.

Очаквани резултати:

- Разбира същността на програмируемо устройство.
- Разбира нуждата от формиране на знания за управление на програмируемо устройство.
- Знае как се управлява програмируемо устройство.
- Разбира, че програмируемото устройство има предварително заложиени възможности, които могат да се управляват програмно.

– Познава функционалността на бутони на конкретно устройство

Учебно съдържание:

В съвременното общество, базирано на информация и технологии, за успешен живот и бъдеща професионална реализация са нужни специфични знания и умения. Важна част сред тях са дигиталните умения, които не се ограничават единствено до използване на информационни технологии, но се отива далеч напред. Нужни са умения за създаване на алгоритми, за създаване на софтуерни приложения, за работа с разнообразни софтуерни платформи. Освен на софтуерно ниво, съвременния човек трябва да може да управлява редица хардуерни роботизирани системи. Това изисква формиране на базови знания за управление на програмируеми устройства. От децата днес се очаква не само да използват технологиите, но и да могат да ги създават и управляват.

Да направим уточняване на понятието „програмируемо устройство“, за да го разграничим от понятието „дигитално устройство“. Под програмируемо устройство ще разбираме устройство, което има предварително програмирани възможности, които можем да управляваме и използваме. То има значително по-малко възможности от компютъра или таблета и може да няма един или няколко от основните компоненти на дигиталното устройство – процесор, памет, екран, клавиатура.

Всяко програмируемо устройство има предварително заложи функционалности и не може да прави неща, които не са предварително дефинирани. Например, има програмируеми устройства, които могат да се движат – те изпълняват команди като напред, назад, наляво, надясно. Ако устройството може да се обръща на 90 градуса в посока, а не е заложи възможност за завъртане на 15 градуса, то устройството не може да бъде програмирано да се завърти на 15 градуса – може да се обръща само на по 90 градуса наляво и надясно. Може устройството въобще не поддържа движение, а само издаване на звукови или светлинни сигнали.

В живота около нас има безброй програмируеми устройства – в кухнята (уреди като перални, фурни, миялни машини, кафе-машини и мн. други), в стаята (програматори за температура в стаята, телевизори и уредби, които могат да бъдат програмирани кога да се включват, кога да се изключват, как да работят), роботизирани прахосмукачки, които могат да бъдат програмирани да се включват в определен ден и час през седмицата и мн. др.

За да се програмира устройството трябва да се знае какви възможности има и как се използват и програмират тези възможности. За да се програмира едно устройство трябва да се разполага с борд за управление – това може да бъдат бутони върху устройството, може да е дисплей с бутони или сензорен дисплей за програмиране на работата на устройството, може да е софтуерно приложение (програма за инсталиране на компютъра или мобилно приложение за смарт-телефон или таблет), от което да се настройва работата на програмируемото устройство. Ако се управлява чрез дигитално устройство, програмируемият робот трябва да се свърже дигиталното устройство. Това може да стане чрез кабел или безжично.

Съществува огромно разнообразие от програмируеми устройства, които се използват в училище за формиране на начални знания по програмиране и управление на програмируеми работи.

За целите на обучението по Компютърно моделиране, авторският екип е избрал робота Blue Bot. Роботът е представител на т. нар. floor robots, т.е. работи, които се движат върху гладка повърхности, обикновено по очертана квадратна мрежа с фиксирана стъпка и могат да изпълняват прости команди за движение – напред, назад, завъртане наляво и

надясно (под прав ъгъл). Роботът Blue Bot се управлява чрез бутони, разположени върху робота или чрез команди от визуална среда за програмиране, която е свободно достъпна и може да се инсталира както на компютър, така и като приложение за мобилен телефон или таблет. Работи с операционни системи Windows и Android. (Може да се избере и роботът InO Bot – разработка на същия производител, както и Blue Bot – InO Bot може да се управлява през средата за визуално програмиране Scratch.)

Изборът на авторския колектив е определен от една страна от широкото разпространение на Blue Bot, сигурната и лесна работа с него, разнообразието от ресурси. От друга страна е добре учениците да се запознаят с още една среда за визуално програмиране (след запознаването с визуалните среди Code.org и Scratch). Предимство на средата е, че може да се използва и като симулатор, когато към компютъра не е свързан робот. Така всеки ученик може да работи самостоятелно на компютъра в училище или у дома, като симулира движението на робота.

Роботът BlueBot се свързва с дигиталното устройство (компютър, смарт-телефон, таблет) чрез Bluetooth връзка.

Може да се работи с един робот и разработени към него дидактични материали – игрално табло и карти, които се поставят върху него. Оптимален вариант е осигуряването на комплект от 6 робота с комплектите към тях, за да се реализира екипна работа в хода на учебния процес и учениците самостоятелно да могат да управляват програмируемите устройства. Когато роботът е един, се зарежда през USB порт на компютър. Когато са 6, роботите са в училищен комплект, включващ удобна станцията за зареждане.



Методически насоки

Учителят провежда беседа с учениците по темата за професиите сега и професиите на бъдещето. Целта е да се стигне до извода, че нещата се променят постоянно. Има професии, които са вечни – като например, професии като лекар, готвач, шивач. За други професии е определяща технологията. В миналото, когато не е имало коли, не е имало и професия шофьор. Когато не е имало гражданска авиация не е имало и професия стюардеса. С бързото развитие на технологиите не можем да знаем със сигурност какви професии ще се появяват в бъдеще. Новите технологии ще доведат до появата на нови професии, както и до изчезването на някои съществуващи днес професии.

Кои професии вече не съществуват? С навлизането на автомобилите изчезват кочияшите. Изчезват и коларите – майстори на каруци, двуколки, файтони и др. Няма ги старите пишещи машини, няма и машинописци, работили с тях. Няма ги старите телефонни централи и ги няма и телефонистите, работили там.

Кои професии може би ще изчезнат в бъдеще? Почта – писмата изпращаме все по-често само в електронен вид по електронната поща. Инкасаторите – повечето

устройства могат да бъдат свързани с интернет и да се отчитат от разстояние, не е нужно човек да обикаля по домовете на хората, за да запише показанията на уред за отчитане (електромер, водомер, потребление на газ и др.). Касиери в магазините – и сега има магазини, в които хората сами сканират покупките и плащат на машина в брой или с банкова карта.

От какви умения ще имам нужда като порасна? Да работя в екип, да представям работата си, да използвам технологии, да имам математически знания, да мога да чета и пиша, да зная езици, да зная за музика, изкуства, литература, да анализирам, да прилагам наученото, да търся информация, да избирам най-важното, да проверявам информацията, да общувам с хората, и още много други.

Да програмирам, да управлявам програмируеми устройства, да работят с дигитални устройства.

Прави се извода, че важна част от личностните умения са дигиталните умения, които не се ограничават единствено до използване на информационни технологии. Нужни са умения за създаване на алгоритми. Хората не само ще използват все повече технологии, но и трябва да могат да ги създават и управляват.

Въвежда се понятието „програмируемо устройство“. Под програмируемо устройство разбираме устройство, което има предварително зададени възможности и функции, които можем да управляваме и използваме чрез даване на команди или писане и изпълняване на програмен код. То може да няма някои от основните компоненти на дигиталното устройство, например – екран или клавиатура.

Всяко програмируемо устройство е предварително определено и не може да прави неща, които не са заложили в него. Например, ако едно програмируемо устройство може да се движи напред-назад и да се завърта в страни на прав ъгъл (90°), то не можем да го програмираме да се завърти на друг ъгъл, например на 15° или 30° . Има устройства, които не могат да се движат, а се програмират само да издават звуци или да излъчват светлини.

В живота около нас има безброй програмируеми устройства – перални, фурни, миялни машини, програматори за температура в стаята, телевизори и уредби, роботи-прахосмукачки и др. – можем да ги програмираме кога да се включват и как да работят.

За да се програмира едно устройство трябва да разполага с борд за управление – бутони и/или дисплеи. Устройството може да се програмира и през софтуерно приложение – от програма, инсталирана на компютъра или от мобилно приложение за смарт-телефон или таблет. Тогава програмируемостта на устройството трябва да е свързана с дигиталното устройство. Това може да стане чрез кабел или чрез безжична връзка (Bluetooth, wi-fi).

В края на часа учителят може да включи програмируем робот, с който учениците ще работят в следващите часове и да направи кратка демонстрация като го управлява – от бутони върху устройството и чрез програмен код в среда за визуално програмиране.

Урок 20-21 Среда за визуално програмиране Blue Bot. Командни блокове за движение. Роботът се движи

Урок за нови знания

Цел. Формиране на знания за интерфейса и възможностите за управление на програмируемо устройство чрез визуална среда за програмиране.

Задачи:

1. Да се формират знания и умения за работа с конкретна среда за визуално програмиране за управление на програмируемо устройство.
2. Да се формират знания и практически умения за свързване на програмируемо устройство с дигитално устройство.

Очаквани резултати:

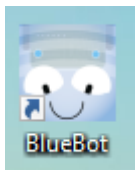
- Познава функционалността на бутони на конкретно устройство.
- Умее да ползва бутоните за задаване на поредица от движения в различни посоки.
- Познава блокове от команди за включване на програмируеми устройства към дигитални устройства.

Учебно съдържание:

Роботът-пчела Blue Bot е програмируемо устройство, което може да се управлява *ръчно* с помощта на бутоните, които се намират върху горната му част или *програмно*, чрез код, създаден в средата за визуално програмиране със същото име – Blue Bot.

Бутоните върху робота-пчела са 7 на брой. 4 са бутони за определяне на посоката на движение – напред, назад, завъртане надясно и завъртане наляво. Има един бутон пауза, с който се задава спиране на движението за 1 секунда. Има бутон X, с който се изтрива последно запомнената последователност от команди за движение. Бутонът в центъра на управление е надписан GO. При неговото натискане се стартира движението на робота. Изпълнява се движението, което е зададено чрез последователното натискане на бутоните за движение и бутона за пауза. Ръчното управление на робота-пчела е същото, както управлението на робота-пчела Bee Bot.

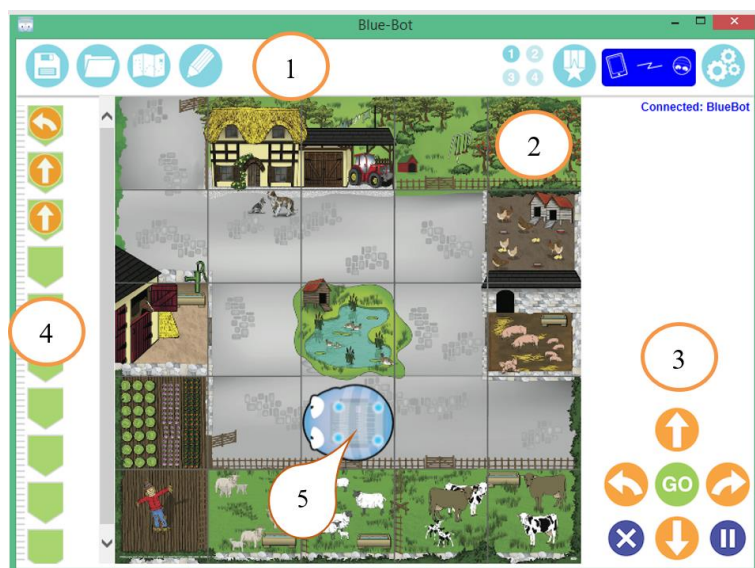
Средата за програмиране на робота може да се изтегли от <https://www.bee-bot.us/>. След инсталиране може да се създаде пряк път към софтуера. Иконата на програмата е:



Освен на компютър/лаптоп, визуалната среда Blue Bot може да се изтегли като приложение за мобилен телефон (Windows, Android, iOS). Иконата е същата.

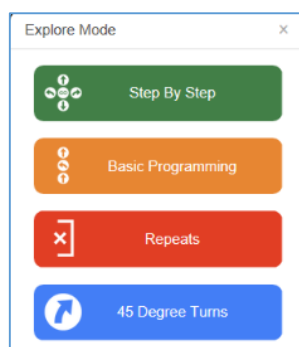
След като се инсталира средата, роботът Blue Bot може да се свърже с дигиталното устройство чрез Bluetooth връзка. Активира се Bluetooth устройството на компютъра/телефона. Включва се роботът-пчела. Софтуерът разпознава робота и предлага свързване. След успешна връзка, се издава звуков сигнал, а пчелата-робот светва в синьо.

Прозорецът на програмната среда Blue Bot има следния вид:



1 – меню; 2 – платно за движение; 3 – блокове за управление на движението; 4 – програмен код; 5 – аватар на робота-пчела Blue Bot.

За да програмираме робота, трябва да изберем табло, по което той да се движи, както и режим на работа. В програмата има вградени табла. Можем да зареждаме и табла от файл.



В режим Explore, можем сами да си създаваме проекти, а в режим Challenge – изпълняваме задачи от различен вид и сложност. Режимът на работа може да се сменя между отделните задачи.

Режим Explore предлага избор от четири възможности:

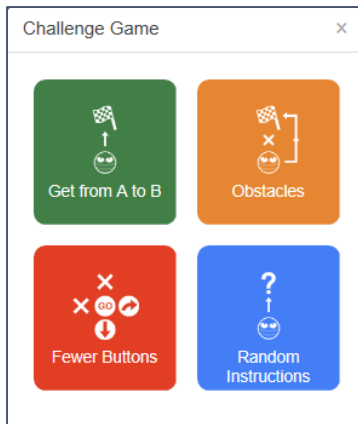
Step by Step – незабавно изпълнение на избраната команда.

Basic Programming – създава се програма, която се изпълнява след стартиране.

Repeats – създаване на програма с повтарящи се действия.

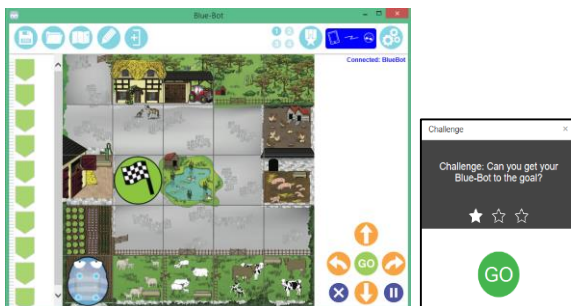
45 Degree Turns – Възможност за програмиране на път, при който има и допълнителни бутони-команди за завъртане на 45 градуса (такива бутони няма върху робота-пчела и не могат да се използват при ръчно управление).

При щракване върху бутоните за движение те се подредат една след друга в програмния код. Ако сме в режим Step by step, пчелата-робот веднага изпълнява зададените команди. В останалите случаи трябва да щракнем върху бутона Go, за да стартира движението на робота.

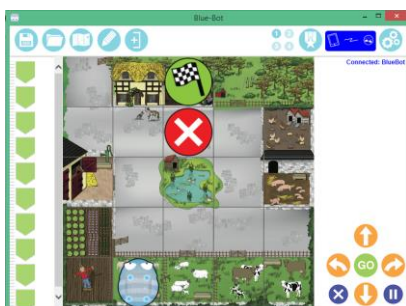


Режим Challenge предлага четири вида предизвикателства:

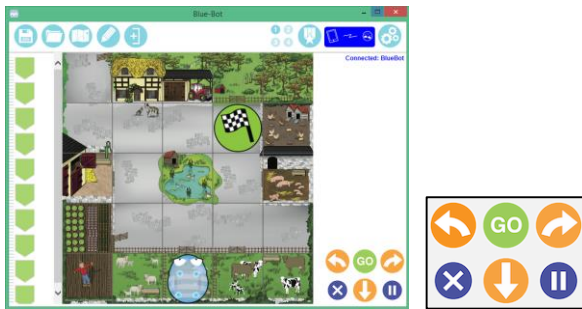
- **Get from A to B** – Променят се положението на роботчето и мястото, до което трябва да стигне. При правилно решение на задачата, се поставя друга мисия. Според броя на решените задачи се увеличава рейтингът на състезателя.



- **Obstacles** – Предизвикателствата са подобни на предходните, но на полето има препятствия, които роботът трябва да заобикаля.



- **Fewer Buttons** – Задачите се усложняват от това, че част от командите стават недостъпни и предизвикателството трябва да се изпълни с по-малко команди.



- **Random Instructions** – Това е най-трудната задача. Роботът е поставен на някаква позиция. Играчът трябва да разчете програмата и да определи местоположение на робота след изпълнението на програмния код.

Програмата Blue Bot има вградени игрални табла, но могат да се добавят и нови графични изображения от файл. Те се съхраняват в библиотеката на програмата и могат да се използват и при следващо стартиране на средата.

За нуждите на обучението по компютърно моделиране са разработени игрални табла, които учителят има в своите ресурси в електронен вид.

Методически насоки

Урокът започва със запознаване с програмируемото устройство Blue Bot. В хода на фронтална работа, учителят показва бутоните върху устройството и обяснява функциите им. Обясняват се светлинните сигнали на робота – когато е свързан чрез Bluetooth с компютъра, роботът свети в синьо. Когато се зарежда – в червено, когато е напълно зареден – в зелено.

Учителят прави демонстрация, като задава ръчно примерно движение на пчелата, за да се проследи нейното действие.

Преминва се към работа със средата за визуално програмиране Blue Bot. Учителят разяснява елементите в прозореца на програмата и прави демонстрация на работата с робота.

Поставя се първата задача. Предварително учителят е заредил работното табло от ресурсните файлове и го избира, за да се програмира движението на пчелата от А до Ъ.

С помощта на картоните с команди за движение от тетрадките си учениците подреждат самостоятелно кодовете за движение. Учителят следи работата и отбелязва ученици с различни кодове, за да може да се проиграт и проверят.

За работа с робота-пчела учителят трябва да разполага и с игрално табло, върху което да се тества хода, записан с програмни код. Таблото може да се направи от картон или да се използват допълнителните, специално разработени към системата дидактични материали.

Учителят свързва робота с компютъра и заедно с учениците съставят кодове и ги тестват. Работи се по задачите от урока. Обръща се внимание на изискването за позиция на робота в началната и в крайната точка. По таблото могат да се предложат и допълнителни задачи, както и да се даде възможност на учениците сами да съставят свои задачи по същото табло.

Последната задача учениците изпълняват самостоятелно на своите компютри (където предварително трябва да е инсталирана програмата и да е съхранено игралното табло с българската азбука). Те използват средата, без свързан робот към нея – в режим на симулация.

Ако се разполага с училищен комплект от 6 робота се използва екипна работа, при която след като се симулира движението на робота-пчела от всички ученици индивидуално, екипът се събира и тества съставени кодове с движението на реалния робот.

Урок 22. Моят робот

Урок за упражнение

Цел. Затвърждаване на знания и умения за управление на програмируемо устройство чрез визуална среда за програмиране.

Задачи:

1. Да се затвърдят знанията и уменията за работа с конкретна среда за визуално програмиране за управление на програмируемо устройство.
2. Да се затвърдят знанията и практическите умения за свързване на програмируемо устройство с дигитално устройство.

Очаквани резултати:

- Познава функционалността на бутони на конкретно устройство.
- Умее да ползва бутоните за задаване на поредица от движения в различни посоки.
- Познава блокове от команди за включване на програмируеми устройства към дигитални устройства.

Методически насоки

Урокът започва с актуализиране на знанията на учениците за програмируемия робот Blue Bot и средата за програмиране на движението му.

Учителят поставя задача 1: *Подреди код. Роботът трябва да премине през всички жълти фигури. Начална позиция долен ляв ъгъл. В крайната позиция да е обърнат на север.*

Първата задача се изпълнява без работа на компютър от учениците. Учителят поставя задачата. Работи се специално разработено табло с геометрични фигури (5x4). Роботът се поставя в долния ляв ъгъл. Задачата е да се програмира движение, при което пчелата да мине през всички жълти фигури.

Обсъждат се различни маршрути. Всеки ученик подрежда код за движение с помощта на картоните с команди за движение, изрязани от тетрадките. След това учителят избира няколко от подредените алгоритми и заедно с учениците ги тестват през средата за визуално програмиране. За работата в урока учителят трябва да разполага с игрално табло. Може да се използва разчертано табло и карти с изображения, които се поставят върху съответните полета или подготвените дидактични ресурси към системата.

При тестването на създадените алгоритми, учителят избира подходящ, за да покаже как се работи в режим Explore, но с модула Repeats – с повторения. Вместо да се дадат три команди Напред, може да се даде една и да се зададе тя да се повтори 3 пъти. Може да се зададе група от команди, която да се повтори няколко пъти – както при командите за цикъл в средата Scratch. Например, повтори 3 пъти напред и наляво. Добре е да се включи молива, за да се види очертаната пътека на движение на робота-пчела по таблото.

При избор на режим с повторение, в менюто (горе вляво) се появява още един бутон. При щракване върху този бутон до всички команди се появява по едно очертано кръгче. Избираме първата команда, от която искаме да стартира цикъла. След това щракваме върху последната команда в цикъла. Поредицата от команди, които искаме да повтаряме се оцветява в червено и се появяват бутони, с помощта на които можем да зададем броя на повторенията – с + увеличаваме с 1, с – намаляваме, с x изтриваме и се отказваме от повторенията. Ако искаме да повторим само една команда, например, командата Надясно, 4 пъти, то щракваме два пъти върху червеното кръгче до нея.



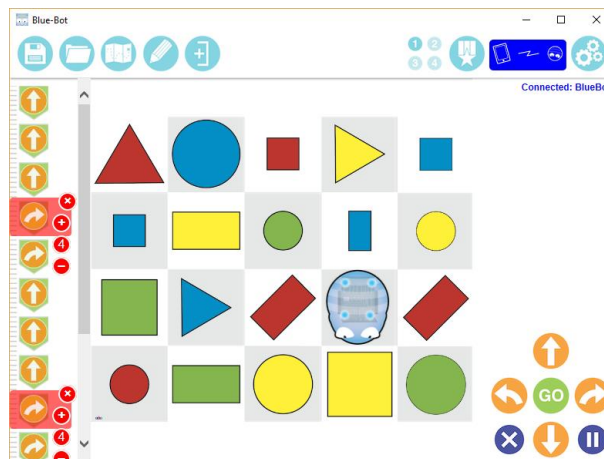
– инструмент за задаване на определен брой повторения на една команда



– командата Надясно ще се повтори 4 пъти

Следващите задачи се изпълняват от учениците на компютрите им, като работят в режим симулатор.

Втората задача се изпълнява в режим Explore – Repeats. Пчелата тръгва от долния ляв ъгъл и трябва да мине през всички триъгълници. Поставя се допълнително условие да се завърти на място, когато стигне в поле с триъгълник (4 пъти завъртане в една посока) и след това да продължи пътя си. Режимът Repeats (повторения) дава възможност да се пише по-кратък код, като се отбележи броят на повторенията на една или няколко поредни команди от кода.



Учителят дава време за изпълнение на задачата, след това избира ученик, който да му продикува кода и заедно тестват верността на алгоритъма с пчелата-робот.

Преминва се към последната задача. Работи се по същото табло. Работи се в режим Challenge Game (предизвикателства) – Obstacles (препятствия). Всеки ученик работи със собствено темпо. В този режим програмата избира начална позиция за пчелата, поставя крайната цел в произволно поле и поставя знак „Забранено“ на някое от полетата. Целта е пчелата да стигне до крайната цел, без да стъпва на забраненото поле.

Урок 23. Програмирам робот. Знам и мога

Урок за обобщение

Цел. Обобщаване на знания за управление на програмируемо устройство чрез визуална среда за програмиране.

Задачи:

1. Да се обобщят знанията и уменията за работа със среда за визуално програмиране за управление на програмируемо устройство.

Очаквани резултати:

- Познава функционалността на бутони на конкретно устройство.
- Умее да ползва бутоните за задаване на поредица от движения в различни посоки.
- Познава блокове от команди за включване на програмируеми устройства към дигитални устройства.

Методически насоки

Урокът започва с обобщаване на наученото. В хода на беседа се обобщава наученото – понятието за програмируемо устройство, блоковете за управлението му, средата, с която роботът може да се управлява, инструментите на средата.

Преминава се към работа по задача 1 – *Напиши код по таблото “Treasure”. Намери път към съкровището, скрито в горния десен ъгъл на позиция E5, като не трябва да минаваш през зловещата гора от A4, пиратския кораб в E1 и омагьосаното дърво в E3. Начална позиция A1.*

Учителят уточнява понятията A1, E5 и т.н. Картата е разчертана на квадратни полета. По хоризонтала полетата са именувани от А до Е на латиница, а по вертикала са номерирани от 1 до 5. E1 е квадратното поле на първия ред в последната колона – там се намира пиратския кораб. В поле A4 е гората – тя се намира в първата колона на 4-тия ред и т.н.

На празно очертано игрално табло, учителят поставя знака за крайна цел в поле E5. В полетата, които трябва да се избягват поставя знака „Забранено“ – това са полетата A4, E1 и E3. Поставя пчелата-робот в поле A1, ориентирана на север (или към поле A5).

Учениците стартират работа по задачата на компютрите си – работят в режим Explore/Basic Programming. Включват моливчето, за да се очертава пътя, през който минават.

Учителят дава време за изпълнение на задачата, след което избира един или няколко от учениците и всички заедно тестват маршрутите им с робота-пчела.

При наличие на училищен комплект, задачата-мисия „Да намерим съкровището“ може да се реализира чрез екипна организация на работата.

Преминава се към задачи 2 и 3. Задача 2 е подобна на първата задача. Може да се изпълни и при екипна организация на работа и в игрова форма с игрални табла и пластмасови фигури и др.

Задача 3 цели затвърждаване на уменията за създаване на алгоритми с повтарящи се действия. Развива се логическо и творческо мислене. В режим на симулация всеки ученик проектира своя траектория на движение и я очертава с робота-пчела. Работи се върху специално подготвено табло от файл – на бели и сиви квадрати. Работният файл е включен в ресурсите на учителя.

Урок 24. Знам и мога

Урок за обобщение

Цел. Обобщаване на знания за работа със Scratch и за управление на програмируемо устройство.

Задачи:

1. Да се обобщят знанията и уменията за работа в среда за визуално програмиране Scratch.
2. Да се обобщят знанията и уменията за работа със среда за визуално програмиране за управление на програмируемо устройство.

Очаквани резултати:

- Познава команди за създаване на последователни, разклонени и алгоритми с повтарящи се действия в Scratch.
- Синхронизира действията на героите.
- Умее да управлява програмируемо устройство през среда за визуално програмиране.

Методически насоки

Урокът прави преговор на наученото в игрова форма. Разработено е специално табло за средата Blur Bot върху което са поставени персонажи от вече реализирани проекти и задачи в средата за визуално програмиране Scratch. Чрез задачи за придвижване на пчелата-робот до определени полета се припомнят съответни знания за алгоритми, команди и начини за тяхното използване.

Урок 25-28. Компютърна игра „На училище“

Комбинирани уроци

Цел. Придобиване на умения и компетентности за разработване на компютърна игра.

Задачи:

1. Да се придобият умения за планиране и подготовката на ресурси за създаване на компютърна игра
2. Да се затвърдят и разширят умения за програмиране на сцената и героите според избрания сценарий и правилата на играта.
3. Да се изградят умения за работа в екип.

Очаквани резултати:

- Умее да подбира герои според зададения сюжет
- Умее да задава основни характеристики – костюми и декори
- Умее да задава правила за играта
- Използва променливи за отчитане на резултата
- Умее да построи код чрез командни блокове за управление на героите и сцената в играта.

Учебно съдържание

Играта групова или индивидуална дейност, организирана по някакви правила. Игрите, които имат познавателна цел наричаме образователни. Чрез игровия подход можем да формираме, затвърждаваме и обобщаваме знания. Ефективността на играта се постига, когато се постигне желание и мотивация на игра. Това зависи от много фактори – сценарий, съобразен с възрастовите особености на играчите, правилата, системата за успех или загуба. При компютърни игри особено важни са дизайна, качеството на изображения, видео, анимация.

Игрите имат ясно зададени правила и цели, които трябва да се спазват от всички играчи.

Създаването на компютърни игри от учениците е особено ефективен подход при формиране на дигитални компетенции у учениците. В начална училищна възраст учениците обичат да играят. През играта те учат не само факти и понятия, но и формират много личностни умения и навици. Създаването на компютърни игри развива критично, творческо и логическо мислене у учениците.

В хода на серия от уроци учителят и учениците разработват и програмират образователни компютърни игри. В тях се включват „сюжетно“ обвързани задачи с определени очаквани резултати – достигане до определена позиция или резултат. В игрите често се използват случайни стойности; когато участват повече състезатели има състезателен елемент и се събират точки.

Процесът на разработка на образователна компютърна игра минава през няколко етапа:

1. Определяне на сюжета и правилата на играта. Определяне началната позиция и целта.
2. Създаване на сценарий с определени герои, сцени и действия. Сценарият и планът за неговата реализация се уточнява в диалог с учениците и се записва на дъската и/или в тетрадките. Ако се работи групово, се определят ролите и отговорностите на учениците в екипите.
3. Компютърна реализация на сценария.
4. Тестване и коригиране на създадените компютърни игри.

Методически насоки

В поредица от уроци се предвижда съвместно създаване на компютърна игра – На училище.

Разглеждаме следния примерен сценарий:

Ани ни показва своето училище и в хода на тази разходка ни поставя различни задачи и въпроси. За всеки верен отговор получаваме точки. Според броя на спечелените точки получаваме звезда на знанието – колкото по-висок е резултата, толкова по-силно свети звездата.

Конкретните параметри на играта определяме по следния начин:

- Играта има три нива – събиране, изваждане, история
- В първото ниво – събиране – трябва да се решат 5 задачи от събиране на естествени числа. Във второто ниво – 5 задачи от действие изваждане. В третото ниво трябва да се отговори на 3 въпроса, свързани със знания за обществената организация.
- В първото ниво за всяка вярна задача получаваме по 1 точка. Във второто ниво – по 2 точки. В третото ниво – по 3 точки.

- Краен резултат. Максималният брой точки е 24 точки. При до 12 точки губим играта – оценка „залягай над книгите“ и получаваме угаснала (сива звезда). При точки от 13 до 19 получаваме оценка „добра работа“ (светло жълта звезда). При точки от 20 до 24 получаваме отлична оценка „отлична работа“ (ярка жълта звезда).

Основни елементи по реализиране на сценария:

- Сцените са пет – една за начало, една за края и по една за всяко ниво на играта.
- Героите са: Ани, която задава задачи и въпроси, бутони Старт и Стоп, и звезда за крайния резултат.
- Сюжетната линия се реализира чрез следните сцени:
 - Сцена 1. Ани се появява и ни предлага да играем в нейното училище за игри.
 - Сцена 2. Кабинет по математика. Урок по събиране. Ани задава 5 задачи от действие събиране. Верен резултат носи по 1 точка.
 - Сцена 3. Кабинет по математика. Урок по изваждане. Ани задава 5 задачи от действие изваждане. Верен резултат носи по 2 точки.
 - Сцена 4. Кабинет по история. Ани задава 3 въпроса, свързани с историята на България. Правилен отговор носи по 3 точки.
 - Сцена 5. Ани е отново пред училището. Обявява крайния резултат. Появява се звездата, която сме спечелили – тя може да е сива, светложълта или ярко-жълта, според броя на спечелените точки.

Програмирането на играта в средата за визуално програмиране Scratch включва избор на декори, герои, логика. Използват се последователни действия, разклонени алгоритми, алгоритми с повтарящи се действия. Използва се случайни числа, оператори за аритметични и логически операции. Използва се синхронизиране на действията на героите в проекта. Редактира се външния вид на героите, използват се анимационни ефекти.

Урок 29-32. Работа по проект

Комбиниранни уроци

Цел. Придобиване на умения и компетентности за работа по проект.

Задачи:

1. Да се придобият умения за планиране и подготовката на ресурси за реализиране на компютърен проект
2. Да се затвърдят умения за програмиране на сцената и героите според избрания сценарий и вида проект – тест, комикс, пъзел, игра, калкулатор и пр.
3. Да се изградят умения за работа в екип.
4. Да се придобият умения за представяне на проекта в реална и виртуална среда.

Очаквани резултати:

- Знае основните стъпки за разработване на проект.
- Умее да планира работата по определен проект и да работи в екип.
- Умее да подбира герои според зададения сюжет.
- Умее да задава основни характеристики – костюми и избира декори на сцената.
- Умее да създава код чрез използване на командни блокове за управление на героите в проекта.

- Знае основните правила за представяне на проекта пред аудитория.

Учебно съдържание

Разработката на проект е процес, който преминава през няколко основни етапа:

1. Анализ на поставената за реализиране задача.
2. Проектиране и планиране на процеса на разработка.
3. Реализиране (кодиране) на проекта в определена програмна среда.
4. Тестване на проекта, използване и поддръжка.

През първия основен етап се анализира поставената задача. Задава се въпроса „Какво?“. Учениците трябва да разберат какво трябва да се реализира.

През втория етап се сформират екипите и се прави план за изпълнение на поставеното задание. Създаденият план се поставя на видно място и се изпълнява според зададените срокове и отговорници. През този етап се задават въпросите „Кога?“ и „С кого?“.

През третия етап се реализират задачите според създадения план. Създават се героите и декорите. Програмират се действията на героите и взаимодействията помежду им. През този етап основния въпрос е „Как?“

През последния етап се правят тестове на създаденото приложение. Разиграват се определените в началото сценарии, откриват се грешки и се коригират. Когато проектът е готов, се представя пред останалите екипи и се предоставя за ползване. Може да се сподели в защитеното интернет пространство на <https://scratch.mit.edu/>

Методически насоки

Представена е подробна разработка на проект на тема „Пътешествие из България“. Избран герой посещава популярни забележителности в България и ни задава въпроси, свързани с тях.

Етапите за разработване на проект и дейностите по тях са обобщение в следната таблица:

Етап	Дейности
Анализ	Диалог с учениците за това как си представят реализацията на проекта.
Проектиране	<p>Уточняват се някои въпроси като:</p> <ul style="list-style-type: none"> - Кои забележителности ще бъдат включени? - Колко и какви герои ще участват в пътешествието? - Какви ще са декорите? - Какви въпроси ще се задават? - Какви анимационни ефекти ще се използват? - Ще има ли и какви звукови ефекти и т.н. <p>Създаване на сценарий с определени герои, сцени и действия.</p> <p>Сценарият и планът за неговата реализация се уточнява в диалог с учениците и се записва на дъската и в тетрадките.</p>

	Учениците могат да се разделят на екипи. Определят се ролите и отговорностите на учениците в екипите.
Кодиране	- Създават се героите. - Определят се декорите на сцената. - Програмират се действията на героите.
Тестване и представяне	- Тества се проектът според сценария. - Коригират се грешки и несъответствия. - Представя се пред останалите ученици. - Предоставя се за използване.

Примерно реализиране на проекта

Сценарий: Калин развежда Зара из България и ѝ разказва за забележителностите из страната ни. Той задава въпроси, чийто отговор е някакво число. За да продължим напред, трябва да отговорим правилно на зададения въпрос.

Нека след диалог с учениците през втория етап да са определени следните параметри на проекта:

- забележителности: Рилски манастир, връх Мусала и паметника Шипка.
- участват няколко герои – момче, момиче, звезда, както и избраните забележителности.
- декорът е един – картата на България.
- въпросите са свързани със забележителностите и изискват числов отговор. Пътешествието продължава само при верен отговор.
- анимационните ефекти са движение на героите, изчертаване на линията на маршрута, появяване и скриване на героите.
- звуков ефект при верен отговор.

Реализираме проекта в Скрач като:

- Работим с подготвен работен файл, в който е даден декор – контурна карта на България. Картата е предоставена в ресурсите на учителя и като файл – векторна графика (формат svg). Форматът е непознат на учениците, поради което е подготвен работен файл. Но декорът може да бъде зареден от дадения файл, така както се работи с файлове png, jpg и т.н. Проектът може да се разработи изцяло самостоятелно.
- Картата може да се оцвети в зелен цвят или да се оцвети в цветовете на българското знаме – по решение на екипа/ученика.
- Героите избираме от библиотеката. Желателно е да работим с векторни герои, за да се получи визуално качествен краен продукт. Подходящи векторни герои-хора са Abby и Dee. С графичния редактор можем да създадем свои герои (желателно с инструментите за векторна графика)
- От интернет сваляме малки графични изображения на забележителностите Шипка, Рилски манастир и връх Мусала. В графичния редактор ги изрязваме и обработваме.
- Определяме следния сценарий:
 - Част 1: Зара (Zara) и Калин (Kai) се срещат и решават да направят пътешествие. Героят „Звезда“ е върху картата. Останалите герои са скрити

- Част 2: Героят „Рилски манастир“ се показва. Калин задава въпрос на Зара за Рилския манастир. При верен отговор продължават пътешествието, при грешен Калин дава джокер – израз от случайно генерирани числа със стойност правилния отговор. Това се повтаря докато се даде верен отговор. Героят „Звезда“ изчертава първата част от маршрута.
- Част 3: Героят „Мусала“ се показва. Калин задава въпрос на Зара за връх Мусала. Ако Зара даде верен отговор, продължават напред. Ако допусне грешка, Калин дава джокер – израз подобен на този от Част 2. Това се повтаря до верен отговор. Героят „Звезда“ изчертава следващата част от маршрута.
- Част 4: Героят „Шипка“ се показва. Калин задава въпрос от типа „нагоре-надолу“. Повтаря се до верен отговор.
- Част 5: Зара и Калин се разделят и си тръгват от сцената.

Могат да се дадат предложения за развиване на примерния проект:

- Допълнително анимиране на движението на героите
- Добавяне на фонов звук и звукови ефекти
- Добавяне на допълнителни забележителности и въпроси към тях
- Други допълнения и идеи на учениците

Урок 33-34. Представи проект. Обобщение и самооценка

Уроци за обобщение и диагностициране

Цел. Обобщение и преценка на наученото.

Методически насоки

Последните два часа за годината са предвидени за обобщаване на наученото и представяне на проекти и компютърни игри, разработени от учениците. Може да се разработи тържествен сценарий, включващ презентирание пред публика.

Чрез представяне на работата публично се развиват редица личностни качества на учениците, като комуникативно-речеви умения, презентационни умения, творчество, концентрация, дисциплина и мн. др.

Изходяща диагностика

Оценяването на знанията и уменията на учениците се осъществява с качествена оценка и съобразно с постигнатите очаквани резултати по обобщени теми на ниво учебна програма. В предметът „Компютърно моделиране“ приоритетно се оценяват уменията на учениците на основата на **постигнат краен продукт от практическа дейност**, като се взема предвид реалното им поведение по време на работа. Отчитат се изпълнението на възложената практическа задача, подбора на средствата, наличието на индивидуално решение и процеса на реализация. Препоръчително е при оценяване постиженията на учениците да се използва карта за оценка. Тя съдържа система от критерии и показатели относно знанията, уменията и отношенията, необходими за реализирането на определена задача.

Основна дидактическа цел на задачите е тренингът. На учениците се предлага „сюжетно“ формулирана задача с точно определени очаквани резултати. Изпълнявайки стъпка по

стъпка предложеният алгоритъм за решаване на поставената задача, обучаваният усвоява определени подходи в компютърното програмиране. Успешното решение на задачите стимулира ученика и му позволява да експериментира и твори, провокира го към самообучение и усъвършенстване.

При представяне на готовите проекти на анимирани истории и образователни игри се прави и самооценка от разработващия ученик или екип (**Карта за самооценяване**)

Критерии за оценяване на проект – Таблица за оценителя / учителя

№ по ред	Критерий	Точки (максимален брой точки 24)	
		0 точки Не е спазено изискването в условието	1 точка спазено изискването в условието
	Отваряне на нов проект в Scratch		
1	Избор на подходящи сцени според зададения в проектното задание сюжет		
2	Самостоятелно изработени сцени		
3	Избор на подходящи герои според зададения в проектното задание сюжет		
4	Самостоятелно изработени герои		
5	Използване на блок за разклонение на алгоритъм		
6	Използва блокове, представлящи аритметични действия		
7	Използва блокове за сравняване на числа		
8	Използва блокове за логически оператори		
19	Влага блок в блок		
10	Управлява движението на героите (изчакване, забавяне или ускоряване).		
11	Използва начини за поява на герой след настъпване на събитие		

12	Манипулира външния вид на героя с цел постигане на различни анимации		
13	Задава последователност от звуци, които да изпълнява героят		
14	Задава текст, който да се изобразява в текстово поле, свързано с героя		
15	Спазва книжовно-езиковите норми на българския език		
16	Синхронизира движенията и разговорите на повече от един герой последователно и с нужното изчакване		
17	Създава и именува променливи		
18	Присвоява стойност и използва числови променливи		
19	Има синхрон между сцените		
20	Има синхрон между героите		
21	Представяне на проекта – 4 точки		

Скала за оценка

Оценка = 2+брой токи/6

Карта за самооценяване на проект

№ по ред	Критерий	Точки (максимален брой точки 24)	
		0 точки Не е спазено изискването в условието	1 точка спазено изискването в условието
	Отваряне на нов проект в Scratch		
1	Избор на подходящи сцени според задания в проектното задание сюжет / Самостоятелно изработени сцени		
2	Избор на подходящи герои според задания в проектното задание сюжет / Самостоятелно изработени герои		
3	Използване на блок за разклонение на алгоритъм		
4	Използва блокове, представлящи аритметични действия		
5	Използва блокове за сравняване на числа		
6	Използва блокове за логически оператори		
7	Използва начини за поява на герой след настъпване на събитие		
8	Манипулира външния вид на героя с цел постигане на различни анимации		
9	Задава текст, който да се изобразява в текстово поле, свързано с героя. Спазва книжовноезиковите норми на българския език		
10	Синхронизира движенията и разговорите на повече от един герой последователно и с нужното изчакване		
11	Създава и именува променливи. Присвоява стойност и използва числови променливи		
12	Има синхрон между сцените и между героите		

Критерии за оценяване на образователна игра – Таблица за оценителя / учителя

№ по ред	Критерий	Точки (максимален брой точки 80)				
		4 точки	3 точка	2 точки	1 точка	0 точки
		Спазени са всички изисквания	Спазен и са 3 от изискванията	Спазен и са 2 от изискванията	Спазено е 1 от изискванията	Не е спазено изискването в условието
	Отваряне на нов проект в Scratch					
1	Определяне на сюжета и правилата на играта.					
2	Определяне частите на сюжета (брой части/теми/задачи)					
3	Избор на подходящи сцени според сюжета на играта (брой сцени)					
4	Самостоятелно изработени сцени					
5	Избор на подходящи герои според сюжета на играта					
6	Самостоятелно изработени герои					
	Описание на сюжета и действията на героите в Част 1 от сюжета					
7	Сцена					
8	Герой					
9	Променливи					
	Описание на сюжета и действията на героите в Част 2 от сюжета					
10	Сцена					
11	Герой					
12	Променливи					
	Описание на сюжета и действията на героите в Част 3 от сюжета					
13	Сцена					
14	Герой					
15	Променливи					
	Допълнения към играта:					
16	Допълнително анимиране на движението на героите					

17	Добавяне на фонов звук и звукови ефекти					
18	Други допълнения и идеи на учениците за доразвиване на играта					
19	Изготвяне на документация с описание условията на играта					
20	Представяне на играта пред публика					

Скала за оценка

Оценка = 2+брой токи/20

Карта за самооценяване на игра

№ по ред	Критерий	Точки (максимален брой точки 24)		
		Не е спазен	Частично спазен	Напълно спазен
1	Избор или самостоятелно изработване на декори и герои според зададеното проектно задание.	0	1	2
2	Използване на блок за разклонение на алгоритъм и блокове за логически оператори	0	1	2
3	Използва блокове, представлящи сравняване на числа и аритметични действия.	0	1	2
4	Използва начини за поява на герой след настъпване на събитие	0	1	2
5	Манипулира външния вид на героя с цел постигане на различни анимации	0	1	2
6	Задава текст, който да се изобразява в текстово поле, свързано с героя. Спазва книжовно-езиковите норми на българския език	0	2	3
7	Синхронизира движенията и разговорите на повече от един герой последователно и с нужното изчакване. Изпращане на съобщения	0	2	3
8	Създава и именува променливи. Присвоява стойност и използва числови променливи	0	2	3
9	Управление на събития, според сценария като докосване на герой, смяна на декор, събития на мишката и клавиатурата	0	2	3
10	Използва звукови ефекти и добро графично оформление	0	1	2