**Сценарий 10 – Хранене на котките**

Задача: Програмирайте игра, при която пазителят на котешкия приют многократно ще пита играча за броя на котките, които може да храни в определена стая. Броят на котките зависи от броя (2 до 10) и размера (2 до 5) на купите. За всяка стая тези два номера трябва да бъдат присвоени произволно. Размерът на купата показва колко котки могат да ядат от нея, например ако размерът на купата е 3, това означава, че 3 котки могат да ядат от нея.

Трябва да имаме и брояч, който да отчита правилните отговори. Създайте игра, в която играчът ще трябва да отгатне точния брой котки, които можем да храним във всяка стая. След като изпълните задачата дайте обратна връзка за това колко пъти играчът е отговорил правилно и колко пъти е сгрешила.

Процесът на създаване на играта:

1. Отворете файла - шаблон:

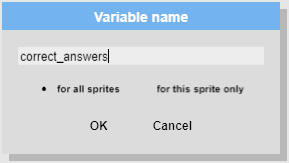
<https://snap.berkeley.edu/project?user=zapusek&project=cat_feeding_template>

в него има картинки за фон на сцената и главния герой – пазачът на приюта.

|  |  |
| --- | --- |
|  |  |

1. Искаме да преброим верните отговори. Ако искаме да съхраним стойност в програма, трябва да използваме променлива. Можем да получим достъп до команди относно променливи в групата „Променливи“. Нова променлива се създава, когато щракнем върху: .

Променливите свързват име и стойност, така че когато правим нова променлива, първо трябва да я наименуваме. Препоръчва се да използваме мнемонично име, това означава, че можем да разберем какъв вид стойност се съхранява в променлива от нейното име. За преброяване на верните отговори можем да изберем името „правилни\_отговори“ (correct\_answers).



Стойността на променливата може да бъде зададена или променена с някаква стойност. Ако искаме да зададем стойността на променливата, използваме блока: . Всеки път, когато зададем стойност на променлива, предишната стойност се заменя.

Ако искаме да променим текущата стойност с друга стойност, трябва да използваме блок: . Текущата стойност на променливата ще бъде променена от стойността, която посочим в празно пространство.

1. Сега ще започнем всъщност да кодираме играта. Първо помислете как действията във всяка стая всъщност ще бъдат много сходни, еднакви дори. Във всяка стая трябва да:
   1. задайте произволна стойност за броя на купите и за размера на купата.

В групата „Оператори“ (Operators) можем да намерим блока, който връща произволната стойност от интервал, който можем да посочим. Например:  ще върне произволното число между 1 и 10.

Ще ни трябват стойностите на броя и размера на купите във всяка стая, съхранявани за по-късно, когато ще трябва да изчислим верния отговор. Това може да се постигне чрез създаване на нови променливи и присвояване на тези стойности към тях.

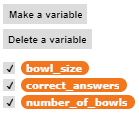
Създайте две променливи:

1. за броя на купите
2. за размера на купите във всяка стая

Задайте им произволни стойности според спецификацията на тази игра. Използвай блокове:  и .

* 1. Играчът трябва да знае кои случайни стойности е избрал компютърът, затова трябва да я информираме. Използвайте блокове:  ,  и стойностите на променливите, които съхраняват броя и размера на купите:  и .

Можете да намерите препратки към променливи в групата „Variables“:



* 1. Сега трябва да подканим играча да въведе нейния отговор. Ако искаме да получим входните данни на играча, трябва да използваме входния блок от групата „Sensing“: .

Когато играчът напише отговора си в полето за въвеждане, той се съхранява в променливата „answer“:.

* 1. След това трябва да проверим дали отговорът на играча е верен или не. Помислете как можете да изчислите верния отговор от броя и размера на купите. Ако сравним стойността, съхранявана в  с броя на котките, които можем да храним (изчисляваме тази стойност), можем да разберем дали отговорът е верен или не.

Изчислете верния брой котки, които можем да храним във всяка стая, като използвате тези блокове:

,  и .

* + 1. ако отговорът е верен: поздравете играча и добавете един към стойността на променливата correct\_answer.
    2. ако отговорът е грешен: дайте подходящата обратна връзка.

Можем да разграничим точно две възможни ситуации с използването на блок if-else:

С две ситуации (правилен или неправилен отговор) с помощта на условния блок за разклонение „if-else“.



Ако отговорът ще бъде верен, той ще бъде равен на изчислената стойност, в противен случай няма да бъде. Попълнете кода, като използвате блока сравнение, който проверява равенството:.

1. Реализирахме кода за една стая, но в приюта има десет стаи. Можем да копираме горния код десет пъти и да го поставим последователно, но това не би било оптималният начин да го направим. Вместо това можем да използваме цикъл, който ще повтори същия код десет пъти. Най-простият цикъл за постигане на това е цикъл reapeat [n]:



1. Когато играта приключи, трябва да предоставите обратна връзка: брой верни и грешни отговори. Броят на верните отговори се съхранява в променлива и броят на грешните отговори може да бъде изчислен. Ако знаем, че играчът ще въведе числото 10 пъти и също така знаем колко пъти е отговорил правилно, можем да разберем колко пъти той греши. Завършете играта, като използвате тези блокове:

,  and .

1. Можете да надстроите играта, като използвате следните предложения:
   1. Пазителят на приюта включва номера на всяка стая във въпроса си. Например: „Познайте броя на котките, които мога да храня в стая 5“.
   2. Ако отговорът на играча е грешен, пазачът казва правилния номер.
   3. Вместо десет стаи, номерът на стаята е случаен или играчът въвежда броя на стаите в началото на играта.
   4. Добавете свои собствени идеи и надстройте играта по ваш вкус.