

Шалений Фелікс (рівень 6)

Нотатки для кервників клубу

Передмова:

У цьому проєкті діти створюватимуть платформу для гри, подібної на Manic Miner або Mario Bros. Він буде найважчим з тих, які вони виконували, тож зберіться з силами! Проєкт виконуватиметься протягом 3-4 тижнів.

Ресурси інтерфейсу

У даному проєкті використовуються ресурси, надані Code Club, тож перевірте наявність відповідної папки. Інші ресурси, необхідні їм, діти створять самостійно.

Необхідні скретч картки:

- Записати послідовність
- Відтворити послідовність

Кілька слів про складнощі для виконання і фіксацію моменту зіткнення:

Manic Miner і Mario Bros - ігрові платформи, побудовані таким чином, що фігурка головного персонажа рухається в межах екрану у різні боки, натикаючись на певні предмети. Деякі предмети, наприклад, стіни і перекриття, просто зупиняють рух фігурки. Інші ж – погані персонажі - вбивають головного героя, якщо він на них натикається. Є також предмети, які треба збирати, наприклад, ключі. Деякі предмети просто формують тло і не впливають на гру взагалі.

Це означає, що важливо визначити момент, коли головний герой натикається на певний предмет (це називається " фіксація моменту зіткнення ", можливо, вам доведеться пояснити учням детальніше). У Скретч існує кілька блоків команд, за допомогою яких можна фіксувати момент зіткнення: (доторкається [спрайт] ?; доторкається кольору [] ?; колір [] доторкається [] ?; відстань до [спрайт]?). Але для того, щоб створити гру, потрібно не лише знати про сам факт зіткнення, а й про те, якою стороною фігурка наткнулась на певний предмет. Якщо фігурка наткнулась на ліву стіну, вона не може далі рухатись вліво.

Але вона ще може рухатись вправо, стрибати, падати, якщо немає перекриття. Зіткнення з негативним персонажем-ворогом може вбити головного героя, але якщо головний герой торкнеться цього персонажа ногами, то ліквідує його. Жоден з

вбудованих блоків Скретч не дає змоги виявити місця та напрямку зіткнення. Для цього треба створити власний детектор зіткнення.

Детектори зіткнення: проблема зіткнення з кулею

Ви, мабуть, помітили, що Фелікс рухається досить повільно. Це зроблено для запобігання так званій проблемі зіткнення з кулею. В іграх, які передбачають стрільбу кулями по тонких мішенях, може статися так, що куля проходить крізь мішень, не зачепивши її. Це відбувається тому, що комп'ютер переміщує кулі з певним кроком і перевіряє, чи відбулось зіткнення, після кожного такого кроку. Тож якщо один крок був до моменту влучання в мішень, а наступний уже після цього моменту, комп'ютер ніколи не зрозуміє, що куля вдарилася об мішень.

Детектори зіткнення у цій грі мають таку ж проблему, адже вони досить тонкі. Якщо збільшити відстань, на яку Фелікс переміщається при кожному своєму кроці (спробуйте 20 і більше), а потім перетягувати його в різні місця, ви зможете побачити, що він іноді проходить крізь перекриття!

Ви можете запобігти переміщенню Фелікса крізь тонкі предмети, збільшивши величину детекторів, але це не заважатиме Феліксу застрягати всередині предметів, перш ніж програма визначить, що він зіткнувся з певним предметом. Є кілька способів подолання цієї проблеми (наприклад, оточити все ґратами, або переміщати Фелікса трохи назад, якщо він наткнувся на щось, або передбачати траєкторію його руху і повільно наближатись до відповідного предмета), але вони або складні для виконання, або повільно діють, або і те, і друге. Я краще примирюся з неквапливим рухом Фелікса у цьому проекті. Якщо ви хочете спробувати щось інше, - вперед!

Намалюйте власні детектори зіткнення

Якщо діти хочуть використовувати інший образ замість Фелікса, їм доведеться створювати свої власні детектори зіткнень для цього. Я створював їх, імпортуючи образ в GIMP і створюючи один шар (рівень) для кожного бокового штриха. Вибираючи, які рівні зробити видимими, ви можете легко експортувати всі чотири детектори зіткнень (як GIF файли). Як і при проектуванні рівнів (див. нижче), переконайтеся, що ви отримаєте правильний колір, коли ви конвертуєте зображення в індексовану палітру.

Зверніть увагу, що детектори верхнього і нижнього краю не повинні перекривати бічних. Це запобігає зависанню спрайта у повітрі, якщо він натикається на перешкоду під час стрибка.

Завдання 4: Ключі і цілі

Вас може здивувати, що з самого початку ми прописуємо місце для ключів, приставки і Фелікса та надсилаємо повідомлення на перемогу. Це зумовлено тим, що коли ви створите свої власні рівні, ви захочете, щоб ці речі з'являлися в різних місцях на різних рівнях. Якщо ми зафіксуємо їх зараз, то потім буде легше зробити так, щоб рівні відображалися правильно.

Тиждень 2: Створення рівнів

Якщо при створенні рівнів ви використовуєте програми для малювання (а не вбудовані інструменти Скретч), будьте обережні під час конвертування їх в GIF. Перехід від RGB-зображення до зображення на основі індексованої палітри може спричинити зміни в кольорах, які неможливо побачити неозброєним оком. Це означає, що деякі з датчиків, які реагують на кольори, більше не будуть працювати. Можливо, вам доведеться вручну редагувати кольорову палітру, щоб зберегти необхідні кольори.

Переконайтеся, що діти ознайомлені з усією інформацією, яка знадобиться для створення кожного рівня. Їм буде потрібно:

- Початкові значення x і y для Фелікса
- Значення x і y для приставки
- Значення x і y для кожного ключа
- Чи ключ видно на кожному рівні? Скільки ключів є на рівні?
- Фонове зображення для рівня кожного негативного персонажа
- Початкові x і y значення
- Кінцеві значення x і y
- Початковий напрям
- Видимий чи ні?
- Горизонтальний чи вертикальний рух. Образ

Тиждень 3

Завдання 1: Відображення нового рівня

Майте на увазі, що виконання цього завдання потребує часу. Є багато змін, які потрібно внести. Майже кожен спрайт повинен реагувати на повідомлення про **перехід на новий**

рівень. Для кожного спрайта треба буде створити кілька списків із зазначенням **перейти на новий рівень** на кожному рівні.

Заохочуйте дітей, щоб вони створювали один спрайт за раз і тестували його. Ми використали кілька фрагментів скрипту на **Сцені** для встановлення поточного рівня і подальшого показу повідомлення **перейти на новий рівень**. Якщо діти створюють одразу кілька ключів або перешкод, попросіть видалити їх всі для кожного типу спрайтів, внести зміни, а тоді дублювати спрайти і змінювати їх параметри.