



*School of
Computer
Science*

ГРАФИКА. ИГРЫ

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



ГРАФИКА

ЧТО ПРЕДСТОИТ ОСВОИТЬ

- Создание графического окна
- Спрайты и манипуляции с ними
- Текст в графическом окне
- Отслеживание столкновений между спрайтами
- Обработка ввода от мыши
- Управление действиями компьютерного противника
- Игра «Сумасшедшая масленица»



БИБЛИОТЕКИ PYGAME И LIVEWIRES

ИНСТРУКЦИИ ПО УСТАНОВКЕ

- Запустите командный интерпретатор (cmd.exe) или Power Shell
- Выполните команду:
 - `pip install pygame`
 - или
 - `python -m pip install <имя_файла>.whl`
- Распакуйте архив `livewires.zip`
- Выполните `livewires/setup.bat`



СОЗДАНИЕ ГРАФИЧЕСКОГО ОКНА

NEW_GRAPHICS_WINDOW.PY

- Создадим пустое графическое окно
- Координата **X** окна растёт слева направо, а **Y** сверху вниз
- Импорт модуля **games** из пакета **livewires**:
`from livewires import games`
- Инициализация графического экрана:
`games.init(screen_width, screen_height, fps)`
- Запуск графического экрана:
`games.screen.mainloop()`



ПОЛЕЗНЫЕ ОБЪЕКТЫ И КЛАССЫ GAMES

Объект	Описание
screen	Предоставляет доступ к графическому экрану
mouse	Предоставляет доступ к мыши
keyboard	Предоставляет доступ к клавиатуре

Класс	Описание
Sprite	Класс для создания спрайтов – объектов, которые могут отображаться на экране
Text	Подкласс Sprite. Для текстовых объектов, отображаемых на экране
Message	Подкласс Text. Для текстов, отображаемых на экране в течение заданного промежутка времени



ПОЛЕЗНЫЕ СВОЙСТВА ОБЪЕКТА SCREEN

Свойство	Описание
width	Ширина графического экрана
height	Высота графического экрана
fps	Частота обновления экрана (количество кадров в секунду, frames per second)
background	Фоновое изображение экрана
all_objects	Список всех спрайтов, содержащихся на графическом экране
event_grab	Логическая переменная, которая показывает, перенаправляется ли пользовательский ввод на графический экран



ПОЛЕЗНЫЕ МЕТОДЫ ОБЪЕКТА SCREEN

Метод	Описание
add(sprite)	Добавляет объект класса Sprite (или одного из его подклассов) на графический экран
clear()	Удаляет все спрайты с экрана
mainloop()	Запускает основной цикл работы графического экрана
quit()	Закрывает графическое окно



НАЗНАЧЕНИЕ ФОНОВОЙ КАРТИНКИ

BACKGROUND_IMAGE.PY

- Загрузка изображения:
Прежде чем получить картинку на экране, нужно загрузить ее в память, в переменную специального типа:
`wall_image = games.load_image("wall.jpg",
transparent = False)`
- Картинка фонового окна непрозрачная
- Загруженная картинка присваивается свойству `background` объекта `screen`:
`games.screen.background = wall_image`



ОТОБРАЖЕНИЕ СПРАЙТОВ

PANCAKE_SPRITE.PY

- **Спрайт** – это графический объект с картинкой в своем составе
- Загружаем картинку для спрайта в память:
`pancake_image = games.load_image("pancake.bmp")`
- Картинка для спрайта – прозрачная
- Создаем спрайт с применением ранее загруженной картинки и добавляем его на графический экран:
`pancake_sprite = games.Sprite(image =
pancake_image, x = 500, y = 300)
games.screen.add(pancake_sprite)`



СВОЙСТВА ОБЪЕКТОВ КЛАССА SPRITE

Свойство	Описание
angle	Наклон в градусах
x	Абсцисса
y	Ордината
dx	Скорость по оси абсцисс
dy	Скорость по оси ординат
left	Абсцисса левого угла спрайта
right	Абсцисса правого угла спрайта
top	Ордината верхнего угла спрайта
bottom	Ордината нижнего угла спрайта
image	Объект-изображение, на основе которого создан спрайт



СВОЙСТВА И МЕТОДЫ СПРАЙТОВ

Свойство	Описание
<code>overlapping_sprites</code>	Список всех объектов, которые на экране перекрываются с данным спрайтом
<code>is_collideable</code>	Логическое свойство, определяющее, будет ли спрайт «сталкиваться», т. е. должны ли регистрироваться его перекрытия с другими объектами на экране как столкновения

Метод	Описание
<code>update()</code>	Обновляет внешний вид спрайта. Автоматически вызывается при каждой итерации цикла <code>mainloop()</code>
<code>destroy()</code>	Удаляет спрайт с экрана



ОТОБРАЖЕНИЕ ТЕКСТА

BIG_SCORE.PY

- Для отображения текста на графическом экране используется объект типа `Text`:
Добавляем импорт модуля `color`:
`from livewires import games, color`
- При создании текста `x`, `y` задают координаты центра объекта, а `size` – высоту текста в пикселах:

```
score = games.Text(value = 1756521,  
                    size = 60,  
                    color = color.black,  
                    x = 910,  
                    y = 30)  
games.screen.add(score)
```



ВЫВОД СООБЩЕНИЯ **YOU_WON.PY**

- Экземпляр класса **Message** выводит сообщение на заданный период времени:
- **Message** – это подкласс от **Text**
- В атрибуте **lifetime** при создании для объекта типа **Message** указывается количество циклов **mainloop**, которые объект должен отображаться
- В параметре **after_death** указывается ссылка на функцию, которая выполнится после исчезновения сообщения



ПОДВИЖНЫЕ СПРАЙТЫ

MOVING_PANCAKE.PY

- Спрайты могут двигаться по экрану, что удобно для создания аркадных игр
- Для задания скоростей движения спрайта укажите ненулевые значения в атрибутах **dx** и **dy** при создании спрайта
- **d** – сокращение от «дельта», т.е. изменение
- **dx** – изменение абсциссы спрайта за один цикл **mainloop** в пикселах
- **dy** – изменение ординаты спрайта за один цикл **mainloop** в пикселах



УЧЕТ ГРАНИЦ ЭКРАНА

BOUNCING_PANCAKE.PY

- При столкновении спрайта с границей окна можно задать разное поведение – остановка, отскок, взрыв с исчезновением
- Наш блин будет отскакивать как мячик
- Для отскока нам нужно при достижении границ обратить ту компоненту скорости, которая привела к столкновению
- При достижении верхней или нижней кромки экрана меняем знак компоненте dy , а при столкновении с левой/правой границей окна – компоненте dx



ОБРАБОТКА ВВОДА С ПОМОЩЬЮ МЫШИ

MOVING_PAN.PY

- Объект `mouse` модуля `games` обеспечивает доступ к мыши
- Вызов `games.init()` создает объект `games.mouse`, который мы используем для чтения координат указателя мыши
- Далее создаем класс `Pan` для спрайта-сковородки
- В методе `update()` перемещаем сковородку в позицию указателя мыши



ПОЛЕЗНЫЕ СВОЙСТВА ОБЪЕКТА MOUSE

Свойство	Описание
x	Абсцисса указателя мыши
y	Ордината указателя мыши
is_visible	Логический параметр, будет ли указатель мыши видимым: True – да, False – нет. По умолчанию равен True





РЕГИСТРАЦИЯ СТОЛКНОВЕНИЙ

SLIPPERY_PANCAKE.PY

- В играх нужно как-то обрабатывать столкновения спрайтов
- В нашем примере блин будет «убегать» в случайную позицию на окне при столкновении со сковородой
- В классе **Pan** для каждой итерации **mainloop** проверяется было ли столкновение с блином
- Если столкновение было, то блину посылается сигнал для изменения его положения через вызов метода **handle_collide()**



ИГРА «СУМАСШЕДШАЯ МАСЛЕНИЦА»

CRAZY_STRAWMAN.PY

- Класс **Pan** – спрайт с изображением сковороды, которая управляется мышью
- За каждый пойманный блин игроку добавляется 10 очков
- Класс **Pancake** служит для генерации экземпляров блинов, которые сбрасываются с забора
- Как только блин коснется земли – игра окончена
- Класс **Strawman** – соломенное чучело, которое сбрасывает блины



ВОЗМОЖНЫЕ ЗАДАЧИ НА ДОМ

- Доработайте игру «Сумасшедшая масленица» так, чтобы сложность игрового процесса постепенно возрастала
Возможны разные способы добиться такого эффекта:
 - увеличить скорость падения блинов и/или перемещения чучела
 - уменьшить расстояние от крыши до сковороды
 - выпустить на экран несколько чучел
- Напишите игру, в которой на персонаж, управляемый игроком с помощью мыши, сверху (с «неба») будут падать какие-нибудь тяжелые объекты, а он должен будет уворачиваться
- Создайте простую игру в пинг-понг для одного игрока. В этой игре пользователь должен управлять ракеткой с помощью мыши, а шарик – отскакивать от трех стенок. Если шарик проскочит мимо ракетки и вылетит за пределы игрового поля, игра должна заканчиваться

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*