

*School of
Computer
Science*

Первые игры

ИЗУЧЕНИЕ PYGAME

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

- Как установить внешние модули?
- Для чего устанавливать внешние модули?
- Приведите пример внешних модулей.



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

- Дайте определение следующим понятиям:
- Игровое окно,
- игровой цикл,
- игровое событие,
- кадр,
- спрайт,
- коллизия,
- поверхность



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

- Как инициализировать модули Pygame?
- Как инициализировать игровое окно
- Как получить событие из очереди событий?
- Как получить тип события?
- Как обновить дисплей?
- Как отрисовать фигуру?



МОДУЛЬ СОБЫТИЯ

PYGAME.EVENT

- Pygame отслеживает все сообщения о событиях через очередь событий.
- Методы в этом модуле помогают управлять этой очередью событий. Входная очередь сильно зависит от модуля отображения (display) pygame.
- Если дисплей не был инициализирован очередь событий не будет работать.



МЕТОДЫ МОДУЛЯ EVENT

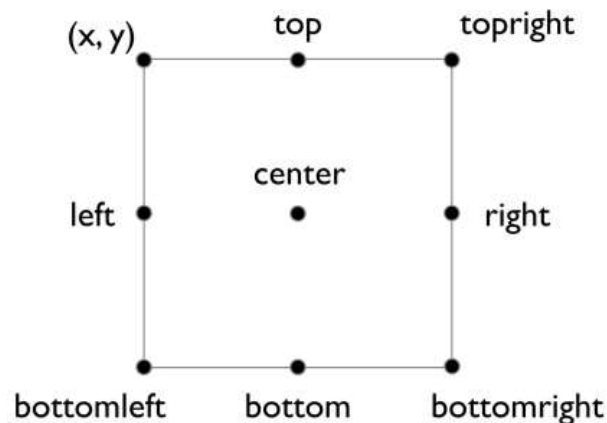
МОДУЛЬ PYGAME.EVENT

Вызов (pygame)	Описание
<code>event.get()</code>	Получает события из очереди
<code>event.wait()</code>	Ждёт одиночного события из очереди
<code>event.clear()</code>	Удалить все события из очереди
<code>event.post</code>	Поместить новое событие в очередь. Обычно в очередь помещают собственные события
<code>event.Event</code>	Создать новый объект события



ОБЪЕКТ RECT PYGAME.Rect

- Pygame использует объекты **Rect** для хранения и манипулирования прямоугольными областями. **Rect** может быть создан из комбинации значений (x, y), ширины и высоты.
- Rect также могут быть созданы из объектов python, которые уже являются **Rect**.





МЕТОДЫ ОБЪЕКТА RECT

МОДУЛЬ PYGAME.RECT

Описание

Вызов(pygame)	Описание
Rect.copy() -> Rect	Возвращает новый прямоугольник, имеющий ту же позицию и размер, что и оригинал.
Rect.move(x, y) -> Rect	Возвращает новый прямоугольник, перемещаемый данным смещением.
Rect.move_ip() -> None	Перемещение заданного прямоугольника
Rect.inflate(x, y) -> Rect	Возвращает новый прямоугольник, размер которого изменяется на заданное смещение



МЕТОДЫ ОБЪЕКТА RECT

МОДУЛЬ PYGAME.RECT

Описание

Вызов

Rect.union (Rect) -> Rect

возвращает новый прямоугольник, полностью покрывающий две предоставленные прямоугольные области

Rect.union_ip(Rect) -> None

Тот же метод, что и Rect.union(), но для заданного прямоугольника

Rect.unionall([Rect]) -> Rect

возвращает объединение прямоугольника и серии прямоугольников.

Rect.unionall_ip([Rect])->None

Тот же метод, что и Rect.unionall(), но для заданного прямоугольника



МЕТОДЫ ОБЪЕКТА RECT

МОДУЛЬ PYGAME.RECT

Описание

Вызов	Описание
Rect.collidepoint(x, y) -> bool	Проверяет, находится ли точка внутри прямоугольника
Rect.colliderect(Rect) -> bool	Проверяет, пересекаются ли два прямоугольника
Rect.collidelist([Rect]) -> bool	Проверяет, пересекается ли прямоугольник в списке. Если хоть один пересекается, вернется True
Rect.collidelistall([Rect]) -> bool	Проверяет, все ли прямоугольники в списке пересекаются. Если хоть один не пересекается вернется False



КЛАВИАТУРА

PYGAME.KEY

- **pygame.key** - это модуль содержит функции для работы клавиатурой. Очередь событий получает события **pygame.KEYDOWN()** и **pygame.KEYUP()** при нажатии и отпускании клавиш клавиатуры соответственно.
- Каждое нажатие и отпускание клавиши является игровым событием, которое мы обрабатываем
- Каждая клавиша имеет собственную клавиатурную константу. Далее приведен список основных клавиатурных констант



КЛАВИАТУРА PYGAME.KEY

Вызов	Описание
K_0, K_1, ... , K_9	1, 2, ..., 9 соответственно
K_a, K_b, ... , K_z	a, b, ... , z соответственно
K_F1, K_F2, ..., K_F9	F1, F2, ... , F9 соответственно
K_UP, K_DOWN, K_RIGHT, K_LEFT	Стрелки на клавиатуре, где: -K_UP - стрелка вверх -K_DOWN - стрелка вниз -K_RIGHT - стрелка вправо -K_LEFT - стрелка влево



ПРАКТИЧЕСКАЯ РАБОТА

MOVE_RECT

- Рассмотрим скрипт обработки события на нажатие клавиши `move_rect.py`
- Почему прямоугольник не движется в нужном направлении, если клавиша зажата?
- Попробуйте исправить этот скрипт таким образом, что персонаж будет двигаться в нужном направлении до тех пор, пока клавиша зажата



МЫШЬ

PYGAME.MOUSE

- **pygame.mouse** - это модуль содержит функции для работы мышью. Очередь событий получает события `pygame.MOUSEBUTTONDOWN` и `pygame.MOUSEBUTTONUP` при нажатии и отпускании клавиш мыши соответственно.
- Колесо мыши будет генерировать `pygame.MOUSEBUTTONDOWN` и `pygame.MOUSEBUTTONUP` события при прокрутке.



МЫШЬ

PYGAME.MOUSE

Вызов	Описание
<code>mouse.get_pressed()</code> -> bool	получить состояние кнопок мыши
<code>mouse.get_pos()</code> -> (x, y)	получить позицию курсора мыши
<code>mouse.set_cursor((width, height), (fire_x, fire_y))</code> -> None	используется для установки пользовательского курсора
<code>mouse.set_visible(bool)</code> -> None	скрыть или показать курсор мыши



МЫШЬ

PYGAME.MOUSE

Вызов	Описание
<code>mouse.set_pos((x, y)) -> none</code>	установить позицию курсора мышь
<code>mouse.get_rel() - > int</code>	получить количество движений мышь
<code>mouse.get_focused ()</code>	проверяет, принимает ли дисплей ввод мышь



МЫШЬ PYGAME.MOUSE

Рассмотрим небольшой скрипт
`draw_circle.py`

Определить какая кнопка была нажата на мышке можно используя значение **`event.button == x`**, где

- 1 - left click
- 2 - middle click
- 3 - right click
- 4 - scroll up
- 5 - scroll down



РАЗРАБОТКА СОБСТВЕННЫХ ИГР

- Рассмотрим две небольшие игры `fast_clicker.py` и `arcade.py`
- Обе игры используют объектно-ориентированный подход программирования
- При работе со спрайтами обязательно конвертируйте размер и предварительно удаляйте задний фон.

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*