

Константы:

1. MAIN\_FONT - шрифт (**consolas** размером 32)

Разметка поля:



Общий игровой класс:

1. Поля класса:

- a. `SCREEN_SIZE = SCREEN_WIDTH, SCREEN_HEIGHT` - размер экрана (800\*1000)
- b. `FIELD_SIZE = FIELD_WIDTH, FIELD_HEIGHT` - размер игрового поля (750\*750)
- c. `TICK` - время ожидания между кадрами
- d. Индексы сцен

2. Поля объекта:

- a. `surface` (из классов объектов обращаться к `self.game.screen`)
- b. Массив сцен
- c. Количество очков в текущей игре (не забыть обнулить при начале новой)
- d. Музыка и звуковые эффекты
- e. Настройки:
  - i. `ghost_speed` - скорость призрака
  - ii. `ghosts_count` - количество призраков
  - iii. `level` - номер уровня
  - iv. `mode` - номер режима
  - v. `field_texture` - текстура поля
  - vi. `coop` - True/False

3. Методы:

- a. `process_all_events()` - фиксирует события и передаёт их в метод обработки событий у сцены
- b. `process_all_logic()`
- c. `process_all_draw()` - вызывает отрисовку сцены
- d. `main_loop()` - основной игровой цикл
- e. `add_scores(n)` - добавить n очков
- f. `set_scores(n)` - установить значение очков n
- g. `set_scene(n)` - установить сцену с индексом n

## Базовые классы:

### 1. Базовый класс сцены:

#### а. Поля:

- i. game - объект игры
- ii. objects - пустой список объектов

#### б. Методы:

- i. process\_event(e), process\_logic(), process\_draw():
  - 1. Проходятся по object и вызывают у каждого объекта соответствующий метод

### 2. Базовый класс объекта:

#### а. Поля:

- i. game - объект игры
- ii. rect - квадрат отрисовки
- iii. image - текстура

#### б. Методы:

- i. process\_event(e), process\_logic() - реализуются у конкретных объектов
- ii. process\_draw() - отрисовка объекта
- iii. collision(obj) - проверка коллизии с другим объектом
- iv. die() - метод "смерти" - объект перестаёт отображаться

### 3. Базовый класс текста:

#### а. Поля:

- i. game - объект игры
- ii. text - текст
- iii. text\_color - цвет текста. По умолчанию - белый
- iv. bg\_color - цвет фона. По умолчанию чёрный
- v. Координаты текстового поля
- vi. padding - Отступ от текста до границ его фона

#### б. Методы

- i. change\_text(t) - заменяет text на t
- ii. process\_draw() - отрисовка текста

iii. `process_event(e), process_logic()` -  
pass. нужно для простого вызова методов  
в сценах

4. Базовый класс кнопки (наследуется от базового  
класса текста с определением метода  
`process_event(e)`)

## Сцены:

1. Меню
  - a. Кнопка запуска
  - b. Кнопка таблицы рекордов
  - c. Кнопка выхода
  - d. Кнопка настройки
2. Меню настроек
  - a. Выбор уровня:
    - i. Сэтап локации (расположение стен)
  - b. Выбор режима:
    - i. Выживание - побеждаем при собрании всех зёрен
    - ii. Набор очков - победа при достижении определенного количества очков
    - iii. Уничтожение - убиваем всех привидений
  - c. Кооператив
    - i. True - появляются 2 Пакмена: зелёный и жёлтый, один управляется на WASD, другой на стрелки
  - d. Текстуры пакмена
  - e. Текстуры локации
  - f. Скорость призраков
  - g. Количество призраков
3. Таблица рекордов
  - a. Загружает первые N записей из файла `highscores.txt`
4. Уровень игры
  - a. Поле:
    - i. Параметр отступа (для поля 750\*750 он равен 25)
    - ii. Собираем все объекты в уровень
    - iii. Массив, содержащий условные координаты стен и зёрен
    - iv. Размер поля - 750\*750 (меняется в константах)
    - v. Размер условной клетки 30\*30 (т.е. при стандартных параметрах получается 25\*25 условных клеток) (При увеличении поля

размер клетки изменится пропорционально,  
или увеличится количество клеток, например  
для уровней большего размера)

vi. Следим за условиями победы/поражения

b. Объекты:

i. Пакмен (наследуется от базового класса  
объекта):

1. Движение по WASD. 2-й игрок - стрелки,  
можно добавить переменную номера игрока и  
проверять по ней
  - a. Движение при одиночном нажатии клавиши
  - b. Плавное
  - c. Поворот при возможности (окно небольшое)

2. Коллизии:

- a. Со стенами - останавливается
- b. С призраками
  - i. В обычном режиме - умирает Пакмен  
(`self.die()`)
  - ii. В режиме страха - умирает  
призрак(`ghost.die()`), начисляются  
очки (`self.game.add_scores(n)`)

3. Текстуры (4 фрейма) (скин)

ii. Призрак (наследуется от базового класса  
объекта):

1. Алгоритмы движения:

- a. Случайный
- b. Двигается напрямую к пакмену
- c. Двигается на несколько клеток впереди  
пакмена

2. Текстуры (4 фрейма) (скин)

3. status - состояния призрака (реакция  
реализуется в классе Пакмена)

- a. 0 - обычное
- b. 1 - страх

4. Метод die()

- a. В режиме уничтожения пропадает навсегда,  
а в других - воскрешается через  
константное время

- iii. Стена (наследуется от базового класса объекта)
- iv. Зерно (наследуется от базового класса объекта)
  - 1. Добавляет N очков при контакте с Пакменом (`self.game.add_scores(n)`)
- v. Большое зерно (наследуется от базового класса объекта)
  - 1. Переводит призраков в режим страха
  - 2. Добавляет  $10 \cdot N$  очков (`self.game.add_scores(n)`)
- vi. Телепорт (наследуется от базового класса объекта)
  - 1. Телепортирует из заданной точки  $x_1, y_2$  в заданную точку  $x_2, y_2$  с сохранением вектора движения
- vii. Кнопка меню (паузы)
  - 1. Пауза при нажатии 'p'
- viii. Текст
  - 1. Время
  - 2. Очки
  - 3. Жизни (можно значками либо текстом)
- ix. Элементы декора (неигровые текстуры)
  - 1. Рамки
  - 2. Жизни (если значками)
  - 3. Дополнительные украшения
    - a. Фон поля
    - b. Фон боковой панели
- 5. Меню окончания игры
  - a. Поле ввода никнейма
  - b. Вывод сообщения 'FAIL' или 'WIN'
  - c. Кнопка возврата в главное меню
  - d. Кнопка завершения игры
  - e. Таблица рекордов