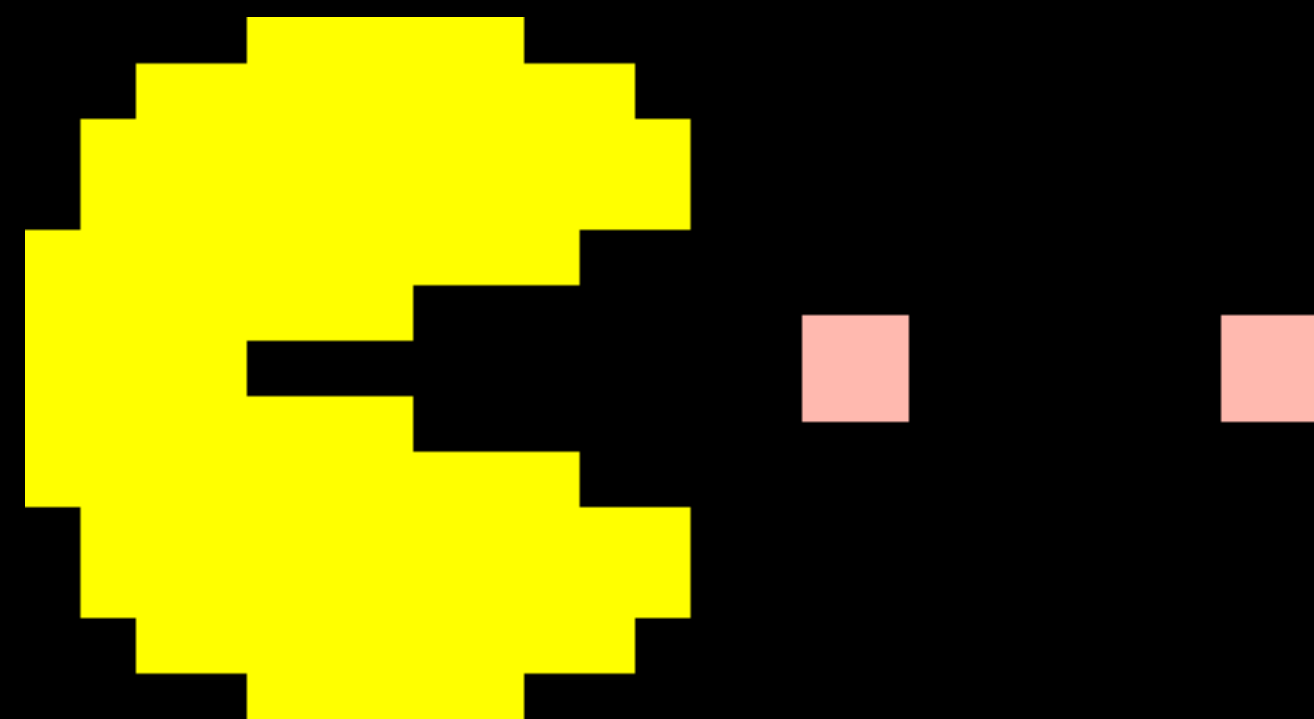


Разработка игры Расстан на Python



Цель:

Создать работающую игру

Задачи:

- Разработать графику
- Реализовать логику игры
- Выпустить игру в свет



Все py-файлы и для чего нужны

- `ghosts.py` — отвечает за поведение призраков
- `pacman.py` — отвечает за поведение пакмена
- `load_data.py` — отвечает за подгрузку данных
- `logic.py` — отвечает за логику игры
- `windows.py` — отвечает за окна, которые видит пользователь
- `main.py` — главный файл программы

pacman.py

Реализует единственный класс: Pacman,
отвечающий за реакцию пакмена на ваши действия



```
import pygame

from logic import Entity
from load_data import pacman
from logic import UP, DOWN, LEFT, RIGHT

class Pacman(Entity):
    # Pacman class code
```

ghosts.py

Класс Ghost — поведение призраков в целом

Классы Blinky, Pinky, Inky и Clyde реализуют поведение одноименных призраков

```
from load_data import blinky, pinky, inky, clyde
from logic import Entity, UP, DOWN, LEFT, RIGHT
from random import randint
import pygame
```

```
HOUSE = 0
CHASE = 1
SCATTER = 2
FRIGHTENED = 3
```

```
class Ghost(Entity):
    # Ghost class code
```

```
class Blinky(Ghost):
    # Blinky class code
```

```
class Pinky(Ghost):
    # Pinky class code
```

```
class Inky(Ghost):
    # Inky class code
```

```
class Clyde(Ghost):
    # Clyde class code
```



logic.py

Классы Matrix (матрицы)
и Field (поля) - нужны для
удобства программиста

Класс Entity - отвечает за
корректное
использование карты
существами игры

```
import pygame.draw
from load_data import items
```

```
LEFT = 0
RIGHT = 1
UP = 2
DOWN = 3
WALL = -1
VERTICAL = UP, DOWN
HORIZONTAL = LEFT, RIGHT
```

```
class Matrix(object):
    # Matrix class code
```

```
class Entity(object):
    # Entity class code
```

```
class Field(Matrix):
    # Field class code
```



load_data.py

Класс Sprite - объект
статического спрайта
Класс AnimatedSprite -
объектная реализация
анимированных спрайтов
Функция load_image отвечает
за подгрузку изображений
Функция screen_print отвечает
за текст в игре

```
import pygame
import os

# Colors
empty = 0, 0, 0, 0
black = 0, 0, 0, 255
white = 255, 255, 255, 255
yellow = 255, 255, 0, 255
red = 255, 0, 0, 255
green = 0, 255, 0, 255
blue = 0, 0, 255, 255

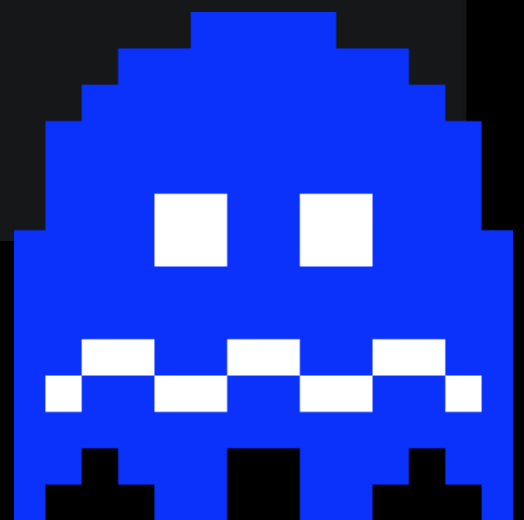
class Sprite(pygame.sprite.Sprite):
    # Sprite class code

class AnimatedSprite(pygame.sprite.Sprite):
    # AnimatedSprite class code

def load_image(image: str) -> pygame.image:
    # load_image function code

def screen_print(screen: pygame.Surface, string: str,
                  position: tuple, color: tuple = None) -> None:
    # screen_print function code

# Data load here
```



windows.py

Все функции

- main_menu (главное меню)
- start_game (меню ввода ника игрока)
- settings (меню настроек)
- lider_board (меню лидеров)

PAC-MAN



START GAME
SETTINGS
LIDER BUTTON
EXIT

Превью main_menu

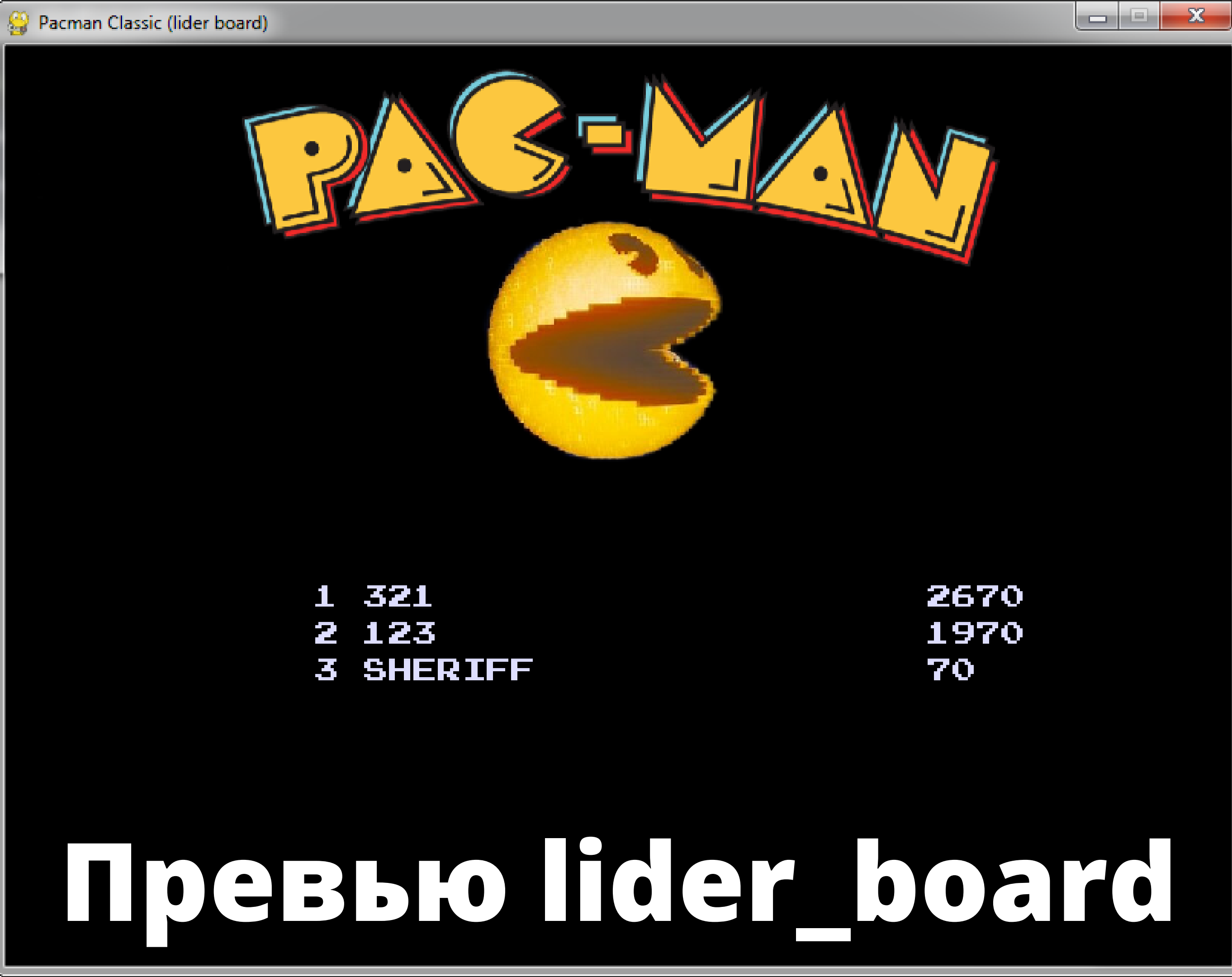


PAC-MAN

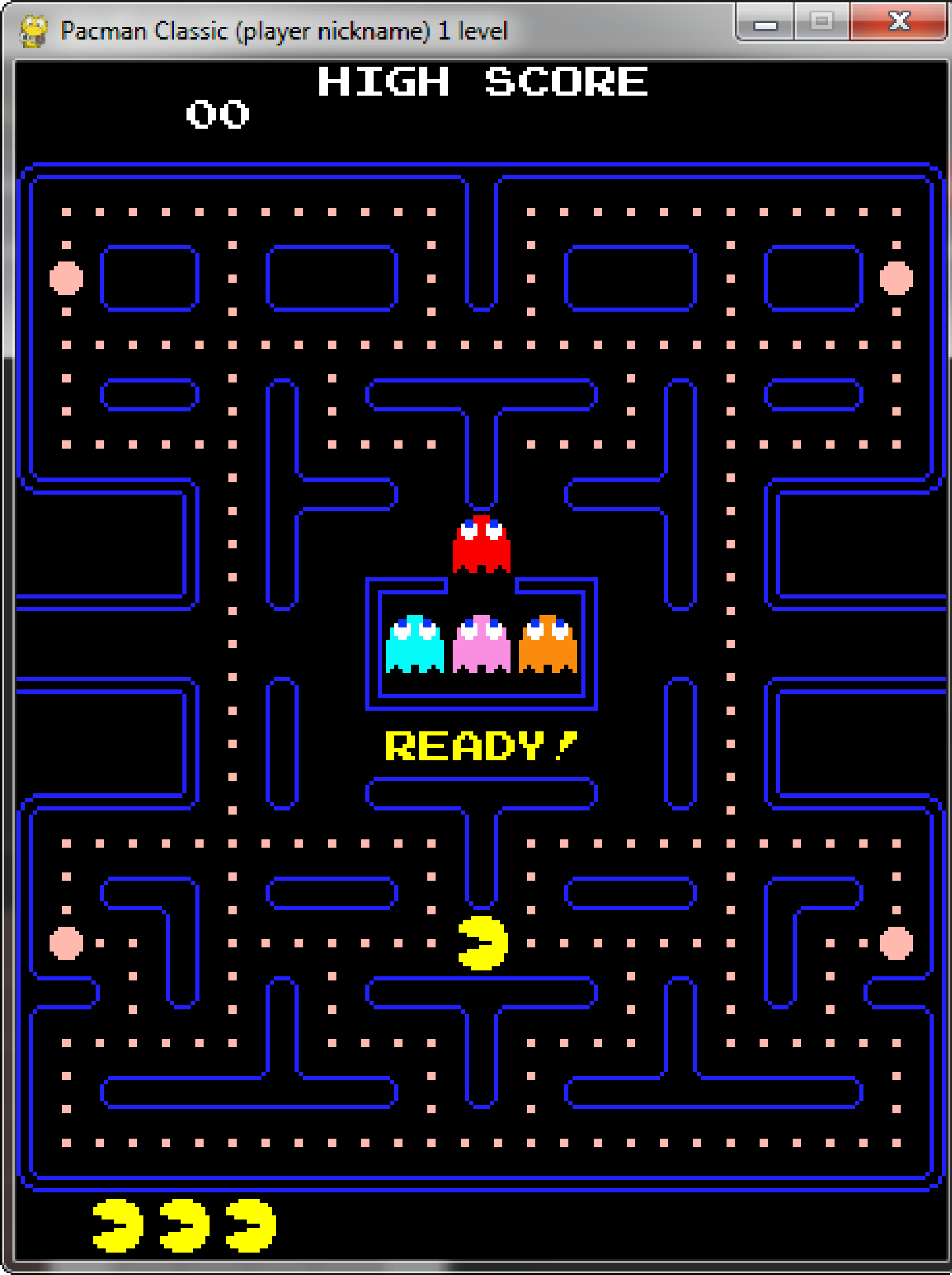


THERE ARE NO SETTINGS HERE YET

Превью settings



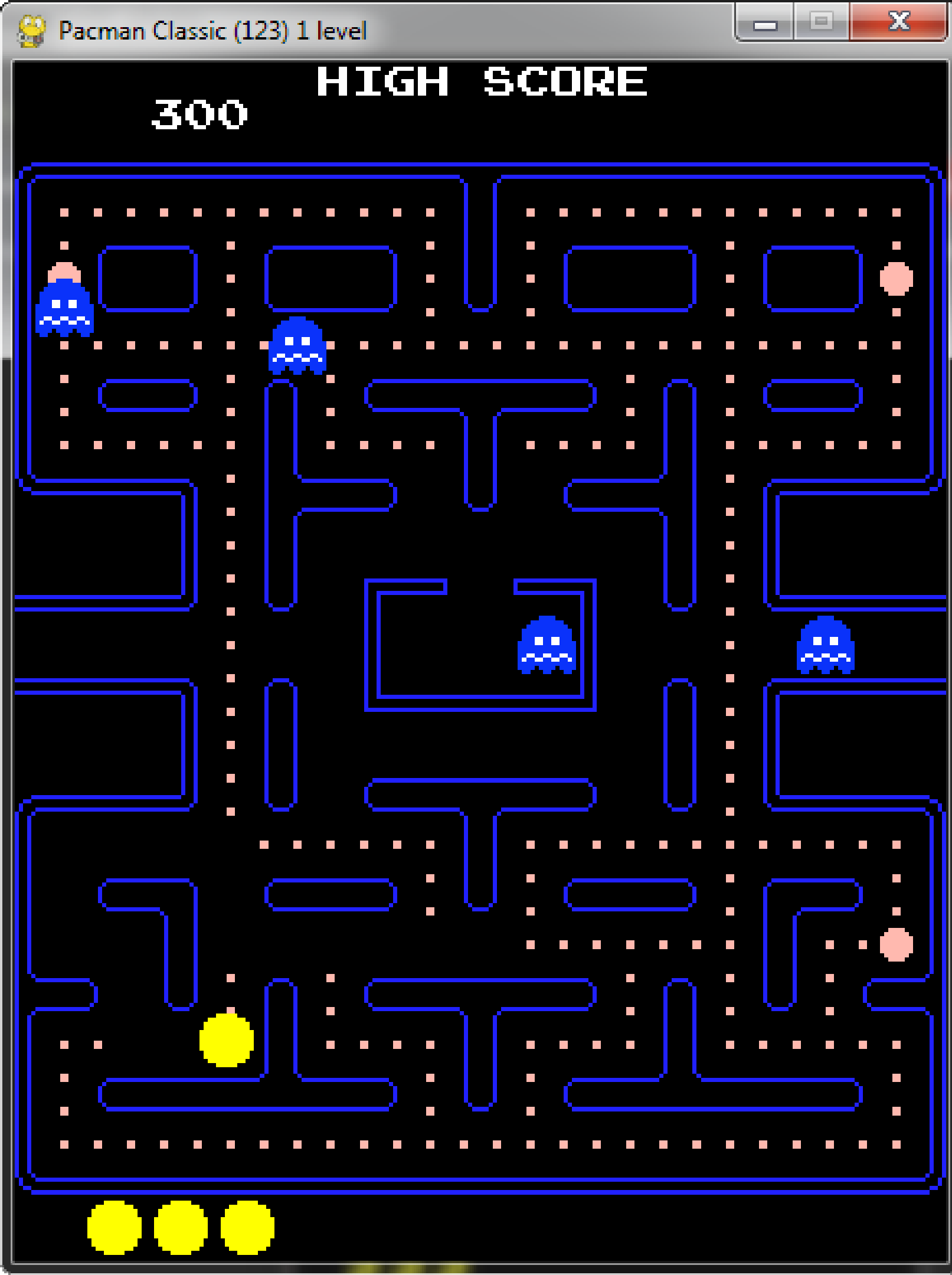
Превью lider_board



Начало игрового процесса

Здесь можно увидеть начальное расположение пакмена, призраков, точек и энерджайзеров. Сама игра начнётся через 3 секунды после появления этого окна.

В заголовке видна информация о названии игры (Pacman Classic), нике игрока (player nickname) и уровне (1 level).



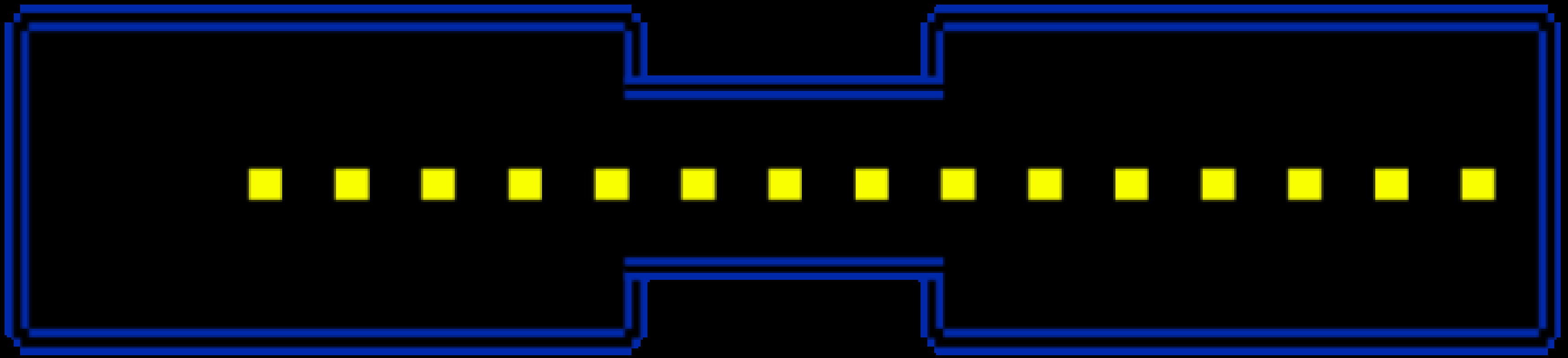
Сам игровой процесс

Здесь видно сколько и каких точек съел пакмен, а также, испуг привидений, ведь игрок заполучил энерджайзер и в течение 10 секунд может их съесть.

Задача пакмена съесть все точки, после чего он перейдёт на следующий уровень. После трёх смертей пакмен начинает с самого начала (с первого уровня).

main.py

Этот файл объединяет весь код



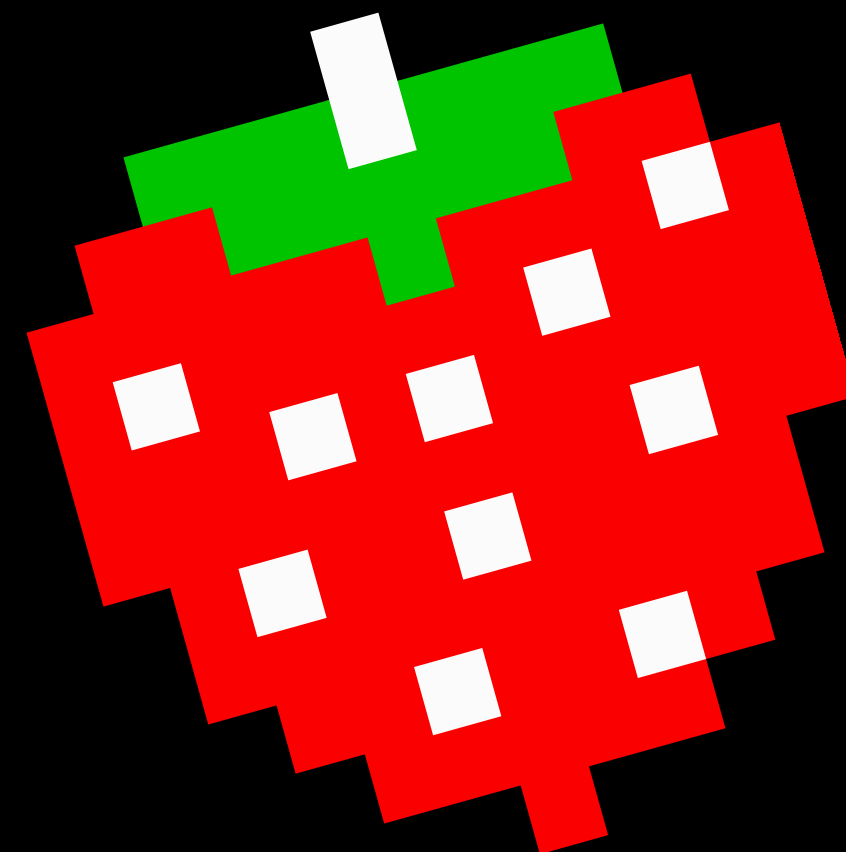
Мы решили не показывать вам этот код, чтобы не нагружать. 

Выводы



В связи со скорым дедлайном и сложностью реализации игры (из-за недопонимания принципов её работы) она была сделана "на скорую руку", поэтому часть задуманного функционала отсутствует.





Спасибо за внимание!