

Отчет о проведении нагрузочного тестирования и стресс-тестирования

1.Резюме.

Нагрузочное тестирование игры «Tower Defence» показало, что программа работоспособна в условиях повышенной нагрузки и ее работа удовлетворяет требованиям к производительности системы, основанным на разумных предположениях и потреблении аппаратной части ресурсов.

Стресс-тестирование игры «Tower Defence» показало, что приложение и система в целом работоспособны в условиях стресса, а также программа способна к возвращению к нормальному состоянию после прекращения воздействия стресса.

Результатом проведения тестирования является отчет, в котором изложены методики тестирования и обнаруженные недостатки.

2. Обнаруженные недостатки

В результате тестирования не было выявлено проблем в устойчивости программы, которые требовали бы внесения существенных изменений.

3. Методики тестирования

В связи с тем, что приложение не предусматривает взаимодействие более чем с одним пользователем, применение стандартных методик и ПО для нагрузочного и стресс тестирования оказалось трудным. Поэтому проведенное тестирование было основано на наблюдении поведения программы и на исследовании FPS в зависимости от нагрузки. Тестирование состояло из нескольких этапов:

- Попытки протестировать вручную
- Попытки автоматизации процесса (Построение класса FPSCounter)
- Проведение тестирования с использованием данного класса
- Подведение итогов

4. Этапы тестирования и наблюдения

(В норме FPS == 62)

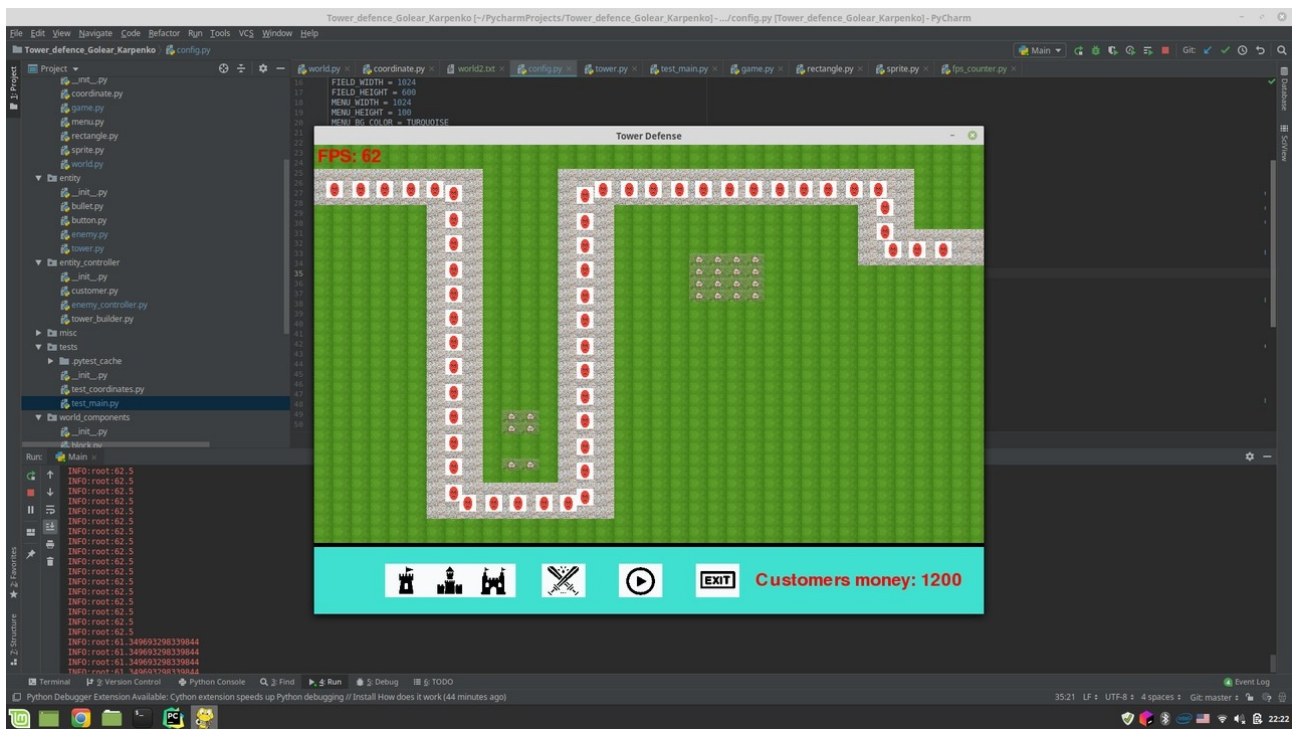
№	Тест	Наблюдение / результат	Изменение FPS	Пройден/ не пройден
1	Увеличение количества противников до 100	Программа работает без изменений	нет	Пройден (приложение 1)
2	Увеличение количества противников до 500, уменьшение скорости их движения до 1	Программа работает без изменений	нет	Пройден (приложение 2)

3	Исследование взаимодействия противников с башнями. Условия такие же, как в тесте 3, + увеличение количества башен до 50	Программа начинает слегка подвисать, что находится в пределах нормы, так как каждая башня должна следить за каждым противником, что отражается на производительности. По завершении теста показатели работоспособности вернулись в норму.	Падение до 50 - 55	Пройден (приложение 3)
4	Автоматизация расстановки башен. Все поле заставлено башнями, количество противников – 200, скорость движения противников – 1	Программа работает намного медленней, противники движутся очень медленно. По завершении теста приложение продолжило работать без видимых изменений	Падение до 15 - 20	Пройден (приложение 4)

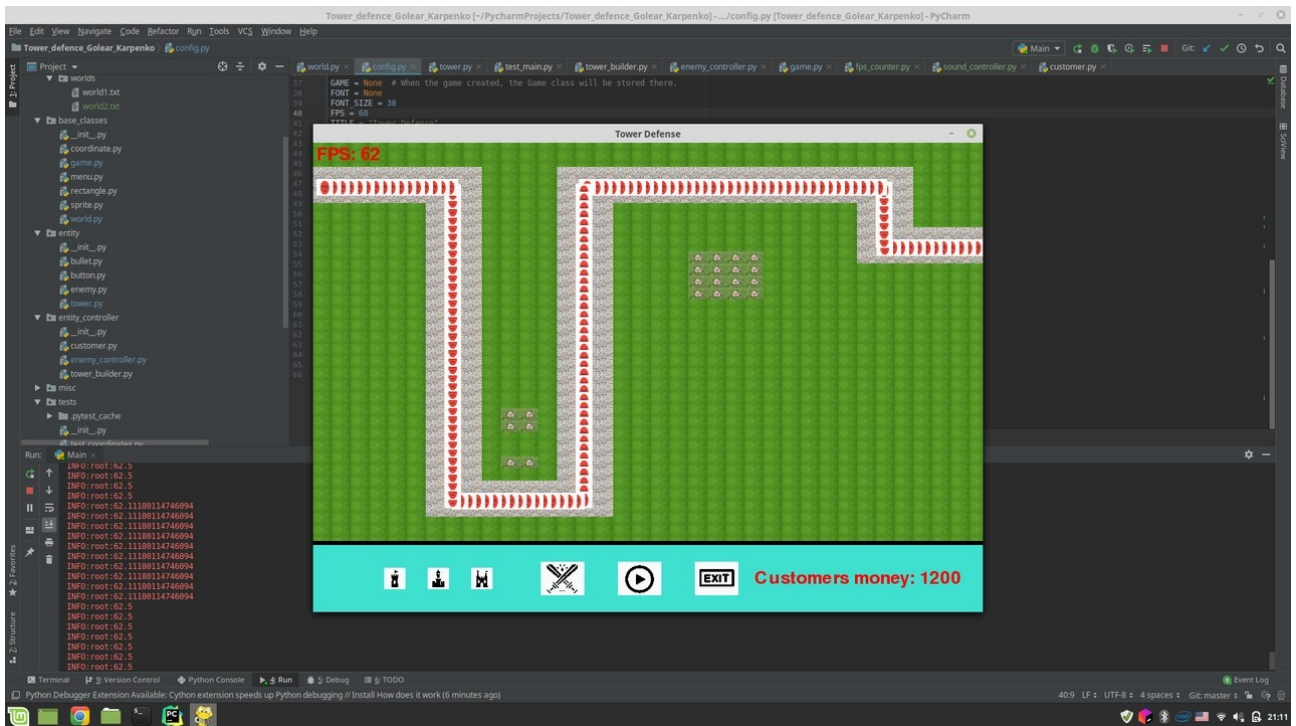
В принципе на данном этапе разработки приложение не предусматривает необходимость введения больше 50 противников и 20 башен, и провисание показателя FPS в последних двух тестах было ожидаемо (т.к используемый алгоритм для поиска противников башнями работает за $O(N * M)$, где N — количество башен, M — количество противников, + применяются дорогостоящие по времени операции, такие как взятие корня и т.д).

Поэтому можно считать, что программа успешно прошла нагрузочное и стресс тестирования.

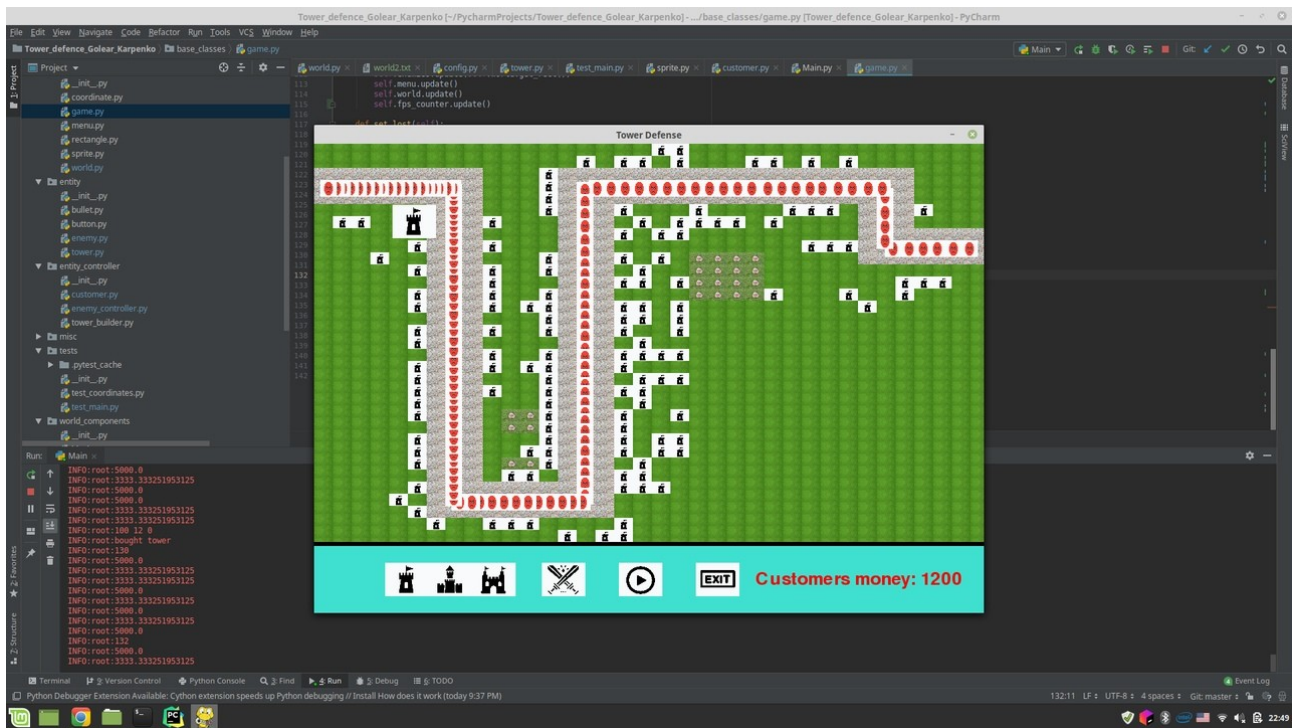
Приложение №1. Тест №1.



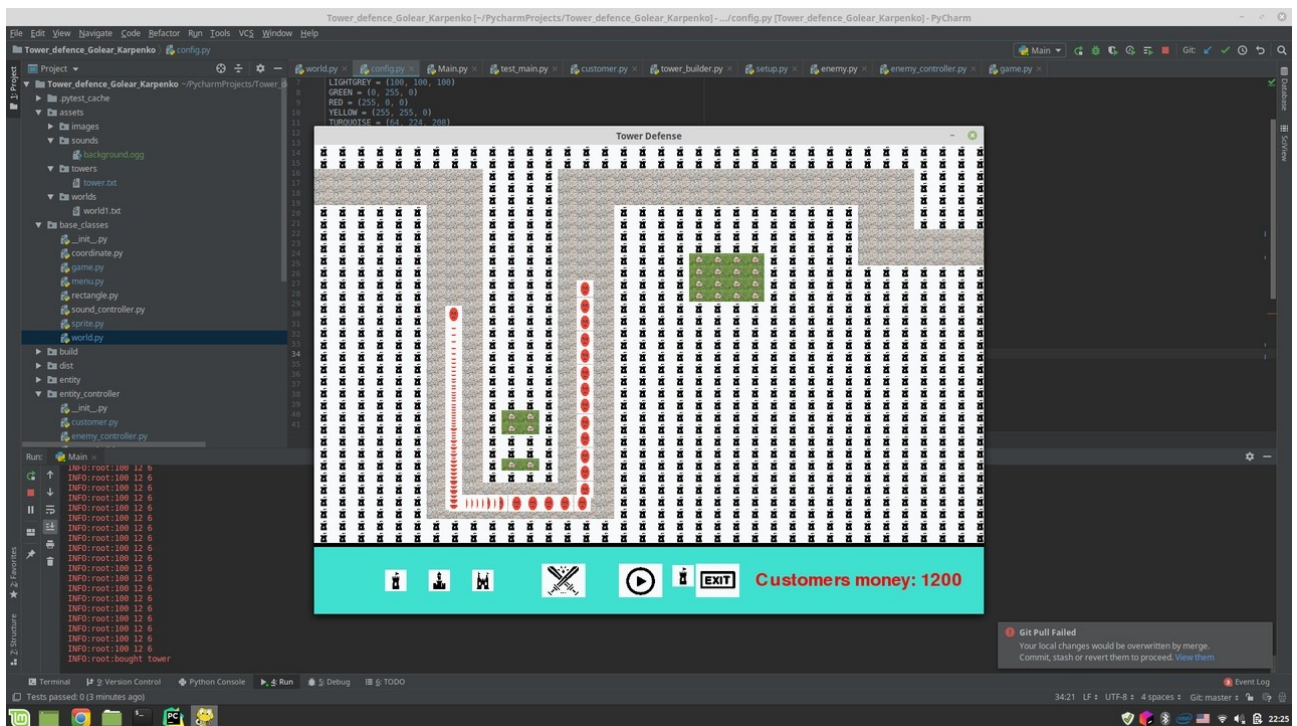
Приложение №2. Тест №2.



Приложение №3. Тест №3.



Приложение №4. Тест №4.



Приложение №5. Автоматизация тестирования.

```
game.py (/home/dgolear/PycharmProjects/Tower_defence_Golear_Karpenko/base_classes)
```

```
6da171bcdbbbbaee46145ca1848f052a3005
```

```
Local
```

```
1 difference
```

```
def check_mouse_pressed(self, pos):  
    """If player pressed left mouse button, checks if he pressed on a cell.  
    :param pos: position of mouse click.  
    """  
    if (self.world.get_rect().collidepoint(pos)  
        and self.customer.item.attached()):  
        for cell in self.tiles.sprites():  
            if (cell.get_rect().collidepoint(pos)  
                and cell.can_build  
                and self.customer.enough_money()):  
                tower = self.customer.buy_tower()  
                self.world.place_tower(tower, self.world.get_cell_position(  
                    rect=cell.get_rect()))  
                logging.info(f'bought tower: {tower}')  
            elif self.menu.get_rect().collidepoint(pos):  
                for item in self.menu.get_items():  
                    if item.get_rect().collidepoint(pos):  
                        item.action(pos)  
def update(self):  
    self.enemies.update(self.world.get_rect())  
    self.menu.update()  
    self.world.update()  
def set_lost(self):  
    self.enemies.clear()  
    self.menu.set_lost()  
    self.customer.money = 0  
    self.game_started = False  
def draw(self):  
    self.all_sprites.draw(self.field)  
    self.enemies.draw(self.field)  
    self.menu.draw(self.screen)  
    if self.customer.item.attached():  
        pos = Coordinate(pygame.mouse.get_pos())  
        pos = (pos - self.customer.item.get_size() // 2)  
        self.screen.blit(self.customer.item.get_image(), pos)  
def main_loop(self):  
    while True:  
        self.clock.tick(self.FPS)  
        self.handle_events()  
        self.draw()  
        self.world.get_cell_position(  
            rect=cell.get_rect())  
        logging.info(f'bought tower: {tower}')  
        menu.get_rect().collidepoint(pos):  
        # in self.menu.get_items():  
        item.get_rect().collidepoint(pos):  
        item.action(pos)  
        self.world.place_tower(tower, self.world.get_cell_position(  
            rect=cell.get_rect()))  
        logging.info(f'bought tower: {tower}')  
        menu.get_rect().collidepoint(pos):  
        # in self.menu.get_items():  
        item.get_rect().collidepoint(pos):  
        item.action(pos)  
        self.world.update()  
        self.draw()  
        self.draw(self.screen)  
        tower.item.attached():  
        coordinate(pygame.mouse.get_pos())  
        pos = self.customer.item.get_size() // 2  
        screen.blit(self.customer.item.get_image(), pos)  
        self:  
        clock.tick(self.FPS)
```

Process finished with exit code 0

Git Pull Failed
Your local changes would be overwritten by merge.
Commit, stash or revert them to proceed. [View diff](#)

96:1 LF : UTF-8 : 4 spaces : Git: master : 22:23