Экранные формы

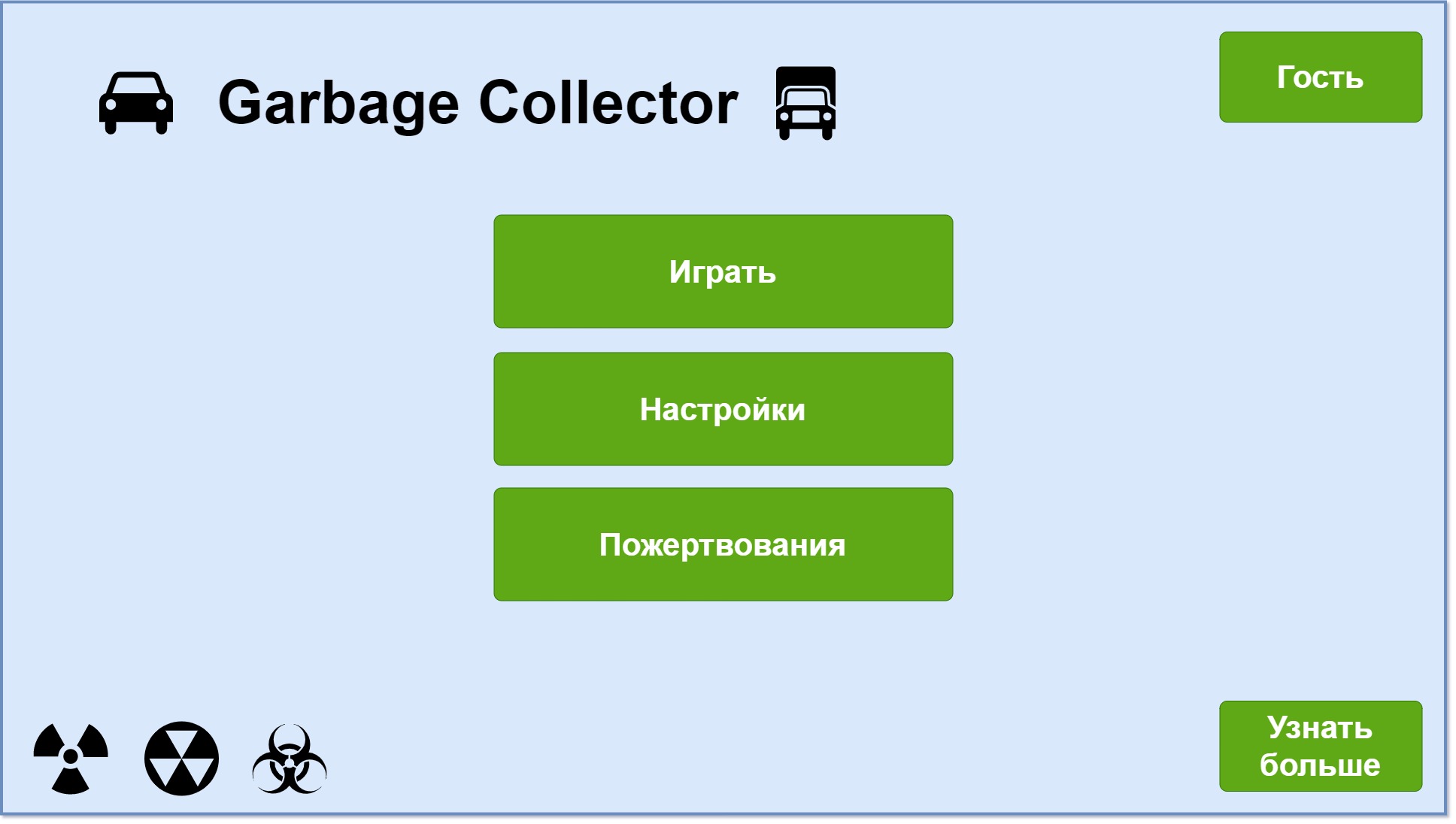


Рисунок 1. Главная форма

Играть – начать игру.

Настройки – настройки звука, уведомлений и рассылки.

Пожертвования – выводит список организаций, которым можно помочь финансовой поддержкой.

Гость – кнопка личного кабинета, когда не выполнен вход.

Узнать больше – ведет на образовательные ресурсы, которые связаны с экологическими проблемами и организациями.

Логотип команды разработчиков (левый нижний угол) ведет на информацию о команде разработчиков.



Рисунок 2. Форма настроек

Позволяет отрегулировать настройки звука слайдерами, и кнопки радио для включения – отключения уведомлений и рассылки.

Кнопку назад, здесь и далее возвращает на главную форму.

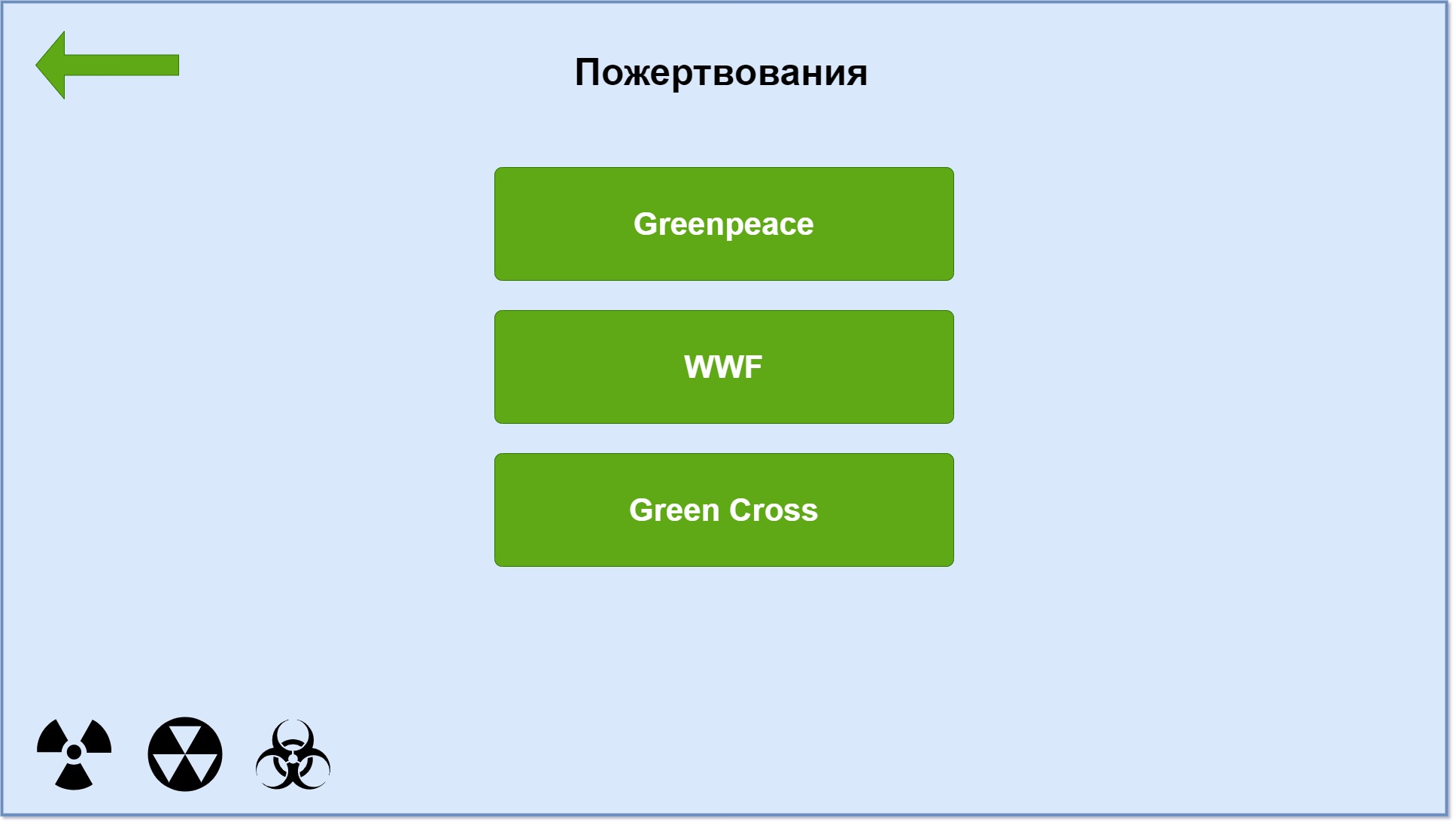


Рисунок 3. Форма пожертвований

Кнопки ведут на сайты перечисленных организаций в раздел Donate.

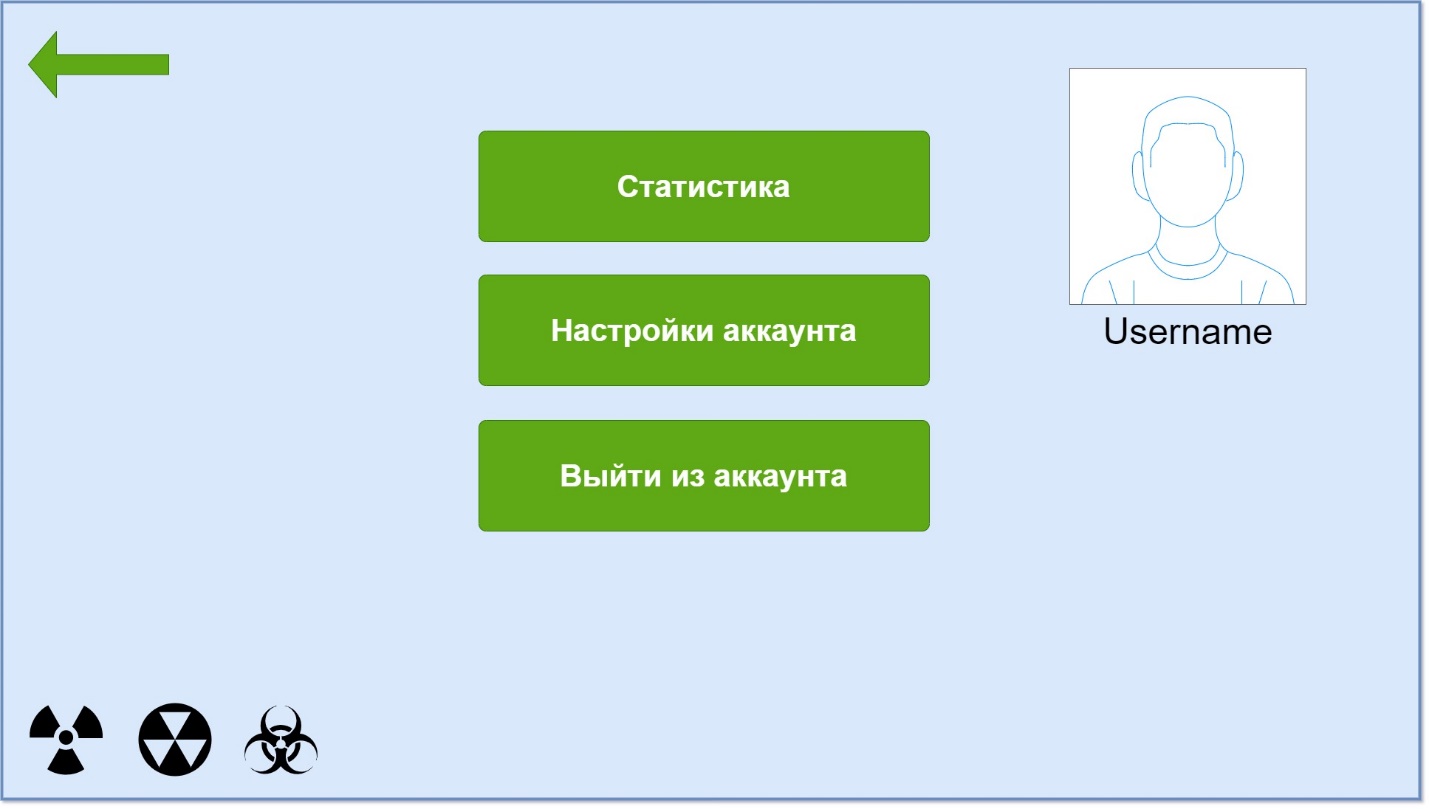


Рисунок 4. Форма личного кабинета игрока

Статистика – отображение статистики игрока за все время.

Настройки – возможность сменить данные для входа.

Выйти из аккаунта – выход из аккаунта.

При клике на аватар есть возможность сменить его.

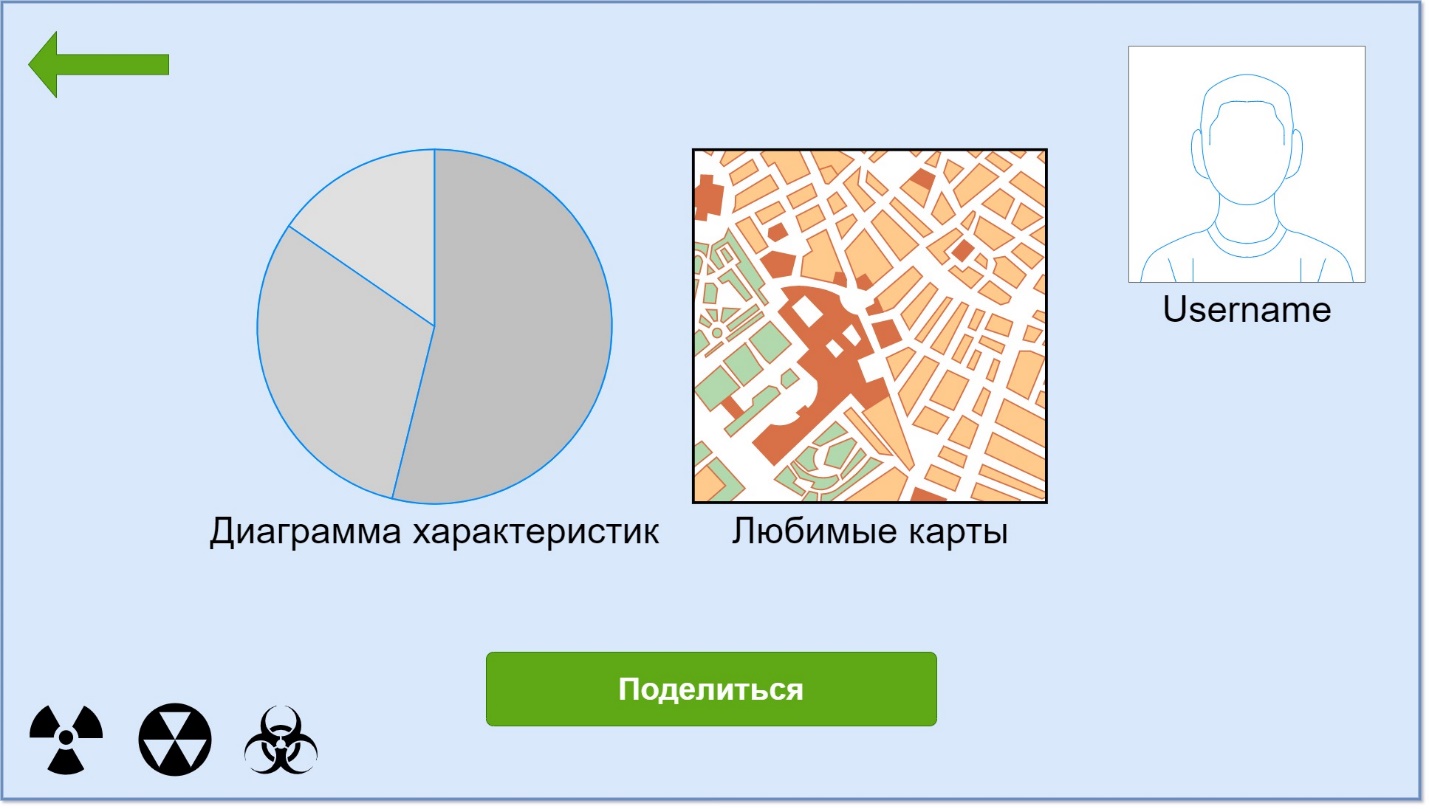


Рисунок 5. Форма статистики

В статистике содержится информация по характеристикам игрока в виде диаграммы. А также любимые карты игрока отображенные на мини карте.

Кнопка поделиться позволяет поделиться статистикой в социальных сетях и мессенджерах.



Рисунок 6. Форма загрузки игры

Форма загрузки игры, отображается при любых действиях требующих ожидания пользователя.

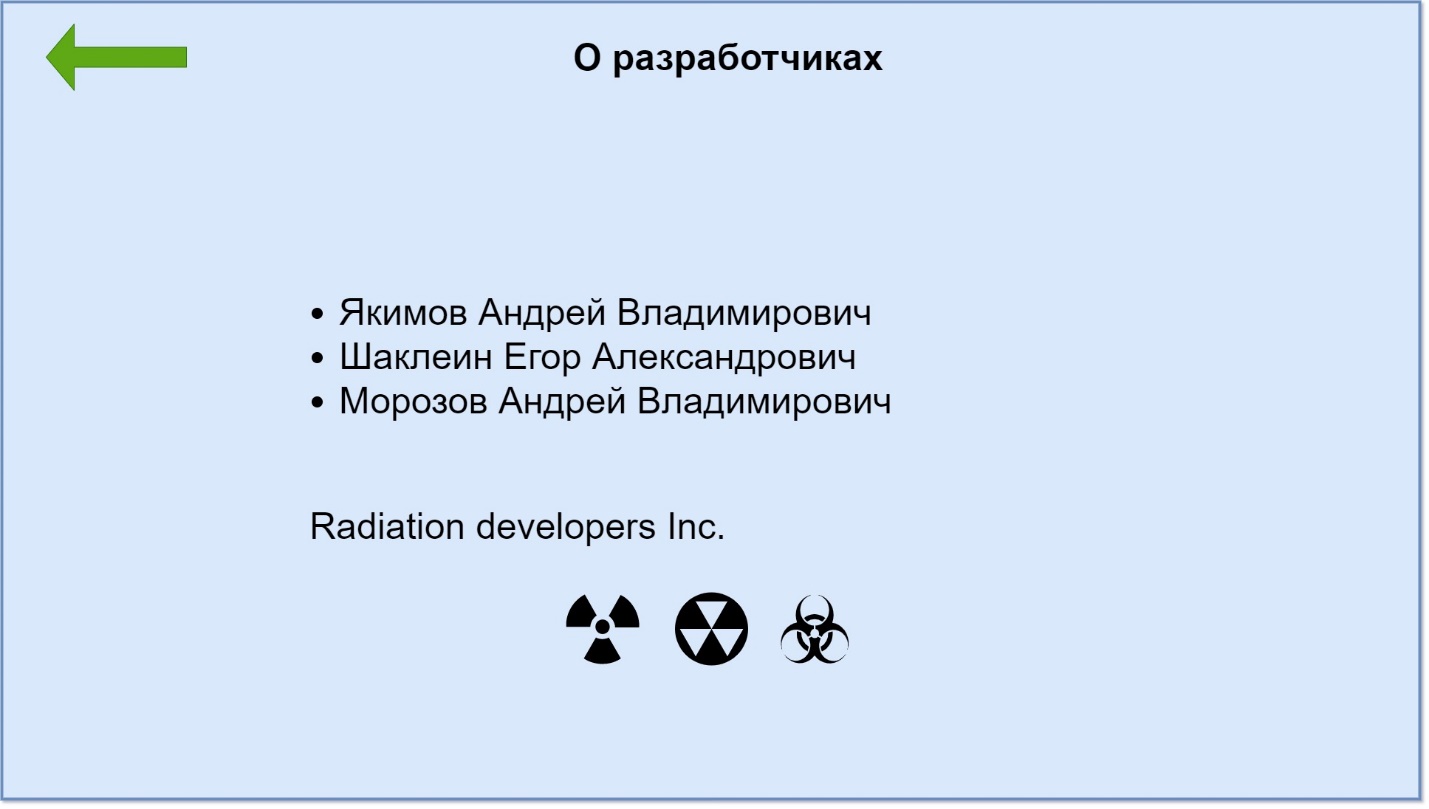


Рисунок 7. Форма «О нас»

Содержит актуальную информацию о разработчиках игры.



Рисунок 8. Форма узнать больше

Содержит актуальные ссылки на образовательные и информационные ресурсы экологических организаций.

ER диаграмма

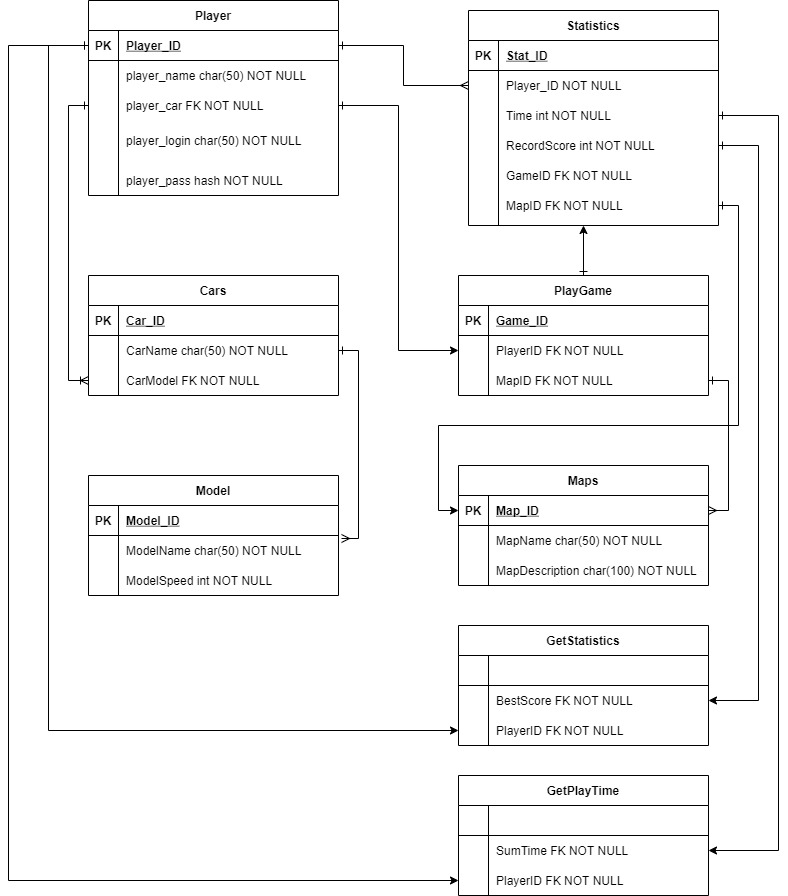


Рисунок 9. Диаграмма сущностей

API

Void Login(username, pass)

\*Авторизация пользователя

> логин и пароль

< авторизация

Void Register(PlayerID attributes)

\*Регистрация пользователя

> атрибуты, необходимые для создания пользователя

< регистрация

Player GetPlayer(PlayerID)

\*Получение игрока и его атрибутов

> id игрока

< атрибуты игрока

Statistics GetStat(PlayerID / GameID)

\*Получение статистики игрока

> id игрока или id игры

< стастика

List<Car> GetCars(PlayerID)

\*Получение списка автомобилей игрока

> id игрока

< список автомобилей

List<Map> GetMaps()

\*Получение списка карт

> не принимает аргументов

< возвращает список карт

GameID StartGame(PlayerID, MapID)

\*Начать игру

> принимает игрока и карту

< возвращает ID начатой игры

GameID EndGame(PlayerID, MapID)

\*Закончить игру

> принимает игрока и карту

< возвращает ID завершенной игры

List<Statistics> GetFullStat()

\*Получение статистики для отображения рекордов за все время

> не принимает аргументов

< возвращает список со статистикой за все время

Bool IsGameInProgress(GameID)

\*Получение состояния игры, true если игра не завершена (в прогрессе), false если рга завершена, null если игры не существует

> принимает GameID

< возвращает true, false или null (bool?)

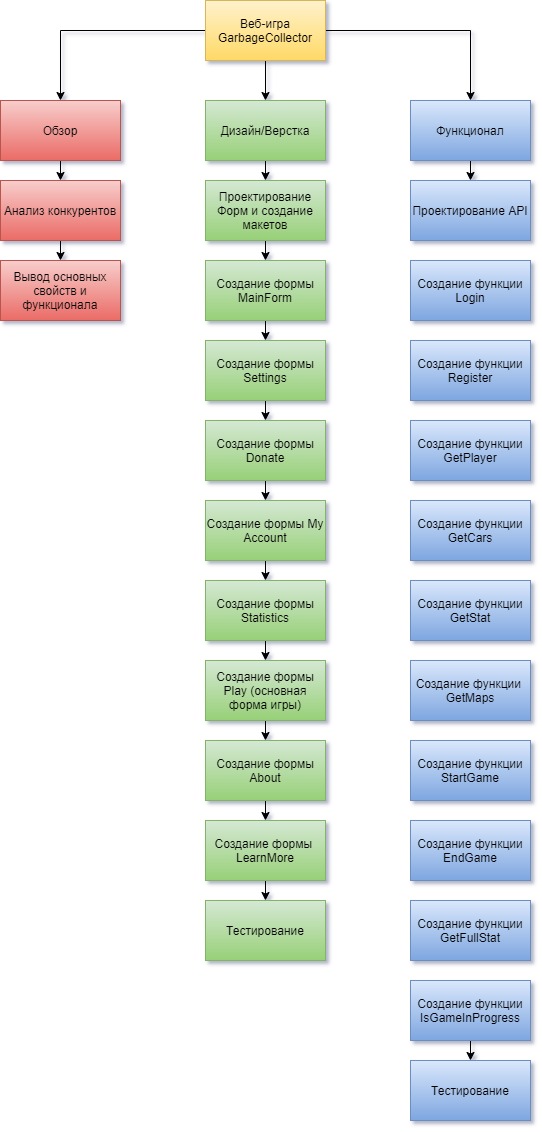
Иерархическая структура работ

Рисунок 10. ИСР

Время выполнения проекта по PERT

Количество сущностей: 8. Количество форм: 8. Количество методов api: 10.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Затраты | Сущность | Форма | Метод API | Проектирование | Тестирование |
| Пессимистичные | 5 ч | 20 ч | 3 ч | 7 ч | 10 ч |
| Оптимистичные | 2 ч | 4 ч | 1 ч | 3 ч | 5 ч |
| Оптимальные | 3 ч | 10 ч | 2 ч | 6 ч | 8 ч |

Ei=(Pi+4Mi+Oi)/6

Eс=(5+4\*3+2)/6=19/6=3,1 ч

Eф=(20+4\*10+4)/6=64/6=10,6 ч

Eм=(3+4\*2+1)/6=12/6=2 ч

Eп=(7+4\*6+3)/6=34/6=5,6 ч

Eт=(10+4\*8+5)/6=44/6=7,3 ч

E=8\*3,1+8\*10,6+10\*2+5,6+7,3=142,5 ч

CKOi=(Pi-Oi)/6

CKOc=(5-2)/6=0,5

CKOф=(20-4)/6=2,6

CKOм=(3-1)/6=0,33

CKOп=(7-3)/6=0,66

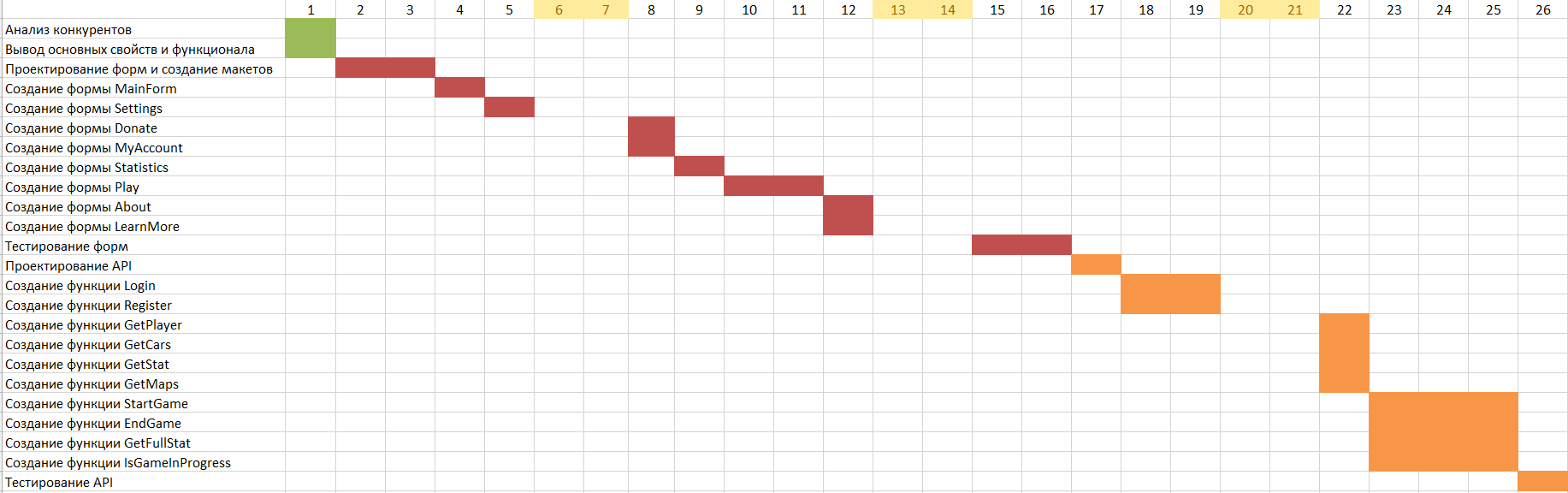
CKOт=(10-5)/6=0,83

CKO==7,63 ч

E95%=(142,5 +7,63\*2)\*4 =631,04 ч

Сотрудник тратит 80% рабочего времени в месяц 165 \* 0.8 = 132

Трудоёмкость проекта в чел \* мес: 631,04 / 132 = 4.78

Базовое расписание (диаграмма Ганта)

