

ТЕСТОВОЕ ЗАДАНИЕ



ОЛЕГ САВЛ

savaal@mail.ua

vk.com/433986090

Доброе время суток, дорогой друг. Твоему вниманию представлено решение задач для статьи на app2top.ru.

*Данная презентация позволила мне получить предложение на позицию «Младший Игровой Дизайнер». Задания 4-5-6 выполнены на языке *Java* и *C++*;*

СПИСОК ЗАДАНИЙ

Тестовое задание на позицию Младший Игровой Дизайнер (Junior Game Designer).

ЗАДАНИЕ 1

Выберите наиболее удачную, на ваш взгляд, free-to-play Farm games, Strategy (клиентскую, браузерную или социальную) и опишите кратко ее core loop, основные используемые механики (Core loop, PvE, PvP, Loot, Craft и т.д.) и методы монетизации.

ЗАДАНИЕ 2

Выберите любую free-to-play игру и предложите три основных улучшения геймплея, монетизации или интерфейса.

ЗАДАНИЕ 3

Напишите спецификацию крафта в Minecraft. Опишите систему ресурсов, предметов и их крафт. Формат: спецификация для разработчика, который будет имплементировать данную механику.

ЗАДАНИЕ 4 (java, c++);

Задача. Из волшебного сундука с шансом 27% выпадает редкий предмет. У игрока есть три сундука. Каковы шансы найти 0-1-2-3 редкие вещи в этих сундуках? Опишите ваши рассуждения.

ЗАДАНИЕ 5 (java, c++);

Задача. 13% игроков покупают Премиум Аккаунт во free-to-play игре. 55% купивших Премиум Аккаунт в начале игры купили Набор Новичка. Также, 9% не купивших Премиум Аккаунт покупали Набор Новичка. Какова вероятность того, что только что пришедший игрок купит Премиум Аккаунт, если он только что купил Набор Новичка?

ЗАДАНИЕ 6 (java, c++);

Задача. В мобильной mid-core игре есть юнит, который имеет базовый урон 35 ед. за удар. Выдаем юниту оружие, которое дает +12% урона, и второе оружие, которое дает +17% урона. Также на юнита наложен бафф, который дает +6% урона. Предположим, у этого юнита с последней битвы есть травма, которая дает -17% урона. Вопрос, сколько суммарно урона нанесет отряд из 40 таких юнитов 1 ударом? Опишите ваши рассуждения и объясните, почему вы выбрали именно этот вариант расчета урона.

ЗАДАНИЕ 7

Придумайте легендарный предмет, оружие или персонажа для гипотетической ММО игры в сеттинге sci-fi: опишите его характеристики и особые возможности, напишите текстовое описание (с которым он будет преподноситься в игре) и подберите референс внешнего вида.

ФОРМА ПОДАЧИ ТЗ:

Презентация PowerPoint



ЗАДАНИЕ 1 – ПРАВИЛА ВОЙНЫ

Выберите наиболее удачную, на ваш взгляд, free-to-play Farm games, Strategy (клиентскую, браузерную или социальную) и опишите кратко ее core loop, основные используемые механики (Core loop, PvE, PvP, Loot, Craft и т.д.) и методы монетизации.

Краткий **Core Loop** игры **Правила Войны**. Каждый из пунктов имеет своё подмножество...

1. **Постройка базы**

- 1.1 **Исследования**
- 1.2 **Дипломатия**
- 1.3 **Торговля**
- 1.4 **Плацдарм**

2. **Добыча ресурсов**

- 2.1 **Выполнение заданий**
- 2.2 **Контроль объектов**
- 2.3 **Торговля**
- 2.4 **Пожертвования**

3. **Тренировка войск**

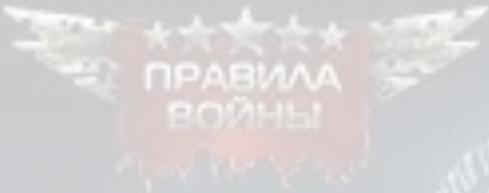
- 3.1 **Разработка боевой ед.**
- 3.2 **Обучение боевой ед.**
- 3.3 **Улучшение боевой ед.**
- 3.4 **Восстановление боевой ед.**

**Тренировка
войск**

**Постройка
базы**

**Добыча
ресурсов**





ЗАДАНИЕ 1 — ПРАВИЛА ВОЙНЫ

Далее механика:

1. CORE LOOP

- **База** — постройка зданий внутри укрепленного периметра. Обычно здесь тренируют солдат, проводят исследования или торговые сделки.
- **Ресурсы** — служат основным материалом для построек, исследований или содержания войск.
- **Армия** — то чем игрок взаимодействует с игровым миром: он может захватывать игровые объекты, защищать друзей от агрессоров или нападать на других игроков.

2. PVP

- **Грабёж игроков** — нападение с целью наживы *ресурсом*.
- **Война кланов** — групповой бой за локальные *ресурсы*.
- **Защита базы** — удержание армии в лагере, защита добытых *ресурсов*.
- **Защита дружеских баз** — удержание армии в лагере союзника, защищает от нападения более сильных противников.

3. PVE

- **Выполнение заданий** — стандартные игровые *квесты*.
- **Сюжетные локации** — уровни с цепочками *квестов*.

4. LOOT

- **Схемы и Чертежи** — материалы необходимые для производства *войск* или исследования *технологий*. Могут быть *обменены/приобретены* на торговой площадке.
- **Артефакты** — особые предметы используемые лагерем, они распространяют эффекты вроде: усиление войска, улучшение работы базы или ускоренная добыча *ресурсов*.
- **Ресурсы** — служат основным материалом для построек, исследований или содержания войск.

5. DONATE

- **Пожертвования** — игрок покупает голоса в сети Вконтакте, затем обменивает их на универсальную внутри-игровую валюту. Уникальные предложения, акции и скидки для стимуляции разовой закупки большого кол-ва игровой валюты.

ЗАДАНИЕ 2 — УЛУЧШЕНИЕ ГЕЙМПЛЕЯ

LINEAGE 2: Classic — Три способа улучшить геймплей!

(1) **Разработать систему ремёсел для персонажей**

Каждый любит что-то создавать и социально ориентированные ремёсла смогут повысить синергию между игроками. К тому же, существует очень большая прослойка игроков, — вроде *социальщиков, исследователей* или *фармеров*, — которым просто интересно пассивно проводить время в игре, *крафтя* разнородные цацки на продажу.

(2) **Разработать систему турниров**

Дополнительный режим турниров, вроде Башни Смерти, позволяет создать условия для получения игроком Эпической Бижутерии или другой высокоуровневой экипировки. Это очень важный момент, т.к инициация условия — «каждый может выиграть ценный приз», позволит замедлить отток игроков. Победитель такого турнира не самый сильный, — а самый хитрый или везучий.

(3) **Максимальный уровень "Пит-Боссов"**

"Нужно давать возможность игрокам побеждать" — именно такой политике следуют в *Blizzard*. Для того чтобы высокоуровневый персонаж мог получить некоторую эпическую бижутерию, ему приходится создавать и прокачивать нежелательных *твинков*. Это очень неудобно, да и к тому же, для чего еще нужен основной персонаж? Синхронизация уровней пит-боссов с максимальным, позволит *фармить* все игровые ценности основным (высокоуровневым) персонажем.

ЗАДАНИЕ 3 — КРАФТ

Напишите спецификацию крафта в Minecraft. Опишите систему ресурсов, предметов и их крафт. Формат: спецификация для разработчика, который будет имплементировать данную механику.

Крафт — способ получения большей части существующих в Minecraft [блоков](#) и [предметов](#).

Для того, чтобы *скрафтить* что-либо, игроку нужно определённым образом разместить в сетке крафта необходимые [ресурсы](#). В инвентаре игрок может воспользоваться сеткой крафта 2x2. Для работы с сеткой 3x3 нужно создать и установить [верстак](#).

Правила крафта:

1. Для крафта нужно иметь определённые ингредиенты.
2. Доски, каменный кирпич, песчаник и шерсть могут быть любого вида.
3. Ингредиенты должны быть расположены определённым образом друг относительно друга, за некоторыми исключениями. Если порядок в данном рецепте не важен, им можно воспользоваться и в инвентаре, если, конечно, все необходимые ингредиенты помещаются в сетку 2x2. Пример таких рецептов: [красители](#), [цветная шерсть](#), [тушёные грибы](#), приготовленный [паучий глаз](#), [огненный шар](#).
4. Расположение ингредиентов в сетке крафта не имеет значения. Например, факел можно скрафтить, поместив уголь над палкой в любом из 2 вариантов в сетке 2x2 или любом из 6 — для 3x3.
5. За каждый крафт (за каждый щелчок по иконке результата) будет использован только один ингредиент из каждой ячейки. Если при этом зажать SHIFT, будет сделано максимально возможное количество предметов.
6. Если рецепт несимметричен относительно вертикальной оси, его можно сделать любым образом. Например, рецепт мотыги можно получить из рецепта лопаты добавлением материала как слева, так и справа. Рецепты, обладающие этим свойством: [топор](#), [мотыга](#), [огниво](#), [ножницы](#), [ступени](#), [удочка](#) и [лук](#).



Ресурсы	Схемы рецептов 2x2			Результат	Описание
<p>Любая древесина (Дуб, Ель, Береза, Акация)</p>				<p>х4 Доски</p>	<p>Используется как строительный материал и для крафта многих вещей.</p>
		<p>дерево</p>			
<p>Любые доски (Дуб, Ель, Береза, Акация)</p>				<p>х1 Верстак</p>	<p>Нужен для крафта в 3x3 сетке.</p>
	<p>доска</p>	<p>доска</p>			
	<p>доска</p>	<p>доска</p>			
<p>Любые доски (Дуб, Ель, Береза, Акация)</p>				<p>х4 Палка</p>	<p>Используются для изготовления факелов, стрел, табличек, лестниц, заборов, ворот и инструментов.</p>
		<p>доска</p>			
		<p>доска</p>			
<p>Палка + (Уголь или Древесный уголь)</p>				<p>х4 Факел</p>	<p>Создаёт свет. Факелы также плавят снег.</p>
		<p>уголь</p>			
		<p>палка</p>			

ЗАДАНИЕ 4 — ТРИ СУНДУКА

Задача. Из волшебного сундука с шансом 27% выпадает редкий предмет. У игрока есть три сундука. Каковы шансы найти 0-1-2-3 редкие вещи в этих сундуках? Опишите ваши рассуждения.

Чтобы вычислить теоретическую вероятность, нужно провести многократный цикл опытов: каждый опыт воссоздаст пространство событий, а все возможные исходы событий — сохранятся в переменные **M0**, **M1**, **M2** и **M3**. Алгоритм следующий:

Задаем цикл в **100** повторений. Чтобы вычислить случилось событие или нет, генерируем случайное число от **1** до **100**. Если случайное число окажется в диапазоне от **1** до **27**, значит событие случилось (**A**), а если больше **27**, — события не случилось (**B**).

- A. (27%) **НАЙДЕНА РЕДКАЯ ВЕЩЬ!** произошло один раз
- B. (83%) **В СУНДУКЕ ПУСТО...** событие произошло два раза

Если произошло событие (**B**), то результат сохранится в переменной **M0(+1)**, а если событие (**A**) — результат сохранится в переменной **M1(+1)**, если событие (**A**) произойдет два раза из трех — результат сохранится в переменной **M2(+1)**, а если три раза из трех — в переменной **M3(+1)**;

После того как цикл опытов завершится, мы подсчитаем вероятность прока событий по формуле: **M / 100 * 27**.

Вот какой результат показал мой компилятор:

0 — 58.59%, 1 — 17.82%, 2 — 4.05%, 3 — 0.54%

ЗАДАНИЕ 4 — ТРИ СУНДУКА [дополнительный код на C++]

Задание дополнительно выполнил на C++, код прикрепил к письму.
Подробнее в "тз_2.зад_4.три_сундука.cpp"

```
out << "*****ИГРА С ВОЛШЕБНЫМИ СУНДУКАМИ*****\n"
out << "(Сколько сундуков вы хотите открыть? (1-100): ");
out << "Задайте количество сундуков, которые вы хотите открыть (1-100): ";
out << "Имя пользователя: ";
out << "Укажите количество циклов: ";
out << "Введите количество сундуков, которые вы хотите открыть (1-100): ";
// Просим указать количество сундуков, которые вы хотите открыть
out << "Сколько сундуков вы хотите открыть? (1-100): ";
in >> exp_cycle;
if(exp_cycle>=101 || exp_cycle==0) { // обрабатываем исключения
    cout << "Введите количество сундуков, которое вы хотите открыть (1-100): ";
    system("pause");
    return 0;
}
for(a=0; a<exp_cycle; a++) { // Цикл со счетчиком, установленным пользователем ранее
    for(b=0; b<3; b++) { // Цикл со счетчиком, установленным пользователем ранее
        cout << "Открываем сундук["<math>26</math><math>+</math><math>27</math>]<math>+</math>";
        if(b==0) {
            cout << "Вы обнаружили редкий предмет!";
        }
        cout << "В сундуке пусто...";
    }
    cout << "Открываем сундук["<math>32</math><math>+</math><math>27</math>]<math>+</math>";
    if(b==0) {
        cout << "Вы обнаружили редкий предмет!";
    }
    cout << "В сундуке пусто...";
}
cout << "Открываем сундук["<math>69</math><math>+</math><math>27</math>]<math>+</math>";
if(b==0) {
    cout << "Вы обнаружили редкий предмет!";
}
cout << "В сундуке пусто...";
}
cout << "Открываем сундук["<math>54</math><math>+</math><math>27</math>]<math>+</math>";
if(b==0) {
    cout << "Вы обнаружили редкий предмет!";
}
cout << "В сундуке пусто...";
}
cout << "Открываем сундук["<math>21</math><math>+</math><math>27</math>]<math>+</math>";
if(b==0) {
    cout << "Вы обнаружили редкий предмет!";
}
cout << "В сундуке пусто...";
}
cout << "Открываем сундук["<math>89</math><math>+</math><math>27</math>]<math>+</math>";
if(b==0) {
    cout << "Вы обнаружили редкий предмет!";
}
cout << "В сундуке пусто...";
}
}
// Статистика за 100 циклов:
cout << "Статистика за 100 циклов:
// Справочная информация:
if(getChar() != '\n') {
    cout << "Для продолжения нажмите любую клавишу . . . ";
    if(match==2) match2++; // Событие произошло два раза
    if(match==3) match3++; // Событие произошло три раза
}
system("pause");
return 0;
}
```

Полученно	редких предметов	-	случаев	Вероятность	%
Полученно 0	редких предметов	-	217 случаев	Вероятность	58.59%
Полученно 1	редких предметов	-	66 случаев	Вероятность	17.82%
Полученно 2	редких предметов	-	15 случаев	Вероятность	4.05%
Полученно 3	редких предметов	-	2 случая	Вероятность	0.54%

М ИСКЛЮ
double) m
(double) m
(double) m
<< (double) m

ЗАДАНИЕ 5 — ПОКУПКА ПРЕМИУМ АККАУНТА

Задача. 13% игроков покупают Премиум Аккаунт во free-to-play игре. 55% купивших Премиум Аккаунт в начале игры купили Набор Новичка. Также, 9% не купивших Премиум Аккаунт покупали Набор Новичка. Какова вероятность того, что только что пришедший игрок купит Премиум Аккаунт, если он только что купил Набор Новичка?

Нас интересует только вероятность совместного элементарного события: купит и Премиум Аккаунт, и Набор Новичка!

Формулы:

$$0.13 \times 0.55 * 100 = 7.15$$

$$13 / 100 * 55 = 7.15$$

Вот какой результат показал мой компилятор:

7,15%

(7\100)

ЗАДАНИЕ 5 —

ПОКУПКА ПРЕМИУМ АККАУНТА

[дополнительный код на C++]

Задание дополнительно выполнил на C++, код прикрепил к письму.
Подробнее в "тз_2.зад_5.покупка_прем_аккаунта.cpp"

```
id displayResult(double chance);
t main(int argc, char *argv[])
setlocale(LC_ALL, "Russian");
double
double
// Инициализируем переменные
xPlayers
pAccUser
pAccUserRecKit
cout << "ЗАДАНИЕ 5: Какова вероятность того, что только что пришедший игрок,
купит Премиум Аккаунт, если он только что купил Набор Новичка?
(c) vk.com/savluchenko, savaal@mail.ua, nunamir@mail.ru
13% от 100 новых игроков = 13 игроков
55% от 13 игроков = 7.15 игроков
Вероятность того, что новый игрок купит
и Премиум Аккаунт, и Набор Новичка - составляет 7.15%
Для продолжения нажмите любую клавишу . . .
// ШАГ 2: Вычисляем сколько игроков, из 13%, способны купить Набор Новичка
cout << "55% от " << temp_event1 << " игроков = " << temp_event1/100*pAccUserRecKit << " игроков\n";
temp_event2=temp_event1/100*pAccUserRecKit; // сохраняем результат в отдельной переменной
displayResult(xPlayers/100*temp_event2); // Выводим результат на экран
system("pause");
return 0;
```

ЗАДАНИЕ 6 — УРОН ЮНИТОВ

В мобильной mid-core игре есть юнит, который имеет базовый урон 35 ед. за удар. Выдаем юниту оружие, которое дает +12% урона, и второе оружие, которое дает +17% урона. Также на юнита наложен бафф, который дает +6% урона. Предположим, у этого юнита с последней битвы есть травма, которая дает -17% урона. Вопрос, сколько суммарно урона нанесет отряд из 40 таких юнитов 1 ударом? Опишите ваши рассуждения и объясните, почему вы выбрали именно этот вариант расчета урона

```
int main()
{
    // Задача. В мобильной mid-core игре есть юнит, который имеет базовый урон 35 ед. за удар. Выдаем юниту оружие, которое дает +12% урона, и второе оружие, которое дает +17% урона. Также на юнита наложен бафф, который дает +6% урона. Предположим, у этого юнита с последней битвы есть травма, которая дает -17% урона. Вопрос, сколько суммарно урона нанесет отряд из 40 таких юнитов 1 ударом? Опишите ваши рассуждения и объясните, почему вы выбрали именно этот вариант расчета урона
    double base_dmg = 35.0; // базовый урон юнита
    double wpn1, wpn2, buff, maim; // объявляем переменные эффектов урона
    double temp_wpn1, temp_wpn2, temp_buff, temp_maim; // объявляем переменные для хранения результатов
```

Сначала вычисляем результирующий урон юнита под объявленными ранее эффектами, затем умножаем полученный результат на количество юнитов.

Формула: $\text{сумм.урон} = (\text{баз.урон} + \text{эфф.оружия1} + \text{эфф.оружия2} + \text{эфф.баффа} - \text{эфф.травмы}) * 40$

Если выбрать другой вариант и сначала умножить базовый урон на 40, а затем прибавить сумму урона от эффектов, — то для 39 юнитов не будет рассчитан бонус и результат окажется неверным.

Вот какой результат показал мой компилятор:

```
// Вычисляем суммарный урон отряда из 40 юнитов
// формула: сум. урон = (баз. урон + эфф. оружия 1 + эфф. оружия 2 + эфф. баффа + эфф. травмы) * 40
cout << "\n\n40 ед. таких юнитов нанесут за один удар: ("
    << base_dmg << " * (1.12 + 1.17 + 1.06 - 0.17) * 40 = " << temp_wpn1 + temp_wpn2 + temp_buff - temp_maim) * 40 << " ед. урона\n\n";

system("pause");
return 0;
}
```

1652 ед. урона

ЗАДАНИЕ 6 — УРОН ЮНИТОВ

[дополнительный код на C++]

Задание также выполнил на C++, код прикрепил к письму.

Подробнее в "тз_2.зад_6.урон_юнитов.cpp"

```
int main(int argc, char *argv[])
{
    setlocale(LC_ALL, "Подробнее в \"тз_2.зад_6.урон_юнитов.cpp\"");

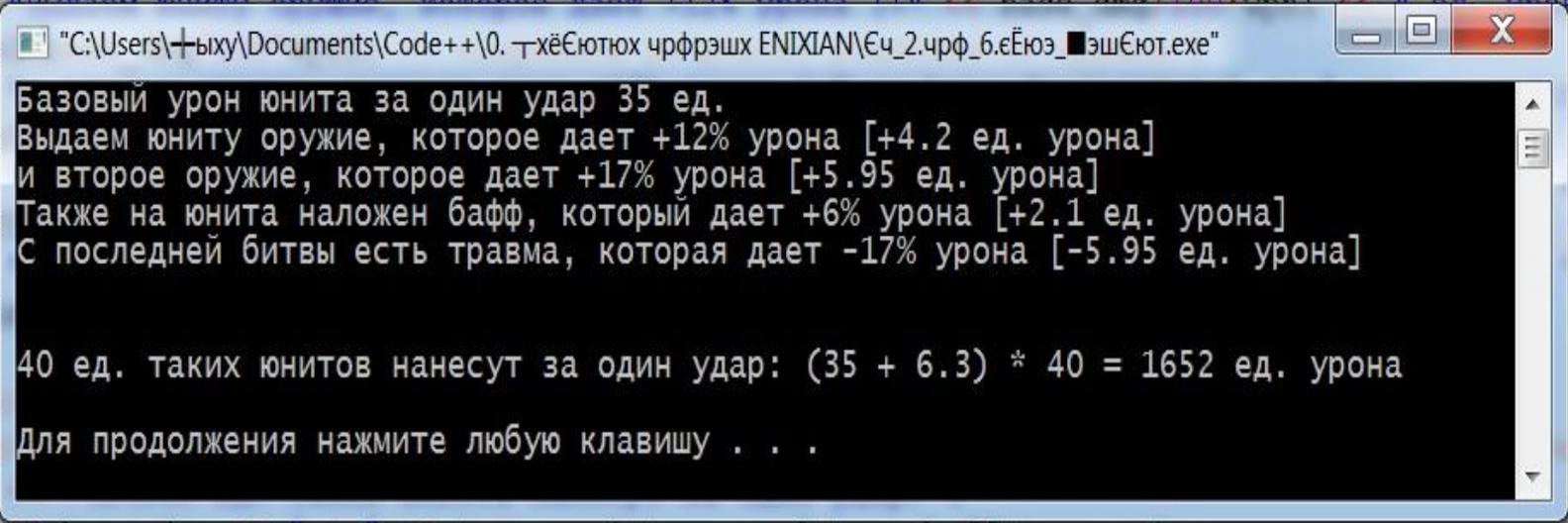
    double base_dmg = 35.0; // Базовый урон юнита
    double temp_wpn1 = 12.0; // +12% урона
    double temp_wpn2 = 17.0; // +17% урона
    double temp_buff = 6.0; // +6% урона
    double temp_maim = -17.0; // -17% урона

    < "Базовый урон юнита за один удар " << base_dmg << " ед.\n";
    < "Выдаем юниту оружие, которое дает +12% урона [" << base_dmg/100*temp_wpn1 << " ед. урона]";
    < "и второе оружие, которое дает +17% урона [" << base_dmg/100*temp_wpn2 << " ед. урона]";
    < "Также на юнита наложен бафф, который дает +6% урона [" << base_dmg/100*temp_buff << " ед. урона]";
    < "С последней битвы есть травма, которая дает -17% урона [" << base_dmg/100*temp_maim << " ед. урона]";

    double total_dmg = (base_dmg + temp_wpn1 + temp_wpn2 + temp_buff + temp_maim) * 40;

    << "40 ед. таких юнитов нанесут за один удар: (" << base_dmg << " + " << temp_wpn1+temp_wpn2+temp_buff+temp_maim
    << ") * 40 = " << (base_dmg+temp_wpn1+temp_wpn2+temp_buff+temp_maim)*40 << " ед. урона\n\n";

    ("pause");
    return 0;
}
```



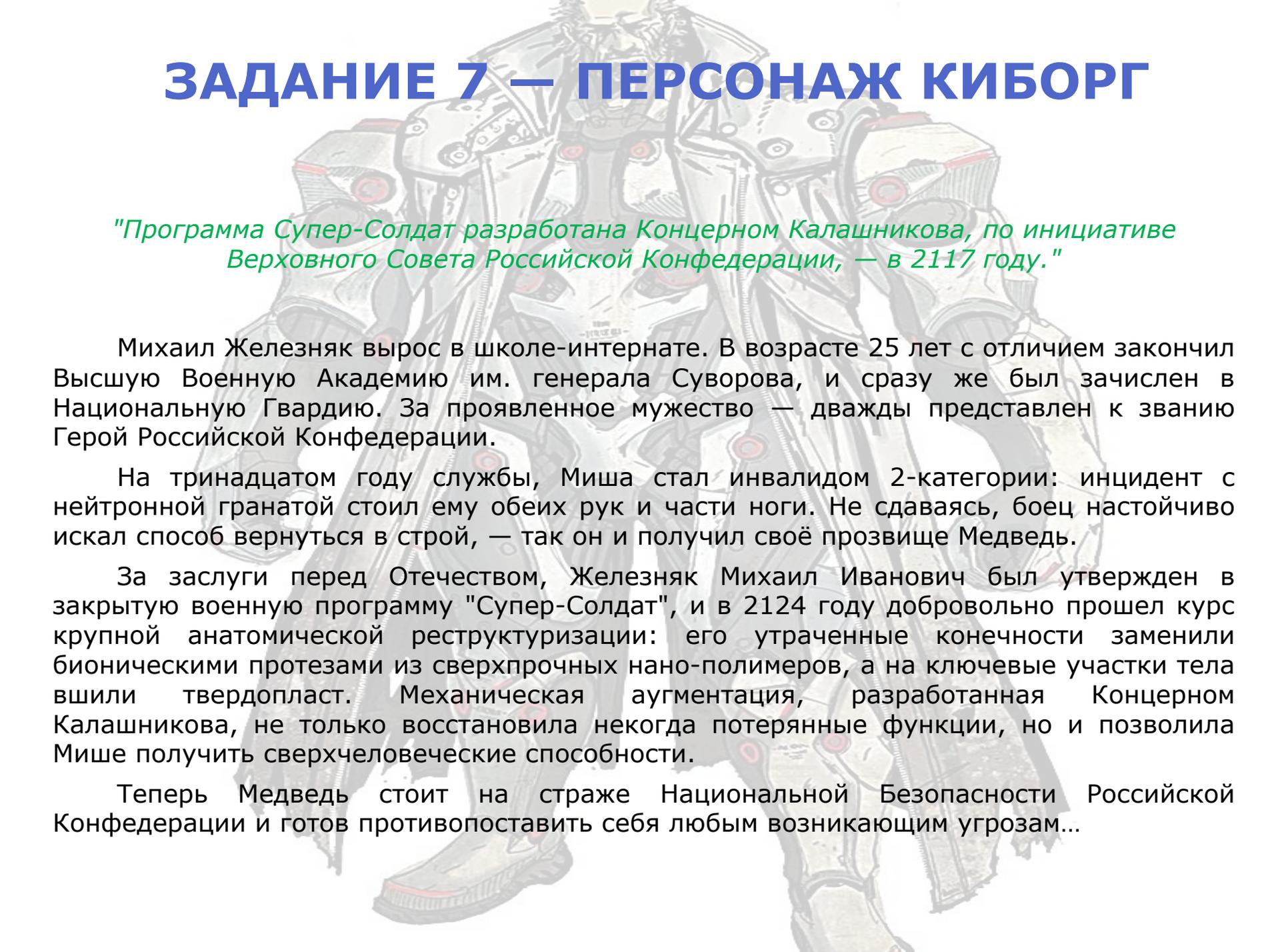
ЗАДАНИЕ 7 — ПЕРСОНАЖ КИБОРГ

Придумайте легендарный предмет, оружие или персонажа для гипотетической MMO игры в сеттинге sci-fi: опишите его характеристики и особые возможности, напишите текстовое описание (с которым он будет преподноситься в игре) и подберите референс внешнего вида.



Ф.И.О : Железняк Михаил Иванович
ДАТА РОЖДЕНИЯ : 2081 год, мп. Москва
ПОЗЫВНОЙ : "Медведь"
КЛАСС : Киборг

ЗАДАНИЕ 7 — ПЕРСОНАЖ КИБОРГ



"Программа Супер-Солдат разработана Концерном Калашникова, по инициативе Верховного Совета Российской Конфедерации, — в 2117 году."

Михаил Железняк вырос в школе-интернате. В возрасте 25 лет с отличием закончил Высшую Военную Академию им. генерала Суворова, и сразу же был зачислен в Национальную Гвардию. За проявленное мужество — дважды представлен к званию Герой Российской Конфедерации.

На тринадцатом году службы, Миша стал инвалидом 2-категории: инцидент с нейтронной гранатой стоил ему обеих рук и части ноги. Не сдаваясь, боец настойчиво искал способ вернуться в строй, — так он и получил своё прозвище Медведь.

За заслуги перед Отечеством, Железняк Михаил Иванович был утвержден в закрытую военную программу "Супер-Солдат", и в 2124 году добровольно прошел курс крупной анатомической реструктуризации: его утраченные конечности заменили бионическими протезами из сверхпрочных нано-полимеров, а на ключевые участки тела вшили твердопласт. Механическая аугментация, разработанная Концерном Калашникова, не только восстановила некогда потерянные функции, но и позволила Мише получить сверхчеловеческие способности.

Теперь Медведь стоит на страже Национальной Безопасности Российской Конфедерации и готов противопоставить себя любым возникающим угрозам...

ЗАДАНИЕ 7 — ПЕРСОНАЖ КИБОРГ

СПОСОБНОСТИ:

ЖЕЛЕЗНЫЙ КУЛАК

Мощный удар, наносящий **35** ед. урона и, с вероятностью **13%**, оглушающий жертву на **3** сек.



СОКРУШИТЕЛЬНЫЙ БРОСОК

Михаил проводит фирменный прием, — хватает противника и швыряет его об землю, нанося **115** ед. урона. В следующие **5** сек., жертва будет получать на **50%** больше урона.



УДАРНАЯ ВОЛНА

Михаил бьет по земле, — вызывая мощную ударную волну, наносящую **20** ед. урона и отбрасывающую противников на **3** метра.



ОСОБЕННОСТИ:

МЕХАНИЧЕСКАЯ АУГМЕНТАЦИЯ

Тяжелый экзоскелет увеличивает урон Михаила на **40** ед., но замедляет его скорость атаки на **12%**



ЗАЩИТНАЯ МАТРИЦА

Встроенный в доспехи Михаила, плазменный щит — поглощает до **55** ед. повреждений. Перезарядка — **7** сек.



МАСКИРОВОЧНОЕ ПОЛЕ

Находясь вне боя в течение **3** сек., Михаил входит в состояние незаметности. Действие прекратится, если он получит урон.



АБСОЛЮТ:

АКСЕЛЕРАЦИЯ ТУРБИН

Позволяет Михаилу расширить предельную нагрузку на протезы, увеличивая его скорость на **100%** — на **15** сек.



Мне еще многому предстоит научиться — я это понимаю. Надеюсь что не потратил ваше время зря... С уважением.

ОЛЕГ САВЛ

savaal@mail.ua

[vk.com / 433986090](https://vk.com/433986090)