

Сложно ли разрабатывать игры?

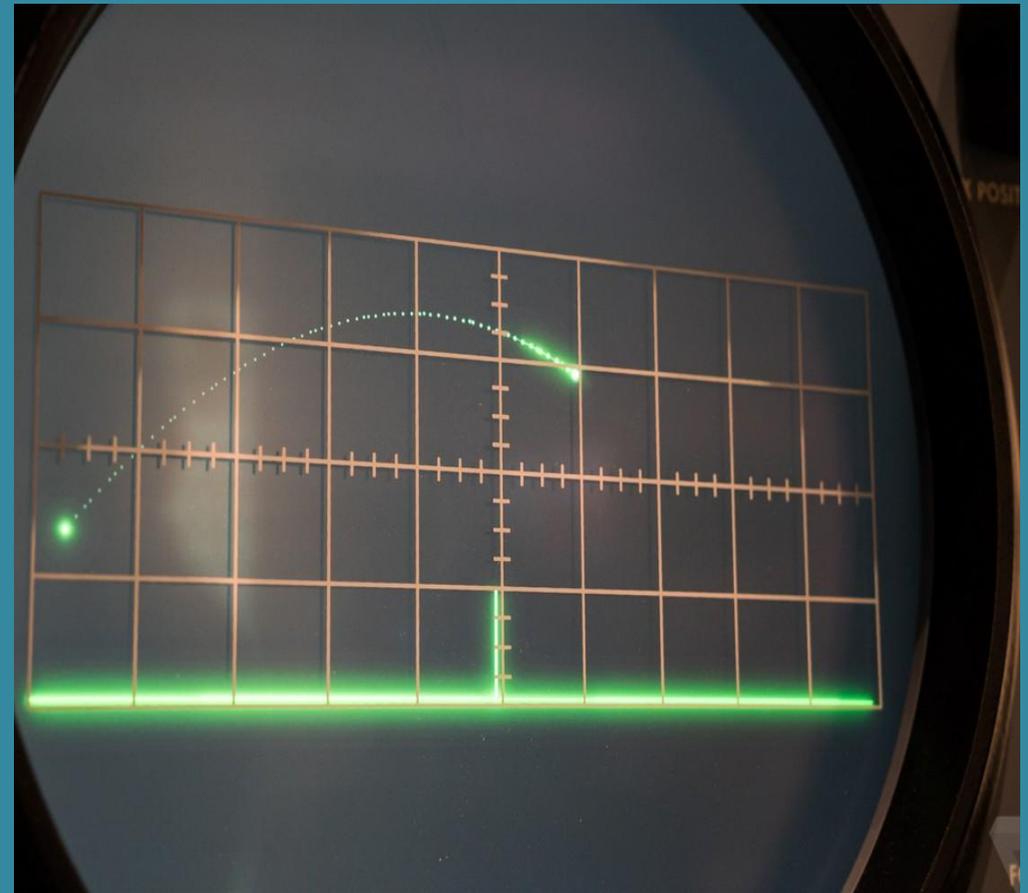
Презентацию подготовил
Ученик 10 Т2 класса:
Есенкулов Телимбек.

Цели проекта:

1. Узнать на сколько сложно/легко создавать игры.
2. Сколько на это тратиться денег?
3. С какого возраста можно пробовать этим заниматься?

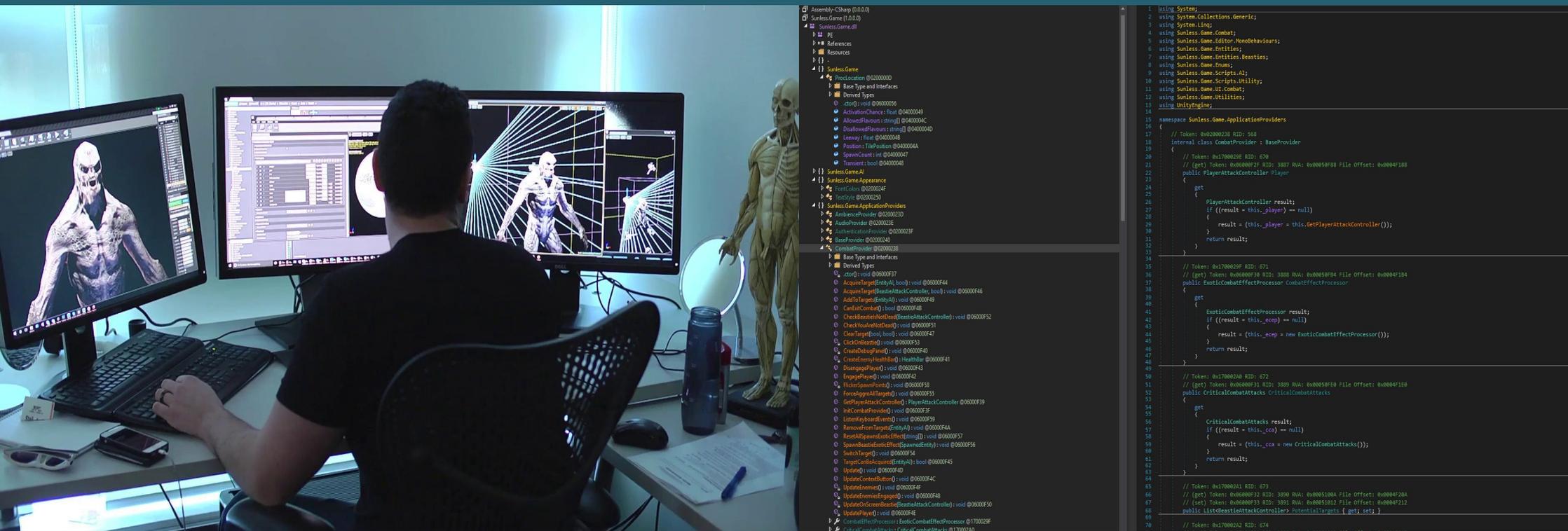
Интересные факты:

1. Первая видеоигра в истории человечества была разработана в **1958** году и называлась она "**Tennis for Two**", выводилась данная игра на осциллограф. Использовалась она для того чтобы развлекать посетителей Брукхейвенской национальной лаборатории.
2. Во время предвыборной кампании Барак Обама заплатил за рекламу в видеоигре **Burnout Paradise**.



РАЗРАБОТКА ИГРЫ

По времени, разработка хорошей игры действительно требует от 3 лет и более. Нужно еще учитывать вероятные задержки. Если будут перебои с финансированием, проект затянется на 5 лет и больше. У крупных компаний и у AAA-проектов затраты в разы выше, но и доход тоже. В их случае цифру в 100 млн рублей можно смело умножить на 30-40.



```
Assembly-CSharp (0.0.0.0)
├── SunlessGame (1.0.0.0)
│   ├── PE
│   ├── References
│   └── Resources
│   └── {}
│       └── SunlessGame
│           ├── PradLocation @02000000
│           │   ├── Base Type and Interfaces
│           │   └── Derived Types
│           │       ├── <ctor>() void @06000056
│           │       ├── ActivationChance float @04000049
│           │       ├── AllowedFlavours string[] @0400004C
│           │       ├── DisallowFlavours string[] @0400004D
│           │       ├── Liveness float @0400004B
│           │       ├── Position TitlePosition @0400004A
│           │       ├── SpawnCount int @04000047
│           │       └── Transient bool @04000048
│           ├── SunlessGameAI
│           ├── SunlessGameAppearance
│           │   ├── FurrColor @0200024F
│           │   ├── Texture @02000250
│           │   └── SunlessGameApplicationProviders
│           │       ├── AmbientProvider @0200023D
│           │       ├── AudioProvider @0200023E
│           │       ├── AuthenticationProvider @0200023F
│           │       ├── BaseProvider @02000238
│           │       └── CombatProvider @0200023B
│           ├── Base Type and Interfaces
│           └── Derived Types
│               ├── <ctor>() void @06000F37
│               ├── AcquireTargetEntryA bool void @06000F44
│               ├── AcquireTarget(BeastAttackController, bool) void @06000F46
│               ├── AddToTargetsEntryA void @06000F49
│               ├── CanAttack bool @06000F4B
│               ├── CheckCanBeNotDead(BeastAttackController) void @06000F52
│               ├── CheckYouAreNotDead void @06000F51
│               ├── ClearTarget(bool, bool) void @06000F47
│               ├── ClearOnDeath void @06000F53
│               ├── CreateDebugPanel void @06000F4D
│               ├── CreateEnemyHealthBar HealthBar @06000F41
│               ├── DisengagePlayer void @06000F43
│               ├── EngagePlayer void @06000F42
│               ├── IsTargetImmune void @06000F58
│               ├── ForceAggroAllTargets void @06000F55
│               ├── GetPlayerAttackController PlayerAttackController @06000F39
│               ├── HitCombatProvider void @06000F3F
│               ├── ListenForBeastDeath void @06000F59
│               ├── RemoveFromTargetsEntryA void @06000F4A
│               ├── ResetAllSpawnsExoticEffect(string) void @06000F57
│               ├── SpawnBeastExoticEffect(SpawneeEntry) void @06000F56
│               ├── SwitchTarget void @06000F54
│               ├── TargetCanBeAcquired(EntityA) bool @06000F45
│               ├── Update void @06000F4D
│               ├── UpdateCombatButton void @06000F4C
│               ├── UpdateEnemy void @06000F4E
│               ├── UpdateEnemyEngaged void @06000F4B
│               ├── UpdateScreenBeast(BeastAttackController) void @06000F50
│               └── UpdatePlayer void @06000F4E
│           └── CombatEffectProcessor ExoticCombatEffectProcessor @1700024F
│           └── CriticalCombatAttacks CriticalCombatAttacks @17000242
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using Sunless.Game.Combat;
5 using Sunless.Game.Editor.Homobehaviours;
6 using Sunless.Game.Entities;
7 using Sunless.Game.Entities.Besties;
8 using Sunless.Game.Enums;
9 using Sunless.Game.Scripts.AI;
10 using Sunless.Game.Scripts.Utility;
11 using Sunless.Game.UI.Combat;
12 using Sunless.Game.Utilities;
13 using UnityEngine;
14
15 namespace Sunless.Game.ApplicationProviders
16 {
17     // Token: 0x0200023B RID: 358
18     internal class CombatProvider : BaseProvider
19     {
20         // Token: 0x1700023E RID: 678
21         // (get) Token: 0x06000F3F RID: 3887 RVA: 0x00050F58 File Offset: 0x0004F188
22         public PlayerAttackController Player
23         {
24             get
25             {
26                 PlayerAttackController result;
27                 if ((result = this._player) == null)
28                 {
29                     result = (this._player = this.GetPlayerAttackController());
30                 }
31                 return result;
32             }
33         }
34
35         // Token: 0x1700023F RID: 679
36         // (get) Token: 0x06000F40 RID: 3888 RVA: 0x00050F58 File Offset: 0x0004F184
37         public ExoticCombatEffectProcessor CombatEffectProcessor
38         {
39             get
40             {
41                 ExoticCombatEffectProcessor result;
42                 if ((result = this._ecpp) == null)
43                 {
44                     result = (this._ecpp = new ExoticCombatEffectProcessor());
45                 }
46                 return result;
47             }
48         }
49
50         // Token: 0x17000240 RID: 672
51         // (get) Token: 0x06000F38 RID: 3889 RVA: 0x00050F58 File Offset: 0x0004F188
52         public CriticalCombatAttacks CriticalCombatAttacks
53         {
54             get
55             {
56                 CriticalCombatAttacks result;
57                 if ((result = this._cca) == null)
58                 {
59                     result = (this._cca = new CriticalCombatAttacks());
60                 }
61                 return result;
62             }
63         }
64
65         // Token: 0x17000241 RID: 673
66         // (get) Token: 0x06000F39 RID: 3890 RVA: 0x00051004 File Offset: 0x0004F28A
67         // (set) Token: 0x06000F33 RID: 3851 RVA: 0x00051012 File Offset: 0x0004F212
68         public List<BeastAttackController> PotentialTargets { get; set; }
69
70         // Token: 0x17000242 RID: 674
```

Со сколько лет можно делать свои игры:

Неважно сколько тебе лет, где ты находишься, например: разработчик серии игр Five Nights at Freddy's работал грузчиком, водителем грузовика и кассиром сети фастфудов, а главный дизайнер Mass Effect вообще мусорщиком. Никто не учил их создавать игры. Они кровью и потом добились своего признания. Поэтому главное иметь желание и свободное время.



Заключение :

Видеоигры глубоко засели в повседневной жизни человека, пройдя долгий путь от ракетного симулятора, интерфейсом которого выступало аналоговое устройство на базе электронно-лучевой трубки, до сложносочиненных проектов, разнообразными внутренними мирами, детальными дизайнами уровней и прочими дарами технологического прогресса. Сейчас уже неизвестно, какое будущее ожидало эту сферу развлечений, если бы к ней не приложили руку все вышеупомянутые деятели индустрии, каждого из которых стоит в очередной раз поблагодарить и поиграть в их вечные шедевры.



ИСТОЧНИКИ:

<https://zen.yandex.ru/media/apocalipsgame/8-samyh-interesnyh-faktov-iz-mira-videoigr-5b8e7e5cbe15b400aed1d2f8>

<https://gamesqa.ru/games/pochemu-dolgo-razrabatyvayut-igry-19423/>

[https://www.playground.ru/five_nights_at_freddys/news/krovyyu_i_potom_kak_istyj](https://www.playground.ru/five_nights_at_freddys/news/krovyyu_i_potom_kak_istyj_hristianin_stal_korolyom_horrora-313701)

[hristianin_stal_korolyom_horrora-313701](https://www.playground.ru/five_nights_at_freddys/news/krovyyu_i_potom_kak_istyj_hristianin_stal_korolyom_horrora-313701)

<https://prosports.kz/news/462674-top-5-razrabotchikov-ostavivshih-sled-v-igrovoi-industrii>