

# 海外ゲーム会社入社問題の傾向と対策

株式会社スクウェア・エニックス  
研究開発部ジェネラルマネージャー  
岩崎哲史

# プレゼンテーションの構成

---

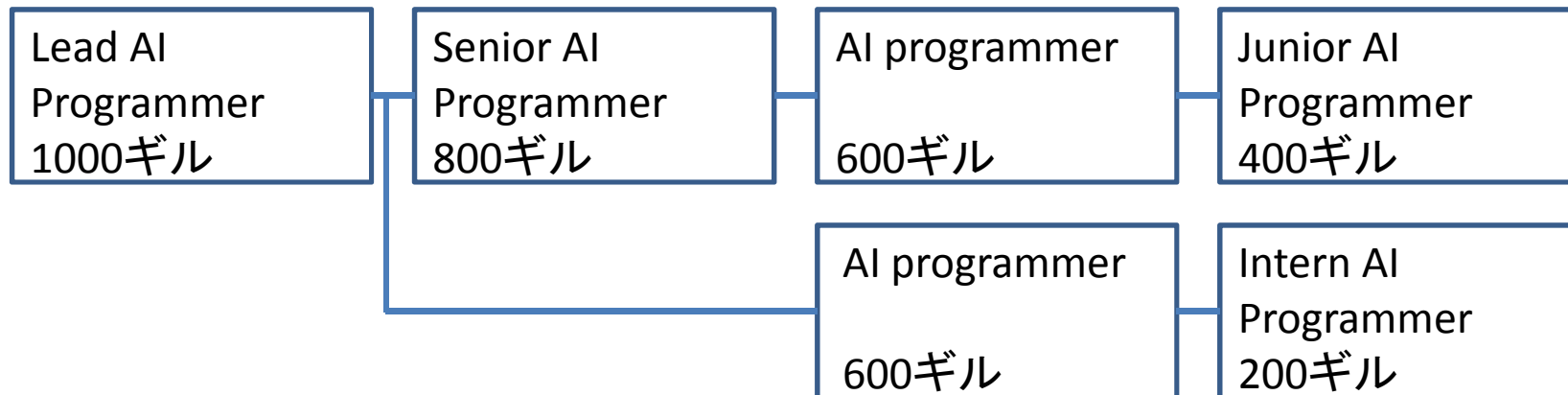
- 海外採用プロセスに関する説明
- ポジションについての考え方
- スペシャリティについての考え方
- 応募前に習得しておきたいこと
- 応募にあたって用意するもの
- 応募後の進行
- ビザについて
- 入社問題の一般的な傾向
- 各種問題

# 海外採用プロセスに関する説明

- 一次面接（電話）
  - 人事レベル、希望給与やポジション、基本的な動機等の説明
  - 自宅で行う何らかのプログラムの課題
- 二次面接（電話）
  - 現場レベル、技術力の査定等
  - 自宅で行う何らかのプログラムの課題
- 現地面接（交通費は多くの場合会社負担）
  - 筆記テスト、幹部の面接等と食事

# ポジションについての考え方1

- 人に給料が張り付いているのではなく、ポジションに給料が張り付いている
- ゲームタイトルと予算によって、あらかじめチームがどのようなポジション構成になるか設計されている



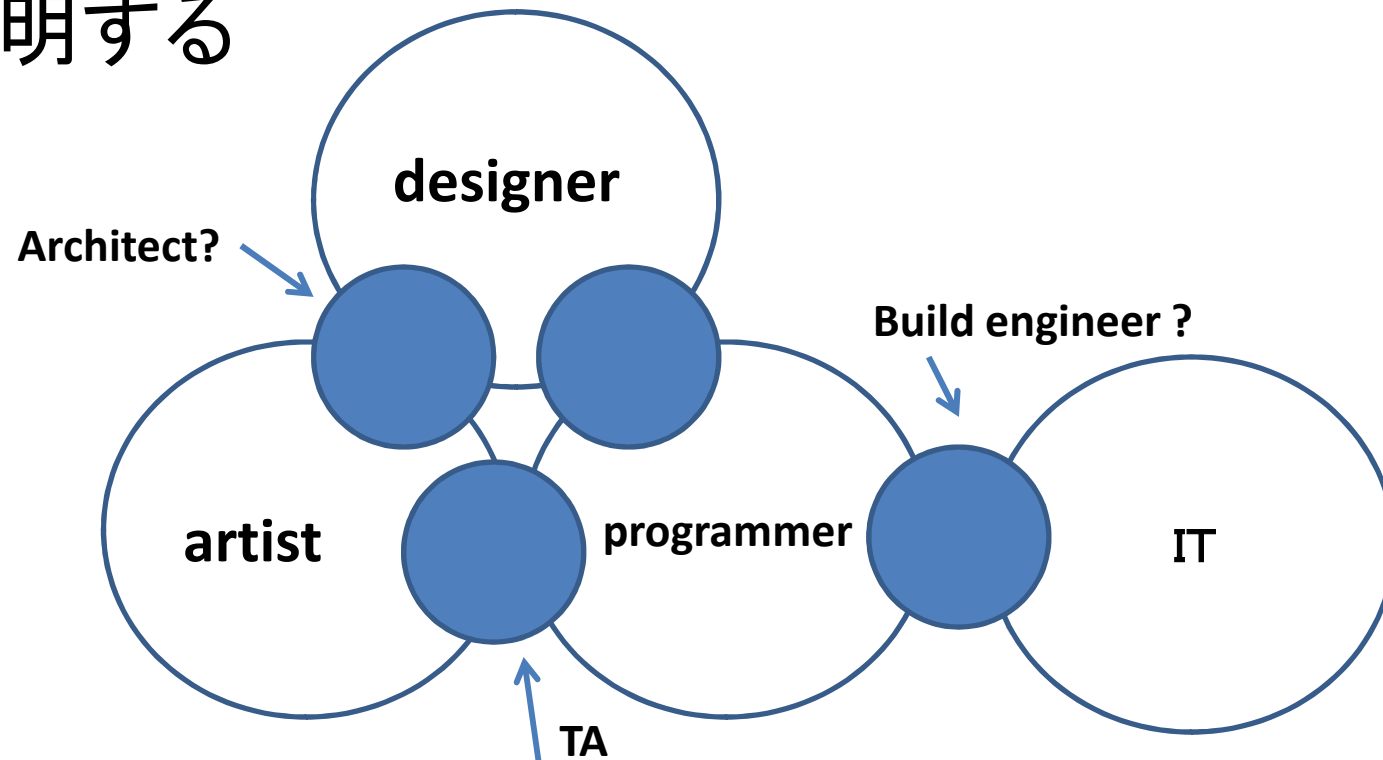
# ポジションについての考え方2

---

- プログラマ職種スペシャリスト系  
Graphic programmer/Shader programmer  
Physics programmer/Network programmer  
Animation programmer/AI programmer
- プログラマ職種ジェネラリスト系  
Game programmer/Console programmer  
UI programmer/Tools programmer  
Build engineer

# ポジションについての考え方3

- 組織上必要な役割からポジションが作られる。必要がある場合、新しいポジション名を発明する



# スペシャリティについての考え方

- 海外では専門性の高い人材を優遇する傾向がある
  - 特に学校で勉強したことと、仕事でしていることの一致は重要
  - キャリアも一貫していることが望ましい
  - マネージャーについてもスペシャリティが必要、つまりマネジメントの勉強をした人
- \* 以上の傾向はUSのほうが大きい

# 応募前に習得しておきたいこと

---

- 英語
  - TOEIC換算で目安は限りなく900点+
  - 英語力 \* コミュニケーション能力 \* 技術力 = 仕事の能力
- テーブルマナー、レディーファースト、表情の作り方など。
- 海外の文化について
  - 特に、キリスト教、イスラム教、神道などを理解すると、西洋文化の理解が深まる。



# 応募にあたって用意するもの

---

- レジюме
  - 履歴書と職務経歴書が一緒になったようなもの
- カバーレター
  - 応募企業へのラブレター
- 応募方法
  - エージェント経由
  - 直接応募
  - 友人を通じた紹介

# カバーレターの例

Tetsuji Iwasaki  
住所をここに書きます

- 
- 
- 
- April 23,2005
- 宛先をここに書きます
- 
- To whom it may concern:
- 
- I would like to apply for a position as Software Engineer. I have profound experience of nearly 20 titles covering many platforms and genres which include:
- "FINAL FANTASY XI" a MMO RPG for Unix server, Play Station2 and windows.
- "FINAL FANTASY TACTICS" a simulation game for Play Station.
- 
- I started game programming when I was 12 years old. Since then, I have dedicated almost all of my life to games. I have adapted to some big changes in the industry, such as 3D, networks and physical simulation, all of which I maintain a high level of expertise. Because of my ability to adapt to new technologies, I have always been designated as a key person for any new project which calls for technical challenges. Although game development tends to require a number of people with specialized skills in different areas, I have wide experience and skill in the development of many games. I can use this knowledge to solve problems in an efficient manner.
- 
- I have been attracted to your company by your products and business method of integrating each process and department through marketing and strong teamwork. Teamwork is a quality I highly value especially in the complex task of game development

# レジュメの例

- **OBJECTIVE: Software Engineer**

- **QUALIFICATIONS**

- · More than ten years experience on approximately 20 projects covering various genres.
- · Experienced in developing graphic engines, animation systems, network games, script systems, interfaces, and physics systems.
- · Majored in Linear Economics which uses high-level linear algebra.
- · Proven track record as a key person for any new project which calls for technical challenges.
- · Worked as a lead server programmer on FINAL FANTASY XI.
- · Have worked with many of the industry's top game developers in Japan.

- **WORK EXPERIENCE**

- **04/1994-10/2002 SQUARE ENIX CO., LTD**

- **"FINAL FANTASY XI"** (Play station2 & Windows)

- Lead server programmer.(Lobby/world server section)

- **"PLAY ONLINE"**

- In charge of the server and client of the patch system.

- · Directed 4 programmers.
- · Constructed protocols for controlling more than 1000 servers.
- · Planned certification(login) protocols.
- · Designed synchronous protocols to communicate between contents.
- · Developed programs for version up servers and clients.
- · Created a client lobby menu program including a communication program (Play station2 and Windows)

# 応募後の進行

---

- 有望であれば大体1週間程度で最初のコンタクト
- 不採用になった場合でも、基本は連絡なし
  - 数ヵ月後や数年後にオファーがある場合も
  - たとえ優秀な人材であっても、ポジションが空いていない場合や、忙しい時期はやとわれない。
  - 疑わしきは採用せず

# ビザについて

---

- 大学卒業もしくは10年単位の実務経験が望ましい。
  - 外国人より自国人を優先するため
- 国によってビザ取得の難易度は違う。
  - カナダ・EU・オーストラリアは比較的容易
  - UK, 特にUSは比較的難しい



# Q & A



# 入社問題の一般的な傾向

---

- プログラミングの知識全般
- C++について、ただ使えるだけでなく、背後の知識を問うもの
- 基礎的な数学
- メジャーなアルゴリズムや用語の知識
- 最適化についての知識
- 設計についての知識

# 個別の問題についての注意

---

- 4年前に20社受けた傾向から作った問題
- 前職で数多くの入社問題作成を担当
- 全てオリジナルの問題で、実際に出題されたものではありません



# プログラミングの知識全般1

---

- 問題1

$0x3d + 0xc3 = 0x \text{????}$

- 問題2

```
short hoge=1;
```

```
hoge ^= -1;
```

```
hoge += 1;
```

```
printf(“%d\n”,(int) hoge);
```

表示される値は？

# 答え

---

- 問題1

0x100

- 問題2

表示される値は-1

0xffffとXORして1を足すと、符号反転となる

# プログラミングの知識全般2

---

```
short hoge[100];  
printf(“%d¥n”,sizeof(hoge));  
printf(“%d¥n”,sizeof(char));  
printf(“%d¥n”,sizeof(short));  
printf(“%d¥n”,sizeof(int));  
それぞれ表示される値は？
```

# 答え

---

<code>printf(“%d¥n”,sizeof(hoge));</code>	.....200
<code>printf(“%d¥n”,sizeof(char));</code>	.....1
<code>printf(“%d¥n”,sizeof(short));</code>	.....2
<code>printf(“%d¥n”,sizeof(int));</code>	.....不明

intのサイズはプラットフォーム依存  
4が多い

# プログラミングの知識全般3

---

- 問題1

devirtualとはなにか説明しなさい

- 問題2

オーバーロードとオーバーライドの違いを説明しなさい

- 問題3

クラステンプレートとテンプレートクラスの違いを説明しなさい

# 答え

---

- 問題1  
動的呼び出しを静的呼び出しに変換する手法
- 問題2  
オーバーロードは同一クラス内で関数名が同一で引数の異なる関数を複数定義すること、  
オーバーライドは基底クラスの関数を派生クラスで再定義すること。
- 問題3  
クラステンプレートは、クラスを作成するためのテンプレート(ひな形)。引数によって異なるクラスを作成するためにある。  
テンプレートクラスは、クラステンプレートによって生成されるクラスのインスタンス。

## C++について、ただ使えるだけでなく、 背後の知識を問うもの1

---

下記のプログラムを高速化しなさい。  
unknownvalueは1000から10000までの不  
定値です。

```
std::vector<int> testVector;  
for( int i=0; i < unknownvalue; i++ )  
    testVector.push_back( i);
```

# 答え

---

```
std::vector<int> testVector;  
testVector.reserve(unknownvalue);  
for( int i=0; i < unknownvalue; i++ )  
    testVector.push_back( i);
```



## C++について、ただ使えるだけでなく、 背後の知識を問うもの2

---

次のコードの問題点を指摘しなさい

```
struct Vec3
{
    int x,y,z;
    Vec3(int a,int b,int c):x(a),y(b),z(c){};
}
void testmain(void)
{
    Vec3& tmp=test();
    printf("%d,%d,%d¥n",tmp.x,tmp.y,tmp.z);
    printf("%d,%d,%d¥n",tmp.x,tmp.y,tmp.z);
}
Vec3 test()
{
    return Vec3(1,0,0);
}
```

# 答え

---

```
struct Vec3
{
    int x,y,z;
    Vec3(int a,int b,int c):x(a),y(b),z(c){};
}
void testmain(void)
{
    Vec3 tmp=test();
    もしくは
    const Vec3& tmp=test();
    printf(“%d,%d,%d\n”,tmp.x,tmp.y,tmp.z);
    printf(“%d,%d,%d\n”,tmp.x,tmp.y,tmp.z);
}
Vec3 test()
{
    return Vec3(1,0,0);
}
```

# 基礎的な数学1

---

- 問題1

ゲームプログラムで、クォータニオンとはどのような場合に使用されますか？

- 問題2

クォータニオンを使うメリットに何がありますか？

# 答え

---

- 問題1

回転の計算

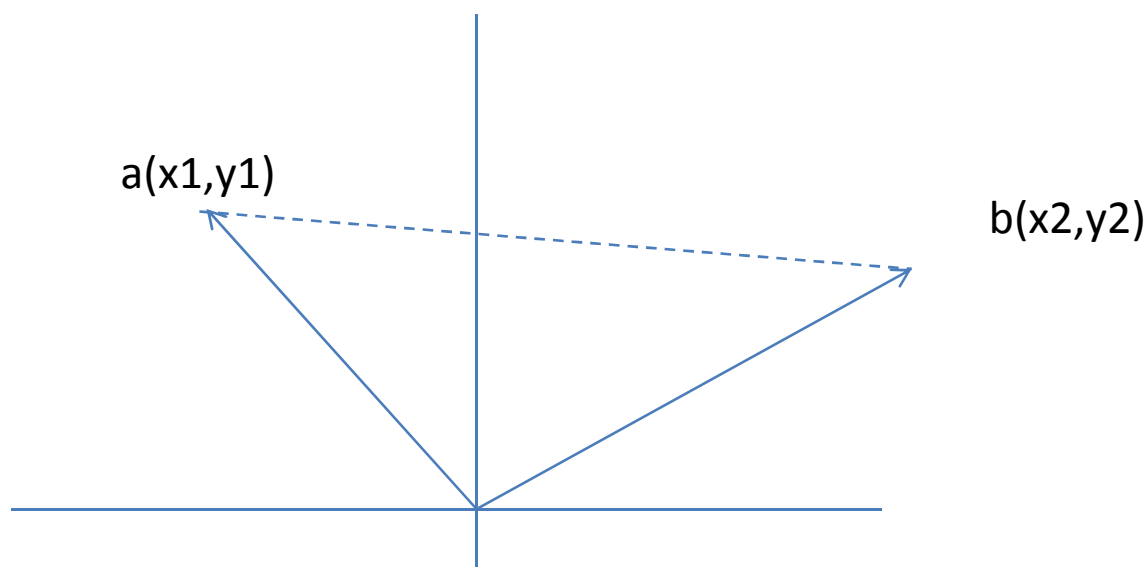
- 問題2

球面線形補完が出来る

ジンバルロックが防げる等

## 基礎的な数学2

XY平面上に下のような2つのベクトル  $a(x_1, y_1)$ ,  $b(x_2, y_2)$  があります。原点とベクトル  $a, b$  で作られる3角形の面積を求めなさい。



# 答え

---

- $| (x_1 * y_2 - y_1 * x_2) | * 0.5$

\* 内積、外積は重要です

# メジャーなアルゴリズムや用語の知識

---

下記のキーワードについて何を表すか説明しなさい。PIDはプロセスIDではありません。

PID

FSM

SSS

OOBB

SSAO

PCA

# 答え

---

PID 比例, 積分, 微分を使用した目標値へ出力制御

FSM 有限状態マシン

SSS 表面下散乱、人間の肌などを表現

OBB Object Orientated Bounding Box

SSAO Screen Space Ambient Occlusion

PCA 主成分分析



# メジャーなアルゴリズムや用語の知識

---

下記のキーワードについて何を表すか説明  
しなさい。

BSP tree

KD tree

Beam tree

OCT tree

Portal

# 答え

---

- いずれも空間分割に使用されるデータ構造もしくはアルゴリズムそのものを指します。
- 詳しい説明は省略させていただきます。

# 最適化についての知識1

---

## 問題1

メモリアクセスにおけるLHSとはなにか説明しなさい

## 問題2

主要コンソールで使用されているPOWER PCでLHSがおこる代表的な例と理由を述べなさい

# 答え

---

## 問題1

LHS:ロードヒットストア

値をメモリに書き込んだ後、すぐ読み込んだ場合、パイプラインがフラッシュ。

## 問題2

floatからint,int からfloatへの代入

float-intとint-floatのレジスタコピーをサポートするインストラクションがないので、これらがメモリを経由して行われるため。

# 最適化についての知識2

---

- 次のプログラムを最適化しなさい

```
static std::vector<int> testVector;
```

```
int sum = 0;
```

```
for( int i=0; i<(int)testVector.size(); i++ )
```

```
{
```

```
    sum +=testVector[i];
```

```
    dosomething(sum);
```

```
}
```

# 答え

---

```
int endCount = (int)testVector.size();  
for( int i=0; i< endCount; i++ )
```

関数がインライン展開されても、ローカル変数でない限りメモリアクセスは省略されない  
ので、レジスタに乗る可能性のあるローカル変数にストアすべき。

# 設計についての知識

---

シングルトンパターンとはなにか説明しなさい

# 答え

---

インスタンスが1つしかないことを保証するためのクラスの構造パターン



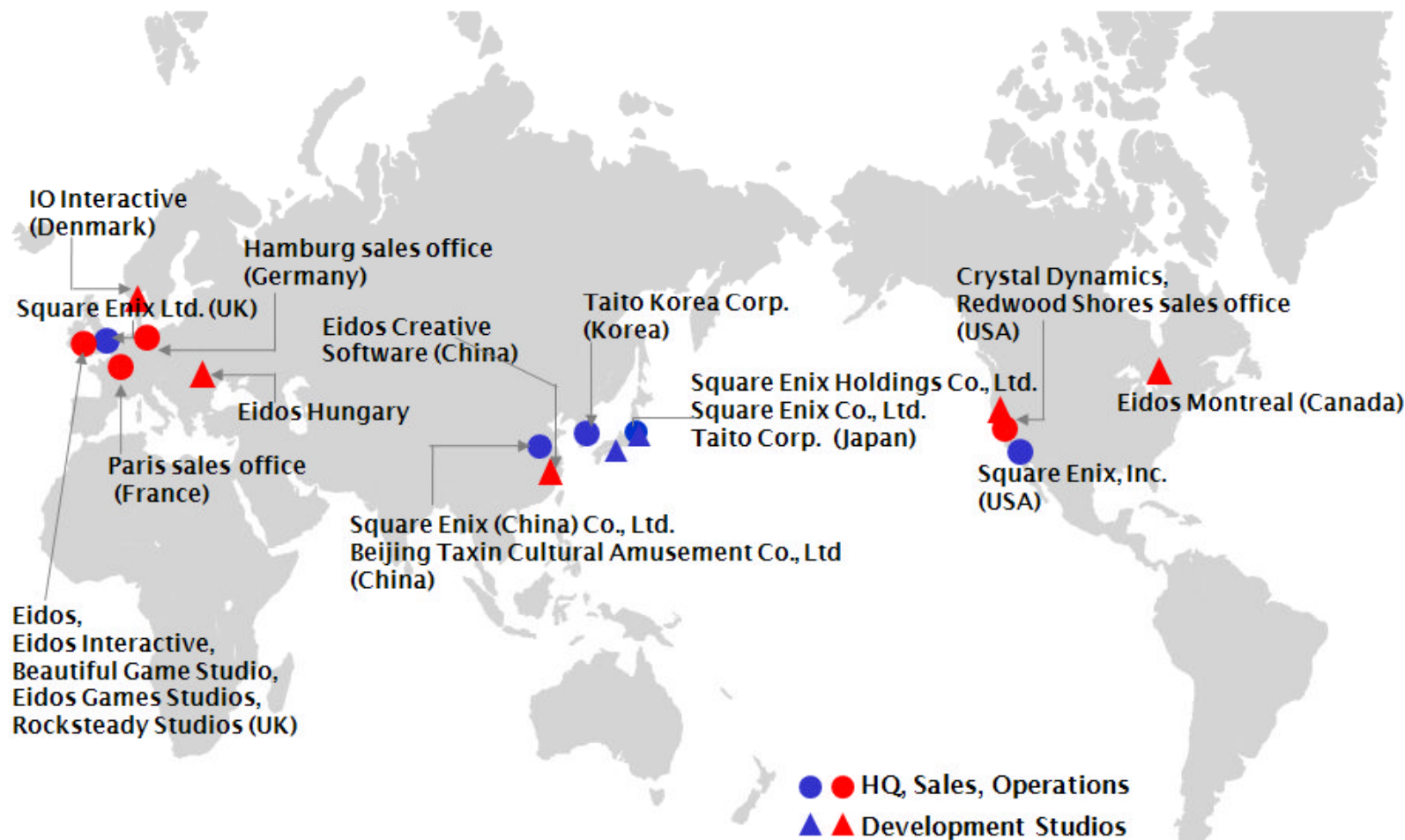


# Q & A



# よろしくお願ひします

## スクウェア・エニックス・グループ(Eidosグループ化後)



2009年8月31日現在