

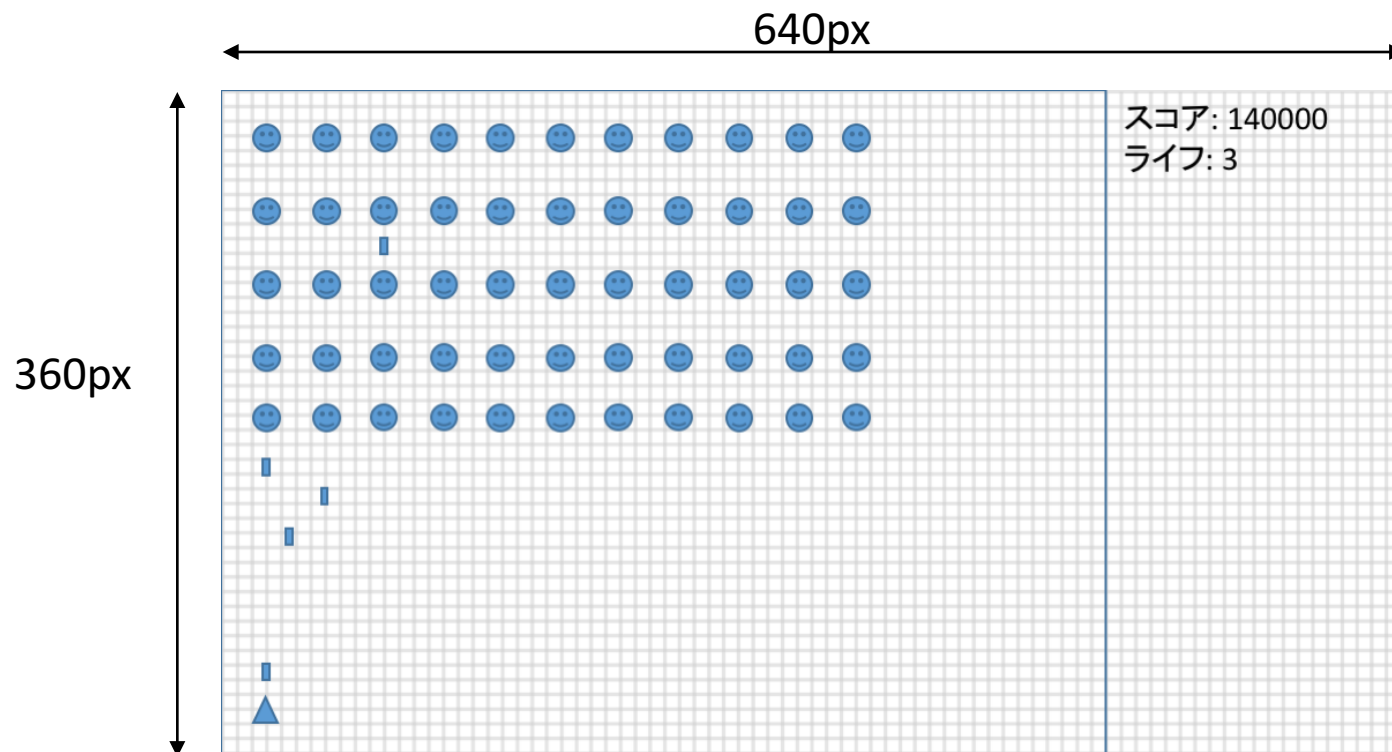
# シューティング ゲーム

---

# 企画:1

---

## ○画面のラフ・スケッチ



# 企画:2

---

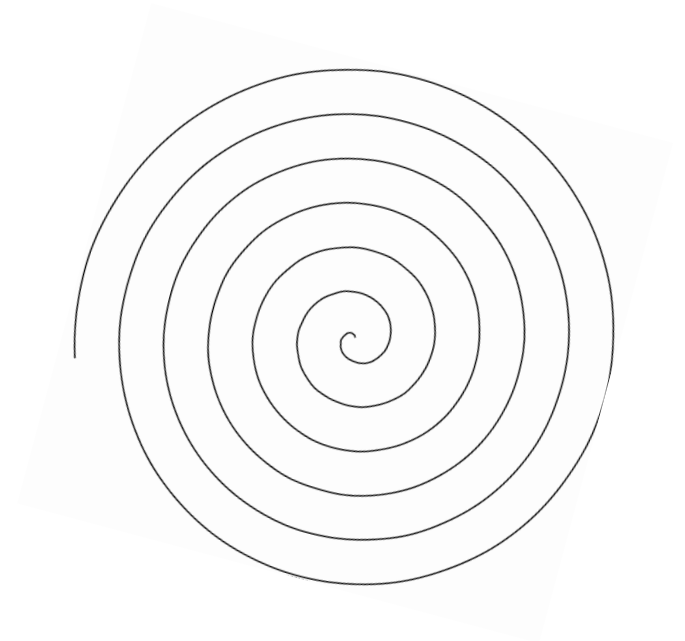
- ゲームのメカニクス(機構)
  - 要素
    - キャノン(自機)、レーザー(自弾)、エイリアン(敵機)、爆弾(敵弾)
  - 自機と敵機で撃ち合う
    - 破壊される組み合わせ
      - レーザーとエイリアン(得点)、爆弾とキャノン(ライフが減る)、キャノンとエイリアン(ライフが減る)
- ゲームの終了
  - ライフ(残機数)が0
  - エイリアン(敵機)をすべて破壊

# スパイラル開発

※spiral らせん

## ◦機能を少しずつ作る

- スプライト: もぐらたたきをベースに
- ステージ
- キャノン
- レーザー
- エイリアン: 1つ→複数
- 当たり判定: レーザーとエイリアン、キャノンとエイリアン
- ゲームの終了
- 爆弾
- 当たり判定: キャノンと爆弾



# 配列リスト

---

要素の個数を変えられる配列

# 配列の欠点

- 要素の数が決まっている
  - `int[] test = new int[10]` → 10個までしか入れられない
- 何個分の値を入れたかを調べるのが面倒

10	2	-1	8	2	0	0	0	0	0
----	---	----	---	---	---	---	---	---	---

↑  
ここから後ろが未使用だが...区別がつかない

- 値を削除すると、虫食い状に穴が空く

0	2	0	8	2	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

どこが未使用なのか、もはやわからない

# Javaのコレクションフレームワーク

---

- 配列よりも高度なデータ構造
  - List: 要素の個数が可変の配列のようなもの
    - ArrayListなど
  - Set: 同じモノを複数入れることができない
    - TreeSetなど
  - Map: 「キー」と「値」の組で要素を保存する
    - HashMapなど

# ArrayListの使い方

<https://docs.oracle.com/javase/jp/8/docs/api/java/util/ArrayList.html>

## ○ 宣言

```
ArrayList<型> 名前 = new ArrayList<型>();
```

## ○ メソッド

size()	リストに含まれている要素の数を返す
add(オブジェクト)	リストの最後にオブジェクトを追加する
get(添字)	添字に対応したオブジェクトを返す
set(添字, オブジェクト)	添字に対応したオブジェクトの内容を更新する
remove(オブジェクト)	オブジェクトを削除する