

SIVA

2D részecske rendszer

[specifikáció](#)

[dokumentáció](#)

[forráskód](#)

[fólia](#)

Engi Erik



Milyen céllal készült?

szoftverlabor házi
későbbi felhasználás

követelmények:
bővíthető
látványos

SDL vagy OpenGL?

könnyen használható

2D-s

lassú

gyors

2D-hez is jó

összetettebb

használat

SIVA

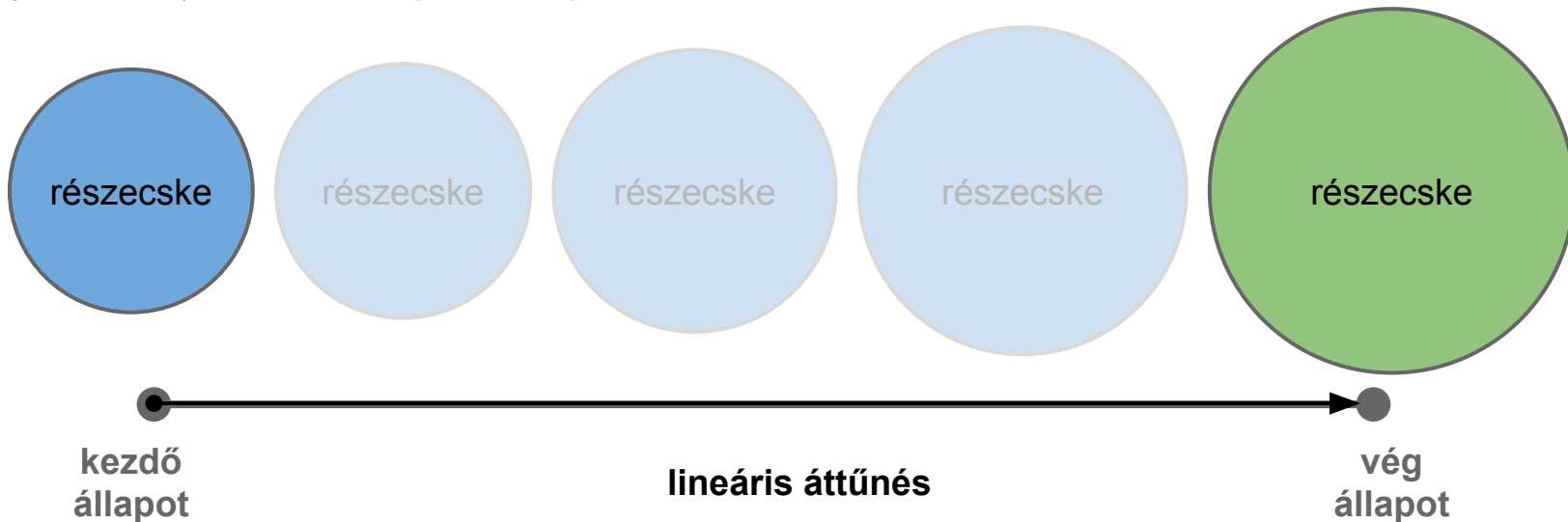
demózás következik

Mi adta az ötlet a működéshez?



Részecske és animáció

```
typedef struct {  
    int duration;  
    int preDuration;  
    long long startTime;    typedef struct {  
    char *type;             char *name;  
    int repeat;            int zIndex;  
    int reverse;           State state;  
    State start;          Animation animation;  
    State end;            Emitter emitter;  
} Animation;              } Particle;
```



Részecskék állapot paramétere

```
typedef struct {  
    double r, g, b, a;  
} Color;
```

```
typedef struct {  
    int size;  
    Color color;  
} Outline;
```

```
typedef struct {  
    double x, y;  
} Point;
```

```
typedef Point Vector;
```

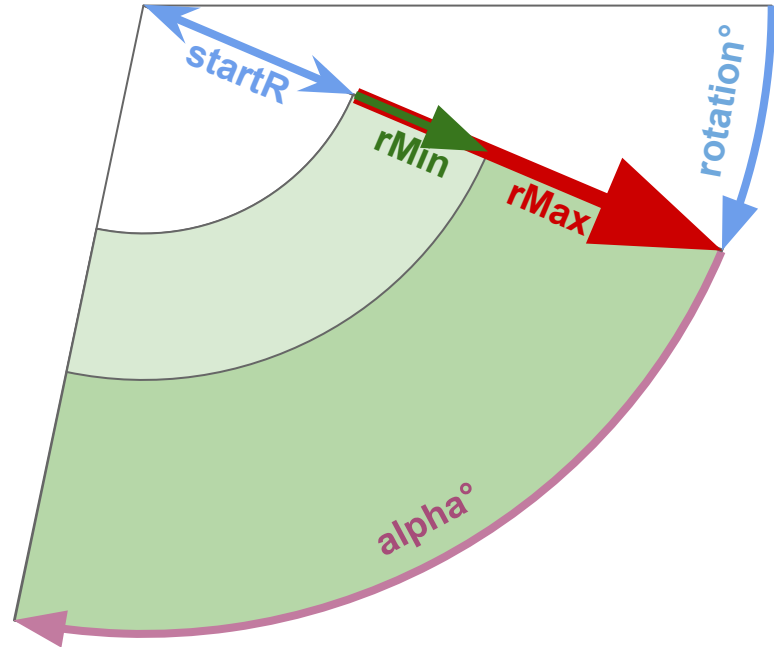
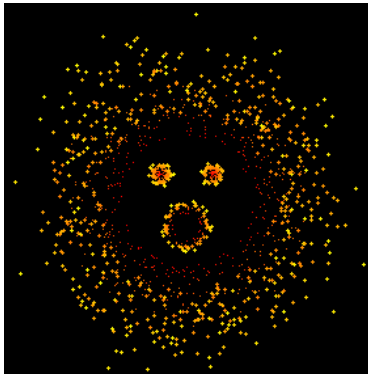
```
typedef struct {  
    Outline outline;  
    Color color;  
} Texture;
```

```
typedef struct {  
    Point p;  
    Vector v;  
    double size, opacity, rotation;  
    Texture texture;  
} State;
```

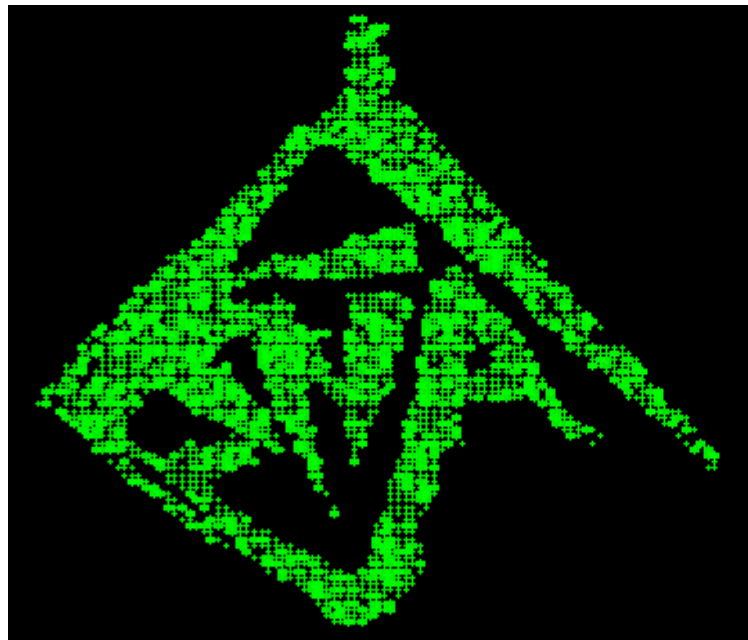
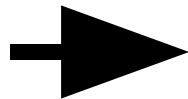


Részecske rendszerek “kilökődők”

```
typedef struct {  
    char *name;  
    int preDuration;  
    int reStart;  
    char *type;  
    int db;  
    Point p;  
    int rotation, alpha, startR, rMin, rMax;  
} Emitter;
```



Kép alapú kirajzolás

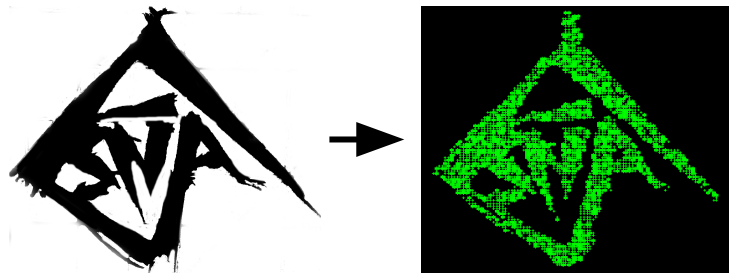


Kép alapú kirajzolás

```
void addEmitterGroup(char *img, double scaleX, double scaleY, double density, Emitter e) {
    SDL_Surface *imgSurface = IMG_Load(img);

    for(int i = 1; i < imgSurface->w; i += (int) (imgSurface->w * density < 1 ? 1 : imgSurface->w * density))
        for(int j = 1; j < imgSurface->h; j += (int) (imgSurface->h * density < 1 ? 1 : imgSurface->h * density)) {
            Color c = getColor(imgSurface, i, j);
            if (c.r == 0 && c.g == 0 && c.b == 0) {
                Emitter tmpE = e;
                tmpE.p.x += i * imgSurface->w * scaleX / imgSurface->w - imgSurface->w * scaleX / 2;
                tmpE.p.y += j * imgSurface->h * scaleY / imgSurface->h - imgSurface->h * scaleY / 2;
                addEmitter(tmpE);
            }
        }

    SDL_FreeSurface(imgSurface);
}
```



SIVA

kísérletezés következik

SIVA

2D részecske rendszer

[specifikáció](#)

[dokumentáció](#)

[forráskód](#)

[fólia](#)

Engi Erik

