Suprannua Engine Architecture Document

1. Introduction

Suprannua Engine is a 2D platformer oriented game framework for compiling simple Windows games or visualisations for algorithms. The hallmark and namesake of this engine is a superannuated design where the visuals are minimally done with legacy OpenGL, while the architecture is structured around a procedural runtime with the C programming language.

This was a first time project that was designed beyond basic C programming exercises. Therefore, it was created with a limited knowledge of standard programming practices or even a solid plan of the architecture before having it implemented. It has since been refined to just get the components working as intended while being lenient on the coding style and original architecture (use of global variables, externs, etc).

2. Core

2.1 Data structures

The game objects are Polygons, Blocks, Texts and the Camera. All game objects are instances of type defined structs. They all contain the Vertex struct to represent x and y positions against a world map. The world map and rectangular objects like the Camera and Blocks all contain the Rect struct to represent width and height.
The world map x coordinate, from 0 to the maximum size, corresponds with the left to right. The y coordinate from 0 to maximum size corresponds with down to up.

```
typedef struct
{
    Property properties;
    Vertex vertices[MAX_POLYGON_SIDES];
    Vertex centre;
    double radius;
}RegularPolygon;

typedef struct
{
    Property properties;
    Vertex vertices[4];
    Vertex centre;
    Rect dimensions;
}Block;

typedef struct
{
    Vertex textPin;
    char textContent[128];
    unsigned char colour[4];
    unsigned char classification;
}Text;
```

Polygons and Blocks are the only objects that contain the Properties struct, which describes the object's;

- •Classification (Whether it is a background, foreground, entity, platform, etc)
- •Colour (Including alpha)
- •Edges (Geometric sides)
- •Angle (Only really applicable to Polygons)
- •BouncePercentage (Elasticity)
- •Mass
- •xVelocity
- •yVelocity

Text is different where in addition to position, it stores a string with colour and either a entity or heads up display classification (to pin to the world map or camera, respectively).

```
typedef struct
{
    unsigned char classification;
    unsigned char colour[4];
    int edges;
    double angle; //for rotation
    double bouncePercentage;
    double mass;
    double xVelocity;
    double yVelocity;
}Property;
```

2.2 Colour palette

There are 21 colours supported by Suprannua that range from black, white, rainbow colours and additional secondary colours with different shades. These values are stored in globally declared, hard coded arrays of unsigned chars:

```
unsigned char black[3] = { 0,0,0 };
unsigned char white[3] = { 255,255,255 };
unsigned char red[3] = { 255,0,0 };
unsigned char green[3] = { 0,255,0 };
unsigned char blue[3] = { 0,0,255 };
unsigned char orange[3] = { 255,165,0 };
unsigned char yellow[3] = { 255,255,0 };
unsigned char violet[3] = { 191,5,247 };
unsigned char purple[3] = { 128,0,128 };
unsigned char brown[3] = { 150,75,0 };
unsigned char skyBlue[3] = { 135,206,235 };
unsigned char gold[3] = { 204,153,0 };
unsigned char seaGreen[3] = { 46,139,87 };
unsigned char pink[3] = { 255,192,203 };
unsigned char grey[3] = { 128,128,128 };
unsigned char darkRed[3] = { 192,0,0 };
unsigned char darkGreen[3] = { 0,128,0 };
unsigned char darkBlue[3] = { 0,0,128 };
unsigned char darkBrown[3] = { 80,40,0 };
unsigned char magenta[3] = { 255,0,228 };
unsigned char darkGrey[3] = { 32,32,32 };
```

The alpha values for the game objects are defined by the editor and text modules of the engine. By default the object is given a full 255 value so that they appear opaque. In the coming chapters, changing alpha values and transparency will be explained.

2.3 States
2.4 Entry point

3. FreeGLUT API
4. Game Loop
5. Modules

5.1 2D Audio
5.2 2D Camera
5.3 2D Renderer
5.4 AI
5.5 Editor
5.6 Events
5.7 Geometry
5.8 Input
5.9 Physics
5.10 Text

6. Game