Floor Plan Language (FPL)
Project Proposal

Xinwei Zhang (xz2663) Manager
Chih-Hung Lu (cl3519) Language Guru
Dondong She (ds3619) System Architect
Yipeng Zhou (yz3169) Tester

I. Introduction

Nowadays, AutoCAD has been a dominated professional software to draw house floor plans. However, AutoCAD is neither easy nor free to use. For someone who wants to decorate the house by himself, it takes hundreds of dollars to buy AutoCAD and dozens of hours to watch AutoCAD tutorials. To avoid this painful experience, our group intends to develop a user-friendly programming language called Floor Plan Language (FPL), which is specifically designed for drawing floor plans.

As we doing research on how AutoCAD works, we find that the basic idea is to build the framework of the floor plan, which is the wall. So, we design our language to be a C-like language. Users can easily and precisely define the position, size, and color of all these elements of the floor plan with FPL. With built-in control flow such as "for", and "if-else", users can easily generate dozens of furniture without repetitive operations. For professional floor plan designers, FPL can be used to generate hundreds of floor plans with just a few code modification. In the following sample code, we show a simple example of how FPL works.

From the high-level view, we will implement a library fplComponent to render all types of our components like a wall, chair, desk,...etc. In the library fplComponent, we will invoke the APIs from OpenGL to do rendering job. Moreover, our FPL compiler will parse our FPL and generate the corresponding object file, which contains function call invoking APIs provided by fplComponent. At the final stage, all the object file (*.o), fplComponent (*.so/*.a), and OpenGL (*.so/*.a) will be linked together and compiled as an executable file FPLRender. Users can execute this FPLRender to render their floor plan graph.

II. Syntax

A. Primitive data type

int	a 32-bit unsigned integer
char	a single ASCII character
bool	a boolean variable (True and False)
string	a null-terminated sequence of characters
dirc	a constant value representing direction (TOP, BOTTOM, LEFT, RIGHT)

B. Built-in data types

1. Wall: define a wall with two coordinates
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
2. Desk: define a desk with two coordinates
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
3. Chair: define a chair with a coordinate, size and direction
    a) int X
    b) int Y
    c) int Size
    d) dirc Direction
4. Bed: define a bed with two coordinates
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
5. Door: define a door with two coordinates and direction
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
    e) dirc direction

6. Window: define a window with two coordinates and width
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
7. Stair: define a stair with two coordinates, len and direction
    a) int X1
    b) int Y1
    c) int X2
    d) int Y2
    e) dirc Direction

C. Function prototype

FPL is a c-like language, providing simple function declaration and definition.

1) Define a function:
```
fun returnType funName(arg1, arg2, arg3)
{
        ....
}
```
2) Call a function:
```
funName(arg1, arg2, arg3);
```

D. Control flow

FPL support basic control flow features including for loop and if condition.

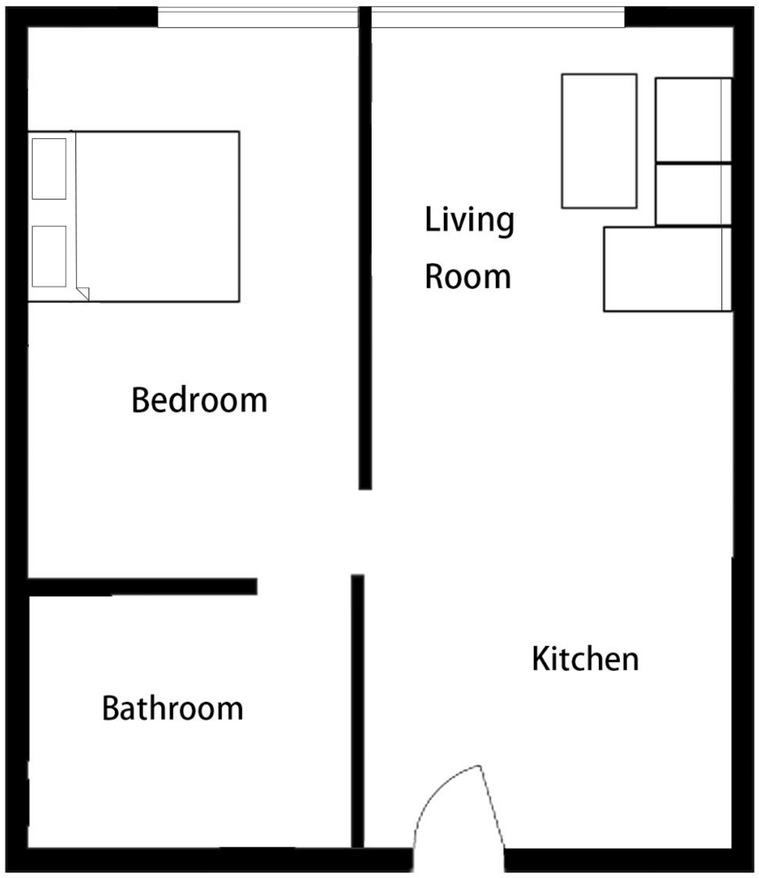
1) For loop statement:
```
for (expr)
{
     …
}
```
2) If conditional statement
```
If (expr)
{
   …
}
```

III. Sample code

This sample code represents a 1b1b apartment, which is 10-feet wide, and 20-feet long. And the corresponding rendered graph is on the section IV.

```
int main(){
        Wall topWall = Wall(0, 0, 10, 1);
        Wall leftWall = Wall(0, 0, 1, 20);
        Wall rightWall = Wall(10, 0, 11, 20);
        Wall bottomWall = Wall(0, 20, 10, 21);
        Wall middleWall0 = Wall(5, 0, 6, 10);
        Wall middleWall1 = Wall(5, 15, 6, 20);
        Wall bedBathWall = Wall(0, 15, 8, 16);

        Door door = Door(17, 20, 19, 21, BOTTOM);

        Bed bed = Bed(1, 2, 5, 6, RIGHT);

        // the two small chairs
        for(int i = 0; i < 1; ++i){
                Chair chair = Chair(18, 2*i + 2, 2, LEFT);
        }

        // the biggest chair
        Chair chair = Chair(18, 6, 4, LEFT);

        Desk desk = Desk(13, 2, 16, 6);

        Window window0 = Window(5, 0, 10, 1);
        Window window1 = Window(11, 0, 16, 1);

        Return 0;
}
```

IV. Sample graph


Living
Room
Bedroom
Kitchen
Bathroom