Timothy Pranoto  
38964311  
ICS161 HW5

Sound/ Voice

Description: I personally think that audio is one of the key components that should be added to sprite class. I will make 2 type of audio functions that can be added in the sprite class, background sound, which will be played throughout; sprite's voice, which will be played when that specific sprite appear in the window.

Design issue: Perhaps making 2 functions of audio can make a confusion and also redundant. Maybe I can make a function for background sound and then called it in the sprite's voice function.

Added to Sprite.h:

//for background sound  
//take the bgSound and play it with the provided volume  
void playBackgroundSound(SDL\_AudioSpec\* bgSound, int volume);

//for sprite's voice  
//when the sequenceName called the spSound played.  
void sequenceSound(std::string sequenceName, SDL\_AudioSpec\* spSound);

Game Programmer's Guide: Need to initialize audio in main function

sprite1->playBackgroundSound(bgSound, volume);

sprite1->sequenceSound("wave",spSound);

Z-index

Description: z-index is used to specify the order of the texture, which positioned the textures in the z axis. SDL did not provide z-index function, so I will do the z-index by making a vector to hold the textures and their position in z-axis. By doing this, I can specify which texture goes first in the window.

Design Issue: I think this can work, but it will not be efficient. Because to make it work, we need to have all the textures available first and put it into the vector. Also in order for this to work, I need to edit add another show method. And we need to take care about empty index in the vector, because we can skip indexes in the vector.

//Z-index  
//specify the z-index and put it in the vector  
void putZIndex(SDL\_Texture \* texture, int zindex);

//Show method for z-index  
//forloop the vector with renderCopy and renderPresent  
//dont need any parameter in it  
void show();

Game Programmer's Guide:

sprite1->putZindex(texture1, 1);  
sprite1->putZindex(texture2, 2);

sprite1->show();