Leon Matsuki

ICS 161 Game Engine Lab

3/4/15

Professor Frost

HW 8

Event Handler/Dispatcher:

The event handler and the dispatcher are completely different from the previous implementation of the event queue. The custom event queue was scrapped, and the new event handler is used instead. These features aim to decouple events in the game and allow the game programmer more control when deciding when and where the events are handled.

Design issues and open questions:

While these might be sufficient for simple games, adding an event manager to manage queues and passing the event object with a pointer using a class instead of a struct may expand the game programmer's options while also increasing efficiency.

Header section:

```
struct Event
{
	int Type;
	int arg1, arg2;
};

class IEventHandler;

class EventDispatcher
{
public:
	static EventDispatcher * Get();

private:
	EventDispatcher() : _deviceList(0) { ; }
	~EventDispatcher() { ; }
	static EventDispatcher * _instanceOf;
public:
	void RegisterHandler(IEventHandler * device);
	void SendEvent(int eventType, int arg1 = 0, int arg2 = 0);
private:
	IEventHandler * _deviceList;
};
```

Programmer's guide:

First, the class that would like to receive events needs to inherit from the IEventHandler:

```
class Sprite : public IEventHandler
```

Then the EventHandler function needs to be implemented:

```
// To handle events in the Sprite class
	void EventHandler(const Event &e);
```

```
void Sprite::EventHandler(const Event &e)
{
	switch (e->Type)
	{
	case E_KEYBOARD_DOWNPRESS:
		std::cout << "Move down" << std::endl;
		break;
	case E_KEYBOARD_UPPRESS:
		std::cout << "Move up" << std::endl;
		currY += -1;
		break;
	case E_KEYBOARD_LEFTPRESS:
		std::cout << "Move left" << std::endl;
		break;
	case E_KEYBOARD_RIGHTPRESS:
		std::cout << "Move right" << std::endl;
		break;
	}
}
```

Then finally send the event in the appropriate places:

```
else if (e.key.keysym.sym == SDLK_UP)
{
	//sprite1->movey(-1);
	direction = "fall up";
	EventDispatcher::Get()>SendEvent(E_KEYBOARD_UPPRESS);
}
```