Game Programming with LUA

afewhee@gmail.com

- File I/O
- Frame Work

1 File I/O

- 자주 사용되는 파일 입출력함수
  - open(): 파일 입출력 핸들 얻음. "r, "w" 모드 필요
  - close(): 파일 핸들 닫기
  - read(): 파일을 읽어 들임. *all을 사용하면 전부 읽음
  - write(): 파일에 쓰기
  - lines(): line 단위로 읽기

```
-- 텍스트 파일 읽기 연습

fr= io.open ("test.txt", "r")              -- 파일 핸들 얻기

while true do
    local line = fr:read("*line")          -- 라인단위로 파일 내용 읽기

    if nil == line then  break end         -- 읽을 데이터가 없으면  read는 nil 반환

    print(line)                            -- 라인 내용 출력
end

fr:close()                                 -- 파일 핸들 반환
```

```
-- 텍스트 파일 읽기/쓰기 연습

fr = io.open ("unit_lvl.csv", "r")          -- 원시 파일을 읽기 모드("r")로 열기
fw = io.open ("unit_lvl.txt", "w")          -- 대상 파일을 쓰기 모드("w")로 열기

while true do
   local line = fr:read("*line")            -- 한 줄 읽기

   if nil == line then break end

   print(line, "₩n")

   fw:write(line, "₩n")                     -- 원시 파일을 읽기 모드("r")로 열기

end

fr:close()
fw:close()
```

- 패턴 문자 분류
  - 매직 문자: characters ^$()%.[]*+-?
  - x: magic 문자가 아닌 문자
  - .: 모든 문자
  - %a: 알파벳 조합, 숫자
  - %c: 제어 문자
  - %d: 숫자
  - %l: 소문자
  - %p: 구두점 문자
  - %s: 공백 문자
  - %u: 대문자
  - %w: 영문자 소문자
  - %x: 16진수

  - '%' 다음의 문자가 대문자이면 반대의 경우
  - %A: 알파벳과 숫자가 아닌 문자

- 마법 문자 표현: %사용
  - % => %%
  - . => %.

- 문자집합:
  - 문자 분류로 표현할 수 없는 패턴 표현
  - 이식성, 효율성 높음

  - [set]: 문자 집합
  - [^set] : 문자 집합여집합
  - %d = [0-9]
  - %x = [0-9a-fA-F]
  - %S = [^%s]
  - 8진수: [01234567] 또는 [0-7]
  - 알파벳, '_', 숫자: [a-zA-Z_0-9]

- 반복 패턴:
  - +: 1번 또는 그 이상의 반복
  - *: 0번 또는 그 이상의 반복(긴 조합)
  - -: 또 다른 0 또는 그 이상의 반복(짧은 조합)
  - ?: 0 또는 1번

- 반복 패턴 예
  - [a-zA-Z_][a-zA-Z_0-9]*: 시작은 반드시 알파벳과 '_'로 시작하고 숫자, 알파벳, _ 들을 0번 이상 반복 사용 => C언어 함수 또는 변수 이름
  - /%*.-%*/ : '/*' 과 '*/' 사이의 모든 문자 0 번이상 반복 => C 주석

```
fr = io.open ("unit_lvl.csv", "r")          -- 파일 핸들 얻기

while true do
   local line = fr:read("*line")            -- 한 줄 읽기

   if nil == line then break end

   local v ={}; local i = 1

   for w in string.gmatch(line,  "([^,^¥t]+)") do
     v[i] = w; i = i + 1
   end
end

fr:close()
```

```
fr = io.open ("unit_lvl.csv", "r")

while true do
    local line = fr:read("*line")
    if nil == line then break end

    local v; local comma = 0 ; local n1 = 1; local n2 = 1

    while true do
      comma = string.find(line, "[,₩t]", comma+1)

      if comma == nil then
                  n2 = #line; s = string.sub(line, n1, n2)
                  break;
      end

      n2 = comma -1
      s = string.sub(line, n1, n2)
      n1 = comma +1
    end
end

fr:close()
```

2 Frame Work

- 분업화
- 표준화, 규격화
- 기획이 수시로 변경되는 프로젝트의 수행에 유리
- 유지 보수에 편리
- 하향식 설계

- 건축물과 유사
  - 큰 단위로 분류
  - 추상 설계 (Abstracted Architecture) ➔ 구체화(Realize) 구현

- 하향식 설계
  - Common Interface
  - 위상(Game Phase) 설정
  - Game Data/ Rendering Data 분리
  - 구현할 파일 분할

- 건축물과 유사
  - 큰 단위로 분류
  - 추상 설계 (Abstracted Architecture) ➔ 구체화(Realize) 구현

- 하향식 설계
  - Common Interface
  - 위상(Game Phase) 설정
  - Game Data/ Rendering Data 분리
  - 구현할 파일의 분리