한국심리학회지 : 실험
The Korean Journal of Experimental Psychology
2005, Vol. 17, No. 4, 529-540

<자료>

감각과 지각 가상실습실 : 자바 에플릿을 이용한 시간제어

곽 호 완

경북대학교 심리학과

본 논문은 자바 에플릿을 이용하여 인터넷 상에서 구동되는 감각과 지각 심리학 가상실험실습 프로그램 세트를 소개하였다. 세부적으로, 자극의 제시시간 및 자극간 시간간격의 조작은 실험실습에서 특히 중요 관건이 되므로, 에플릿 상에서 이를 어떻게 제어하는지에 대해 논의하였다. 부가하여, 시각검색과 같이 실습자의 반응을 처리하는 실험과제에서 반응시간을 측정하는 방법에 관하여 논의하였다.

주요어: 감각, 지각, 자바 에플릿, 가상 실험실습실, 시간제어

교신저자 : 곽호완, (702-701) 대구광역시 북구 산격동, 경북대학교 사회과학대학 심리학과
E-mail: kwak@mail.knu.ac.kr

감각과 지각 심리학은 자극의 형태, 색, 움직임 등을 유기체가 어떻게 처리하는지를 연구하는 분야이므로, 다양한 지각 현상에 대한 관찰과 실험실습이 필수적이다. 무엇보다도 우리의 지각과정의 특징을 잘 드러내는 여러 가지 지각현상들을 관찰하다 보면 아주 흥미로운 사실들을 깨닫게 된다. 특히, 우리의 지각체계의 특징을 잘 드러내는 예들 중에는 다양한 착시현상, 잔상, 대비효과 등이 있다. 이러한 지각현상들을 관찰하는 과정에서 우리 자신의 지각과정을 이해하게 된다.

예전에는 착시현상을 보여주는 모형기구나, 색 혼합 등을 보여주는 모터가 달린 색채혼합기 등을 사용하여 지각현상에 대한 동적인 실습을 행하였다. 이러한 기기들은 값도 비싸고 개별적인 실험실습을 하기에는 불편한 점이 많았고, 다양한 조작을 행하기 곤란하였다. 최근에는 컬러모니터를 장착한 컴퓨터로 구동되는 여러 실험실습 프로그램들이 선보였다[1]. 이 컴퓨터 실습프로그램들은 다양한 색채 및 역동적인 동영상들의 제공이 용이하여 빈번히 사용되게 되었지만, 사용자들의 다양한 컴퓨터 사양에 맞추어 설치되어야 하는 단점이 있었다.

반면에 여기에 소개되는 실험실습프로그램이 기초한 자바 에플릿은 웹 브라우저에서 구동되는 실행프로그램이므로 사용자는 별도의 프로그램을 설치하지 않아도 실행이 가능하다. 또한 사용자의 키 입력이나 마우스 입력 조작에 의해 프로그램의 제시환경과 상호작용이 용이하다. 자바는 선마이크로시스템즈사(http://www.sun.com)에서 1995년 개발한 프로그래밍 언어로서 개발환경과 적용시스템들이 독립적으로 구현될 수 있도록 만들어졌다. 자바 에플릿(Java Applet)은 HTML문서에서 참조되고 로딩되며, 사용자 측(client-side) 컴퓨터의 웹브라우저 상에서 자바프로그램이 구동되어 사용자와 동적인 상호작용이 가능하도록 함으로써 서버의 부하 및 보안문제를 해결하였다. 사용자와 프로그램간의 동적인 상호작용을 가능하게 하는 프로그램 환경으로는 자바에플렛 외에도 Perl CGI나 자바스크립트 등이 있다. CGI는 서버에서 사용자의 반응을 받아서 처리할 수 있도록 해주는 서버 측(server-side) 프로그램으로서 폼 입력이나 설문조사 등에 사용되는데, 인터넷을 통한 정보교환이 일어나므로 실시간제어가 곤란하고, 그래픽 등의 사용에 제한이 있다. 자바스크립트는 HTML문서 안에서 동적인 화면구성이 가능하도록 만들어진 간이 프로그램언어인데, 그 기능이 제한적이어서 다양한 동적 실습프로그램의 제작에는 제한적이다.

앞서 언급한 대로, 심리학과 관련된 웹기반 실습 또는 검사 프로그램들은 사용자와 동적으로 상호작용을 하도록 만들어져 있다. 그러한 프로그램들은 여기서 언급하는 지각심리학 실습 외에도 웹을 통한 온라인 성격검사[2]나, 통계프로그램[3], 그리고 온라인 사전프로그램[4] 등이 있다. 이러한 프로그램들은 사용자의 입력 또는 선택 자료를 바탕으로 성격프로파일을 만든다든지 통계처리의 결과를 보여준다든

1) 예를 들어 E. B. Goldstein(1996)의 Sensation & Perception 보조교재로서 'Exploring Perception CD rom', PshchNow, 그리고, Psyk_Trek 등이 있다.

2) 예: ASP를 이용한 온라인 성격검사 사이트 http://psychtest.educyber.org/

3) 예: 자바스크립트를 이용한 심리 통계 실습실 http://bh.knu.ac.kr/~kwak/psy_method/stat/

4) 예: Perl CGI 및 자바스크립트를 이용한 온라인 실험심리학 사전 http://www.cogpsych.org/dict/

지, 입력한 용어에 대한 학문적 정의와 관련 어 등을 보여주는데, 제시시간이나 반응시간 등을 측정하는 목적이라기보다는 빈도계산 및 반응유형을 처리하면 되기 때문에 구현하는 방법이 비교적 용이하다. 반면에 지각심리 실습 프로그램의 경우 화면의 색상 및 크기가 정확히 제시되어야 하고, 밀리 초 단위의 정확성으로 자극을 제시하며, 제시된 자극에 대한 반응시간을 정확히 측정해야 하므로 웹상에서 정확히 구현하기는 쉽지 않다. 실제로 자바 에플릿은 사용자측 컴퓨터에서 직접 구동되므로 온라인 상의 비동시성의 문제를 극복하기는 하지만, 현 윈도우즈 시스템의 다중작업 환경 및 사용자 컴퓨터의 다양성으로 인해 그 엄밀성이 컴퓨터 자체 실험프로그램에 비해 약간 떨어진다고 할 수 있다. 이러한 약점에도 불구하고, 자바 에플릿이 지각현상을 실습하는 정도의 정확성은 가지며, 실제로 외국의 실험심리학자 사이트[5]에서도 몇몇의 자바에플릿 실습프로그램이 소개되어 있으므로 그 효용성이 인정되고 있다.

본 연구는 자바 에플릿을 이용한 지각실습 프로그램의 구성에서 중점적으로 다루어야 할 부분 중 특히 시간제어에 관해서 그 구현방법 및 제한점에 관하여 논의하고자 한다. 대부분의 역동적인 지각심리학 실습의 경우에 자극화면의 순차적인 제시시간이나, 중다 화면의 시간적 배열 등에 따라 경험되는 지각현상이 매우 달라진다. 특히, 시각검색과제나 빠른 순차제시과제 (RSVP, rapid serial visual presentation) 등에서는 제시화면의 노출시간 뿐만 아니라, 피험자의 반응을 밀리 초(msec) 단위로 측정해야 한다. 이러한 측면을 제대로 구현하기 위해서는 첫째, 컴퓨터 화면으로 제시되는 자극의 물리적 측면뿐만 아니라, 컴퓨터 환경이 가지는 소프트웨어적인 측면을 잘 이해할 필요가 있고, 여러 가지 자극사상이 제시되는 실습프로그램의 흐름도(flow chart)를 잘 제어할 필요가 있다.

실습 프로그램의 구성과 알고리즘

하드웨어의 구성 통상적으로 사용되는 IBM 호환 펜티엄급 컴퓨터와 17인치 LCD 모니터가 본 프로그램의 개발에 사용되었다. 자바 에플릿의 개발 및 구동에 필요한 컴퓨터는 32비트 컬러 및 1024x768의 해상도를 제공하는 컬러모니터와 펜티엄급 컴퓨터 정도이며, 사실 사용자들의 다양한 컴퓨터 환경을 고려하면 지나치게 고급사양을 요구하지 않는 것이 좋다.

소프트웨어의 구성 웹에서 무상 제공되고 있는 선 마이크로시스템즈사의 자바 컴파일러 버전 1.2를 사용하여 프로그램들을 개발하였다. 본 가상실험실습실은 86개의 개별 에플릿 프로그램으로 구성하여 저자의 홈페이지에 구현하였다(부록2 참고). 현재 이 프로그램들은 저자의 감각과 지각 가상강좌에 활용되고 있다[6]. 통상적으로 16주 강의기간 동안 매주 5개 이상의 실험실습 프로그램이 기용되어야 하므로 총 80-90개 정도의 프로그램이 구현되어야 최적의 멀티미디어 실습실이 가동될 수 있다. 이 정도의 프로그램이 완성되기 위해서는 감각·지각 심리학의 학문내용에 친숙한

5) 예: Donald D. Hoffman의 사이트 http://www.cogsci.uci.edu/personnel/hoffman/hoffman.html

6) 지각심리 실습 프로그램을 시연하려면 개발자의 홈페이지 http://bh.knu.ac.kr/~kwak/vlab/vlab_demo.html 참고.

사람들과 프로그래머들이 팀을 이루어 수행되어야 한다. 이를 위해서 본 저자는 전자공학과 영상처리 전공교수를 영입하여, 그 영상처리 실험실과 지각공학실험실 공동으로 프로그래밍 작업을 진행하였다. 부가하여, 영상처리 실험실에서 개발한 Water-marking 알고리듬은 그림파일 및 동영상 파일의 지적소유권 보호에 기여하므로 이 기술을 개발된 영상 파일에 부분적으로 적용하였다.

프로그램의 구조 약간의 다양성이 있지만 대부분의 실습 에플릿은 몇가지 구조로 나누어진다. 우선 프로그램 실행상의 측면에서 보면 사용자 입력부분, 프로그램 실행부분 및 결과제시부분이 그것이다. 사용자는 자극의 제시시간이나 자극간 간격 등을 결정하고, 시작버튼을 누르면 프로그램이 실행한다. 실행 후에 사용자는 지각현상의 변화를 보기 위해 색 또는 길이를 조정할 수 있다. 중지 또는 결과확인 버튼을 누르면 실행 후의 결과가 나온다.

프로그램 소스 내부의 구조는 약간 더 복잡하다. 대부분의 프로그램은 상수 및 변수 초기화 부분, 사용자 입력을 통한 변수변경 부분 (텍스트 박스, 버튼, 체크박스 등), 반응입출력 활성화 (버튼 및 마우스), 자극 그리기 및 제시부분, 반응측정 및 기록부분, 마지막으로 실험결과의 요약 및 제시부분 등이 있다. 본고에 소개된 가현운동 프로그램의 경우 무한루프로 자극이 교대로 제시되며, 사용자의 중지버튼 누르기를 백그라운드로 기다리게 된다. 따라서 사용자의 반응을 컴퓨터가 기다리는 도중에 자극을 계속적으로 제시할 수 있다. 다른 예로, 시각검색 에플릿의 경우 자극이 제시된 후 일단 자극화면은 종료되는데, 실제로는 백그라운드로 마우스 이벤트를 기다리고 있다가 마우스 누르기가 일어나면 반응시간을 계산·측정하게 된다.

프로그램의 실행 가상실험실에 인터넷으로 접속하면 몇 가지 메뉴방식을 선택하는 화면이 뜬다. 여기서 사용자의 선호에 따라 좌변 프레임메뉴 또는 노프레임 메뉴 등을 선택하게 된다. 그 다음 실험실습은 총 6개의 대주제로 나뉘어서 제공된다 (개요, 생리적 기초, 색채지각, 형태지각, 크기 및 깊이지각, 운동

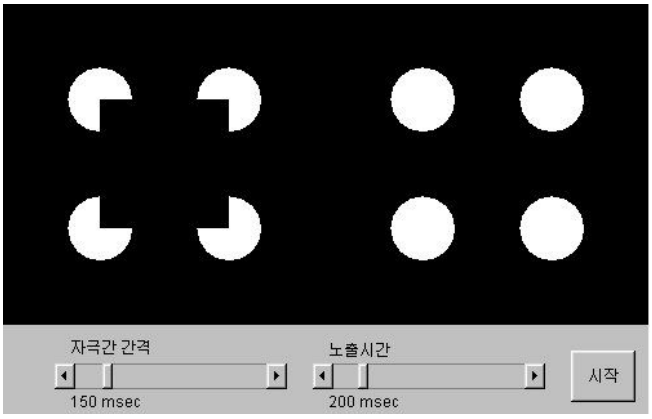

그림 1. 가상실험실의 실행 예 (착각적 윤곽의 가현운동)

지각). 그 대 주제 중 하나를 선택하여 그 중 하나의 실험실습프로그램으로 들어가면 먼저 프로그램에 대한 지각심리학적 설명 및 실행 방법이 제시된다. 그 후 실행하기 버튼을 누르면 자바 에플렛 프로그램이 실행된다[7]. 대부분의 프로그램들은 실험실습에 필요한 모수치들을 변경시키는 버튼이나 스크롤바 또는 텍스트 입력상자들을 가지고 있다. 예를 들어 그림1의 '착각적 윤곽의 가현운동' 실습에서, 자극의 노출시간 및 자극 간 시간간격[8](ISI)를 스크롤바를 통해 조작하고, 시작 버튼을 누르면 착각적 사각형이 화면의 양쪽에 교대로 제시된다. 이 실습에서 노출시간 및 자극간 시간간격의 조작이 중요한 것은 가현운동의 지각에서 이 두 가지 시간변인이 현상적으로 운동지각을 경험하거나, 아니면 개별적인 정지화면의 교대로만 나타나거나를 결정하기 때문이다(Goldstein, 1996).

시간제어 프로그램 자극화면의 정밀한 시간제어의 프로그래밍에서 중요한 측면은 특정 언어가 시간제어의 명령어를 기본함수로서 가지고 있느냐의 문제이다. 다행히 자바 에플릿에서는 멀티쓰레딩(multi-threading) 기능이 포함되어 프로그램 내부에서의 자원을 재할당할 수 있는 기능이 있어서 특정 프로세서를 정지 또는 중지시킬 수 있다. 예를 들어, 두 개의 자극화면을 각각의 노출시간 동안 제시하게 하면서 두 자극간의 공백화면(blank field)을 특정 자극간 시간간격으로 제시하고자 할 때 그림 2의 예와 같이 프로그램 될 수 있다 (전체 소스는 부록 참고).

여기에 제시된 부분 소스프로그램에서 볼 수 있듯이 각 화면이 제시되면 thr.sleep() 명령어에 의해 특정 시간동안 프로그램이 정지하게 된다. 이 경우 자극을 화면에 그리는 명령어가 실행되는 시간은 요즘의 컴퓨터 처리속도에서 보면 무시할 정도로 작지만[9], 실제로 화면에 자극이 모두 보이는 시간은 모니터의 재생(refresh) 속도 및 사용자 측 컴퓨터의 다중작업 상황에 따른다[10].

자극의 제시시간이나 시간차를 조작하는 것 이외에도 컴퓨터에 내장된 시계를 사용하는 것이 필요한 실습프로그램이 있는데 바로 피험자의 반응시간을 측정하는 것이다. 예를 들어 시각검색 과제 실험실습에서, 특정 화면을 제시한 후 피험자의 반응시간을 측정하려면 그림 3의 예와 같이 할 수 있다[11].

예에서 알 수 있듯이 System.currentTimeMillis() 명령어를 사용해서 각각의 시각을 밀리초 단위로 두 번의 시점에 걸쳐 측정하여 그 차이

7) 사용자의 컴퓨터 환경에 따라 선마이크로시스템즈사에서 제공하는 업데이트된 자바 런타임 모듈을 설치할 필요가 있다. 업데이트되지 않은 컴퓨터의 경우 제대로 실행되지 않을 수 있다.

8) 자극간 시간간격(inter-stimulus interval) 대신에 자극제시시간차(SOA, stimulus-onset-asynchrony) 를 사용하기도 한다. 실제로, SOA는 ISI 와 노출시간의 합과 동일하다.

9) 저자의 계산으로는 대부분의 화면제시 명령어의 실행시간은 1 msec 이하였음.

10) 예컨대 모니터 주사율이 75Hz로 세팅되어 있다면 화면 전체가 새로 제시되는데 걸리는 시간은 13.3msec이다. 즉 이 경우 컴퓨터로 제시되는 화면의 시간 제어는 1000/75 이상의 정밀도를 가질 수 없다. 특히 LCD 모니터는 잔상이 남기 때문에 정밀도가 더 떨어진다. 그리고 정확한 시간제어 및 프로그램의 실행을 위해서 사용자 컴퓨터에서 현재 실행되는 여타의 프로그램 및 백그라운드 프로그램을 가능하면 종료시키는 것이 좋다.

11) 시각검색 에플릿의 소스는 http://bh.knu.ac.kr/~kwak/vlab/applets/Visual_Search/Visual_Search.java 참고. 본문에 제시된 소스는 단순화시킨 것임.

```
draw1();  // 첫번째 자극 제시
        try{
                Thread.sleep(duration);  // 첫번째 자극면을 duration msec
                                                동안 노출
        } catch(InterruptedException e){}; //중지버튼을 누르면 정지
draw3(); // 공백화면
        try{
                Thread.sleep(ISI);  // 두번째 자극이 제시되기 전에 ISI msec
                                                동안 공백화면제시
        } catch(InterruptedException e){};
draw2(); // 두번째 자극제시
        try{
                Thread.sleep(duration); // 두번째 자극면을 duration msec
                                                동안 노출
        } catch(InterruptedException e){};
draw3(); // 공백화면
        try{
                Thread.sleep(ISI);   // 첫번째 자극이 제시되기 전에 ISI msec
                                                동안 공백화면제시
        } catch(InterruptedException e){};
```

그림 2. 자극 제시시간 제어 프로그램 예

```
    //마우스 버튼 왼쪽/오른쪽 반응확인, 마우스 입력상태를 활성화시키지만, 백그라운드로
진행됨
    dPanel.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            if(dPanel.is_start==true){
                if (e.getModifiers() ==
MouseEvent.BUTTON1_MASK){
                    dPanel.response_yes();
                }else if(e.getModifiers() ==
MouseEvent.BUTTON3_MASK){
                    dPanel.response_no();
                }
            }
        }
    });

    Thread.sleep(iti); // 시행간 공백화면
    draw_FIX();  // 응시화면 제시
    Thread.sleep(fix_dur); //응시화면 제시시간
    draw_BLANK();  // 빈화면 제시
    Thread.sleep(blank_dur); //빈화면 제시시간

    draw1();  // 자극 제시

    old_timestamp=System.currentTimeMillis(); // 자극제시 당시의 시각 기록

    // 자극제시 후 화면에 자극이 지속되면서 마우스 반응을 기다리고 있다가, 마우스
       버튼을 누르면 아래 루틴이 활성화하면서 반응시간을 측정하게 됨.
    class DrawPanel extends Panel{
        public void response_yes(){
            rt_data[trial_no]=System.currentTimeMillis()-old_timestamp; //반응시간 측
정
        }
    }
```

그림 3. 반응시간 측정 프로그램 예

를 계산하면 피험자의 반응시간을 측정할 수 있다. 여기서도 주의할 점은 자극을 제시하기 전에 시각을 측정하는 것 보다 자극제시 명령 후에 시각을 측정하는 것이 더 정밀하다. 부가하여, 마우스의 두 버튼에 대한 반응을 따로 받도록 하면 선택반응에서 정확도를 측정할 수 있다.

부가해서 언급할 점은, 여기 소개된 자바에플릿 프로그램들은 프로그램의 구성으로 보면 자극의 제시시간이나 간격 등은 계열적인 순서에 의해 실행되지만, 반응을 받는 키보드나 마우스 입력은 백그라운드로 진행된다는 점이다. 다시 말해서, RSVP과제[12] 처럼 매우 많은 자극이 제시되면서 자극이 다 제시되기 전에 반응할 수 있도록 프로그램 될 수 있다는 점이다. 예를 들어 위의 소스에서 dPanel.addMouseListener() 함수가 바로 그 기능을 수행한다.

논 의

본고에 소개된 감각과 지각 가상실험실은 현재의 인터넷 시대에 적합한 수업보조 방법으로서의 의의가 있다. 실제로 유사한 컴퓨터상의 실험실습프로그램들이 시중에 나와 있다. 그러나, 대부분의 실험실습 프로그램들이 웹상에서 지원되는 것이 아니고 독자적(stand-alone) 프로그램이어서 복사 및 실행이 어렵고, 판권의 문제도 있다. 이에 반해 본 프로그램은 웹상에서 누구나 접속하여 실습할 수 있는 장점이 있다. 현재 구현된 것이 감각과 지각 현상 전체를 포괄하지는 못하지만 86개 정도로 비교적 풍부하며, 모든 프로그램이 소스코드와 함께 제공되므로 추후 누구라도 프로그램을 개선하고, 이를 토대로 유사한 다른 실습프로그램을 개발할 수도 있다.

본 프로그램은 사용자들의 다양한 요구에 부응하기 위해 별도의 코멘트 게시판을 만들어서 사용자들간 및 개발자-사용자간의 상호작용의 극대화에 노력하였다. 부가하여 개별 실습프로그램에 연계된 코멘트 프로그램을 제공하여 각 프로그램의 사용정보 및 내용정보의 교환을 할 수 있도록 하였다. 마지막으로, 자바스크립트 및 펄 CGI를 서버에 구현하여 위계적 메뉴의 구성 및 자동 인덱스 기능을 첨가하여 개발자의 개발환경을 향상시켰다.

본 가상실험실 프로그램이 앞으로 구현하고 해결하여야 할 문제점은 다음과 같다. 첫째, 여러 개발자들에 의하여 프로그래밍되었기 때문에, 각 프로그램마다 인터페이스 및 전체적 메뉴배치가 일관성이 결여되어 있다. 이 문제는 현재 수정 중에 있다. 둘째, 각 실험실습 프로그램의 실행 후 그 결과의 통계적 분석 및 해석에 관한 자동적 구현이 부족하다. 셋째, 원격실험프로그램의 장점을 살릴 필요가 있다. 예를 들어, 한 실험심리학자가 피험자를 구하기 어려운 상황에서 웹상에서 에플렛 실험을 수행한다고 하자. 이 경우 각 접속자의 실험실습 결과를 자동적으로 서버에서 추출하여 분석할 수 있다면 좋을 것이다. 이 문제는 선마이크로시스템즈사에서 자바를 개발할 때부터 대두된 문제였는데, 보안상 문제로서 사용자측의 정보가 서버로 함부로 넘어오지 못하도록 결정된 것이다[13].

12) http://bh.knu.ac.kr/~kwak/vlab/applets/RSVP/RSVP.java

13) 구체적으로, 보안문제 때문에 자바 에플릿 프로그램은 자료를 하드디스크 같은 저장매체에 기록하거나 읽을 수 없고, 서버로 자료를 자동전송할 수도 없다. 그러나 이 문제를 해결하는 방법을 강구

마지막으로, 이와 같은 에플릿 프로그램을 이용한 실험실습 프로그램 기법은 실험심리학의 범위 밖으로도 적용될 수 있다. 예를 들어 신경심리 평가 및 검사도구 개발이나, 질문지형 또는 투사형 심리검사 도구를 웹상에서 개발할 수 있다.

참고문헌

Reynolds, M. C., & Wooldridge. A. (1997). 알기쉬운 자바스크립트 활용. (김영훈 역). 서울: 정보문화사.

Harlan, D., Ó Foghlú, M., Doyle, P., Powers, S., & Healy, M. D. (1997). Perl 5 웹 프로그래밍 활용. (조성현, 전승철 역). 서울: 정보문화사.

GoldStein, E. B. (1996). *Sensation and Perception.* Pacific Grove, CA: Brooks/Cole Publishing Company.

Hoffman, D. D. (1998). *Visual Intelligence: How We Create What We See.* New York: W. W. Norton & Company, Inc.

IDO (2001). Java: Java2 SDK 1.3.1. (편저) 서울: 영진.Com.

Lemay, L., & Cadenhead, R. (1999). Java2. (곽용재, 손우상, 최현덕, 천영환 역). 서울: 인포북.

[웹 소스]

1. 감각과 지각 가상실험실: http://bh.knu.ac.kr/~kwak/
2. 자바: http://www.sun.com
3. ASP를 이용한 온라인 성격검사 사이트 http://psychtest.educyber.org/
4. 자바스크립트를 이용한 심리 통계 실습실: http://bh.knu.ac.kr/~kwak/psy_method/stat/
5. Perl CGI 및 자바스크립트를 이용한 온라인 실험심리학 사전 : http://www.cogpsych.org/dict/
6. Donald D. Hoffman의 사이트 http://www.cogsci.uci.edu/personnel/hoffman/hoffman.html

1차원고 접수: 2005. 11. 10
최종게재결정: 2005. 12. 26

할 수 있는데, 예를 들어, 결과를 사용자의 결정에 의해 전자우편으로 보내게 하는 방법이 있다.

한국심리학회지 : 실험
The Korean Journal of Experimental Psychology
2005, Vol. 17, No. 4, 529-540

<Data>

Virtual Lab for Sensation and Perception: Timing Control using Java Applet

Ho-Wan Kwak

Department of Psychology, Kyungpook National University

This study introduced a web-based virtual lab for sensation & perception using Java Applet. Specifically, since it is critical to manipulate exposure durations and inter-stimulus intervals for lab experimentations, we discussed how to control them in the Java programming. In addition, it was also discussed how the reaction time can be measured in the tasks such as the visual search paradigm.

Keywords: sensation, perception, java applet, virtual lab, timing control

부록 I

착각적 윤곽의 가현운동 자바 에플릿 소스

http://bh.knu.ac.kr/~kwak/vlab/applets/Apparent_Illusory_Contour_Motion/Apparent_Illusory_Contour_Motion.java

```
/* Apparent_Illusory_Contour_Motion.java */

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Apparent_Illusory_Contour_Motion extends Applet{
    DrawPanel dPanel=new DrawPanel();
    Label l1=new Label("자극간 간격");
    Label l2=new Label("노출시간");
    Label l3=new Label(""+ dPanel.ISI+" msec");
    Label l4=new Label(""+ dPanel.duration+" msec");

    Scrollbar sc_ISI=new Scrollbar(0,dPanel.ISI,0,0,1000);
    Scrollbar sc_exposure=new Scrollbar(0,dPanel.duration,0,10,1400);
    Button btn_start=new Button("시작");
    public void init(){
            this.setLayout(null);
            dPanel.reshape(dPanel.xb, dPanel.yb, dPanel.xsize, dPanel.ysize);
            l1.reshape(50,dPanel.ysize+ 10,100,20);
            sc_ISI.reshape(40,dPanel.ysize+ 30,180,20);
            l3.reshape(50,dPanel.ysize+ 50,100,20);
            l2.reshape(250,dPanel.ysize+ 10,100,20);
            sc_exposure.reshape(240,dPanel.ysize+ 30,180,20);
            l4.reshape(250,dPanel.ysize+ 50,100,20);
            btn_start.reshape(440,dPanel.ysize+ 20, 50,40);
            dPanel.img=createImage(dPanel.xsize, dPanel.ysize);
            dPanel.img_g=dPanel.img.getGraphics();
            add(dPanel);
            add(sc_ISI);
            add(sc_exposure);
            add(l1); add(l2); add(l3);add(l4);
            add(btn_start);
            btn_start.addActionListener(new java.awt.event.ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                    dPanel.is_start=!dPanel.is_start;
                            if (dPanel.is_start==true) { btn_start.setLabel("중지");
                                                    dPanel.start();
                            } else {btn_start.setLabel("시작");
                                    dPanel.stop();
                            }
                    }
            });
            sc_exposure.addAdjustmentListener(new
java.awt.event.AdjustmentListener() {
                    public void adjustmentValueChanged(AdjustmentEvent e) {
                            l4.setText(""+  (e.getValue())+"ms");
                            dPanel.duration=e.getValue();
                            dPanel.repaint();
                    }
            });
            sc_ISI.addAdjustmentListener(new java.awt.event.AdjustmentListener() {
                    public void adjustmentValueChanged(AdjustmentEvent e) {
                            l3.setText(""+(e.getValue())+"ms");
                            dPanel.ISI=e.getValue();
                            dPanel.repaint();
                    }
            });
    }
}
class DrawPanel extends Panel implements Runnable{
    /**********************쓰레드 관련 변수 함수**********************/
    Thread thr;
    int xsize=500, ysize=250, xb=0, yb=0, x_c=xsize/2, xc=x_c/2, yc=ysize/2, os=50;
    boolean is_start=false;
    Image img;
    Graphics img_g;
    boolean is_ok=false;
    int duration=200,ISI=150;
```

```
    public void start(){
            if(thr==null){
                    thr=new Thread(this);
            }
            thr.start();
    }
    public void stop(){
            thr.stop();
            thr=null;
    }
    public void run(){
            while(true){
                    draw1();
                    try{
                                    thr.sleep(duration);
                            } catch(InterruptedException e){};
                    draw3(); //blank
                    try{
                                    thr.sleep(ISI);
                            } catch(InterruptedException e){};
                    draw2();
                    try{
                                    thr.sleep(duration);
                            } catch(InterruptedException e){};
                    draw3(); //blank
                    try{
                                    thr.sleep(ISI);
                            } catch(InterruptedException e){};
                    }
            }
public void drawBg(){
    img_g.setColor(Color.black);
    img_g.fillRect(xb,yb,xsize,ysize);
    img_g.setColor(Color.white);
    img_g.fillOval(xc-os-os/2,yc-os-os/2,os,os);
    img_g.fillOval(xc+ os-os/2,yc-os-os/2,os,os);
    img_g.fillOval(xc-os-os/2,yc+ os-os/2,os,os);
    img_g.fillOval(xc+ os-os/2,yc+ os-os/2,os,os);
    img_g.fillOval(x_c+ xc-os-os/2,yc-os-os/2,os,os);
    img_g.fillOval(x_c+ xc+ os-os/2,yc-os-os/2,os,os);
    img_g.fillOval(x_c+ xc-os-os/2,yc+ os-os/2,os,os);
    img_g.fillOval(x_c+ xc+ os-os/2,yc+ os-os/2,os,os);
    }
public void draw1(){
    drawBg();
    img_g.setColor(Color.black);
    img_g.fillRect(xc-os,yc-os,os*2,os*2);
    this.getGraphics().drawImage(img,0,0,this);
    }
public void draw2(){
    drawBg();
    img_g.setColor(Color.black);
    img_g.fillRect(x_c+ xc-os,yc-os,os*2,os*2);
    this.getGraphics().drawImage(img,0,0,this);
    }
public void draw3(){
    drawBg();
    this.getGraphics().drawImage(img,0,0,this);
    }
public void paint(Graphics g){
    draw1();
    img_g.fillRect(0,0,500,400);
    img_g.setColor(Color.white);
    img_g.fillOval(75,125,50,50);
    img_g.fillOval(150,125,50,50);
    img_g.fillOval(250,125,50,50);
    img_g.fillOval(325,125,50,50);
    img_g.fillOval(75, 200,50,50);
    img_g.fillOval(150,200,50,50);
    img_g.fillOval(250,200,50,50);
    img_g.fillOval(325,200,50,50);
    this.getGraphics().drawImage(img,0,0,this);
    }
}
```

부록 II

주별 강의요목에 따른 감각과 지각 실험실습 프로그램 구성

주	강의 요목	자바에플릿 실험실습 프로그램
1	왜 감각·지각심리학을 공부하는가? 지각의 일반적 특징, 지각심리의 생리적 접근법	신경흥분 전달, 흥분/억제 시냅스, 구멍으로 들여다보기
2	감각이란? 지각이란? 감각·지각의 관계는? 지각심리의 여러 접근법 : 정신물리, 인지적 접근.	베버-페크너 법칙, 스티븐스 법칙, 맥락효과
3	시지각의 생리적 기초1 - 눈의 구조, 수용기	눈의 기제, 수용장
4	생리적 기초2 - 신경회로, 수용장, 중성색의 지각기제, 밝기대비	밝기대비, 헤르만 격자, 흐린윤곽, 마하밴드, 외측억제, 마하카드, 마하카드/경계효과
5	생리적 기초3 - 외측 슬상체, 시각피질, 생리와 지각 관계	뇌 영역
6	색채지각1- 색자극, 색 부호화, 색 혼합, 삼원색설과 대립과정설	색혼합, 등휘도색, 맥클로프, 색맹검사, 사진 색필터링, 색대응, 네온효과
7	색채지각2- 색채이론의 생리적 증거, 색채대비현상, 색채잔상효과 실습 밝기 항상성과 색항상성	색잔상, 색대비, 색 동화/대비, 벽돌착시, 화이트착시,
8	형태지각1- 형태지각의 물음, 지각체제화, 형태주의 Gestalt 심리학	13개의 얼굴, 착각윤곽/네커, 집단화:좋은 형태, 집단화:유사/인접성, 자동 집단화, 역전성 전경배경, 역전가능한 계단, Escher, 결분리,
9	형태지각2- 구성주의접근, 형태지각에서의 주의모형, 3차원 모양지각, 공간빈도접근,	시각검색, 착각적 결합, 사각/사인파, 사진공간주파변형 공간빈도잔여효과, 푸리에분석, 형태/공간주파, 착각적 원, 시각차폐, RSVP
10	깊이지각 및 입체시- 깊이단서, 입체지각, 양안부등	그림자 효과, 무선점 입체상, 전통적 입체상,
11	크기지각- 시각도와 거리관계, 다양한 크기 착시, 생태학적 접근	뮐러라이어착시, 폰조착시, 쪄르너착시, 포겐돌프착시, 분트착시, 올비슨착시, 헤링착시, 에른스타인착시, 무늬착시, 사각형착시, 선분기울기착시, 격자착시, 수직수평착시, 책상착시, 까페벽 착시, 티치너착시, 대각선 착시, 왜곡 사각형 착시, 각도 착시, 뒤틀린 선 착시, 채워진 공간착시, 왜곡된 원 착시
12	운동지각1- 운동지각의 기제, 안구운동	가현운동, 가현방향운동, 가현/착각윤곽, 폭포착시, 회전잔상효과, 수직 사선운동,
13	운동지각2- 생물학적 운동, 운동에 의한 구조, 환경에서의 운동	운동 3유형, 운동에의한구조, 원운동, 광학흐름, 원통운동, 착각적 회전, 가려진 사각형회전
14	종합정리 및 전망	범용 에플릿