(1페이지)

안녕하세요. 6조의 발표를 맡은 강승훈입니다.

저희 팀은 김아연, 김희주, 그리고 저로 이루어져 있으며, 지금부터 저희가 구현한 Pups Game에

대해 발표하겠습니다.

(2페이지)

발표는 보시는 바와 같이 프로그램 개요 및 설명, 전체 클래스 구조도 설명, 각 클래스 설명,

프로그램 시연 그리고 질의응답의 순으로 진행하겠습니다.

(3페이지)

\*아무것도 없는 페이지에서\*

저희가 만든 Pups라는 게임은 Pop up Points의 줄임말로, 이름 그대로 화면에 나타나는 포인트를

클릭해 점수를 얻는 게임입니다.

\*클릭해서 처음 기본 UI가 나오면\*

게임 기본 화면입니다.

왼쪽에 난이도, 점수, 제한시간을 나타내는 라벨들과 Start, Exit 버튼이 있고, 오른쪽 부분이 게임이 실행되는 곳입니다.

\*스타트 버튼을 클릭\*

스타트 버튼을 클릭하면 난이도를 정할 수 있는 창이 뜨고

\*난이도 LOW로 선택\*

난이도를 선택하면 왼쪽 난이도를 나타내는 라벨이 해당 난이도로 바뀌고, 그에 맞게 게임이

진행됩니다.

보이는 것과 같이 화면에 뜨는 아이콘을 클릭하면 그에 해당하는 점수를 얻게 됩니다.

(4페이지)

난이도 별로 나타나는 아이콘의 종류와 아이콘이 사라지는 속도가 다릅니다.

\*클릭\*

Low 레벨일 때, 나타나는 아이콘의 종류와 그에 해당하는 점수입니다.

\*클릭\*

Middle 레벨일 때, 나타나는 아이콘의 종류와 그에 해당하는 점수입니다.

\*클릭\*

High 레벨일 때, 나타나는 아이콘의 종류와 그에 해당하는 점수입니다.

(5페이지)

제한 시간 내에 제한 점수를 넘지 못하면 이와 같은 창이 뜨고, \*클릭\*

제한 점수를 달성하면 이와 같은 창이 뜹니다. \*클릭\*

게임이 끝나고 나면 계속할 건지 묻는 Continue 창이 뜨고, \*클릭\*

예를 누르면 게임 초기화면으로 돌아갑니다.

Continue 창에서 아니오를 누르거나, 기본 화면의 Exit 버튼을 누르면 게임을 끝낼 건지 묻는

창이 뜨고, \*클릭\*

예를 누르면 게임 화면이 꺼지게 됩니다.

(6페이지)

프로그램의 개요에 대해 설명했으니, 다음으로 저희 프로그램의 전체 클래스 구도에 대해 설명하겠습니다.

저희 코드의 UML 다이어그램입니다.

PupsGamePanel은 전체 게임의 기본 화면을 나타내고, 각 클래스의 객체들을 선언하여 모든 기능과 이벤트를 처리해줍니다. TimeThread 클래스의 limitTime 객체를 사용하여 제한 시간을 나타내고, IconThread 클래스의 gamePanel 객체를 이용하여 아이콘들이 나타나고 사라지는 시간을 조정합니다. GameListener 클래스는 PupsGamePanel 클래스 안에 있는 종속 클래스로, 이 게임에서 발생하는 모든 이벤트를 처리해 줍니다.

(7페이지)

각 클래스에 대해 설명하겠습니다.

첫번째로 PupsGamePanel 클래스입니다.

\*클릭\*

이 클래스에서는 기본 GUI를 생성하는 것이 주된 기능이고, 보이는 데이터들은 이를 보여주기 위해 쓰이는 패널, 라벨 그리고 버튼입니다.

데이터 중 Point 객체 nLocation은 아이콘을 임의의 좌표에 띄우기 위해, 좌표의 값을 설정하기 위한 것입니다. X와 y 값을 랜덤 함수를 사용해 지정하고, 그 좌표를 사용해 화면에 보여줍니다.

또한, int형 객체 index는 해당 아이콘의 점수를 계산하기 위하여 점수가 저장된 배열의 인덱스를 넘겨받기 위한 변수입니다. 이는 PupsConstants 클래스를 설명할 때 자세히 얘기하도록 하겠습니다.

addIcon 메소드는 IconThread 클래스에서 생성된 임의의 아이콘을 parameter로 받아서, 임의의 좌표를 설정해 JButton 객체인 아이콘을 gamePanel에 add하는 역할을 합니다. 그리고 아이콘에 해당하는 점수를 저장한 배열의 인덱스도 Parameter로 같이 받아 아이콘이 점수를 가지고 있을 수 있도록 합니다.

exitGame은 게임을 끝낼 것인지 묻는 제이다이얼로그 박스 창을 처리하기 위한 메소드입니다.

continueGame은 게임을 계속할 것인지 묻는 제이다이얼로그 박스 창을 처리하기 위한 메소드입니다.

\*클릭\*

GameListener 클래스는 PupsGamePanel 클래스에 종속된 클래스입니다.

여기서 start, exit 버튼 뿐만 아니라 JButton 객체인 아이콘에 대한 이벤트를 처리합니다.

(8페이지)

TimeThread는 제한시간을 위한 클래스입니다.

startNumber부터 finishNumber까지 시간이 흐를 때 그 안에서 처리할 기능들을 정의하고 delayTime을 두어 일정 시간 동안 일정한 간격을 두고 기능을 반복할 수 있도록 했습니다.

스래드 객체 numberThread는 게임을 다시 시작할 경우에 스래드를 초기화하기 위한 객체로, setNumberThread 메소드에서 초기화를 해줍니다.

\*클릭\*

Run 메소드에서 앞에서 설명한 기능들을 처리합니다.

(9페이지)

IconThread는 아이콘을 생성하는 클래스 입니다. 스레드를 사용한 이유는 아이콘들이 시간에 따라 나타나고 사라지기 때문입니다.

TimeThread 클래스와 같이 numberThread로 setNumberThread메소드를 이용하여 게임이 다시 시작할 경우를 위해 스레드 초기화를 시켜주고, delayTime으로 일정 간격을 두게 합니다.

이 클래스에서 게임에 필요한 아이콘을 생성하게 됩니다. JButton 객체인 btnIcon으로 아이콘들을버튼으로 생성하고, 정수형 객체 nIndex로 임의의 인덱스를 설정하여 아이콘을 그때그때에 랜덤하게 생성합니다.

\*클릭\*

Run 메소드에서 시간 간격에 따라 아이콘을 생성합니다.

\*클릭\*

여기에서 PupsGamePanel클래스 때 설명드린 addIcon이라는 메소드가 사용이 됩니다. 이 primary는 생성자에서 넘겨받은 PupsGamePanel 클래스의 객체입니다.

\*클릭\*

생성된 아이콘을 넘겨받아 PupsGamePanel에 추가합니다. nIndex도 같이 넘겨주어 생성된 버튼이 어느 것인지 알려줍니다.

(10페이지)

\*클릭\*

PupsGameEx 클래스는 게임을 실행하기 위한 드라이버 클래스입니다.

\*클릭\*

PupsConstants 클래스는 아이콘에 대한 상수 클래스로, ICON 객체는 아이콘의 이미지들을 저장해놓은 상수 배열이고, ICONSCORE는 각 아이콘에 해당하는 점수를 저장해 놓은 상수 배열입니다. 인덱스로 각 버튼과 점수를 구분합니다.

(11페이지)

프로그램을 시연하겠습니다.

~~~~~~~

(12페이지)

질문 있으신 분은 질문 해주시기 바랍니다.

(13) 끝!