

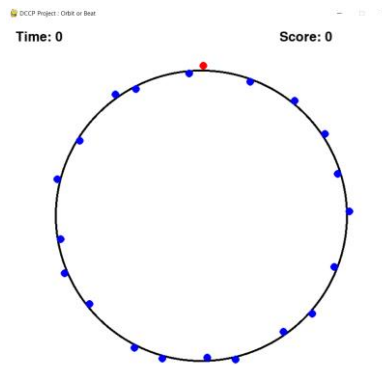
# 2021-2 DCCP Final Project

서울대학교 자유전공학부 양석훈

## 1. 프로그램 소개

Orbit or Beat라는 게임을 구현하고자 하였습니다. 이 게임은 특정 궤도를 돌고 있는 원이 있으며 이 원이 가시를 피해야 하는 방식의 리듬 게임입니다. 오랫동안 생존해내는 것이 이 게임의 목표가 됩니다.

## 2. 실행 및 조작 방법



### 1) 게임의 실행방법 및 구성요소

파일을 다운받은 후 압축을 풀어서 'Orbit\_or\_Beat.py'파일을 실행하면 게임을 시작할 수 있습니다. 위 사진은 게임을 실행시킨 후 처음 나오는 화면을 캡처한 화면입니다.

먼저, 빨간색의 작은 원이 플레이어입니다. 플레이어는 빨간 원이 장애물에 닿지 않도록 하는 것이 게임의 목표입니다. 다음으로, 파란원은 가시입니다. 빨간 원이 파란원에 닿게 될 경우 저장된 장소로 돌아가서 다시 시작하게 됩니다. 검은색의 원주는 플레이어가 조작하는 빨간 원이 돌게 될 궤도입니다. 즉, 플레이어는 검은색의 원주를 도는 빨간 원이 파란원에 닿지 않도록 조작하여 최대한 오래 살아남아 높은 점수를 따면 됩니다.

왼쪽 위 Time은 현재까지 버틴 시간을 보여주는 것입니다. 또, 오른쪽 위의 Score는 획득한 점수로 최대한 많이 버틴 시간을 의미합니다.

### 2) 조작방법 및 게임설명

기본적인 조작은 '스페이스바'로 이루어집니다. 스페이스바를 누를 경우 빨간 원이 원주 안쪽에서 바깥쪽으로, 혹은 바깥쪽에서 안쪽으로 플립하게 됩니다. 가시 역할인 파란원은 원 내부,

혹은 원 바깥쪽에 위치하기에 이를 플립을 통해 피해서 장애물에 닿지 않게 지나가면 됩니다. 만약, 실수로 빨간 원이 파란원에 닿게 된다면 가장 마지막에 저장된 위치로 돌아가서 다시 시작하게 됩니다. (기본적인 세이브포인트는 시작시점 입니다.)

게임을 플레이하다 보면 일정시간이 지날 때 마다 세이브포인트가 지정된 위치에 등장하게 됩니다. 플레이어가 세이브포인트를 지나게 된다면 마지막 세이브포인트가 해당 세이브포인트로 갱신되어서 이후 가시에 닿는다면 이 지점에서부터 시작할 수 있습니다.

마지막으로 가시의 위치는 일정 주기로 변화하니까 방심하지 않아야 피할 수 있습니다. 또, 일정시간이 지날 때 마다 빨간 원의 속도가 약간씩 빨라지게 됩니다. 많이 틀리실 경우 가시의 변화가 덜 일어나게 프로그래밍 되었기에 아마 적응하실 수 있을 것이라고 생각합니다.

중간에 잠깐 게임을 정지하고 싶으시다면 키보드에서 p키를 눌러주시면 20초간 게임이 정지하게 됩니다. 이 동안은 일정주기로 발생하는 모든 종류의 변화가 일어나지 않기에 편하게 쉬셔도 될 것 같습니다. 또, 게임을 종료하게 되면 화면 정중앙에 초록색 글씨로 점수를 표시해준 후 5초뒤에 게임이 종료됩니다.

### 3) 난이도 조절방법

만일, 빨간 원이 너무 빠르거나 느려서 난이도가 적절하지 않다고 생각하신다면, 방향키를 통해서 속도를 조절할 수 있습니다. 시작시에 위 방향키를 누르면 속도가 1.25배가 되며 아래 방향키를 누르면 속도가 0.8배가 됩니다. 단, 이는 중간에 조절하게 될 경우 빨간 원이 조금씩 빨라지기에 난이도가 원활히 조절되지 않을 수 있어서 처음에만 조절하고 이후 계속 진행 해주시면 좋을 것 같습니다.

## 3. 프로그램 구조, 알고리즘 설명

프로그램의 핵심이 되는 각 클래스에 대해서 하는 역할을 간략하게 설명해보고자 합니다.

### 1) Game class

게임의 기본이 되는 클래스로 배경을 먼저 설정한 후, 모든 종류의 업데이트가 Game 클래스를 통해서 이루어집니다.

### 2) Player class

플레이어가 조절할 수 있는 빨간 원에 대한 클래스입니다. 스페이스키를 누를 때 플립이 되고, 방향키를 누를 때 각속도가 변화하도록 구현하였습니다. 이때, 위치는 속도 값을 구해서 계속 더하는 방식으로 오차가 크게 나타나서 매시간 위치를 계산하고 대입하여 매번 위치를 설정해주는 방식으로 구현하였습니다. 또, 다른 물체와 접촉을 관리하는 것도 Player 클래스에 메소드로 포함되어 있습니다.

### 3) Block class

가시 역할을 하는 파란원에 대한 클래스입니다. 가시의 개수를 20개로 설정하였으며, 전체 각인 360도를 겹치지 않게 partition을 나누어서 각 partition에 가시를 할당하는 방식으로 가시의 위치가 겹치지 않고, 상대적으로 난이도가 쉬울 수 있도록 구현하였습니다.

### 4) Save class

세이브포인트를 만들어주는 클래스입니다. 다른 클래스들과는 다르게 기본적으로 active가 False에서 시작하고, 특정시간이 될 때 마다 active가 True로 전환되어 화면에 그려질 수 있도록 하였습니다.

### 5) Pause class

일시정지를 할 수 있도록 해주는 클래스입니다. p키를 누르면 time.sleep(20)이 나뉘어서 실행되고, 이때 시간을 계속 초기화 시켜주며 계산해서 최종적으로 흐른 시간 역시 일시정지 전과 동일하게 계산되도록 맞추었습니다.

## 4. 참고문헌 및 자료

### 1) 타이머 구현

<https://blog.naver.com/PostView.nhn?blogId=trudiala&logNo=221973644998>

이 링크에서 기본적인 타이머를 구현하기 위해 get\_ticks()를 하는 것과 이를 이용해서 진행시간을 구하는 방식을 확인하고 이를 응용해서 타이머와 점수를 만들었습니다.

### 2) 프로그램의 기본적인 구조

컴퓨터의 개념 및 실습 강의의 지난 4번째 과제에서 여러 오브젝트를 만들고 관리하고 함수를 구성하는 방법을 참조해서 사용하였습니다. 지난 과제는 주어진 영상을 보고 참조하였기에 여러 함수명과 변수명, 함수의 역할이 비슷하게 구현되었습니다.

### 3) 음원파일 삽입

Apink의 %%라는 노래의 파일을 wav의 형식으로 다운받아서 이를 게임에 삽입하기 위해 다음 링크의 내용을 활용하였습니다. <https://makerejoicegames.tistory.com/101> 이 링크를 통해 음악을 게임에 삽입하는 내용을 알았고 이를 제 코드에 활용하였습니다.