BeatRoads 아이디어

1. 오디오 스펙트럼 분석해서 비트 만들기

a. 적절한 비트 만들기 문제

i. window를 통해 앞뒤 50개의 spectrum의 평균을 구하고 적절한 beat 판별 ratio를 곱함

>> 현재 스펙트럼의 값이 i의 결과보다 크다면 beat로 간주

>> 단 이전 비트 생성 시간보다 일정 시간 이후부터 생성 가능

>> 추가할 사항으로 노래의 bpm에 따른 판별식도 고려

ii. 기존의 (https://medium.com/giant-scam/algorithmic-beat-mapping-in-unity-real-time-audio-analysis-using-the-unity-api-6e9595823ce4) 코드 활용시 해당 샘플의 비트는 잘 뽑아내지만 비트가 너무 많이 생성된다

- 기존의 방식 :

>> 직전 프레임과 현재 프레임의 스펙트럼 음역대별 비교로 스펙트럼 증가분(flux) 합 계산

>> 현재 스펙트럼 기준 앞뒤로 window/2개 만큼에 대해

>> window 전체의 flux 평균에 비트 계산 상수를 곱하여 현 스펙트럼의 비트 기준값 결정

>> 해당 비트 기준값을 기준으로 현 프레임의 스펙트럼이 얼마나 큰지 계산 - prune

>> 이때 크지 않다면 0으로 저장

>> 이후 index - 1의 prune값을 기준으로 Index -2, Index를 비교

>> index -1의 prune 값이 가장 크다면 피크값으로 판정

- 내 보정

>> 곡의 실제 bpm과 비교해보면 거의 10배 가까이 더 생긴다(10배 이상)

>> sol1. 비트 상수의 크기를 기운다

>> 테스트 결과 비트 상수의 크기가 2일 때 실제 비트와 유사한 결과를 보여준다

>> sol2. 비트가 과추출되는 것을 방지하기 위해 && 현실적인 플레이를 위해 비트 갱신 최소시간을 둔다

>> sol3. 비트 갱신 최소 시간을 넘기는 비트를 기준으로 주변의 비트들 중 일정 시간 내의 비트들 중 가장 스펙트럼의 값이 큰 것을 비트로 결정한다

>> 노트를 min 노트 주기 이상 && 주변 peak들 중 최대값으로 정한다(winddow이용해 체크)

b. 적절한 길이의 노트 만들기

>> i. 이후 비트의 시작시간 = 이전 비트의 종료시간

>>ii. note에 디버깅 용으로 str와 ed 시간 및 길이, 좌표를 public으로 에디터상에 표기하는 스크립트 만들어서 붙이기

2. 비트의 시작시간, 노트 만들 위치를 엑셀로 뽑기

3. 엑셀을 통해 수정해서 유니티에 JSON으로 불러오기

4. 3에서 불러온 JSON 데이터를 커스텀 에디터를 통해 노트 위치 / 시작 / 끝을 수치로 보기

5. 4의 커스텀 에디터에서 위치, 시작, 끝을 수정할 수 있도록 하기