Final Peer Review for Creative Coding Prototypes

Reviewer Name: Isabella Murray

Project Creator: Ivory heidenreich

Project Creator Email : ivory.heidenreich@student.uts.edu.au

Date: 26/10/2025

Project: Never Tear Us Apart - Tunepad (NXS modern variation)

1. What Do You Like About This Project?

This was a fabulous project. I loved how the kick, snare groove immediately locks in and carries the track while the tight hi-hat pattern adds momentum, the electric bass anchors the harmony and the chords keep the INXS mood intact but modernised. The moment the full kit drops under the chords genuinely lifts the energy. The contrast between lighter and fuller sections keeps it engaging instead of “just looping”. Your layering choices are clear and musical, so each element has space and purpose. Overall, it’s a confident, clever remix that respects the original while sounding distinctly yours.

2. What Is the Project Trying to Say or Do?

By rewriting "Never Tear Us Apart" in Python, Ivory connects classic Australian rock with modern digital craft, speaking to both a sense of nostalgia and to issues of authorship. Musically, the project sequences chords, kick, snare, hat patterns, and electric bass in order to reconstruct the heartbeat of the song, then uses layering and section changes-lighter hats versus full kit drops-to tell a mini-story of tension and release. In plain words, programmatically, the code arranges and plays a modernised version of the track; it triggers chord progressions, drives a drum groove, locks in a bassline, and evolves the energy over time so the piece feels fresh while staying true to the core feeling of the original.

3. How Well Does It Work?

What really surprised me most was how tight the drum engine feels, the kick and snare lock in, the hats keep the pulse moving, and the bassline neatly outlines the harmony so the groove “breathes” rather than just looping. The only places that felt rough were some section handoffs, which come across a bit abrupt, a 1–2 bar fill, a crash, or light velocity/swing on the hats would smooth those transitions and add human feel. From a usability angle, it wasn’t immediately clear how to switch sections or tweak BPM/lengths, brief inline prompts (e.g., print tips at start) and a couple of lines in the README would remove that friction. Overall, the prototype works well and already communicates the idea, a few transition cues and small humanisation tweaks would take it from solid to polished.

4. Git Repository?

The repository link works and I can access the README.md and project files however there are only links and instructions to the tunepad page, no other sources or evidence. The repository has been documented satisfactory to the README.md where It could’ve provided more details regarding rationale surrounding the task, screenshots showing the project, as well as the video explainer for audiences to view. The preview explains how to play the code on Tunepad however URLs need to be copied and pasted in a new browser for them to be opened. The repository could’ve included more external project details such as images and videos. The repository is not clear in the way that it doesnt have folders / directories to assist in navigating areas of the project. It also does not include extra materials to further demonstrate the project

5. How Clear is the Code?

The creator has not added a code file in the repository therefore it is not applicable to rate whether it is easy or difficult to read. It is also hard to determine if the code is broken into logical

Final Peer Review for Creative Coding Prototypes

small sections as reading the code off Tunepad looks more complex. Based on Tunepad, the code appears very complex and intricate for the melody and chords section. It appears each cell varies in length which seems applicable to the song of the project. The variables and functions make sense to me as I also did a musical composition however for other audiences it may have been more difficult to navigate.

6. How Clear is the Creator's Explanation of the Core Innovation or Functionality?

Your explanation was clear, you walked through how lists store the chord progression, loops drive the kick, snare, hat patterns, and a function schedules bars so the bass locks to the harmony, good use of variables (BPM/section) and beat based timing. To go deeper, include a tiny code snippet for your beat/chord generator and explain your choices for swing/velocity and how you keep tracks in sync. Also note how section switches work and add brief comments/docstrings mapping musical ideas to code.

7. Ask a Thoughtful Question

How did you translate the song's emotion into specific code choices, tempo, chord voicings, drum velocity/swing, and section transitions. If you could only change one live parameter to shift the mood, eg. swing %, hat density, or bass note length, which would you choose and why?

8. Final Thoughts

A clever, coded reimagining of INXS's "Never Tear Us Apart" that retains the heart of the song while updating it with a bassline that really glues the chords together. Both respectful and inventive, I'm impressed by how cleanly the Python structure translates into real musical payoff.