Game Project Proposal

Team Audio Enema presents

Tatums

"Phase"

"Need For Speed"

"Half Life 2"

"Ninja Gaiden"

"Take it all in stride"

Tatums is a group of mini games all implementing audio analysis and procedural level generation. The player inputs an audio file which our level generating library then turns into a track compatible with any of our three mini games. Mini games include "Beat Match Racing", "Beat Match Range Weapons", and "Beat Match Melee Combat".

Software Engineering

- *Deliverables Analysis*

The system is expected to take in an audio file, analyze its tonal qualities and output data based on those qualities. This information is used to procedurally generate a 2D map with a starting point, ending point and a path between the two. As the next step the system populates this map with 3D assets which reflect the information gathered from the audio analysis. The system uses the beat data from audio analysis to synchronize the mini game's scenario with the music.

Racing – In this mini game the race track is populated with coins and obstacles which correspond with the up-beat and down-beat of the song. The objective is to score points by collecting as many coins as possible while avoiding the obstacles. Maneuvering further down the track before the end of the song will also increase the players score.

Ranged Weapons – In the style of the racing mini game, this one will also be based on beat. The player will be given an interface combining the note dropping aspect of "DDR" with the FPS aspect of arcade shooters like "Time Crisis". Notes will be mapped to the W, A,S, and D keys and the player will be shown a sequence of key strokes that is synchronized with the beat. By accurately replicating these strokes with proper timing the player's multiplier will increase. The player will also be aiming with the cursor and firing at enemies who popup on the screen. The player will gain points based on enemies shot and the current state of the multiplier when an enemy is shot.

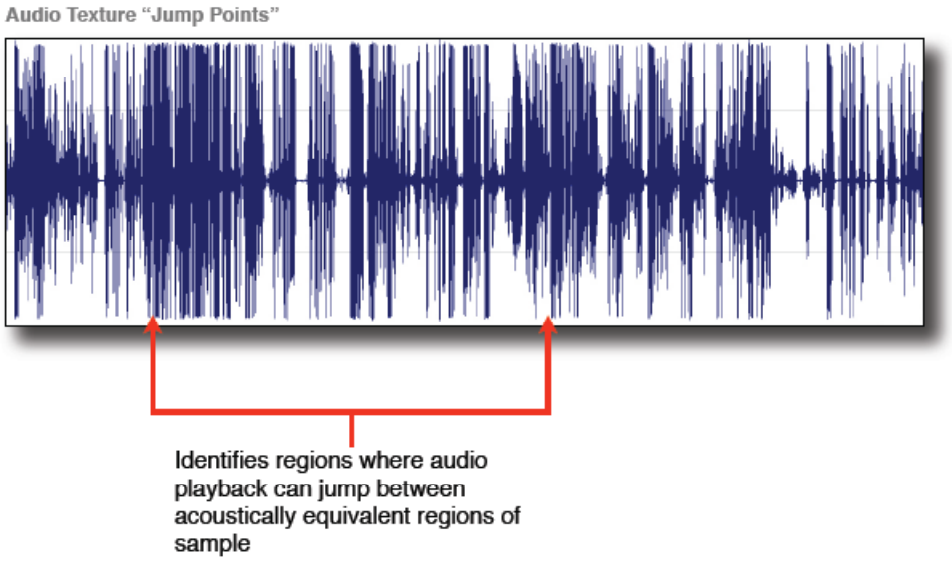
Melee Combat – In this mini game the objective is to attack on the beat and defend on the offbeat. The player will be able to perform basic offensive moves like punch and kick and basic defensive moves like block and dodge. The objective is to score points by defeating enemies.

- Design

  - 3D graphics rendering engine: **Ogre**
  - Physics Engine: **Newton**
  - Sound API: **fMod**
  - Audio Analysis API: **The Echo Nest – music analyzer API**
  - Wii Remote Library: **Wiiyourself**

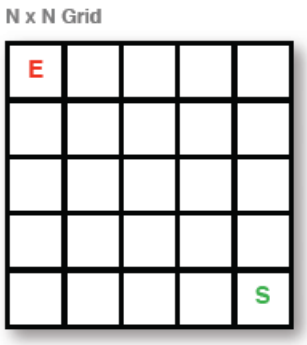
Step 1: Audio Analysis

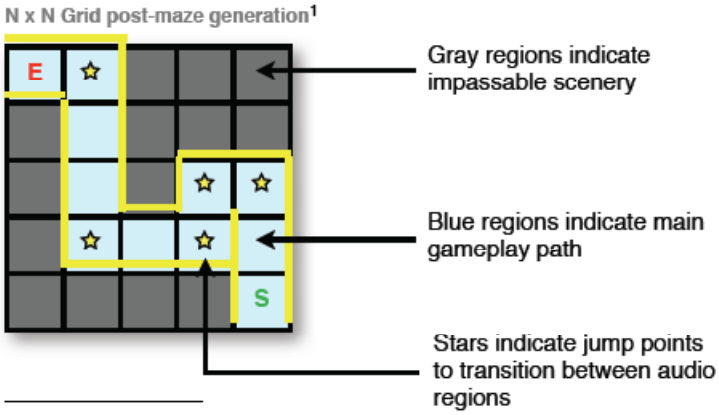
- Player selects a song to "play". The song is then sent over the Web for server-side audio analysis (with EchoNest — analyze.echonest.com)
- Beat information is extracted (tempo/BPM, number of beats, etc.)
- Potential audio textures analyzed
- This results in a set of known "jump points" where various, non-consecutive sequences of the sample can be juxtaposed to give an artificial sense of continuity



Step 2: Map Generation

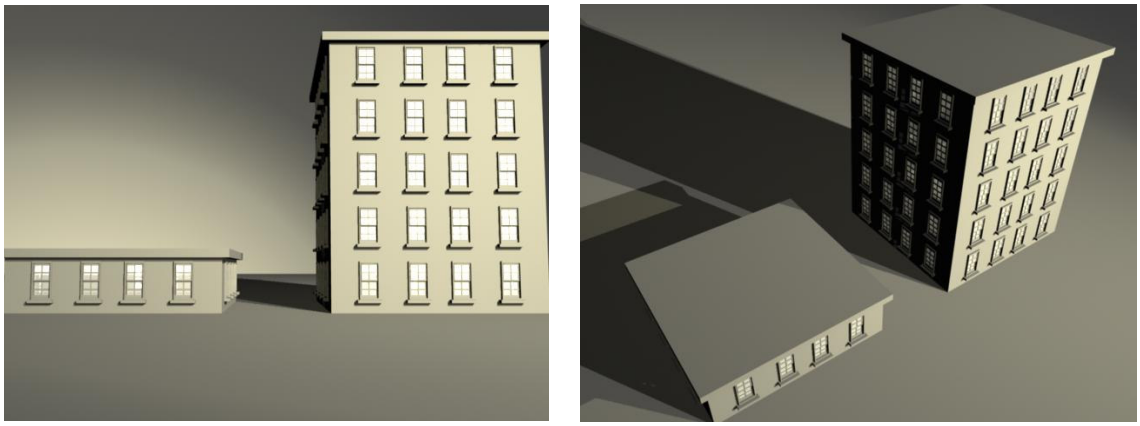
- Map is initially treated as a n x n 2D grid with a single "Start" and "End" point
- 2D grid serves as a generic plane that can be shared by each mini-game implementation.
- Grid only represents X and Z axes; Y axis map information is determined on a per-mini game basis.
- Basic maze generation algorithm would be applied to grid, with the number of walls based on how many audio texture jump points are available in the sample
    - i.e. each straight "hall" or pathway in the maze would be for each audio region
    - transitions from hall to hall would be the spatial equivalents of jumps from audio region to audio region at the matching "jump points"





Step 3: Level Generation

- Map data is used to extrude a 3D level.
- The height value is determined by the variable such as amplitude and pitch. This operation is performed dynamically to reflect the current sonic properties of the song. e.g. the buildings bounce to the beat by changing height.
- Tillable geometry will be used in all situations where height is variable, such as skyscrapers.
- Building and scenery geometry will be placed outside the bound of the path as defined by the map and only the area within the path will be playable.



Tillable geometry

Step 4: Application of Map, Level, and Beat Data to Minigame

- Racing Game
- Melee Combat Game
- First-Person Shooter Game