Jazzy Beach Critters

A Demonstration of Real-Time Music Generation with Application to Games

Donya Quick and Christopher N. Burrows

http://www.donyaquick.com/jazzy-beach-critters/

What Jazzy Beach Critters Explores

- Can we generate novel game music on the fly **at the score level**?
  - The vast majority of games use pre-rendered audio for music.
    - Transitions just involve cross-fading, not choosing notes.
  - Score-level generation in games usually still has a pre-composed database.
    - Generating good, novel musical scores can be a very hard (slow) task!
    - But what about improvisational music like jazz?
- What technical/perceptual issues have to be addressed?
  - Can a commonly used game platform like Unity deal with the timing constraints of generative music with strict metrical structure?
  - How to adapt to changes in the environment and user actions?

Jazzy Beach Critters Overview

- A single Unity scene
- Three critters play music together: a crab, hermit crab, and snail.
  - Each animal is a different part/instrument.
- Critters have 3 moods: happy, neutral, and angry.
  - A critter's mood affects the music it plays. Users can change those moods.
- User actions:

Place food to move the critters.

Pet a critter to make it happier and bring it towards you.

Poke a critter and it will move away and become angrier.

Trade solos with the crab.

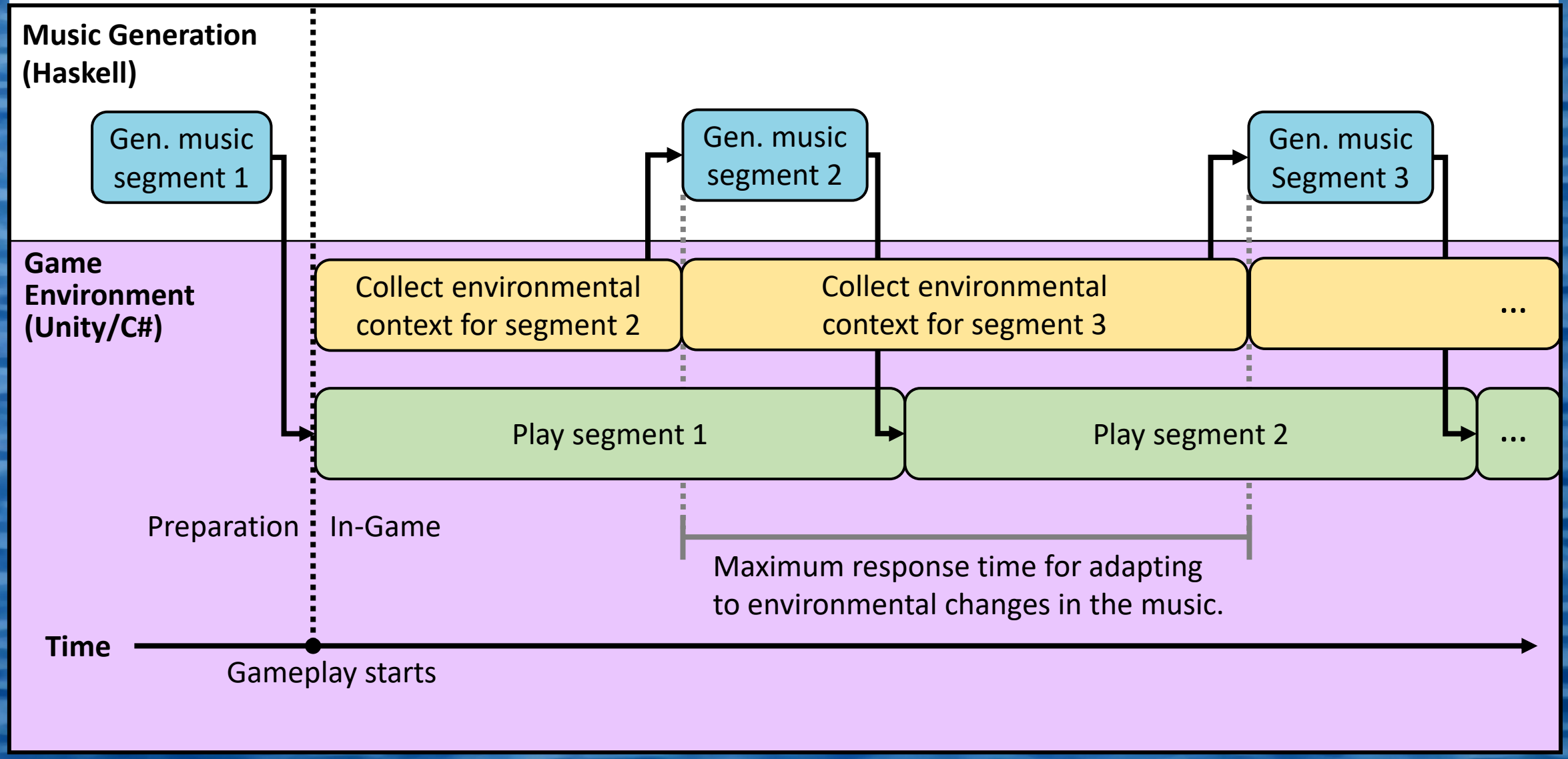
Newer than the paper

Click the beach ball to change styles.

Jazzy Beach Critters Overview

- Real-Time Generation of Musical Scores
  - Nothing is pre-composed / pre-rendered to audio.
  - Sound synthesis happens within the game environment.
  - Played notes spawn 3D note objects.
  - All critters have a shared key context – which is chosen in real time just ahead of the score generation.
- Multiple levels of temporal granularity:
  - **Sound effects** corresponding to visual events or user actions. Only graphical synchrony is important.
  - **Sound for musical notes** requires more precise timing (<10msec variance). Only musical synchrony is important.
  - **Musical mood/style changes** must wait for the next sensible musical boundary.
    - Transitions happen at measure/bar lines in this case.

Jazzy Beach Critters Workflow



Server

Generative Music Service

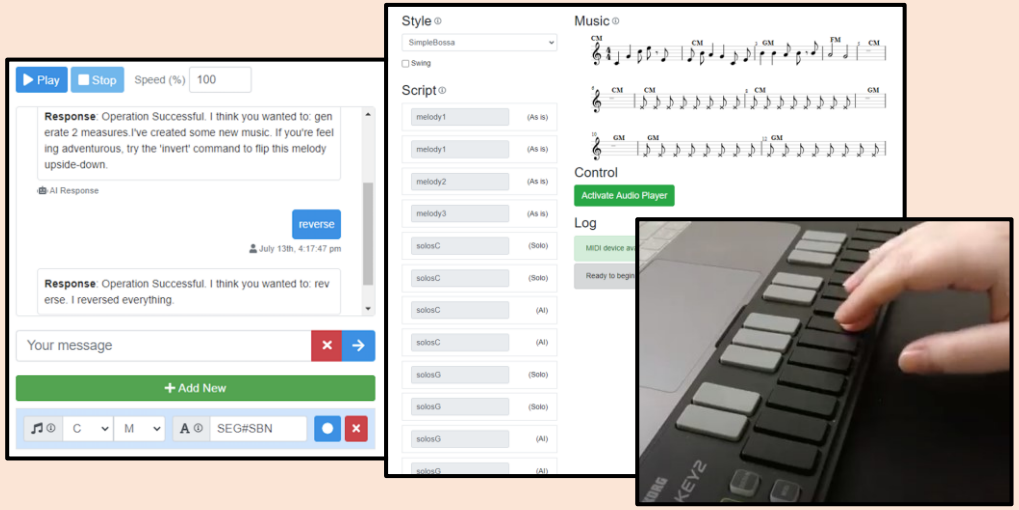
Database

Other Services with Similar Interfaces

Jazzy Beach Critters Game Interface (Unity / C#)

Donya Quick & Chris Burrows

MUSICA Project Interface (Python, JS)



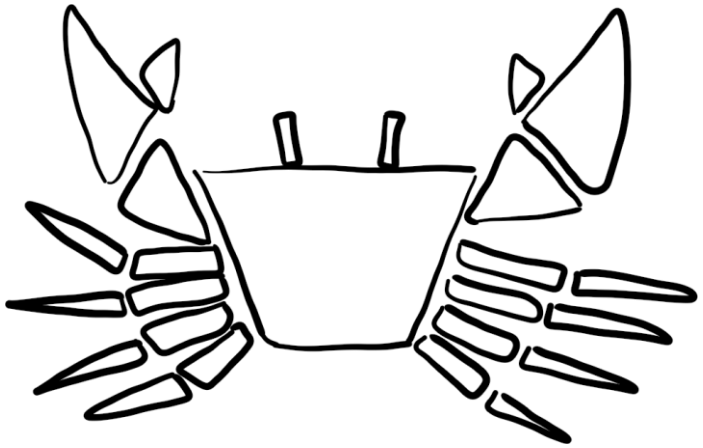
Donya Quick, Chris Kim, & David Burke
http://www.musicaresearch.org

Jazzy Beach Critters can also run in a completely offline, self-contained capacity, but our online demo works with a client-server model and can run in a browser.

Future Directions

- Other game elements within this project:
  - Points system to score user-played music
  - More complex critter behavior
    - Critters move around on their own, some critters don't like each other, etc.
  - Let the user add and remove critters that play different parts/styles.
    - Some critters may be earned by performing well with simpler/easier ones.
    - Incorporation of more recently designed jazz algorithms from the MUSICA project.
- Larger / longer-term possibilities:
  - Story-driven, peter-and-the-wolf themed games where the soundtrack is generated by the characters in real time.
  - Educational tools for children that use game elements to make learning music fun (like guitar hero, but musically interactive & improvisational).

Thank You!



http://www.donyaquick.com/jazzy-beach-critters

Donya Quick, donyaquick@gmail.com

- Music generation, system design, critter design

Christopher N. Burrows, christopher.n.burrows@gmail.com

- Unity & C# scripts, 3D modeling & animation