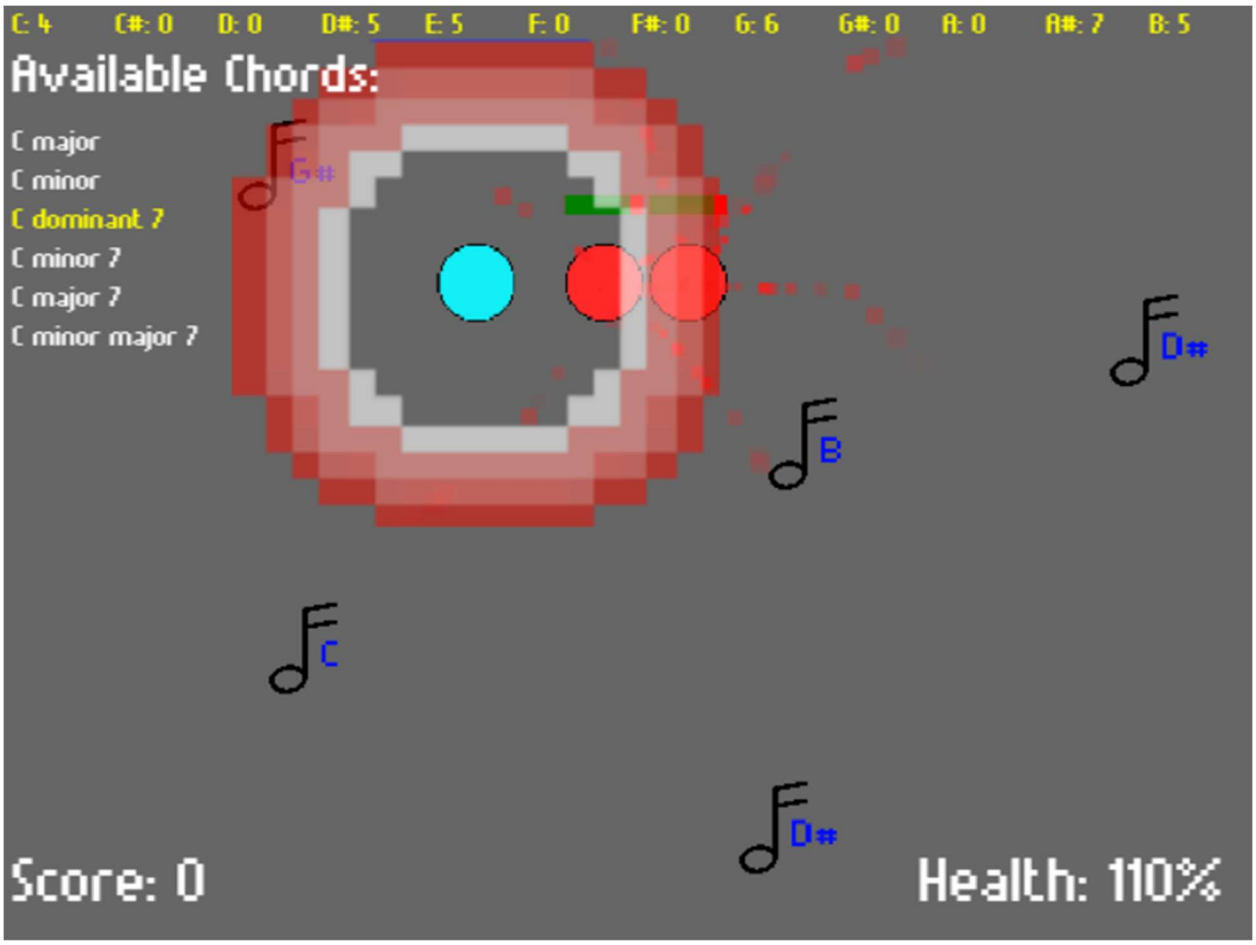
Chord Wars

Alec Bielanos & John Palermo
IGME330 – Professor Cody Van De Mark
April 2017



Original Idea

Our original idea stemmed from the concept of fighting enemies using musical notes/chords, an idea which persists in our final design. We also started out with the idea that different chord combinations would have different sized blasts and damages, but not everything made it to the final product. We wanted to have a library of differently pitched notes from the same instrument or synth so that the program would be able to build chords and play them on the fly. One of our stretch goals was to be able to select the musical 'key' that you played in which would alter the sound output and the background music.

Final Design

The final design took much of the essence of what we were trying to accomplish and simplified the design behind it. We realized that it would have been too complex computationally to check for all possible types of chords, so we shortened the algorithm to only look for combinations in the key of C major. Additionally, we condensed our array of notes into pre-recorded sounds of completed chords, which allowed for some addition of melody and rhythm to the final sound cues.

The background music, with fear that it would not line up with the chord 'attacks' was limited to percussion, which did not prove to be detrimental. Because of time constraints and available software, we went with an 8-bit type retro feeling (see 'Media' section below) and found great results in the cohesiveness of the finished product.

As we developed the initial concept, we moved pretty far away from a lot of what we originally wanted because of practicality, but as we did we began to realize the potential for improvements and future work that resulted from these changes.

Elucidation of Criteria

Media

- Sound:
  - Designed and recorded in-house.
  - Background sound is a looping percussion track only, since the variety of chords does not always fit one key signature.
  - Effect sounds include all 6 of the currently implemented chords (major, minor, major 7, minor 7, dominant 7, minor major 7). There are two sounds for each chord, one that plays when charging and the other that plays when the effect occurs. In addition, a sound effect has been added for when notes are picked up off the ground.
  - Sound effect files are of type WAV, with maximum file size of 690kb. Background file is of type MP3, with maximum file size of 2022kb. The background file, although a loop, had to be extended due to limitations/inability for seamless looping in the HTML audio element.
- Images:
  - Designed and/or modified in-house.
  - The entire project uses the consistent 'theme' of retro pixel art.
  - Spritesheet sourced under Creative Commons 0 license and altered – see "Non-Course Resources" section below.
  - Image files are of type PNG – the largest (spritesheet) is 10kb, the notes themselves range from 2-3kb per file.
- Text/Fonts:
  - 2 Embedded fonts – Munro and Droid Sans, see non-course resources for attributions.

    - Munro – Decorative font used for titles and headings as well as the UI in-game.
    - Droid Sans – Clearer sans-serif font used for description of game.
- Canvas Drawing/Animation:
    - DeltaTime method implemented to ensure smoothness and timing of animations – we actually timed the audio to be the same length as the animations, and thus a time delta had to be used.
    - As mentioned above, a sprite sheet was used in the animation of the explosions. Different chords are represented by different colored explosions
    - Tony Jefferson's emitter.js was used to construct a particle system that emits square pixels outward from the enemies as they move.
    - Everything drawn to the canvas with the exception of the emitter and background is a PNG image.

Interaction (Game)

- Below is the list of current controls, also below the game window on the project page:
    - WASD – move character sprite up, down, left, and right, respectively.
    - 'I' and 'K' – Move chord selector up and down list of available chords (selected chord will be highlighted in yellow)
    - Space Bar – Fire the currently selected chord
    - 'P' – Pause and un-pause the game (screens and game states reflect this change)
    - 'C' – Cheat that will give the player all of the notes currently used to build chords – this enables easy debugging and also grading by instructors
- All key input is handled in a "key daemon" within keys.js and accessed throughout the rest of the program.

Usability

- The game will be paused and un-paused when the window goes out of and returns to focus (respectively).
- Limited instructions are provided, but the game is intuitive enough that we are confident that most will be able to pick up the game with the information provided.
- Feedback and game screens ensure that the 'state' of the game is clear. The player's score and health at the bottom of the screen are updated in real time.
- The difficulty of the game is increased as more enemies spawn. Since the damage stacks, a group of 4 enemies will do 4 times the damage, so making a pile of enemies to kill in one attack may not be the best strategy for this game.
- Four screens (title, main game, paused, and game over) screens are functional and provide information to the user.

Game Design

- The game is a one player survival game, in the true spirit of pixel art "arcade" games. There is no "win" condition, as one must simply go for the high score. However, the player will lose if their health reaches zero.
- User choices do play a part in game outcomes. Although all of the notes can be used in the construction of chords, these chords have different damage and healing values. The player may wish to hold off on their damage chord if it would mean they wouldn't have enough leftover notes for a healing chord should they get into a sticky situation. In addition, as mentioned above, grouping enemies together may prove to be detrimental to the player because of stacking damage.
- The game is not as deep as it could be. This is one shortcoming of the game, and is explained in more detail below in "future work" and "what went wrong".

- As the player plays more, they will improve by knowing which chords do the most damage/healing and how to accurately avoid enemies. As a byproduct, they may learn a bit more about music theory as well (see “What Went Right”)
- We are confident that this game will be engaging to a wide range of players, especially if it is developed more and a wider range of attacks and enemies are added.

Coding

- No external scripts were used in the making of this project (unless you are to count course acquired scripts)
- ‘use strict’ has been included at the top of every javascript document.
- The sprite sheet is pre-loaded in the HTML document and is acquired by the player script later in the initialization process.
- Proper capitalization and indentation/bracketing standards have been followed throughout all scripts.
- This project uses the module pattern – several different js files with varying styles (IIFE, Function constructor, json) were used and pieced together at load time in loader.js.
- Code is clean and of high quality, however some things may be improved later (see “Future Work” below).

Documentation

- You’re reading it right now.

What Went Right

- Art Style – We didn't really have one planned out in the original concept but it all sort of "happened" at once. Due to limitations in software and time, we were able to create a quality, cohesive product (both visually and audibly)
- Tested Our Limits – We learned a lot about the practical application of javascript and different ways to approach game design through this language that we just picked up.
- Huge Potential – This game opened up many possibilities of future work as it continued to be developed.
- Unique Idea – You won't see too many other games like it.

What Went Wrong

- Coding – We never really had a standardized way of sharing our code, which proved troublesome later. In addition, some of the module pattern could be designed better. For instance, there is little to no use of apply, bind, and call, which would have saved us an immense amount of stress on our T H I and S keys.
- Lack of Assets – Some deeper sprites could have been used for player and the enemies, as well as background, however we simply did not have enough time and didn't realize it until we were in over our heads.
- Lack of Depth – It's no secret that this game lacks depth. Certain aspects, such as a grid system that the player would be able to explore to find different notes, as well as enemies dropping valuable notes on death, were planned features that may have helped but did not make it into the final iteration due to time constraints. Looking back, we may have spent too much time on designing the core mechanics and not enough on implementing gameplay.

Future Work

- The first order of business would be to polish the version of the game we currently have, whether it be adding more sounds, sprites, cleaning up the code structure, etc.
- From there, we can begin to plan and implement new features, such as more chords or varying types of enemies. In addition, another mechanic other than avoiding enemies and running over notes would be hugely beneficial in the spaces where the player is not planning an explosion.
- There is a large potential for this game to become an educational resource for people who wish to learn music theory. If that is the case, it's possible for the focus to be shifted to adding more key signatures as was the plan of the original concept, as well as promoting a more casual and instructional approach to the gameplay.

Non-Course Resources

Explosion Sprite Sheet – Although it was modified, the original version can be found on opengameart.org from user "Master484". The files are licensed under Creative Commons 0 (public domain):

https://opengameart.org/content/explosion-set-1-m484-games

Font "Munro" – From Ed Merritt at "Ten By Twenty". The font is free for personal and commercial use, provided that it is not redistributed or sold through said uses:

http://tenbytwenty.com/?xxxx_posts=munro

Font "Droid Sans" – From Steve Matteson, a part of the Google Fonts open source library and embedded via style tag:

https://fonts.google.com/specimen/Droid+Sans

Alec's Contributions

- All image drawing and manipulation – this includes preloading image(s), animations, particle system, photoshop editing and web optimization, etc.
- Audio design and implementation (FL Studio), exporting and hooking up to HTML audio elements and sound.js
- Sourcing and embedding fonts
- Design and creation of the data structures that were used in the storage of notes and overall design of the different modules.
- Collision detection for note pickup, adding notes to the array and algorithm to check for available chords.

John's Contributions

- Enemy design and implementation, including collision detection
- Random generation of notes and enemies over time
- Game screens, game progression logic (pause, etc.)
- Key daemon (keys.js)
- Utilization and manipulation of array data (activating chords, removing notes from inventory)

Total Collaboration – Prototype 1

- The first prototype of this project was a 100% in-person collaboration and as such I cannot speak about separation of work or duties for any code/concepts that had already been implemented in the submission of prototype 1.

Overall Grades – 90%

For both the project and peer evaluation, I would give flat 90's. There are faults with this project, and it is obvious that not every last request on the project document has been fulfilled to a T. Yet, a unique implementation of gameplay concepts through javascript using most of the concepts we have covered until the time of starting this project will be submitted. The work was distributed fairly between us and I feel like we both did an equal job at fulfilling our duties, including helping each other out when necessary.

What impresses me most about this project is the potential. We agree that with more time, this could be turned into an over-the-top portfolio piece. It's not really above-and-beyond YET, but stay tuned! This is a project that we both tried our absolute hardest on and we're immensely proud to be submitting what we are today.