Final Project Requirements

Basic Requirements:

Meeting the following basic requirements for your final project will get a 70/100.

For any requirement highlighted in blue, these requirements can be met in ***ANY state, but it must make sense to meet that requirement there***. That means you can meet the background requirement in the Win state, for example, if it flows with your game design, game mechanics, and game feel. *All other requirements must be met in the Game state.*

- At least 4 unique sprites, 2 of which must be animated.
    - You must manage 4 independently-used OAM indices at once to satisfy the requirement for 4 unique sprites.
        - They should *preferably* have unique appearances as well.
    - The 2 animated sprites must have two completely different animations, which are shown on screen at the same time.
    - "Animated" means three or more frames of a looping animation.
- Two backgrounds that move/scroll independently.
    - They must both move! (*Hint:* try parallax movement).
- Modify tile images/tilemaps at runtime.
    - Using DMA to replace what is currently in memory with a new version **does not** meet this requirement, you must make actual modifications to the charblock/screenblock memory at runtime (meaning it can't all just come from Usenti exports).
- Modify palette and/or sprite palette (either) at runtime.
    - Using DMA to replace what is currently in memory with a new version or changing the palette row of a tile/sprite **does not** meet this requirement, you must make actual modifications at runtime (meaning it can't all just come from Usenti exports).
- The following states: START, INSTRUCTIONS, GAME, PAUSE.

    - The GAME state must be implemented in a tiled mode, such as Mode 0
    - You may (but are not required to) use other modes for other states (ask your TAs for tips if you're confused).

- A WIN and/or LOSE state.
    - You must have at least one or the other.
    - You must be able to restart the game from this end state without re-running your .gba file.

- At least two sounds that can play simultaneously, of which at least one must use digital sound.
    - At least one must be looping.
    - At least one must not loop.

- A "cheat" to make the game easier.
    - The game must be winnable (if your game includes a win state) *without* the cheat.
    - The cheat should not be mistaken for normal gameplay. (ex: a powerup item that's easily accessible on the map)
    - Do *not* just skip to the win state, toggle an "invincible" mode, or remove a conditional. The cheat must meaningfully change existing mechanics or introduce an entirely new one– please be creative (and ask your TA for ideas/guidance).

- The ability to play the game correctly without being told how to do so ahead of time.
    - Aided only by the in-game instructions and menus.

- Bug-free gameplay.

- A recorded demo of gameplay (unless you volunteer for an early demo).
    - More details on this coming soon!

Subjective Requirements:

The other 30 points will be earned based on the creativity, quality, and completeness of your project.

- **Gameplay**: we prefer to see innovation; games like pong, breakout, and "collect X items" won't score very high here. A game like Frogger won't earn many more points than these. You should always strive to make something better than the

last thing you made, and we've already made games like Missile Command, Breakout, etc. Two good standards for any medium are “is it fun to play?” and “is it catchy?”. Unless explicitly worked out with your TA(s), you **CANNOT** recreate or extend an assignment already submitted for this class. Be original!

- **Quality/Completeness**: no bugs, smooth movement/transitions, natural controls, and a general feeling of polished work. Everything “makes sense” and “feels right”. To earn points here, have your friends playtest it and spend time tweaking numbers and timing. Additionally, we will consider how complete your game feels— you should scope your project appropriately and make sure you don’t bite off more than you can chew. See a TA if you feel like the size of your final project is getting out of hand.
    - We will not penalize you for not being an "artist"/being able to have perfect visuals and colors. (We don't even require you to create your own pixel art!) However, if the visual quality of your game detracts from the experience and gameplay, or makes it hard to grade your assignment, points will be taken off. Any effort put into making the visuals more appealing will definitely help you.
- **Extra Credit**: Finally, we will reward extra credit on the Final Project for extraordinary effort. This includes interesting graphical effects, use of advanced programming or GBA features (e.g. mosaic effect, malloc, or an XXL background— AKA a background that is ≥ 512 pixels in BOTH width and height), and general coolness.
    - Note: This is a programming course, not an art course. Amazing custom graphics may be rewarded, but technical achievements will be rewarded much more.

Subjective Deductions:

- **Bugs**: Up to *30 points* may be deducted for bugs. Small bugs and glitches occur even in production level games, and we understand this! As such, we are not going to play your game just to break it and find every little teensy bug that may exist. Instead, bugs that are apparent from playing the game will result in subjective deductions.

Whew, that was a lot!

This project takes all that you've learned this semester and turns it into a one-of-a-kind playable game! We want you to come out of this class with something you're proud of (show this off to your friends and family – have them play it too!). We want this class to feel meaningful and not like a waste of your time. Whether you're here to just eventually get a job or change digital media forever, here is an opportunity to draw from your inspirations and achieve something worth doing.