ChatGPT

Shellwalker

Windwalker Monk Rotation Timeline Planner

Shellwalker is an interactive planning tool for **World of Warcraft** players, specifically designed for the **Windwalker Monk** DPS specialization. It allows you to visualize and optimize your skill rotation on a timeline, helping to maximize damage output by aligning abilities, cooldowns, and buffs in an ideal sequence. With Shellwalker, you can map out your rotation for a boss fight—second by second—ensuring that major cooldowns and buff windows are utilized to their fullest potential.

Features

- **Interactive Skill Timeline:** Plan out your abilities on a visual timeline. Each skill you add is represented in chronological order with its duration (cast time or channel) shown, and abilities are grouped into categories (e.g. major cooldowns, minor cooldowns, fillers) for clarity. This gives a clear overview of when each skill will be used during a fight.
- **Cooldown Alignment & Syncing:** Easily synchronize **cooldowns** by viewing when they become available. Shellwalker displays vertical markers for cooldown completion times, so you can align big abilities (like Storm, Earth, and Fire or Invoke Xuen, the White Tiger) with each other or with specific event timings. This makes it simple to adjust your plan so that important skills are ready exactly when you need them.
- **Buff Interaction Logic:** The planner accounts for **buffs and haste effects** automatically. You can include major buffs such as Bloodlust or the Evoker’s Blessing of the Bronze in your timeline – when active, Shellwalker will speed up your casts and GCDs based on the increased haste. It also handles special Windwalker effects (like set bonuses or talents that modify skill behavior), ensuring your timeline reflects accurate ability cooldown reductions or duration changes during those buff windows.
- **Custom Boss Timelines & Phases:** Plan for real raid scenarios by overlaying **boss events and phases** on your rotation. Shellwalker provides a dedicated “Boss Timeline” track where you can mark events like boss phase transitions or major mechanics. Use the **Phases** track to segment the fight into Phase 1, 2, 3, etc. This feature helps you coordinate your cooldown usage with encounter-specific timings – for example, you might ensure Touch of Death is ready for a burn phase, or delay Fists of Fury if the boss will become untargetable.
- **SimulationCraft APL Export:** After perfecting your rotation timeline, you can export it as a **SimulationCraft APL** (Action Priority List) script. With one click on “Export SimC APL,” Shellwalker generates a sequence of actions (with properly inserted wait times) that mirrors your planned rotation. This allows you to simulate the DPS of your custom rotation in SimulationCraft or share the rotation script with others for feedback and further analysis.
- **Compact & Detailed Views:** The interface supports switching between a full detailed timeline and a simplified view. In **detailed view**, all actions (including minor fillers and auto-attacks if applicable) are shown – great for thorough planning. In **compact view**, only core abilities and major events are displayed, which helps focus on the primary rotation sequence without distraction. You can toggle between these modes depending on the level of detail you need.
- **Intelligent Rotation Enforcement:** Shellwalker is built with Windwalker Monk mechanics in mind and prevents impossible scheduling. For example, it won’t let you stack two abilities at the same timestamp (respecting the global cooldown), and it will warn you if you try to use an ability before its cooldown is ready (displaying a “CD not ready” warning or automatically adjusting the

timing to the next available moment). It also respects channeling rules – if an ability (like Fists of Fury) is channeling, the tool ensures no other spell is placed during its channel time. This intelligent logic saves you from manual guesswork and reflects a realistic sequence of actions.

- **User-Friendly Interface:** Shellwalker runs as a web application with a clean React-driven UI. It features drag-and-drop editing – you can **drag** abilities along the timeline to reposition them, or **right-click** an entry to edit or remove it. Timeline navigation (zooming and panning) is supported for examining different fight lengths, and a **Light/Dark mode switch** is available to suit your visual preference. The interface is available in both English and Chinese, making it accessible to a global audience.

Installation and Setup

Shellwalker is a **Node.js**-based web application. To set up and run the project locally, ensure you have Node.js (version 18+ recommended) and npm (or yarn/pnpm) installed on your system. Then follow these steps:

1. **Clone the Repository:** Download or clone the `Shellwalker` repository to your local machine.
2. **Install Dependencies:** In the project directory, run the command:

   ```
   npm install
   ```

   This will install all required packages (React, Vite, etc.) as listed in the project's package.json. (You may use `yarn` or `pnpm install` if you prefer, as the project supports those as well.)
3. **Run the Development Server:** Start a local dev server by running:

   ```
   npm run dev
   ```

   This will launch Vite's development server. By default, it should tell you the local address (typically http://localhost:5173) where the application is running. Open that address in your web browser to access Shellwalker. You should see the Shellwalker interface load in the browser.
4. **(Optional) Build for Production:** If you want to host the app or test the optimized build, use:

   ```
   npm run build
   ```

   This will compile the app into static files in the `dist/` directory. You can then serve those with any static file server (or use `npm run preview` to test the production build locally).

Environment Requirements: Shellwalker has been tested with Node.js 18+ and modern browsers (Chrome, Firefox, Edge). Ensure your browser is up to date for the best performance. The app uses modern JavaScript and WebGL for timeline visuals, so using an updated browser is important. No special OS-specific setup is required—any system capable of running Node and a web browser (Windows, macOS, Linux) should work.

Usage Guide

Once you have the application running in your browser, you can start creating and optimizing your Windwalker Monk rotation timeline. Below is a typical workflow for using Shellwalker effectively:

1. **Add Abilities to the Timeline:** Begin by adding your Monk's abilities into the timeline. You might have an empty timeline at the start (or a basic opener pre-loaded). Use the interface controls to insert abilities – for example, there may be a menu or list of Windwalker abilities (Tiger Palm, Rising Sun Kick, Fists of Fury, etc.) that you can click or drag onto the timeline. Each added ability will appear as a block on the timeline, positioned at the time you specify (defaulting to the end of the last action if added sequentially). Shellwalker automatically places each ability in the appropriate group track (e.g., major cooldowns, fillers) and displays its duration bar corresponding to cast time or channel time.

2. **Adjust Timing via Drag-and-Drop:** Fine-tune the schedule by dragging abilities left or right along the timeline. As you adjust, pay attention to the timeline grid and spacing – Shellwalker snaps moves to reasonable increments (aligning to seconds or specific time steps) to make timing easier. When you move an ability, the tool will enforce cooldown and GCD rules. For example, if you try to overlap two abilities too closely, Shellwalker will push them apart to respect the 1.0-second global cooldown (or a shorter GCD if haste is active). If an ability is moved before its cooldown is finished (based on a previous use), the timeline will highlight this conflict or automatically delay that ability until its cooldown completion. This interactive adjustment lets you experiment with different sequences quickly – for instance, you can try swapping **Rising Sun Kick** and **Blackout Kick** order, or delaying **Fists of Fury** by a second to see if it aligns better with a buff window.

3. **Incorporate Buffs and Haste Effects:** To simulate realistic combat scenarios, add any relevant **buffs** or external effects into the timeline. For instance, if your raid will use **Bloodlust** at 0:00 or lust at a later time, you can indicate that on the timeline (typically by adding a buff bar in the Bloodlust track starting at the appropriate time and lasting 40 seconds). Similarly, if you have the Evoker's **Blessing of the Bronze** (which grants a 15% haste aura for a few seconds periodically), you can add those in the Blessing track. Shellwalker will adjust the cast times and cooldown speeds during those intervals automatically (you'll notice ability bars shrink in length during Bloodlust, reflecting faster casts). The **Haste** track can also be used to visualize your base haste or temporary haste fluctuations. By modeling buffs, you ensure your rotation takes full advantage of increased attack speed or reduced cooldowns. (Shellwalker's buff logic covers stacking cases too – e.g., if Bloodlust and Blessing overlap, their combined effect on haste is accounted for.)

4. **Mark Boss Phases (Optional):** If you want to tailor your rotation to a specific boss fight, use the **Boss Timeline** and **Phases** tracks to mark encounter events. You can insert custom markers or labels for things like "Phase 2 starts at 60s" or "Boss becomes untargetable at 90s for 10s". For example, you might add a Phase marker on the Phases track at 60 seconds to denote Phase 2, and perhaps another at 120 seconds for Phase 3. Likewise, on the Boss Timeline track, you could label an event like "Boss casts big AoE" or "Add spawns" at the times those occur. These visual cues on the timeline help you adjust your rotation – you might decide to save Storm, Earth, and Fire for after an immunity phase, or plan your Serenity usage to line up with a specific boss vulnerability window. By planning around phases, you ensure that your burst is ready at critical moments of the fight.

5. **Review Cooldown Synchronization:** As you build your timeline, regularly check how your **cooldowns** line up. Shellwalker's vertical cooldown indicators (blue or gray vertical lines dropping down across tracks) show when an ability's cooldown finishes. For instance, after you use Rising Sun Kick, you'll see a marker 10 seconds later (or less, with haste) indicating when it will be available again. Use this information to line up subsequent uses or to decide when to delay an ability for better synergy. A common optimization is to align your heavy-hitters with each other: e.g., ensuring that Fists of Fury comes off cooldown just as Dance of Chi-Ji (buff) procs, or timing Invoke Xuen to coincide with Weapons of Order if that were relevant. In the timeline, these alignments become visually apparent. Adjust the positioning of abilities until you are satisfied that all major cooldowns, buffs, and procs overlap in the most advantageous way. The goal is a smooth sequence with minimal downtime and maximum buffed uptime for your key skills.

6. **Export and Simulate:** Once you're happy with the planned rotation, you can export it for simulation or sharing. Click the **"Export SimC APL"** button – Shellwalker will generate a text output following SimulationCraft's Action Priority List syntax. This output lists your abilities in the exact order and timing you planned (including `wait` commands for delays between actions). Copy this APL script into SimulationCraft to **simulate** your DPS with the planned rotation. This is a powerful way to validate your plan – you can compare DPS results of different timeline configurations to see which yields the highest output. It's also useful for sharing your rotation with others: other theorycrafters can run the same APL or tweak it further. (Note: When simulating, ensure the SimC settings like fight length and conditions match what you assumed in your timeline for best results.)

Throughout the process, feel free to tweak and iterate. Shellwalker is meant to be interactive – add, remove, and shift abilities as needed. You can zoom the timeline view in or out (using your mouse wheel or trackpad) to get a detailed or high-level view of the rotation, and pan across longer fight durations. The **View Range** control lets you quickly jump to specific time spans (e.g., focusing on the first 30 seconds for an opener, or viewing the full 5-minute plan). The bottom line: experiment with different strategies – Shellwalker will faithfully reflect the timing, so you can theorycraft the optimal rotation before trying it in-game.

Advanced Usage & Customization

While Shellwalker works out-of-the-box for Windwalker Monks with the current game data, advanced users or developers might want to customize it further. Below are some advanced topics:

Adding New Skills or Talents

As the game evolves, new abilities or talents might be introduced (or existing ones may change). Shellwalker is built with a data-driven approach, so updating it for new skills is straightforward for those familiar with code. Windwalker Monk abilities and their properties (cooldowns, energy/Chi cost, cast times, etc.) are defined in the project's data files – for example, there is a JSON file that contains the list of monk spells and their stats. If a new ability needs to be added, you can edit this spells database (e.g. add the new spell with its ID, name, cooldown, etc. in `monk_spells.json`) and Shellwalker will include it in the planner. The application logic (written in TypeScript/React) will automatically incorporate the new spell if its data is present. In some cases, you might also need to update logic for special interactions (for instance, if a new talent modifies how two spells interact, you may add a check in the logic or add a buff effect in the data files). The codebase is modular and commented (with many unit tests covering the timing logic), so implementing and testing such changes can be done with minimal friction. **Tip:** after adding or changing spells, run `npm run dev` to launch the app and verify the new skills appear and

function correctly. If you have the test suite set up, you can run `npm test` to ensure that all timing and cooldown calculations still pass with your changes.

Utilizing Boss Timeline Overlays

The **Boss Timeline** feature is especially powerful for advanced encounter planning. To use it effectively, click on the boss timeline track in the UI or open the Boss Timeline Options panel (if provided). Here, you can create custom events that mirror your raid's strategy or a specific boss's script. For example, you might input a sequence for a raid boss like: "Boss casts Deadly Attack at 45s, Soft Enrage at 3:00, Adds spawn at 1:30 and 2:30, Boss becomes immune from 2:45 to 2:55," etc. Each event can be given a label and a time, and it will appear on the Boss Timeline track as a marker or a bar (for a duration event like immunity phases). By laying out these events, you essentially overlay the **encounter timeline** on top of your rotation timeline. Advanced players can then adjust their ability usage around these events – e.g., delaying a cooldown if an immunity is coming up, or planning a burst sequence during a vulnerability window. Shellwalker doesn't come pre-loaded with every boss's timeline (that level of detail is usually up to the user), but it provides the tools for you to input them. If you have a text file or known schedule of boss abilities, you can translate that into Shellwalker events. This level of customization ensures that your optimized rotation isn't just ideal in a vacuum, but also practical and optimized for the actual boss fight conditions you'll face.

With Shellwalker, Windwalker Monk players and theorycrafters have a powerful sandbox to theorycraft "**what-if**" scenarios. Whether you're working on the perfect opener, testing how a new trinket or set bonus fits into your rotation, or planning cooldown usage for a progression boss, this tool gives you a clear visual timeline to experiment on. We hope this README helps you get the most out of Shellwalker. Happy planning, and may your fists forever flurry in the most optimal sequence!
