6v6 Adjustments

- 6v6 Adjustments Project
- Recent Patch Compendium
- Contribution Guide

6v6 Adjustments Project

Welcome to the 6v6 Adjustments Project, an innovative Overwatch Workshop gamemode designed to bring back the classic 6v6 format. This project aims to enhance the gameplay experience by making thoughtful adjustments to the game, providing a balanced and engaging environment for all players.

About the 6v6 Adjustments Project

The 6v6 Adjustments Project is a community-driven initiative that revives the traditional 6v6 format in Overwatch, offering players a nostalgic yet refined gameplay experience. By leveraging the Overwatch Workshop, we have developed custom game modes and tweaks that ensure a fair and enjoyable match for everyone.

Key Features

- **6v6 Format:** Experience the original team dynamics and strategies with six players on each team, just like the early days of Overwatch.
- **Balanced Gameplay:** Our team continuously monitors and adjusts the game settings to maintain balance and fairness, ensuring that every match is competitive and fun.
- **Community Involvement:** We value feedback from our players and encourage community participation in the development process. Join our Discord to share your thoughts and suggestions.

How to Join

- **Gamemode Code:** To play the 6v6 Adjustments gamemode, use the code `7HSKW` in the Overwatch Workshop.
- **Google Doc Notes:** Google Doc
- **Patch Notes:** Patch notes
- **Discord:** Join our community on Discord at 6v6.info for real-time updates, support, and to connect with other players.
- **GitHub:** Follow the development and contribute to the project on our GitHub repository at https://github.com/6v6-Adjustments/6v6-adjustments.

Why Play 6v6 Adjustments?

Continuous Improvement

Our dedicated team is committed to constantly improving the gamemode based on player feedback and the latest game updates, ensuring a balanced and enjoyable experience.

Community Driven

Be a part of a passionate community that shares your love for Overwatch. Your feedback and participation help shape the future of the 6v6 Adjustments Project.

Join Us Today

Rediscover the excitement of 6v6 Overwatch with the 6v6 Adjustments Project. Connect with fellow players, enjoy balanced gameplay, and contribute to a project that values community input and continuous improvement.

For more information and to stay updated, join our Discord at 6v6.info and check out our GitHub repository at https://github.com/6v6-Adjustments/6v6-adjustments.

We look forward to seeing you in the game!

Recent Patch Compendium

https://docs.google.com/document/d/15nnv5-FbSJ302Fnxdirq2BnilNJFT2p3PrarX_j5gUA/edit

Audit the Code

Contribution Guide

If you're interested in submitting code to this gamemode, read this guide carefully.

Required Tools

- **Visual Studio Code**
  - Text Editor for developing OverPy source code
- **OverPy**
  - VScode extension for compiling OverPy source code -> workshop script
- **Git**
  - Version control system
- **GitHub**
  - Remote archive of this source code repository
  - Issue tracking and Pull Request
- **Discord**
  - Communication between developers

Setup Instructions

1. Download Visual Studio Code.
2. Go to extension tab in VScode and install OverPy extension.
3. Clone the repository to your local computer (use GitHub Desktop if you don't know how to clone using terminal).
4. Open cloned repository folder on VS Code.
5. Try building gamemode using instructions below.
6. Follow general workflow to start development.

Building

For instructions on how to use OverPy, see here

1. Compile `main.opy` using the compiling instructions at OverPy Wiki.
2. Open a custom game in Overwatch.
3. Paste the compiled gamemode code.

General workflow

1. Find an issue to fix (or submit one yourself).
2. Create new branch dedicated to that issue.
3. Modify codebase using VScode. Be sure to follow styling guidelines.
4. Compile and test new codebase.
5. Commit working changes.
6. Submit Pull Request (PR) to merge your branch into `dev` branch.
7. Wait for owner to approve and merge your Pull Request.

Release Cycle

- For every 3~5 features added to staging branch, the devs will compile a staging build and test each feature manually.
- If your feature fails to meet quality expectations, the MR corresponding to that feature will be reverted to drop the changes.
- After each feature in staging branch is confirmed, the staging branch will be merged to main branch and released as latest build. If the latest build plays without crashing, it will be set as the stable build.

Coding Style Guidelines

This codebase follows Python's PEP 8 style guide since OverPy follows Python syntax.

- Variable names should be snake_case. Example: `hero_health_armor`
- Function names should be camelCase. Example: `destroyBarrier()`
- File names should be snake_case. Example: `custom_heroes.opy`
- No magic numbers. Constants should always be referred to by the names defined in `src/constants`.
- Rule and subroutine names should follow the format rule `"[file_name.opy]: My custom rule"` and `@Name "[file_name.opy]: mySubroutineFunc()"` respectively.
- Keep rules simple; each rule should only perform one task. Try to limit rules to at most 10 lines of code and group large blocks of code into subroutines whenever possible. See `src/heroes/bastion` as an example.