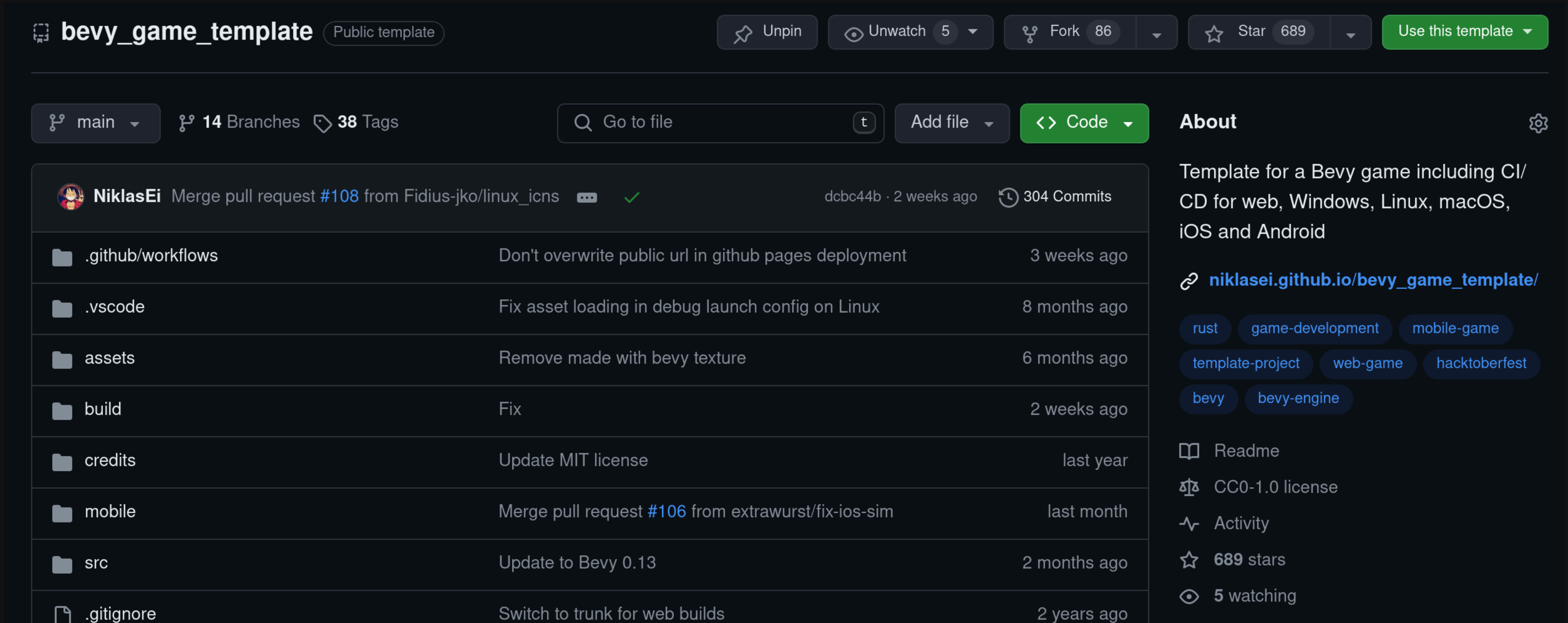
Learnings from a Bevy game template

A third-party starting point for Bevy apps

- cross-platform
- CI/CD
- extendable



NiklasEi Merge pull request #108 from Fidius-jko/linux_icns ✓ dcbc44b · 2 weeks ago 304 Commits

.github/workflows	Don't overwrite public url in github pages deployment	3 weeks ago
.vscode	Fix asset loading in debug launch config on Linux	8 months ago
assets	Remove made with bevy texture	6 months ago
build	Fix	2 weeks ago
credits	Update MIT license	last year
mobile	Merge pull request #106 from extrawurst/fix-ios-sim	last month
src	Update to Bevy 0.13	2 months ago
.gitignore	Switch to trunk for web builds	2 years ago

About

Template for a Bevy game including CI/CD for web, Windows, Linux, macOS, iOS and Android

niklasei.github.io/bevy_game_template/

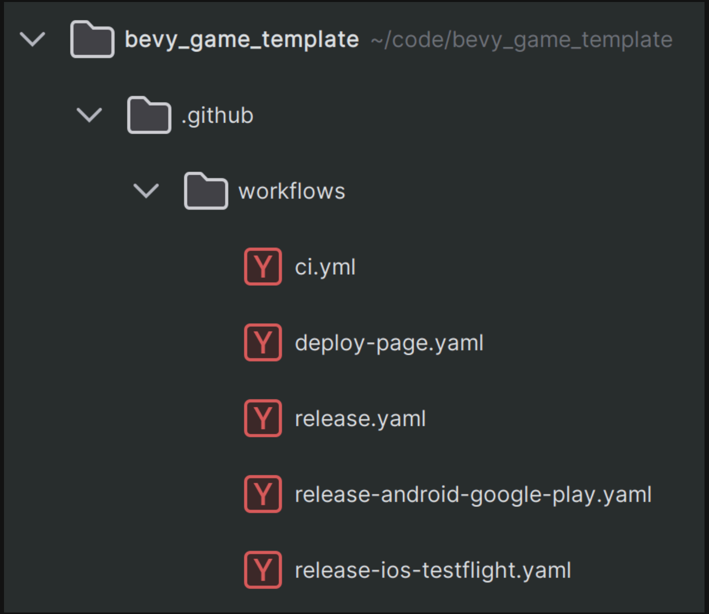
rust · game-development · mobile-game · template-project · web-game · hacktoberfest · bevy · bevy-engine

Cross-platform setup

- Support for all "official" Bevy platforms
- Platform specific files for packaging
- Minimal structure with two crates

Automate what you can

- Simple CI pipeline
- Build pipeline for all target platforms
- Publish pipelines for web, android[1] and iOS[2]



```
bevy_game_template ~/code/bevy_game_template
  .github
    workflows
      ci.yml
      deploy-page.yaml
      release.yaml
      release-android-google-play.yaml
      release-ios-testflight.yaml
```

[1] https://www.nikl.me/blog/2023/github_workflow_to_publish_android_app/

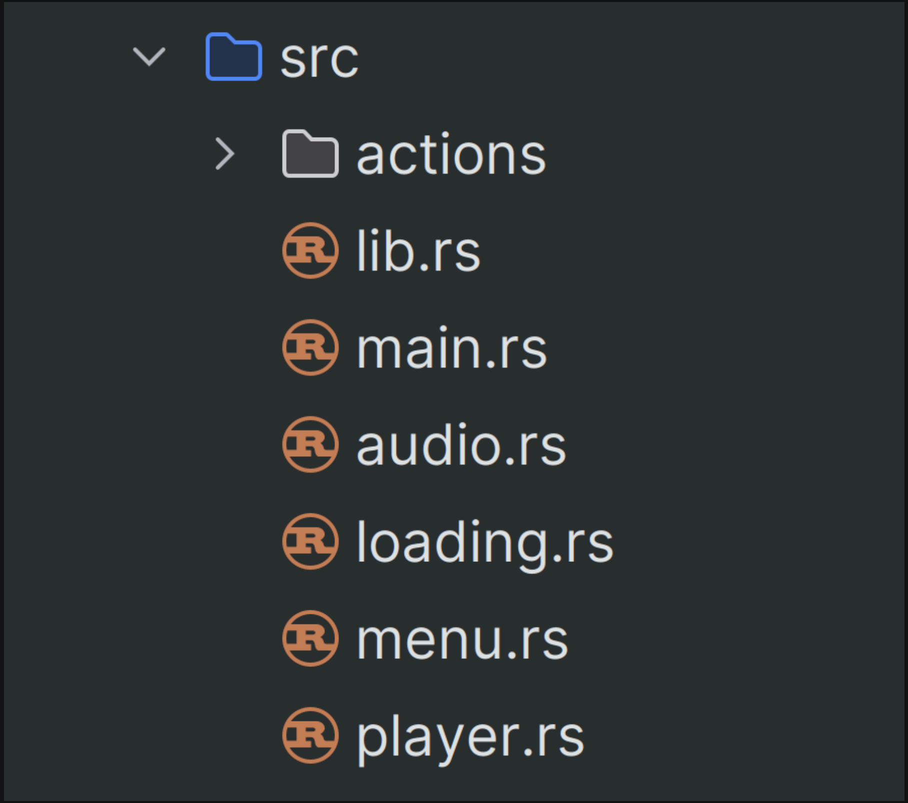
[2] https://www.nikl.me/blog/2023/github_workflow_to_publish_ios_app/

Extendable

- Embrace Bevy plugins for code organisation
- It’s a workspace; You can add more crates
- Plugins are organized by domain

Project structure

Try to cut plugins by domain



```
src
  actions
  lib.rs
  main.rs
  audio.rs
  loading.rs
  menu.rs
  player.rs
```

Project structure - example audio.rs

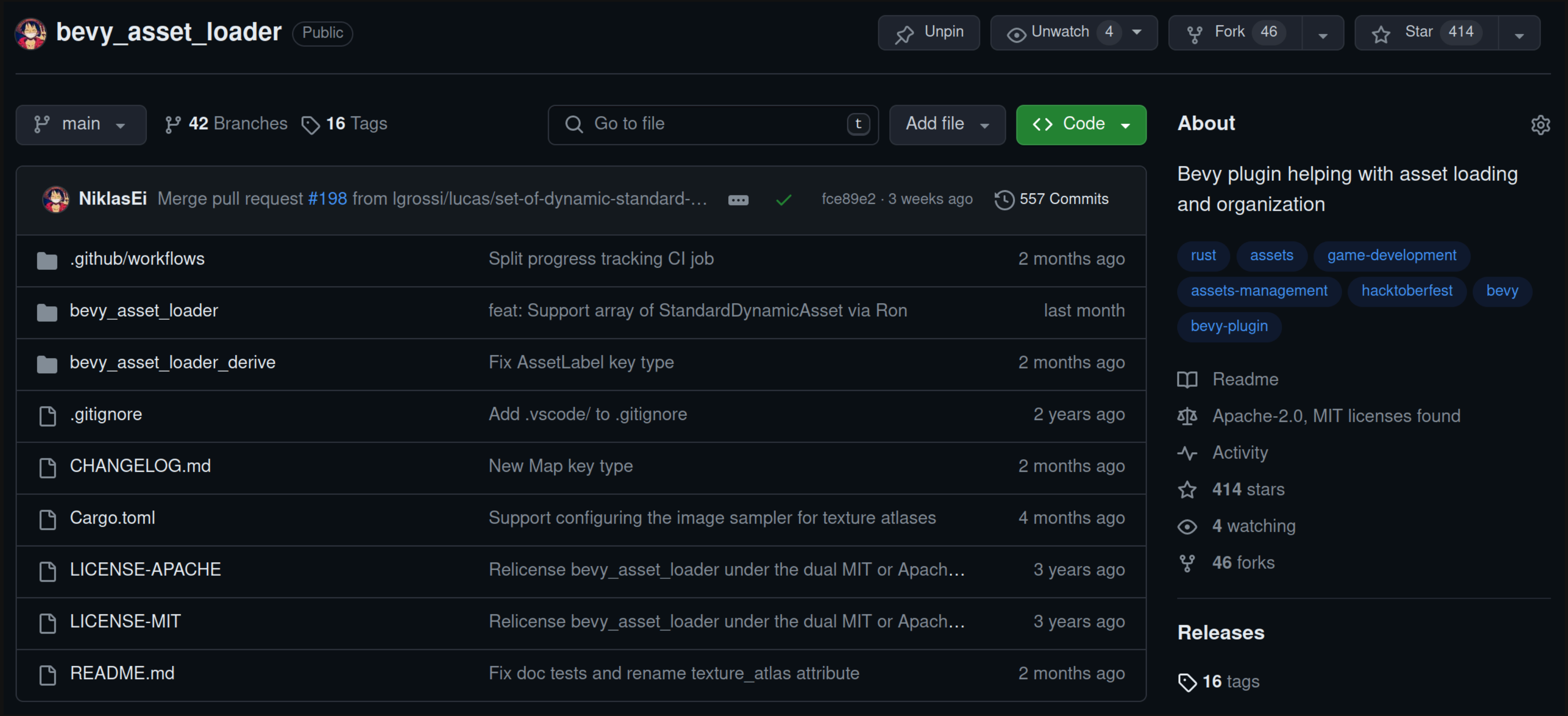
```rust
pub struct InternalAudioPlugin;

// This plugin is responsible for controlling the game audio
impl Plugin for InternalAudioPlugin {
    fn build(&self, app: &mut App) {
        app.add_plugins(AudioPlugin)
            .add_systems(OnEnter(GameState::Playing), start_audio)
            .add_systems(
                Update,
                control_audio
                    .after(set_movement_actions)
                    .run_if(in_state(GameState::Playing)),
            );
    }
}

#[derive(Resource)]
struct FlyingAudio(Handle<AudioInstance>);

fn start_audio(mut commands: Commands, audio_assets: Res<AudioAssets>, audio: Res<Audio>) { ... }

fn control_audio(actions: Res<Actions>, audio: Res<FlyingAudio>, mut instances: ResMut<Assets<AudioInstance>>) { ... }
```

Project structure - example (old; Bevy 0.5) loading.rs

```rust
pub struct LoadingPlugin;
impl Plugin for LoadingPlugin {
    fn build(&self, app: &mut AppBuilder) {
        app.add_system_set(SystemSet::on_enter(GameState::Loading).with_system(start_loading.system()))
           .add_system_set(SystemSet::on_update(GameState::Loading).with_system(check_state.system()));
    }
}
struct LoadingState { handles: Vec<HandleUntyped> }

pub struct AudioAssets { pub flying: Handle<AudioSource> }
pub struct TextureAssets { pub texture_bevy: Handle<Texture> }

fn start_loading(mut commands: Commands, asset_server: Res<AssetServer>) {
    let handles = vec![asset_server.load_untyped("flying.ogg"), asset_server.load_untyped("sprite.png")];
    commands.insert_resource(LoadingState { handles });
}

fn check_state(mut commands: Commands, mut state: ResMut<State<GameState>>, server: Res<AssetServer>, loading_state: Res<Lo
    if LoadState::Loaded != server.get_group_load_state(loading_state.handles.iter().map(|handle| handle.id)) { return; }
    commands.insert_resource(AudioAssets { flying: server.get_handle("flying.ogg") });
    commands.insert_resource(TextureAssets { texture_bevy: server.get_handle("sprite.png") });
    state.set(GameState::Menu).unwrap();
}
```

Some patterns deserve their own crate

Cutting "internal" plugins by domain makes it easy to move them to other crates



bevy_asset_loader

NiklasEi Merge pull request #198 from lgrossi/lucas/set-of-dynamic-standard-… fce89e2 · 3 weeks ago 557 Commits

.github/workflows	Split progress tracking CI job	2 months ago
bevy_asset_loader	feat: Support array of StandardDynamicAsset via Ron	last month
bevy_asset_loader_derive	Fix AssetLabel key type	2 months ago
.gitignore	Add .vscode/ to .gitignore	2 years ago
CHANGELOG.md	New Map key type	2 months ago
Cargo.toml	Support configuring the image sampler for texture atlases	4 months ago
LICENSE-APACHE	Relicense bevy_asset_loader under the dual MIT or Apach…	3 years ago
LICENSE-MIT	Relicense bevy_asset_loader under the dual MIT or Apach…	3 years ago
README.md	Fix doc tests and rename texture_atlas attribute	2 months ago

About

Bevy plugin helping with asset loading and organization

rust · assets · game-development · assets-management · hacktoberfest · bevy · bevy-plugin

- Readme
- Apache-2.0, MIT licenses found
- Activity
- 414 stars
- 4 watching
- 46 forks

Releases

16 tags

Project structure - loading.rs with bevy_asset_loader

```rust
pub struct LoadingPlugin;
impl Plugin for LoadingPlugin {
    fn build(&self, app: &mut App) {
        app.add_loading_state(
            LoadingState::new(GameState::Loading)
                .continue_to_state(GameState::Menu)
                .load_collection::<AudioAssets>()
                .load_collection::<TextureAssets>(),
        );
    }
}

#[derive(AssetCollection, Resource)]
pub struct AudioAssets {
    #[asset(path = "audio/flying.ogg")]
    pub flying: Handle<AudioSource>,
}
#[derive(AssetCollection, Resource)]
pub struct TextureAssets {
    #[asset(path = "textures/bevy.png")]
    pub bevy: Handle<Image>,
    #[asset(path = "textures/github.png")]
    pub github: Handle<Image>,
}
```

cross-platform things in code

No console on Windows [1]

Add the following to your `` `main.rs` `` file:

```rust
#![cfg_attr(not(debug_assertions), windows_subsystem = "windows")]
```

[1] https://bevy-cheatbook.github.io/platforms/windows.html#disabling-the-windows-console

Icons everywhere

- `` `build.rs` `` file for exe icon [1]
- `` `window.set_window_icon()` `` to set the icon in the task bar/window on Windows and X11 Linux [2]

[1] https://bevy-cheatbook.github.io/platforms/windows.html#setting-the-exe-icon

[2] https://bevy-cheatbook.github.io/window/icon.html