

BugsBuzzy

BugsBuzzy is a small 2D platformer/demo made with the Godot Engine. It contains scenes, scripts, sprites and audio assets to demonstrate basic player movement, enemies, damage areas, menus and level/end conditions.

Preview

- Scenes are located under `Scenes/` (including `Main.tscn`, `player.tscn`, `Enemy.tscn`).
- Scripts are under `Scripts/` and `Scenes/` (project and scene logic are implemented in GDScript `.gd` files).
- Sprites and audio are in `Sprites/`.

Requirements

- Godot Engine (recommended: Godot 3.5+ or Godot 4.x). If you know the exact engine version used to create the project, open `project.godot` in a text editor to confirm and use that version.
- A desktop platform for running the editor (Linux, Windows, macOS).

Notes: I made a conservative compatibility assumption (Godot 3.5+ / Godot 4.x). If you see any version-specific issues, I can update this README with exact instructions for the project version.

Run the project (editor)

1. Install Godot if you don't have it already: <https://godotengine.org/download>
2. Open Godot and choose "Import" or "Open" and select the folder containing `project.godot` (the project root).
3. Once the project loads, open the main scene `Main.tscn` (or `Main.tscn` under `Scenes/`) and press Play (F5) to run.

Run an exported build (Linux)

—
PROF

This repository contains an exported package file and a shell script: `Gaol Creeper.pck` and `Gaol Creeper.sh` — they may be an exported build or helper files. Typical ways to run a `.pck` export:

- If you have a Godot binary, you can run a `.pck` with:

`godot --main-pack "Gaol Creeper.pck"`
- If `Gaol Creeper.sh` is an included self-contained runner, make it executable and run it:

`chmod +x "Gaol Creeper.sh"`
`./"Gaol Creeper.sh"`

Adjust the commands above to the specific Godot binary you have (e.g., `godot`, `godot4`, or a full path to the executable).

Project structure

- `project.godot` — Godot project file.
- `Scenes/` — main scenes and UI scenes (`Main.tscn`, `player.tscn`, `Enemy.tscn`, menus, etc.).
- `Scripts/` — reusable script code (e.g., `character_body_2d.gd`).
- `Sprites/` — image and audio assets used by the game.
- `.godot/` — editor metadata (ignored by version control in many projects).
- `Gaol Creeper.pck`, `Gaol Creeper.sh` — included exported package and runner (if applicable).

Key scene & script files to look at when editing game logic:

- `Scenes/Main.tscn` — the top-level scene (entry point).
- `Scenes/player.tscn` and `Scripts/character_body_2d.gd` — player node and movement code.
- `Scenes/Enemy.tscn` and `Scenes/enemy.gd` — enemy logic.
- `Scenes/game_state.tscn` and `game_state.gd` — overall game state and level flow.

Contributing

If you want to improve the project:

1. Fork or branch the repo.
2. Make changes (scenes, scripts, assets).
3. Test locally in Godot.
4. Open a pull request with a short description of your changes.

Small, low-risk improvements I can help with: update engine version in README, add a screenshot, add a small test scene, or tidy scripts and add inline comments.

Troubleshooting

- If scenes fail to open: confirm the Godot editor version matches the project version in `project.godot`.
- Missing resource errors: run the project from the project root so Godot resolves relative paths correctly.
- Scripts crash: open the Output/Debugger panel in Godot to see the error and stack trace.

PROF
—
If you want, I can:

- Detect the exact Godot version used by the project and update this README with precise run/export instructions.
- Add a screenshot to the README.
- Add a short contribution guide or a GitHub Actions workflow to run simple checks.

License

No license file is included in the repository. If you want a specific license (MIT, Apache-2.0, GPL, etc.), tell me and I can add a `LICENSE` file with recommended text.

If you'd like any changes to the README (more technical detail, screenshots, engine-version specifics, or contributor rules), tell me what to include and I'll update it.