

Novellium

A modern, web-based visual novel engine with cloud integration, visual builder, and seamless game sharing capabilities. Create, play, and share interactive stories without any setup required.

Author: James Hill

Final year RMIT student building pet projects in my spare time.

Features

Visual Builder

- **Intuitive Interface:** Easy-to-use forms for creating stories
- **Real-Time Preview:** See your story as you build it
- **Character Management:** Create characters with sprites and emotions
- **Asset Upload:** Support for images, audio, and multiple file formats
- **Export System:** Package games as ZIP files for sharing

Cloud Integration

- **Anonymous Sharing:** Upload games to the cloud without registration
- **Global Library:** Discover games shared by other creators
- **Automatic Asset Hosting:** Images and assets stored securely
- **Dual Mode:** Support both local and cloud game deployment

Core Engine

- **Modern Web Technology:** Built with vanilla JavaScript
- **Responsive Design:** Works on desktop, tablet, and mobile
- **Save System:** Automatic and manual save/load functionality
- **Event-Driven:** Flexible story progression with choices and branching
- **Character System:** Dynamic character sprites with multiple expressions

Quick Start

Option 1: Use Online (Recommended)

Visit novellium.vercel.app to start playing and creating immediately.

Option 2: Local Development

```
# Clone the repository
git clone https://github.com/SpeedyDuck790/Novelluim.git
cd Novelluim

# Start local server (choose one method)
npx http-server -p 8000 -c-1
```

```
# OR
python -m http.server 8000
# OR
npm install -g http-server && http-server -p 8000 -c-1
```

Open <http://localhost:8000> in your browser.

How to Use

Playing Games

1. **Browse Library:** View available games on the home page
2. **Click to Play:** Select any game to start playing immediately
3. **Save Progress:** Use manual saves or rely on auto-save functionality
4. **Import Games:** Drag & drop ZIP files to add new games

Creating Games

1. **Open Builder:** Click "Builder" or visit </build.html>
2. **Set Game Info:** Add title, author, and description
3. **Create Characters:** Add characters with names, colors, and sprites
4. **Build Story:** Create events and connect them with choices
5. **Test Locally:** Use "Deploy Local" to test your game
6. **Share to Cloud:** Use "Deploy Cloud" to share globally

Project Structure

```
Novelluim/
├── index.html           # Main library page (play games)
├── build.html           # Game builder interface
├── styles.css           # Global styles and themes
├── README.md            # This documentation
├──
├── src/                 # Core engine code
│   ├── engine.js        # Main game engine
│   ├── managers/        # System managers
│   ├── models/          # Data models
│   └── ui/              # User interface components
├──
├── config/              # Configuration files
│   └── games-list.json  # Registry of available games
├──
├── gamefolder/          # Game storage directory
│   ├── adventure-game/  # Example adventure game
│   └── dating-game/     # Example dating sim
├──
├── scripts/             # Utility scripts
├── docs/                 # Documentation
└── NovelliumLogo/       # Brand assets and icons
```

Game Format

ZIP Structure

```
game-name.zip
├─ config.json          # Game configuration
├─ characters.json      # Character definitions
├─ story.json           # Events and story flow
└─ assets/              # Game assets
    ├─ backgrounds/     # Background images
    ├─ sprites/         # Character sprites
    └─ audio/           # Sound effects and music
```

Event Types

- **Dialogue:** Characters talking with speech bubbles
- **Narration:** Story text without a character
- **Choice:** Let players make decisions that affect the story
- **Scene:** Change backgrounds, music, or character sprites

Development

Adding New Features

1. **Frontend:** Modify engine or UI components in `src/`
2. **Builder:** Update `build.html` for creation tools
3. **Documentation:** Update relevant docs in `docs/`

Testing

- **Local Games:** Test with example games in `gamefolder/`
- **Builder:** Create test stories using the visual builder
- **Cross-Platform:** Test on different devices and browsers

Cloud Features (Advanced)

- **Database:** Supabase PostgreSQL for game storage
- **CDN:** Automatic asset delivery via Supabase Storage
- **API:** Vercel serverless functions for game management

Contributing

1. Fork the repository
2. Create a feature branch: `git checkout -b feature-name`
3. Make your changes and test thoroughly
4. Commit with descriptive messages

5. Push and create a Pull Request

Development Guidelines

- **Code Style:** Use consistent formatting and meaningful names
- **Documentation:** Update README and docs for new features
- **Testing:** Test both local and cloud functionality
- **Backwards Compatibility:** Maintain compatibility with existing games

Documentation

- **README:** Overview and quick start (this file)
- **User Guide:** Complete walkthrough for building visual novels
- **Game Format:** Technical specifications and advanced features

Access documentation through the Settings menu in the application.

License

This project is open source. Feel free to use, modify, and distribute according to the license terms.

Support

- **Issues:** Report bugs via GitHub Issues
- **Documentation:** Check [docs/](#) folder for detailed guides
- **Examples:** Study games in [gamefolder/](#) for reference

Future Goals

- **Include Audio**
- **typewriter and sound settings beyond dummy parts**
- **preview in builder**
- **chatboard**

Novellium - Empowering storytellers to create and share interactive narratives effortlessly.