



Stake Engine Game Builder

Professional Casino Game Development Platform




Build complete casino games for stake.com and stake-engine.com with a visual interface. Supports slots, table games, dice games, and more with full math modeling, RTP simulation, and compliance features.





What's Included

This package contains everything you need to build desktop and mobile applications:

Desktop Applications

-  **Windows** - .exe installer with NSIS
-  **macOS** - .dmg and .app bundle (Intel & Apple Silicon)
-  **Linux** - AppImage, .deb, .rpm, and Snap packages

Mobile Applications (Optional)

-  **iOS** - Native app for iPhone/iPad
 -  **Android** - APK and AAB for Google Play
-



Quick Start

Prerequisites

- **Node.js** v18 or higher ([Download](#))
- **npm** 9.0 or higher (included with Node.js)
- **Git** (optional, for version control)

Installation

1. Download All Files

- Copy all artifact files into your project directory
- Ensure directory structure matches the guide

2. Create Project Structure

```
bash
```

```
mkdir stake-engine-builder
```

```
cd stake-engine-builder
```

3. Copy Files

- `package.json` → Root directory
- `vite.config.js` → Root directory
- `index.html` → Root directory
- `build-all.sh` → Root directory (Mac/Linux)
- `build-all.bat` → Root directory (Windows)
- `electron/main.js` → Create electron folder
- `electron/preload.js` → electron folder
- `src/App.jsx` → Create src folder (your game builder component)
- `src/main.jsx` → src folder
- `src/index.css` → src folder

4. Install Dependencies

```
bash
```

```
npm install
```

5. Test Development Mode

```
bash
```

```
# Web version
```

```
npm run dev
```

```
# Desktop version
```

```
npm run electron:dev
```

6. Build Production

Windows Users:

```
batch
```

```
build-all.bat
```

Mac/Linux Users:

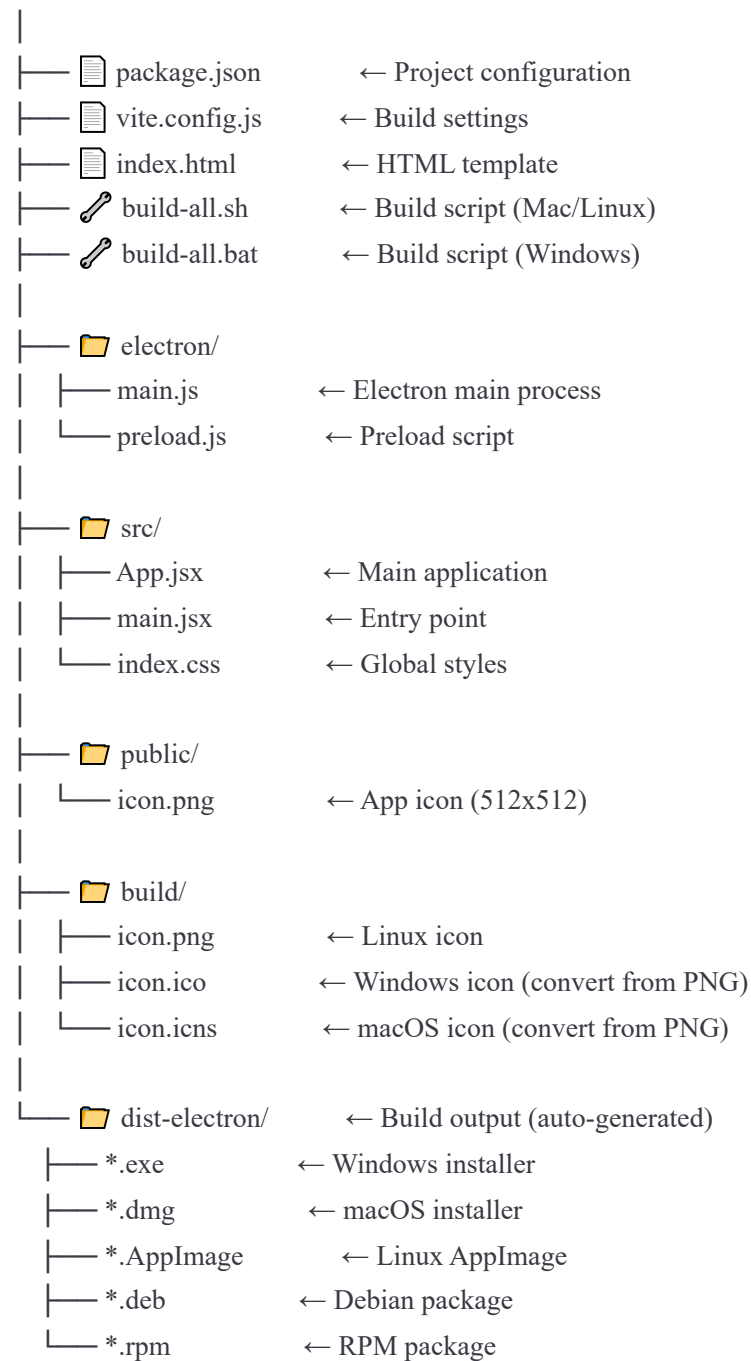
```
bash
```

```
chmod +x build-all.sh
```

```
./build-all.sh
```

📁 File Structure

stake-engine-builder/



Build Commands

Development

bash

`npm run dev` *# Start Vite dev server (web)*

`npm run electron:dev` *# Start Electron in dev mode*

Production Builds

bash

`npm run build` *# Build frontend only*

`npm run build:win` *# Build Windows installer*

`npm run build:mac` *# Build macOS DMG (Mac only)*

`npm run build:linux` *# Build Linux packages*

`npm run build:all` *# Build all platforms*

Automated Build

bash

Mac/Linux

`./build-all.sh`

Windows

`build-all.bat`

Build specific platform




`./build-all.sh --windows`

`./build-all.sh --mac`

`./build-all.sh --linux`

Features

Game Development

-  50+ Game Templates (Slots, Table, Dice, Wheel, Casual)
-  Visual Math Engine with RTP Configuration
-  Symbol & Paytable Designer

- 🎬 Animation & Graphics Configuration
- 🔊 Audio System with Sound Library
- 🎁 Bonus Game Configuration
- 📊 RTP Simulator (1M+ spins)
- 🛠️ Testing & Verification Suite

Professional Tools

- 📈 Analytics & Metrics Dashboard
- 🔒 Compliance & Responsible Gaming
- 🌐 Multi-language Support (6+ languages)
- 💰 Multi-currency Support (USD, BTC, ETH, etc.)
- 📱 Mobile Optimization
- 🎮 Auto-play & Quick Spin
- 💾 Project Save/Load
- 📦 Complete Export Package

Integration

- ⚡ Stake API SDK Integration
- 🎲 Provably Fair Engine
- 🏠 Wallet Integration
- 🛡️ Security & Rate Limiting
- 📊 Performance Monitoring
- 🐞 Error Tracking

📱 Mobile Development

iOS Build (Requires Mac)

1. Setup

```
bash
npm install -g react-native-cli
npx react-native init StakeEngineBuilder
cd StakeEngineBuilder
```

2. Install Dependencies

```
bash
cd ios
pod install
cd ..
```

3. Run on Simulator

```
bash
npx react-native run-ios
```

4. Build for Device

- Open `ios/StakeEngineBuilder.xcworkspace` in Xcode
- Select your team in Signing & Capabilities
- Product → Archive
- Distribute App

Requirements:

- Mac with macOS 12+
- Xcode 14+
- Apple Developer Account (\$99/year)

Android Build

1. Setup

```
bash
# Install Android Studio
# Set ANDROID_HOME environment variable
```

2. Build APK

```
bash
cd android
./gradlew assembleRelease
```

Output: `android/app/build/outputs/apk/release/app-release.apk`

3. Build for Google Play

```
bash

./gradlew bundleRelease
```

Output: `android/app/build/outputs/bundle/release/app-release.aab`

Code Signing

Windows

Use SignTool with a code signing certificate:

```
bash

signtool sign /f cert.pfx /p password /t http://timestamp.server.com app.exe
```

macOS

Sign with Apple Developer certificate:

```
bash

codesign --deep --force --verify --sign "Developer ID" app.app
```

Linux

Generally doesn't require signing for distribution.

Distribution

Desktop

- **Windows:** Microsoft Store or direct download
- **macOS:** Mac App Store or notarized DMG
- **Linux:** Snap Store, Flathub, or direct download

Mobile

- **iOS:** Apple App Store
 - **Android:** Google Play Store
-

Configuration

App Icon

Replace `public/icon.png` with your 512x512 icon.

Generate platform icons:

```
bash

npm install -g app-icon
app-icon generate -i public/icon.png
```

App Name

Edit `package.json`:

```
json

{
  "name": "your-app-name",
  "productName": "Your App Display Name"
}
```

Auto-Update

Install electron-updater:

```
bash

npm install electron-updater
```

Add to `electron/main.js`:

```
javascript

const { autoUpdater } = require('electron-updater');
autoUpdater.checkForUpdatesAndNotify();
```

Troubleshooting

Build Fails

1. Ensure Node.js v18+ installed
2. Run `npm install` again
3. Delete `node_modules` and reinstall
4. Check error messages in console

Windows Build on Mac/Linux

Install Wine:

```
bash

# Mac
brew install wine

# Linux
sudo apt install wine
```

macOS Build Requires Mac

Cannot build macOS apps on Windows/Linux. Use:

- Mac computer
- macOS virtual machine
- Cloud Mac service

App Won't Open

- Check if app is signed
- Check console for errors
- Try running from terminal

Documentation

- [Quick Setup Guide](#) - 5-minute setup
- [Complete Build Guide](#) - Detailed instructions

- [Electron Docs](#)
 - [React Docs](#)
 - [Vite Docs](#)
-

Tech Stack

- **Framework:** Electron 28 + React 18
 - **Build Tool:** Vite 5
 - **Packager:** Electron Builder 24
 - **Icons:** Lucide React
 - **Styling:** Custom CSS with Stake theme
 - **Languages:** JavaScript/JSX
-

Build Checklist

Before releasing:

- ☐ Update version in `package.json`
 - ☐ Test all features in dev mode
 - ☐ Test production build
 - ☐ Add app icon (512x512)
 - ☐ Sign application (if distributing)
 - ☐ Test on clean machine
 - ☐ Create user documentation
 - ☐ Set up auto-update (optional)
 - ☐ Create LICENSE file
 - ☐ Test installer on target OS
-

Features Overview

Included in Build

- ☒ Complete game development suite
- ☒ Math engine with RTP simulator
- ☒ Symbol & payable designer
- ☒ Animation configurator

- ✓ Audio system
- ✓ Bonus game builder
- ✓ Testing & verification
- ✓ Analytics dashboard
- ✓ Compliance tools
- ✓ Export system
- ✓ Project save/load
- ✓ Multi-platform support

System Requirements

Development:

- Node.js v18+
- 16GB RAM
- 50GB disk space
- Modern processor (i5/Ryzen 5+)

Runtime:

- Windows 10+, macOS 10.15+, or modern Linux
- 4GB RAM minimum
- 500MB disk space

Support

Common Issues

"Cannot find module"

```
bash
npm install
```

"Permission denied"

```
bash
chmod +x build-all.sh
```

Build takes too long

- Normal for first build (5-15 min)
- Subsequent builds faster (2-5 min)

App crashes on startup

- Check console errors
 - Verify all dependencies installed
 - Try rebuilding: `npm run build && npm run electron:start`
-

License

MIT License - See LICENSE file for details

Quick Links


- [Node.js Download](#)
 - [Electron Documentation](#)
 - [React Documentation](#)
 - [Electron Builder](#)
 - [Stake Engine Docs](#)
-

Credits

Built with:

- Electron
 - React
 - Vite
 - Electron Builder
 - Lucide Icons
-

Ready to build? Start with the [Quick Setup Guide!](#)

 Happy Game Building! 