

Mögliche Programmierschnittstellen:

Kriterienkatalog

Muss:

- C#
- 2D-Rendering
- Map \Rightarrow Interaktive Bilder
- Keystrokes müssen erkannt werden
- Crossplatform (Windows \leftrightarrow Mac)
- Ausführen im Browser

Kann:

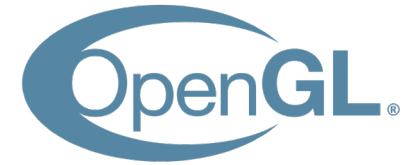
- Animationen
- Audio (Soundeffekte / Musik)

Mögliche Kandidaten:

- Blazor: Client- oder Serverbasiert
- OpenGL/WebGL
- Monogame: Normalerweise Client
- UnityWebGL



WEBASSEMBLY





Monogame



| | Main Points | Pro | Contra |
|----------------|--|--|--|
| Monogame-Lokal | <ul style="list-style-type: none">• Lokal• .NET Game Framework• Re-Implementierung von Microsoft XNA Framework | <ul style="list-style-type: none">• C#• Einfach• Cross-Platform• Aktive Community | <ul style="list-style-type: none">• Nicht Web• Mögliche Komplikation bei verschiedenen Plattformen |
| Monogame-Web | <ul style="list-style-type: none">• Monogame Code• WebGL2 + Bridge.NET (fork) | <ul style="list-style-type: none">• Bekannter Monogame Code• Web | <ul style="list-style-type: none">• Transpiled zu JS• Bridge.NET archiviert• → müssten Fork nutzen, z.B h5 |

OpenGL/WebGL

| | Main Points | Pro | Contra |
|--|---|--|---|
| <div>OpenGL</div>  | <ul style="list-style-type: none">• Client-basierende Grafikbibliothek• Wird oft für Desktopspiele verwendet• Ursprünglich 3D | <ul style="list-style-type: none">• Grosse Community => Viele Tutorials, regelmässige Wartung• Hohe Leistung | <ul style="list-style-type: none">• Evtl. Treiberinstallationen• Schwieriger zu bedienen als WebGL• Läuft standardmässig auf C++ => Open ToolKit |
| <div>WebGL</div>  | <ul style="list-style-type: none">• JavaScript-basierende API von OpenGL• Erstellt Graphiken direkt im Browser | <ul style="list-style-type: none">• Einfach zugänglich für User => Website• Plattformunabhängig• Grundkenntnisse fürs Web reichen aus | <ul style="list-style-type: none">• Leistung begrenzt durch Browser |

Blazor



| | Main Points | Pro | Contra |
|---------------|---|---|---|
| Blazor Server | <ul style="list-style-type: none">• Server-Basiert• Logik läuft auf Server• HTML wird an Client gestreamt | <ul style="list-style-type: none">• Kürzere Ladezeit• Weniger belastend für Client | <ul style="list-style-type: none">• Kann Latenz haben• Benötigt konstante Internetverbindung |
| Blazor WASM | <ul style="list-style-type: none">• Client-Basiert• Logik läuft im Browser via WebAssembly | <ul style="list-style-type: none">• Bessere Performance (je nach dem wie leistungsstark Client)• Offline Support | <ul style="list-style-type: none">• Lange Ladezeit• Belastender für Client |