# 디지털 공화국: Unity 기반 "바이마르"를 위한 아키텍처 및 UX 청사진

## 서론: 디지털 공화국

보드게임 "바이마르: 민주주의를 위한 투쟁(Weimar: The Fight for Democracy)"을 디지털로 전환하는 작업은 단순한 기술적 이식을 넘어, 복잡한 정치적 서사와 시스템을 스크린에 재현하는 예술적 도전을 수반합니다. 이 게임의 핵심은 기계적인 메커니즘이 아니라 "이벤트 카드, 테이블 토크, 변화하는 동맹, 즉 롤플레잉 측면"에 있기 때문입니다.1 따라서 디지털 버전은 이러한 상호작용과 긴장감을 저해하는 것이 아니라 오히려 촉진해야 합니다. 또한 3시간에서 6시간에 달하는 긴 플레이 타임이라는 장벽을 게임플레이의 효율화를 통해 극복해야 하는 과제를 안고 있습니다.2

이러한 과제를 해결하기 위해, 본 보고서는 다음의 세 가지 핵심 설계 원칙을 제안합니다. 이 원칙들은 프로젝트의 모든 기술적, 디자인적 결정의 기준이 될 것입니다.

* **주제적 무결성 (Thematic Integrity):** 이 디지털 적응은 추상적인 엔진이 아니라 1920년대 독일의 긴장감 넘치는 정치 투쟁처럼 느껴져야 합니다. UI 아트부터 음향 효과에 이르기까지 모든 디자인 요소는 이 주제를 강화해야 합니다.
* **전략적 깊이 (Strategic Depth):** 게임의 핵심 의사결정 구조, 특히 다용도 카드 시스템은 반드시 보존되고 강화되어야 합니다.4 디지털 인터페이스는 선택의 무게를 감소시키지 않으면서도 그 과정을 명확하게 만들어야 합니다.
* **사용자 접근성 (User Accessibility):** 게임의 복잡성은 가장 큰 진입 장벽입니다. 디지털 버전의 핵심 가치는 자동화, 뛰어난 튜토리얼, 지능적인 UI를 통해 이 깊이 있는 경험에 쉽게 접근할 수 있도록 만드는 것입니다. 이는 \*쓰루 디 에이지스(Through the Ages)\*와 같은 성공적인 디지털 보드게임 적응 사례에서 얻은 교훈입니다.6

## 1부: Unity 기반의 기초 아키텍처

이 파트에서는 게임의 방대한 복잡성을 관리하기 위해 필수적인, 깔끔하고 확장 가능하며 테스트 용이한 소프트웨어 아키텍처를 상세히 설명합니다.

### 1.1 핵심 아키텍처 패턴: 모델-뷰-프리젠터(MVP) 프레임워크

이 정도 규모의 프로젝트에서 관심사 분리(Separation of Concerns)는 타협할 수 없는 원칙입니다. 이를 위해 MVC(Model-View-Controller) 및 MVP(Model-View-Presenter)와 같은 디자인 패턴이 고려될 수 있습니다.8

"바이마르" 프로젝트에는 MVP 패턴을 강력히 권장합니다. MVP는 MVC와 유사하지만, 뷰(View)가 유니티 엔진의 생명주기에 강하게 종속된 MonoBehaviour로 구성되는 유니티의 컴포넌트 기반 특성에 더 적합합니다. MVP는 더 깔끔한 분리를 가능하게 하여, 게임의 복잡한 로직(프리젠터)을 유니티 엔진 런타임과 독립적으로 순수 C# 코드로 개발하고 단위 테스트할 수 있게 해줍니다.8

* **모델 (Model):** GameState, PlayerState, BoardState(도시 지배, 유닛 위치), ParliamentState, IssueState 등 전체 게임 상태를 나타내는 순수 C# 클래스입니다. 데이터와 상태 변경 알림 이벤트(예: OnUnitPlaced, OnSeatGained)만을 포함하며, 유니티에 대해 전혀 알지 못합니다.
* **뷰 (View):** CardView, MapView, UIPanelView 등 모든 유니티 MonoBehaviour 스크립트와 UI 요소를 포함합니다. 뷰는 모델의 이벤트를 구독하여 시각적 요소를 업데이트하고, 클릭이나 드래그 같은 원시 사용자 입력을 받아 프리젠터에 전달하지만, 자체적으로는 게임 로직을 포함하지 않습니다.11
* **프리젠터 (Presenter):** GamePresenter, CardPlayPresenter, PoliticsPhasePresenter 등 "두뇌" 역할을 하는 순수 C# 클래스입니다. 뷰로부터 입력을 받아 규칙에 따라 유효성을 검사하고, 모델을 조작하며, 게임의 흐름을 조율합니다.

이러한 아키텍처 선택은 단순히 코드 구성을 넘어 프로젝트의 위험을 줄이는 전략적 결정입니다. 비대칭적 규칙과 수백 개의 이벤트 카드를 가진 게임은 버그가 발생하기 쉽습니다.1 전통적인 유니티 아키텍처는 테스트를 어렵게 만들지만, MVP를 채택함으로써 프리젠터와 모델에 있는 전체 규칙과 게임 로직을 자동화된 단위 테스트에 적용할 수 있습니다. 이는 개발 후반부에 버그 수정 시간을 극적으로 줄여주는 전문적인 접근 방식입니다.

**표 1: 아키텍처 패턴 비교 (Unity 환경에서의 MVP vs. MVC)**

| 기능 | Unity에서의 MVC | Unity에서의 MVP | "바이마르"를 위한 권장 사항 |
| --- | --- | --- | --- |
| **코드 결합도** | 뷰와 컨트롤러가 강하게 결합될 수 있음 | 뷰와 프리젠터가 인터페이스를 통해 약하게 결합됨 | **MVP**: 낮은 결합도는 유지보수성과 확장성을 향상시킴 |
| **단위 테스트 용이성** | 컨트롤러가 MonoBehaviour에 종속되어 테스트가 어려움 | 프리젠터가 순수 C# 클래스이므로 로직 테스트가 매우 용이함 | **MVP**: 규칙이 복잡한 "바이마르"에서 테스트 용이성은 필수적임 |
| **확장성** | 로직과 뷰가 섞여 새로운 기능 추가 시 복잡성이 증가함 | 각 부분이 명확히 분리되어 기능 추가 및 수정이 용이함 | **MVP**: 향후 확장팩이나 수정 가능성을 열어둠 |
| **Unity 통합 마찰** | 전통적인 패턴으로, Unity의 컴포넌트 시스템과 충돌할 수 있음 | Unity의 이벤트 기반 및 컴포넌트 시스템과 자연스럽게 통합됨 | **MVP**: Unity 개발 환경에 더 최적화된 접근 방식임 |

### 1.2 유한 상태 머신(FSM)을 통한 게임 상태 관리

게임은 공화국, 의제, 실행, 정치 단계 등 엄격한 순서의 라운드와 페이즈를 통해 진행됩니다.13 이러한 구조는 게임 흐름을 관리하기 위한 표준 패턴인 유한 상태 머신(FSM)을 적용하기에 완벽한 대상입니다.15

중앙 GameManager 싱글톤이 관리하는 계층적 FSM을 구현할 것을 제안합니다.

* **최상위 상태:** GameSetupState, GamePlayState, GameEndState
* **GamePlayState 하위 상태:** RoundState가 존재하며, 이 상태는 다시 RepublicPhaseState, AgendaPhaseState, ImpulsePhaseState, PoliticsPhaseState의 순차적인 하위 상태를 포함합니다.

각 상태 클래스는 해당 페이즈에 특화된 로직을 포함합니다(예: AgendaPhaseState는 플레이어의 의제 카드 선택을 처리). 또한, 상태 전환 조건을 검증하고 다음 상태로 전환하는 책임을 집니다. FSM은 단순히 흐름을 제어하는 것을 넘어, 플레이어 상호작용의 규칙을 강제하는 역할을 합니다. 예를 들어, ImpulsePhaseState에서는 카드 플레이 액션이 허용되지만, PoliticsPhaseState로 전환되면 카드 플레이는 비활성화되고 해당 페이즈의 순차적인 액션(이슈 평가, 점수 계산, 정부 구성 등)만 활성화됩니다.14 이는 FSM이 불법적인 행동을 방지하고 플레이어가 게임의 복잡한 턴 구조를 따라가도록 안내하는 핵심 메커니즘임을 의미하며, 이는 룰북을 상호작용 코드로 직접 변환하는 것과 같습니다.13

### 1.3 카드 및 이벤트 시스템

모든 카드 데이터(제목, 진영, 이미지, 이벤트 텍스트, 행동력 값)는 유니티의 ScriptableObject 에셋에 저장해야 합니다.17 이를 통해 디자이너는 코드 변경 없이 유니티 에디터에서 직접 수백 장의 카드를 생성, 수정, 밸런싱할 수 있어 반복 작업 속도를 극적으로 높일 수 있습니다. 이 시스템은 기본 덱, 진영별 카드, 그리고 의제 카드를 통해 추가되는 카드 세트를 모두 처리해야 합니다.12

플레이어가 카드의 다용도 옵션(이벤트, 토론, 거리 행동) 중 하나를 선택하면, 프리젠터는 해당 **커맨드(Command)** 객체(예: PlayEventCommand, DebateIssueCommand)를 인스턴스화해야 합니다. 이 패턴은 요청을 객체로 캡슐화합니다. 예를 들어, "바이에른 질서의 보루(Cell of Order Bavaria)" 카드의 PlayEventCommand는 중앙당(Zentrum) 플레이어에게 어려운 선택지를 제시하는 모든 로직을 포함하게 됩니다.18 이 방식은 복잡한 다단계 액션을 관리하고 테스트하기 쉽게 만들며, 각 커맨드에

Undo() 메서드를 구현함으로써 디지털 보드게임에서 매우 중요한 "실행 취소" 기능을 지원할 수 있습니다.7

이 아키텍처의 핵심은 카드 데이터와 카드 로직을 분리하는 것입니다. ScriptableObject는 카드가 *무엇*인지만 알고, 커맨드 객체는 그 효과를 *어떻게* 실행할지 압니다. 이러한 분리는 핵심 게임 엔진이 특정 카드에 대해 알 필요 없이 일반적인 ICommand 인터페이스만 실행하면 되도록 만들어, 향후 확장 및 모딩 가능성을 위한 필수적인 기반을 마련합니다.

### 1.4 비대칭성 설계

네 개의 진영(사회민주당, 중앙당, 공산당, 국가인민당)은 서로 다르거나 때로는 상반되는 목표를 가집니다.4 민주 진영은 안정화를 통해 점수를 얻는 반면, 극단주의 진영은 불안과 쿠데타로 점수를 얻습니다.18

이러한 비대칭성은 인터페이스와 스트래티지 패턴(Strategy Pattern)을 사용하여 해결할 수 있습니다. 모든 진영별 로직 클래스가 구현할 공통 IFaction 인터페이스를 정의합니다. 이 인터페이스에는 CalculateVictoryPoints(), CheckInstantWinCondition(), GetAgendaCards()와 같은 메서드가 포함됩니다.

각 진영(예: KPDFaction, SPDFaction)은 이 인터페이스를 구현하는 자체 클래스를 가지며, 고유한 점수 계산 규칙과 승리 조건을 포함합니다. 중앙 GamePresenter는 IFaction 객체 목록을 가지고 정치 페이즈 동안 각 진영의 구체적인 규칙을 알 필요 없이 단순히 faction.CalculateVictoryPoints()를 호출합니다.

비대칭성은 단순히 점수 계산에만 국한되지 않습니다. "중앙당은 정부에 참여하지 않는 한 경찰 유닛만 통제할 수 있다"는 규칙처럼, 진영들은 게임 메커니즘과 고유한 관계를 맺습니다.19 이는 프리젠터 내의 핵심 규칙 검증 로직이 특별한 권한이나 제약을 확인하기 위해 현재 플레이어의

IFaction 객체를 자주 조회해야 함을 의미합니다. 따라서 IFaction 인터페이스는 단순한 점수 계산기를 넘어, 진영의 고유한 규칙 예외와 능력을 포괄하는 저장소 역할을 해야 합니다.

## 2부: 사용자 인터페이스(UI) 및 사용자 경험(UX) 디자인

이 파트에서는 게임의 물리적 존재감을 직관적이고 정보가 풍부하며 주제적으로 공감대를 형성하는 디지털 경험으로 전환하는 데 중점을 둡니다.

### 2.1 디지털 보드: 물리적 공간의 변환

모든 것을 한 화면에 억지로 담으려는 시도는, 특히 작은 기기에서 실패하기 쉽습니다. 이는 \*미궁(Labyrinth)\*의 아이폰 UI 사례에서 얻은 교훈입니다.20 목표는 가장 자주 필요한 정보를 우선적으로 보여주는 깔끔하고 가독성 있는 메인 뷰를 만드는 것입니다.

* **"조종석(Cockpit)" 뷰 레이아웃 제안:**
  + **중앙:** 크고 상호작용 가능한 독일 지도. 도시는 명확하고 클릭 가능한 대상으로 표시됩니다. 경제, 위협과 같은 주요 트랙은 지도 위에 겹쳐지거나 근처에 눈에 띄게 배치됩니다.
  + **상단:** 현재 토론 중인 이슈를 보여주는 여론 트랙. 이는 핵심적인 "정치적 전장"이므로 매우 잘 보여야 합니다.
  + **하단:** 플레이어의 손에 있는 카드를 명확하게 표시합니다.
  + **좌/우:** 플레이어별 정보(플레이어 매트 데이터)와 상대방 요약을 위한 접이식 패널을 배치합니다.

정적인 화면 대신, UI는 플레이어의 현재 행동에 맞춰 동적으로 적응해야 합니다. 이는 최고 수준의 디지털 적응 사례들에서 배운 핵심 교훈입니다.6 플레이어가 카드를 여론 트랙으로 드래그하면 지도는 약간 축소되거나 흐려지고, 이슈 트랙과 관련 카드 세부 정보가 중심으로 부각됩니다. 유닛을 이동할 때는 지도가 주된 초점이 됩니다. 이처럼 UI는 단순한 디스플레이가 아니라, 플레이어의 집중력을 적시에 적절한 "전장"(정치적 또는 거리)으로 유도하는 동적인 렌즈 역할을 함으로써 인지 부하를 줄이고 복잡한 게임 상태를 관리 가능하게 만듭니다.

### 2.2 정보 계층 구조와 접근성

플레이어에게 과도한 정보를 한 번에 제공하는 것을 피하기 위해, 정보는 계층적으로 접근 가능해야 합니다.

* **1단계 (한눈에 보기):** 메인 화면은 아이콘과 명확한 시각적 요소(예: 지도 위의 유닛 토큰, 트랙 위의 정당 마커)를 통해 가장 중요한 상태 정보를 보여줍니다.
* **2단계 (마우스 오버):** 툴팁을 광범위하게 사용합니다. 손에 있는 카드 위에 마우스를 올리면 전체 이벤트 텍스트가 표시됩니다. 도시 위에 올리면 이름, 안정도, 주둔 병력 요약이 나타납니다.
* **3단계 (클릭):** 요소를 클릭하면 상세 모달이나 패널이 나타납니다. 상대방의 초상화를 클릭하면 공개된 게임 상태 요약이 열립니다.

길고 복잡한 게임을 위해서는 게임 로그가 필수적입니다. 단순한 텍스트 피드를 넘어, 필터링 기능(예: "공산당 행동만 보기")과 상호작용 기능(로그 항목 클릭 시 보드의 관련 부분 강조)을 제공해야 합니다.

**표 2: UI 정보 표시 전략**

| 게임 요소 | 1단계 (한눈에 보기) | 2단계 (마우스 오버/툴팁) | 3단계 (클릭/확장 패널) |
| --- | --- | --- | --- |
| **도시 공간** | 유닛 토큰, 지배 마커 | 도시 이름, 안정도, 유닛 상세 | 도시 내 모든 유닛과 효과의 전체 목록 |
| **플레이어 핸드 카드** | 카드 아트, 행동력 값 | 전체 이벤트 텍스트, 플레이 조건 | 카드 확대 보기, 관련 규칙 참조 |
| **이슈 토큰** | 이슈 아이콘, 위치 | 이슈 이름, 승리 시 보상 | 이슈의 전체 효과 및 역사적 배경 설명 |
| **의회 의석** | 정당별 의석 수 | 정확한 의석 수, 총 의석 대비 비율 | 의석 변동 내역, 연정 가능성 분석 |
| **플레이어 매트** | 핵심 자원 수치 아이콘 | 자원 이름 및 정확한 수치 | 자원 획득/소비 로그, 관련 능력 설명 |
| **상대방 요약** | 정당 색상, VP | 공개된 자원, 의석 수, 지배 도시 수 | 상대방의 플레이어 매트 전체 공개 정보 |
| **위협 마커** | 마커의 존재 및 위치 | 위협의 종류 및 현재 효과 | 위협이 발생한 원인 및 제거 조건 |

### 2.3 플레이어 상호작용 및 피드백

카드 플레이에는 드래그 앤 드롭 방식을 우선적으로 사용해야 합니다. 플레이어가 카드를 드래그하기 시작하면, 유효한 드롭 대상(지도, 이슈 트랙, 이벤트 플레이 영역)이 명확하게 강조 표시되어야 합니다. 이는 플레이어의 실수를 방지하고 좌절감을 줄여줍니다.

모든 플레이어 행동은 만족스러운 피드백으로 확인되어야 합니다.

* **애니메이션:** 유닛은 부드럽게 위치로 이동해야 합니다. 의회 의석을 얻으면 말이 트랙 위로 움직이는 애니메이션이 재생됩니다.
* **음향 디자인 (SFX):** 카드 플레이, 유닛 배치, 토론 승리, 자원 획득에 대해 각각 다른 소리를 사용합니다. 소리는 주제에 맞아야 합니다(예: 정치적 영향력에는 군중의 웅성거림, 준군사 행동에는 군화 소리).
* **시각 효과 (VFX):** 강조 표시된 대상에 미묘한 빛 효과, 점수 획득 시 짧은 섬광, 주요 이벤트 발생 시 지도에 파문 효과 등을 추가합니다.

이러한 상호작용 피드백은 단순히 미적인 요소를 넘어, 중요한 학습 도구 역할을 합니다. 카드를 선택했을 때 유효한 행동을 강조 표시함으로써, 시스템은 플레이어에게 해당 카드와 관련된 규칙과 가능성을 암묵적으로 가르쳐 줍니다. 이는 새로운 플레이어에게 게임을 더 쉽게 발견하고 덜 위협적으로 느끼게 만들어, 접근성이라는 핵심 원칙을 직접적으로 해결합니다.

### 2.4 예술적 방향과 주제적 무결성

UI는 바이마르 공화국 시대의 미학에 깊이 뿌리내려야 합니다.

* **핵심 스타일 - 신즉물주의(Neue Sachlichkeit):** 패널, 버튼, 창과 같은 주요 UI 요소는 신즉물주의의 냉철하고 사실적인 스타일을 채택해야 합니다.22 이는 깔끔한 선, 차분한 색상 팔레트(회색, 갈색, 진한 빨간색), 장식보다는 기능성에 중점을 두는 것을 의미하며, 게임의 정치적이고 관료적인 성격을 반영합니다.
* **강조 스타일 - 표현주의(Expressionism):** 거리 투쟁, 쿠데타, 불안 고조와 같은 혼돈의 순간에는 표현주의적 요소를 잠시 도입할 수 있습니다.24 이는 분쟁 중인 도시 주변에 왜곡되고 들쭉날쭉한 테두리를 추가하거나, 지도에 고대비의 극적인 조명 효과를 주는 방식으로 나타날 수 있습니다.
* **타이포그래피 - 포스터 양식(Plakatstil) & 신타이포그래피(New Typography):** 모든 텍스트 요소는 1920년대 독일 포스터 디자인에서 영감을 받은 굵고 기하학적인 산세리프 글꼴을 사용해야 합니다.26 텍스트 블록과 제목에 비대칭 레이아웃을 적용하여 시대에 맞는 강력한 시각적 정체성을 만듭니다.

실물 게임은 이중 레이어 보드와 목재 토큰 등 고품질 구성 요소로 호평을 받았습니다.28 디지털 버전은 이를 존중해야 합니다. 텍스처는 고해상도여야 하고, 디지털 토큰은 입체감을 가져야 합니다. 역사적 사진을 사용하는 카드 아트는 명확하고 정중하게 제시되어야 합니다.

이 두 가지 다른 역사적 예술 스타일을 의식적으로 사용하여 두 개의 다른 "전장"을 표현함으로써, UI는 게임의 핵심 주제를 강력하게 강화할 수 있습니다. 정치적 영역의 차분하고 질서 있는 "신즉물주의" 스타일은 거리 투쟁 영역의 혼란스럽고 감정적인 "표현주의" 스타일과 날카롭게 대조됩니다. 이러한 시각적 부조화는 바이마르 공화국 자체의 사회적 긴장, 즉 거리에서 끓어오르는 폭력적인 혼돈에 끊임없이 위협받는 연약한 민주적 질서를 반영하며, UI를 단순한 장식에서 게임 서사의 핵심 엔진으로 격상시킵니다.

## 3부: 전략적 구현 로드맵

이 마지막 파트에서는 개발 과정 자체에 대한 실행 가능한 조언을 제공합니다.

### 3.1 개발 우선순위와 수직 분할(Vertical Slice)

전체 게임을 한 번에 구축하는 대신, 첫 번째 주요 마일스톤은 "수직 분할"이어야 합니다. 이는 민주 진영(사회민주당)과 극단주의 진영(공산당) 단 두 개의 진영만으로 구성된, 한 라운드를 완전히 플레이할 수 있는 버전을 의미합니다.

이 수직 분할의 목표는 FSM, MVP 루프, 카드 시스템, 그리고 정치 및 거리 전장을 위한 기본 UI 등 전체 핵심 아셔를 구축하고 검증하는 것입니다. 이 핵심 루프가 안정적으로 작동함을 증명하는 것이 최우선 과제입니다. 일단 이 분할이 안정되면, 나머지 두 진영을 추가하는 것은 기반을 재설계하는 것이 아니라 콘텐츠와 특정 규칙을 추가하는 문제가 됩니다.

### 3.2 멀티플레이어 및 AI 고려사항

게임의 긴 플레이 타임은 비동기 멀티플레이어(플레이-바이-이메일과 유사)를 필수적으로 만듭니다. 아키텍처는 처음부터 서버 권위적 모델을 기반으로 구축되어야 합니다.29 클라이언트는 플레이어의 의도(예: "플레이어 1이 X 카드를 이벤트로 사용하기를 원함")만을 서버에 보내고, 서버는 이동을 검증하고 마스터 게임 상태를 업데이트한 후 모든 플레이어에게 변경 사항을 알립니다.

이렇게 복잡하고 비대칭적인 게임을 위한 AI 설계는 상당한 도전입니다. 간단한 "스크립트" AI로 시작할 것을 권장합니다. 더 발전된 AI는 플레이어 유형을 모델링해야 합니다. 예를 들어, "신중한 사회민주당" AI는 안정성을 우선시하고, "공격적인 공산당" AI는 높은 위험을 감수하고서라도 쿠데타를 끊임없이 시도할 것입니다. 이는 솔로 플레이어에게 다양한 도전을 제공합니다. 게임에는 이미 학습을 위한 기본적인 솔로 시나리오가 포함되어 있어, 이를 출발점으로 삼을 수 있습니다.2

### 3.3 최종 권장 사항

* **핵심 결정 요약:** MVP 채택, 계층적 FSM 사용, 커맨드 패턴을 사용한 데이터 기반 카드 시스템 구축, 그리고 바이마르 시대의 이중 미학에 기반한 UI 디자인이라는 가장 중요한 사항들을 다시 한번 강조합니다.
* **최종 체크리스트:**
  + UI가 인지 부하를 성공적으로 줄이는가?
  + 카드의 다용도성이 명확하게 제시되는가?
  + 신규 플레이어가 튜토리얼과 UI 피드백을 통해 게임의 흐름을 이해할 수 있는가?
  + 게임이 필사적인 정치 투쟁처럼 느껴지는가?
* **결론:** "바이마르"의 성공적인 디지털 적응은 디지털 테이블탑 분야에서 기념비적인 성과가 될 것입니다. 이를 위해서는 기술적 우수성뿐만 아니라, 게임의 역사적 주제와 복잡하고 서사 중심적인 디자인에 대한 깊은 존중이 필요합니다. 이 청사진은 그 성공을 달성하기 위한 기반을 제공합니다.

#### 참고 자료

1. Weimar - The Fight for Democracy : r/boardgames - Reddit, 9월 15, 2025에 액세스, <https://www.reddit.com/r/boardgames/comments/1d1cdlz/weimar_the_fight_for_democracy/>
2. Weimar - The Fight for Democracy by Skellig Games GmbH » FAQ - Kickstarter, 9월 15, 2025에 액세스, <https://www.kickstarter.com/projects/skellig-games/weimar-the-fight-for-democracy/faqs>
3. Weimar: The Fight for Democracy - Board Game Bliss, 9월 15, 2025에 액세스, <https://www.boardgamebliss.com/products/weimar-the-fight-for-democracy>
4. Weimar - The Fight for Democracy (ENGLISH) - WarGen Wargames, 9월 15, 2025에 액세스, <https://wargenwargames.com/en/resource-management/6258-weimar-the-fight-for-democracy-english.html>
5. Weimar: The Fight for Democracy - YouTube, 9월 15, 2025에 액세스, <https://www.youtube.com/watch?v=NJxDeItFHF8>
6. Through The Ages - Digital - The Cardboard Republic, 9월 15, 2025에 액세스, <https://www.cardboardrepublic.com/cr_reviews/pixel-provinces/through-the-ages-digital>
7. Through the Ages on the App Store, 9월 15, 2025에 액세스, <https://apps.apple.com/us/app/through-the-ages/id966245474>
8. Build a modular codebase with MVC and MVP programming patterns - Unity Learn, 9월 15, 2025에 액세스, <https://learn.unity.com/tutorial/build-a-modular-codebase-with-mvc-and-mvp-programming-patterns-1?uv=6&projectId=67bc8deaedbc2a23a7389cab>
9. Level up your code with game programming patterns - Unity, 9월 15, 2025에 액세스, <https://unity.com/resources/level-up-your-code-with-game-programming-patterns>
10. Design Patterns in Unity With Among Us — Fix Wiring Task | by Omid Reza Izadi - Medium, 9월 15, 2025에 액세스, <https://medium.com/swlh/design-patterns-in-unity-with-among-us-fixing-wires-task-dab8c6e5e691>
11. A Model-View-Controller (MVC) Pattern for Unity - JacksonDunstan.com, 9월 15, 2025에 액세스, <https://www.jacksondunstan.com/articles/3092>
12. Weimar: SPD Agenda Cards Ranked for Round 1 | Clio's Board Games, 9월 15, 2025에 액세스, <https://cliosboardgames.wordpress.com/2024/09/01/weimar-spd-agenda-cards-ranked-for-round-1/>
13. Rulebook | Capstone Games, 9월 15, 2025에 액세스, <https://capstone-games.com/wp-content/uploads/2023/10/Weimar_Rulebook_EN_web.pdf>
14. Playbook | Capstone Games, 9월 15, 2025에 액세스, <https://capstone-games.com/wp-content/uploads/2023/10/Weimar_Playbook_EN_web.pdf>
15. How are you all managing game flow and state? : r/Unity3D - Reddit, 9월 15, 2025에 액세스, <https://www.reddit.com/r/Unity3D/comments/15xqd5i/how_are_you_all_managing_game_flow_and_state/>
16. State Machines in Unity (how and when to use them) - Game Dev Beginner, 9월 15, 2025에 액세스, <https://gamedevbeginner.com/state-machines-in-unity-how-and-when-to-use-them/>
17. Card game code architecture : r/gamedev - Reddit, 9월 15, 2025에 액세스, <https://www.reddit.com/r/gamedev/comments/n1q0zb/card_game_code_architecture/>
18. Pandemonium (Weimar 1923, #3) - Clio's Board Games, 9월 15, 2025에 액세스, <https://cliosboardgames.wordpress.com/2023/09/03/pandemonium-weimar-1923-3/>
19. Weimar: The Fight for Democracy [Tabletopia] - Steam Community, 9월 15, 2025에 액세스, <https://steamcommunity.com/sharedfiles/filedetails/?l=swedish&id=3122530488>
20. Labyrinth: The War on Terror - App Store, 9월 15, 2025에 액세스, <https://apps.apple.com/se/app/labyrinth-the-war-on-terror/id1438485859?l=en-GB>
21. Blasphemy: Board Games That Play Better Online Than In Real Life - TheGamer, 9월 15, 2025에 액세스, <https://www.thegamer.com/board-games-play-better-online-real-life/>
22. New Objectivity: Modern German Art in the Weimar Republic, 1919–1933 - LACMA, 9월 15, 2025에 액세스, <https://www.lacma.org/sites/default/files/New%20Objectivity%20-%20Didactics_0.pdf>
23. New Objectivity: Modern German Art in the Weimar Republic, 1919–1933 | LACMA, 9월 15, 2025에 액세스, <https://www.lacma.org/art/exhibition/new-objectivity-modern-german-art-weimar-republic-1919-1933>
24. Expressionist Art and Drama Before, During, and After the Weimar Republic - PDXScholar, 9월 15, 2025에 액세스, <https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=3513&context=open_access_etds>
25. Weimar culture - Wikipedia, 9월 15, 2025에 액세스, <https://en.wikipedia.org/wiki/Weimar_culture>
26. The New Typography - MoMA, 9월 15, 2025에 액세스, <https://www.moma.org/calendar/exhibitions/1013>
27. Plakatstil - Wikipedia, 9월 15, 2025에 액세스, <https://en.wikipedia.org/wiki/Plakatstil>
28. Which games in your collection have the best/worst production value : r/boardgames - Reddit, 9월 15, 2025에 액세스, <https://www.reddit.com/r/boardgames/comments/1ibf55g/which_games_in_your_collection_have_the_bestworst/>
29. Handling Turn States in Turn-Based Multiplayer Games : r/gamedev - Reddit, 9월 15, 2025에 액세스, <https://www.reddit.com/r/gamedev/comments/16q2d53/handling_turn_states_in_turnbased_multiplayer/>
30. Game state management - Unity documentation, 9월 15, 2025에 액세스, <https://docs.unity.com/ugs/en-us/manual/cloud-code/manual/game-state-management>