

Declaration System Implementation: Constrained Player Agency

Overview

This document provides complete implementation specifications for the **Hybrid Declaration System** - a game mechanic where historical milestones (HR H-B creation, RS declaration, RBiH independence, war outbreak) occur through **constrained player agency** rather than automatic scripting or free choice.

Core Philosophy

“Players are harried institutional leaders reacting to structural forces beyond their control.”

Key Principles:

- Events are **structurally inevitable** (cannot be prevented)
- Players control **timing and manner** (strategic choices)
- **Cascading triggers** create realistic pressure
- **Consequences for delay** create meaningful decisions
- **War is unavoidable** once dominoes fall

Part 1: System Architecture

1.1 Declaration State Data Structure

```
javascript
```

```
gameState.declarations = {  
  hrhb: {  
    available: false,      // Can player declare now?  
    declared: false,      // Has it been declared?  
    declarationDate: null, // When was it declared?  
    declarationTurn: null, // Which turn?  
  
    window: {  
      start: "1991-09-01", // Earliest declaration date  
      end: "1991-12-31",   // Latest declaration date  
      optimal: "1991-11-18" // Historical date  
    },  
  
    pressure: {  
      base: 0,              // Base pressure to declare  
      turnsSinceAvailable: 0, // Turns player has waited  
      externalEvents: 0     // Pressure from other events  
    },  
  
    effects: {  
      // What happens when declared  
      ivpIncrease: 1,  
      organizationBonus: 10,  
      triggersRSPressure: true  
    }  
  },  
  
  rs: {  
    available: false,  
    declared: false,  
    declarationDate: null,  
    declarationTurn: null,  
  
    window: {  
      start: "1991-11-01", // After HR H-B declared  
      end: "1992-02-29",   // Before BiH independence  
      optimal: "1992-01-09"  
    },  
  
    pressure: {  
      base: 0,  
      turnsSinceAvailable: 0,  
      hrhbBonus: 0,        // Extra pressure if HR H-B declared  
    }  
  }  
}
```

```
    externalEvents: 0
  },

  effects: {
    ivpIncrease: 2,
    enableCoercion: true,
    triggersRBiHReferendum: true
  }
},

rbih: {
  available: false,
  referendumScheduled: false,
  referendumHeld: false,
  declared: false,
  declarationDate: null,
  declarationTurn: null,

  window: {
    start: "1992-01-01", // After RS declared
    end: "1992-03-31", // Deadline
    optimal: "1992-03-01"
  },

  referendum: {
    scheduled: false,
    scheduledTurn: null,
    turnout: null, // Calculated when held
    yesVote: null,
    boycottedBySerbs: false
  },

  pressure: {
    base: 0,
    turnsSinceAvailable: 0,
    rsBonus: 0,
    internationalPressure: 0
  },

  effects: {
    ivpIncrease: 3,
    internationalRecognition: true,
    triggersWarCountdown: true
  }
}
```

```

    }
};

gameState.warStatus = {
    warInevitable: false,    // Set to true after RBiH independence
    countdownStarted: false,
    turnsUntilWar: null,    // 2-4 turns countdown
    warStartDate: null,
    phase: null             // "escalation" → "incidents" → "war"
};

```

1.2 Turn Processing Flow

```

javascript

function processTurnDeclarations() {
    // 1. Check if declaration windows should open
    checkDeclarationWindows();

    // 2. Increase pressure for delayed declarations
    increasePressureForDelays();

    // 3. Check for automatic triggers (missed deadlines)
    checkMissedDeadlines();

    // 4. Process war countdown if active
    processWarCountdown();

    // 5. Update UI with available decisions
    updateDeclarationUI();
}

```

Part 2: HR H-B Declaration (November 1991)

2.1 Activation Conditions

```

javascript

```

```
function checkHRHBDeclarationAvailability() {  
  const currentDate = gameState.currentDate;  
  const windowStart = new Date("1991-09-01");  
  const windowEnd = new Date("1991-12-31");  
  
  // Open window in September 1991  
  if (currentDate >= windowStart &&  
    currentDate <= windowEnd &&  
    !gameState.declarations.hrhb.declared &&  
    gameState.currentEntity === 'hrhb') {  
  
    gameState.declarations.hrhb.available = true;  
  
    // First time available - show major decision prompt  
    if (gameState.declarations.hrhb.pressure.turnsSinceAvailable === 0) {  
      promptHRHBDeclaration();  
    }  
  }  
}
```

2.2 Decision Modal

javascript

```

function promptHRHBDeclaration() {
  const turnsWaited = gameState.declarations.hrhb.pressure.turnsSinceAvailable;
  const currentDate = gameState.currentDate;
  const optimalDate = new Date("1991-11-18");

  // Calculate timing modifier
  let timingAnalysis = "";
  if (currentDate < optimalDate) {
    timingAnalysis = "EARLY: Increased international pressure but organizational advantage.";
  } else if (currentDate.getTime() === optimalDate.getTime()) {
    timingAnalysis = "OPTIMAL: Historical timing balances risks and benefits.";
  } else {
    timingAnalysis = "DELAYED: Risk losing control of mixed municipalities.";
  }

  showDecisionModal({
    id: "hrhb_declaration",
    title: "DECLARE HRVATSKA REPUBLIKA HERCEG-BOSNA?",
    urgency: turnsWaited > 5 ? "HIGH" : "MEDIUM",

    context: `
      HDZ has organized Croatian-majority municipalities in Herzegovina and
      central Bosnia. SDS is consolidating Serb areas. The Yugoslav state
      is collapsing.

      Croatian leaders propose declaring a separate Croatian entity within
      Bosnia-Herzegovina to protect Croatian interests.

      Timing Analysis: ${timingAnalysis}
      Turns since decision available: ${turnsWaited}
    `,

    options: [
      {
        label: "DECLARE HR H-B NOW",
        hotkey: "D",
        effects: {
          immediate: [
            "Create Hrvatska Republika Herceg-Bosna",
            "Establish parallel Croatian institutions",
            "Enable HVO (Croatian Defense Council) formation",
            `International Visibility +${1 + calculateEarlyPenalty(currentDate)}`,
            "Trigger: SDS accelerates RS preparations"
          ]
        }
      }
    ]
  });
}

```

```

    ],
    delayed: [
        "SDS will likely declare RS within 2 months",
        "Consolidate control in Croatian-majority areas",
        "May create tension with SDA in mixed municipalities"
    ],
    },
    consequences: calculateHRHBDeclarationEffects(currentDate),
    style: "primary"
},
{
    label: "WAIT (Review Next Turn)",
    hotkey: "W",
    effects: {
        immediate: [
            "Delay costs organizational momentum",
            `Risk of losing ${calculateMunicipalityRisk(turnsWaited)} municipalities`
        ],
        warning: turnsWaited > 8 ?
            "⚠ CRITICAL: Window closes December 1991. SDS may act first." :
            "Pressure increases each turn."
    },
    consequences: {
        pressureIncrease: 10,
        organizationDecay: -2,
        municipalityRisk: calculateMunicipalityRisk(turnsWaited + 1)
    },
    style: "secondary"
}
],

```

historicalNote: `

Historically, HR H-B was declared on November 18, 1991, establishing Croatian political structures separate from Sarajevo. This was a response to the collapse of Yugoslavia and preceded the declaration of Republika Srpska by two months.

`

});

}

```

function calculateHRHBDeclarationEffects(currentDate) {
    const optimal = new Date("1991-11-18");
    const monthsEarly = (optimal - currentDate) / (1000 * 60 * 60 * 24 * 30);
    const monthsLate = (currentDate - optimal) / (1000 * 60 * 60 * 24 * 30);

```

```
const effects = {
  ivp: 1,
  organizationBonus: 10,
  municipalitiesSecured: []
};

// Early declaration penalty
if (monthsEarly > 0) {
  effects.ivp += Math.floor(monthsEarly); // +1 IVP per month early
  effects.organizationBonus += 5; // Better preparation
}

// Late declaration penalty
if (monthsLate > 0) {
  effects.organizationBonus -= Math.floor(monthsLate * 3); // Lose momentum
  effects.municipalityRisk = Math.floor(monthsLate * 2); // Lose contested areas
}

return effects;
}
```

2.3 Declaration Execution

javascript


```
function executeHRHBDeclaration() {
  const decl = gameState.declarations.hrhb;

  // Mark as declared
  decl.declared = true;
  decl.declarationDate = gameState.currentDate;
  decl.declarationTurn = gameState.currentTurn;

  // Apply effects
  const effects = calculateHRHBDeclarationEffects(gameState.currentDate);

  // Increase IVP
  gameState.constraints.ivp += effects.ivp;

  // Secure Croatian-majority municipalities
  secureCroatianMunicipalities();

  // Enable HVO formation
  gameState.enabledActions.push('form_hvo', 'croatian_parallel_institutions');

  // Trigger RS pressure
  gameState.declarations.rs.pressure.hrhbBonus = 20;
  gameState.declarations.rs.available = true; // Open RS window early

  // Create event log
  gameState.events.push({
    turn: gameState.currentTurn,
    date: gameState.currentDate,
    type: "DECLARATION",
    faction: "hrhb",
    title: "Hrvatska Republika Herceg-Bosna Declared",
    description: "Croatian Democratic Union (HDZ) declares separate Croatian entity.",
    effects: effects
  });

  // Show result modal
  showResultModal({
    title: "HR H-B DECLARED",
    description: `Hrvatska Republika Herceg-Bosna has been established with
      headquarters in Mostar. Croatian parallel institutions now active.`,
    effects: effects,
    nextSteps: [
      "SDS will likely respond by accelerating RS declaration",
    ]
  });
}
```

```

        "Begin organizing HVO (Croatian Defense Council)",
        "Consolidate control in Croatian-majority areas"
    ]
});
}

function secureCroatianMunicipalities() {
    // Municipalities with Croatian majority or HDZ control
    const croatianStrongholds = [
        10626, // Ljubuški
        10405, // Grude
        10243, // Čapljina
        10260, // Čitluk
        10570, // Široki Brijeg
        // ... etc
    ];

    croatianStrongholds.forEach(munID => {
        const mun = gameState.municipalities[munID];
        if (mun && mun.ethnicComposition.croat > 50) {
            // Strengthen HR H-B control
            if (typeof mun.effectiveControl === 'string' && mun.effectiveControl === 'hrhb') {
                mun.stability.score += 15; // Strengthen stability
            } else if (mun.effectiveControl.factions?.includes('hrhb')) {
                // Shift contested → HRHB control
                mun.effectiveControl = 'hrhb';
            }
        }
    });
}

```

Part 3: RS Declaration (January 1992)

3.1 Activation Conditions

javascript

```

function checkRSDeclarationAvailability() {
    const currentDate = gameState.currentDate;
    const windowStart = new Date("1991-11-01");
    const windowEnd = new Date("1992-02-29");
    const hrhbDeclared = gameState.declarations.hrhb.declared;

    // Window opens in November OR immediately if HR H-B declared
    if ((currentDate >= windowStart || hrhbDeclared) &&
        currentDate <= windowEnd &&
        !gameState.declarations.rs.declared &&
        gameState.currentEntity === 'rs') {

        gameState.declarations.rs.available = true;

        // Calculate pressure
        let pressure = gameState.declarations.rs.pressure.base;
        if (hrhbDeclared) {
            pressure += gameState.declarations.rs.pressure.hrhbBonus;
        }
        gameState.declarations.rs.pressure.base = pressure;

        // First time available - show prompt
        if (gameState.declarations.rs.pressure.turnsSinceAvailable === 0) {
            promptRSDeclaration();
        }
    }
}

```

3.2 Decision Modal

javascript

```

function promptRSDeclaration() {
    const turnsWaited = gameState.declarations.rs.pressure.turnsSinceAvailable;
    const currentDate = gameState.currentDate;
    const optimalDate = new Date("1992-01-09");
    const hrhbDeclared = gameState.declarations.hrhb.declared;

    // Urgency increases if HR H-B declared
    const urgency = hrhbDeclared ? "CRITICAL" :
        turnsWaited > 5 ? "HIGH" : "MEDIUM";

    // Calculate timing
    let timingAnalysis = "";
    if (currentDate < optimalDate) {
        timingAnalysis = "EARLY: Heavy international condemnation but organizational advantage.";
    } else if (currentDate.getTime() === optimalDate.getTime()) {
        timingAnalysis = "OPTIMAL: Maximum legitimacy window before BiH referendum.";
    } else {
        timingAnalysis = "DELAYED: Risk BiH gaining independence first. Lose legitimacy.";
    }

    showDecisionModal({
        id: "rs_declaration",
        title: "DECLARE REPUBLIKA SRPSKA?",
        urgency: urgency,

        context: `
            SDS has consolidated control over Serb-majority municipalities.
            ${hrhbDeclared ? "Croatia has declared HR H-B. " : ""}
            BiH government is preparing independence referendum.
            JNA support is still available but time is limited.

            Serbian leaders propose declaring Republika Srpska to ensure
            Serbian self-determination within Yugoslavia.

            Timing Analysis: ${timingAnalysis}
            ${hrhbDeclared ? "⚠ HR H-B already declared - pressure to respond." : ""}
        `,

        options: [
            {
                label: "DECLARE REPUBLIKA SRPSKA NOW",
                hotkey: "D",
                effects: {

```

```

    immediate: [
      "Create Republika Srpska",
      "Establish Serbian parallel institutions",
      "Enable VRS (Army of Republika Srpska) formation",
      `International Visibility +${2 + calculateRSEarlyPenalty(currentDate)}`,
      "Trigger: R BiH will hold independence referendum",
      "Enable: Police takeovers in mixed municipalities",
      "Enable: TO weapons depot seizures"
    ],
    warning: [
      "This will trigger BiH independence referendum within weeks",
      "War will become inevitable",
      "International condemnation is certain"
    ]
  },
  consequences: calculateRSDeclarationEffects(currentDate),
  style: "primary"
},
{
  label: "WAIT (Review Next Turn)",
  hotkey: "W",
  effects: {
    immediate: [
      "Delay allows BiH to consolidate",
      "JNA support may weaken",
      "Risk losing legitimacy window"
    ],
    warning: turnsWaited > 6 ?
      "⚠ CRITICAL: BiH referendum imminent. Must declare before March 1992." :
      "Pressure increases. BiH independence approaching."
  },
  consequences: {
    pressureIncrease: 15,
    jnaSupport: -5,
    legitimacyRisk: calculateLegitimacyRisk(turnsWaited + 1)
  },
  style: "secondary"
}
],

```

historicalNote: `

Historically, Republika Srpska was declared on January 9, 1992, following the example of HR H-B and in anticipation of BiH's independence referendum. This declaration made war virtually inevitable.

```

});
}

function calculateRSDeclarationEffects(currentDate) {
  const optimal = new Date("1992-01-09");
  const daysEarly = (optimal - currentDate) / (1000 * 60 * 60 * 24);
  const daysLate = (currentDate - optimal) / (1000 * 60 * 60 * 24);

  const effects = {
    ivp: 2,
    jnaSupport: 10,
    enableCoercion: true,
    triggersReferendum: true
  };

  // Early = more IVP but better preparation
  if (daysEarly > 30) {
    effects.ivp += 1;
    effects.jnaSupport += 5;
  }

  // Late = less IVP but weaker position
  if (daysLate > 30) {
    effects.ivp -= 1;
    effects.jnaSupport -= Math.floor(daysLate / 30) * 3;
    effects.legitimacyPenalty = Math.floor(daysLate / 15);
  }

  return effects;
}

```

3.3 Declaration Execution

```
javascript
```

```
function executeRSDeclaration() {
  const decl = gameState.declarations.rs;

  decl.declared = true;
  decl.declarationDate = gameState.currentDate;
  decl.declarationTurn = gameState.currentTurn;

  const effects = calculateRSDeclarationEffects(gameState.currentDate);

  // Apply effects
  gameState.constraints.ivp += effects.ivp;

  // Secure Serbian-majority municipalities
  secureSerbianMunicipalities();

  // Enable coercive actions
  gameState.enabledActions.push(
    'police_takeover',
    'to_depot_seizure',
    'form_vrs',
    'checkpoint_intimidation'
  );

  // CRITICAL: Trigger RBiH referendum pressure
  gameState.declarations.rbih.pressure.rsBonus = 30;
  gameState.declarations.rbih.pressure.internationalPressure = 20;
  gameState.declarations.rbih.available = true;

  // Event log
  gameState.events.push({
    turn: gameState.currentTurn,
    date: gameState.currentDate,
    type: "DECLARATION",
    faction: "rs",
    title: "Republika Srpska Declared",
    description: "Serbian Democratic Party (SDS) declares Republika Srpska.",
    effects: effects
  });

  // Show result
  showResultModal({
    title: "REPUBLIKA SRPSKA DECLARED",
    description: `Republika Srpska proclaimed with capital in Pale.`
  });
}
```

```
        Serbian parallel institutions now active.',
effects: effects,
warning: `
    ⚠ BiH government will respond with independence referendum.
    ⚠ War is now virtually inevitable.
`,
nextSteps: [
    "Prepare for BiH independence referendum",
    "Consolidate control in Serb-majority areas",
    "Coordinate with JNA for security operations",
    "Seize TO weapons depots in mixed municipalities"
]
});
}
```

Part 4: RBiH Independence Referendum (March 1992)

4.1 Activation Conditions

```
javascript
```



```

function checkRBiHReferendumAvailability() {
  const currentDate = gameState.currentDate;
  const windowStart = new Date("1992-01-01");
  const windowEnd = new Date("1992-03-31");
  const rsDeclared = gameState.declarations.rs.declared;

  // Opens in January OR immediately after RS declaration
  if ((currentDate >= windowStart || rsDeclared) &&
    currentDate <= windowEnd &&
    !gameState.declarations.rbih.referendumScheduled &&
    gameState.currentEntity === 'rbih') {

    gameState.declarations.rbih.available = true;

    // Calculate international pressure
    if (rsDeclared) {
      gameState.declarations.rbih.pressure.internationalPressure = 20;
    }

    // First prompt
    if (gameState.declarations.rbih.pressure.turnsSinceAvailable === 0) {
      promptRBiHReferendum();
    }
  }
}

```

4.2 Decision Modal

javascript

```
function promptRBiHReferendum() {  
  const turnsWaited = gameState.declarations.rbih.pressure.turnsSinceAvailable;  
  const currentDate = gameState.currentDate;  
  const rsDeclared = gameState.declarations.rs.declared;  
  const hrhbDeclared = gameState.declarations.hrhb.declared;
```

```
  const urgency = rsDeclared ? "CRITICAL" : "HIGH";
```

```
  showDecisionModal({  
    id: "rbih_referendum",  
    title: "HOLD INDEPENDENCE REFERENDUM?",  
    urgency: urgency,  
  
    context: `  
      ${rsDeclared ? "Republika Srpska has been declared. " : ""}  
      ${hrhbDeclared ? "HR H-B has been declared. " : ""}
```

The European Community has offered recognition to BiH if it holds
a referendum on independence. International pressure is mounting.

SDS has announced Serbs will boycott any referendum. This will
undermine legitimacy but is unavoidable.

⚠ WARNING: Holding this referendum will lead directly to war.
There is no diplomatic solution once independence is declared.

```
`,
```

```
  options: [  
    {  
      label: "SCHEDULE REFERENDUM",  
      hotkey: "R",  
      effects: {  
        immediate: [  
          "Schedule referendum for next turn (Feb 29-March 1, 1992)",  
          "Serbs will boycott (expected ~35% no participation)",  
          "International recognition if successful",  
          "International Visibility +3",  
          "⚠ War will begin within 2-4 weeks after independence"  
        ],  
        delayed: [  
          "BiH gains international recognition",  
          "Establish Army of RBiH (ARBiH)",  
          "War countdown begins immediately after referendum"        ]  
      }  
    }  
  ],
```

```

    ]
  },
  consequences: {
    schedulesReferendum: true,
    warInevitable: true,
    ivpIncrease: 3
  },
  style: "critical"
},
{
  label: "DELAY (Review Next Turn)",
  hotkey: "W",
  effects: {
    immediate: [
      "Lose international sympathy",
      "SDS consolidates further",
      "JNA positions for conflict"
    ],
    warning: turnsWaited > 4 ?
      "⚠ CRITICAL: International community losing patience. Must decide." :
      "Cannot delay past March 1992."
  },
  consequences: {
    internationalSupport: -2,
    pressureIncrease: 20
  },
  style: "secondary"
}
],

```

historicalNote: `

The independence referendum was held February 29-March 1, 1992.
 Serbs boycotted as announced. Turnout was 63.4% with 99.7% voting
 for independence. BiH declared independence on March 3, 1992.

War broke out in early April 1992.

```

  });

```

```

}

```

4.3 Referendum Execution

javascript

```
function scheduleRBiHReferendum() {
  const decl = gameState.declarations.rbih;

  decl.referendumScheduled = true;
  decl.referendum.scheduled = true;
  decl.referendum.scheduledTurn = gameState.currentTurn + 1;

  gameState.events.push({
    turn: gameState.currentTurn,
    date: gameState.currentDate,
    type: "REFERENDUM_SCHEDULED",
    faction: "rbih",
    title: "Independence Referendum Scheduled",
    description: "Referendum set for February 29-March 1, 1992."
  });

  showResultModal({
    title: "REFERENDUM SCHEDULED",
    description: "Independence referendum will be held next turn.",
    warning: "⚠ Serbs have announced boycott. War is now inevitable.",
    nextSteps: [
      "Referendum will resolve next turn",
      "Prepare for Serbian boycott (reduces legitimacy)",
      "International recognition likely if successful",
      "Begin war preparations immediately"
    ]
  });
}

function conductRBiHReferendum() {
  const decl = gameState.declarations.rbih;
  const ref = decl.referendum;

  // Simulate referendum results (historical)
  ref.boycottedBySerbs = true;
  ref.turnout = 63.4; // Due to Serb boycott
  ref.yesVote = 99.7;

  // Referendum passes
  decl.referendumHeld = true;
  decl.declared = true;
  decl.declarationDate = gameState.currentDate;
  decl.declarationTurn = gameState.currentTurn;
}
```

```

// Effects
gameState.constraints.ivp += 3;
gameState.declarations.rbih.effects.internationalRecognition = true;

// Enable ARBiH
gameState.enabledActions.push('form_arbih', 'territorial_defense_mobilization');

// CRITICAL: Start war countdown
gameState.warStatus.warInevitable = true;
gameState.warStatus.countdownStarted = true;
gameState.warStatus.turnsUntilWar = 3; // 2-4 weeks
gameState.warStatus.phase = "escalation";

// Event log
gameState.events.push({
  turn: gameState.currentTurn,
  date: gameState.currentDate,
  type: "REFERENDUM_RESULT",
  faction: "rbih",
  title: "BiH Independence Referendum Passes",
  description: `Turnout: ${ref.turnout}% (Serbs boycotted).
    Yes vote: ${ref.yesVote}%.
    BiH declares independence.`,
  effects: {
    internationalRecognition: true,
    ivp: 3,
    warCountdown: 3
  }
});

// Show result with dramatic warning
showResultModal({
  title: "BOSNIA-HERZEGOVINA INDEPENDENT",
  description: `
    Referendum results: ${ref.turnout}% turnout, ${ref.yesVote}% yes.

    BiH has declared independence from Yugoslavia.
    International recognition granted.

    ⚠️ ⚠️ ⚠️ WAR COUNTDOWN INITIATED ⚠️ ⚠️ ⚠️
  `,
  warning: `
    Republika Srpska has rejected the referendum.
  `
});

```

Tensions are escalating rapidly.

War will begin in \${gameState.warStatus.turnsUntilWar} turns.

There is no diplomatic solution.

,

nextSteps: [

"IMMEDIATE: Mobilize Territorial Defense",

"IMMEDIATE: Secure key municipalities",

"URGENT: Prepare for attacks on vulnerable areas",

"Incidents and provocations will escalate each turn"

],

style: "critical"

});

}

Part 5: War Outbreak (April 1992)

5.1 War Countdown System

javascript

```
function processWarCountdown() {
  if (!gameState.warStatus.countdownStarted) return;

  const countdown = gameState.warStatus;

  if (countdown.turnsUntilWar > 0) {
    // Each turn during countdown
    countdown.turnsUntilWar--;

    // Escalation events
    switch(countdown.phase) {
      case "escalation":
        processEscalationPhase();
        if (countdown.turnsUntilWar === 2) {
          countdown.phase = "incidents";
        }
        break;

      case "incidents":
        processIncidentPhase();
        if (countdown.turnsUntilWar === 1) {
          countdown.phase = "imminent";
        }
        break;

      case "imminent":
        processImminentPhase();
        break;
    }

    // Show countdown warning
    showCountdownWarning(countdown.turnsUntilWar);

  } else {
    // WAR BEGINS
    initiateWar();
  }
}

function processEscalationPhase() {
  // Turn 1 of countdown: Tensions rising
  const events = [
    "Roadblocks appear in mixed municipalities",
  ]
}
```

```
    "Checkpoint incidents reported",  
    "Weapons distributed to local militias",  
    "Displacement begins in contested areas",  
    "JNA repositions units"  
];
```

```
// Increase coercion levels
```

```
Object.values(gameState.municipalities).forEach(mun => {  
    if (mun.stability.score < 40) {  
        mun.coercionLevel = Math.min(5, mun.coercionLevel + 2);  
    }  
});
```

```
// Log events
```

```
events.forEach(event => {  
    gameState.events.push({  
        turn: gameState.currentTurn,  
        type: "WAR_ESCALATION",  
        description: event  
    });  
});  
}
```

```
function processIncidentPhase() {
```

```
// Turn 2 of countdown: Shooting incidents
```

```
const incidents = [  
    {  
        location: "Sarajevo",  
        description: "Shots fired at peace demonstration. Casualties reported.",  
        type: "SHOOTING"  
    },  
    {  
        location: "Bijeljina",  
        description: "Serbian paramilitary units enter city. Fighting erupts.",  
        type: "PARAMILITARY"  
    },  
    {  
        location: "Bosanski Brod",  
        description: "Artillery exchange across Sava river.",  
        type: "ARTILLERY"  
    }  
];
```

```
// Apply incidents
```



```
incidents.forEach(incident => {
  gameState.events.push({
    turn: gameState.currentTurn,
    type: "WAR_INCIDENT",
    title: incident.description,
    location: incident.location
  });

  // Find and pressure relevant municipalities
  // (implementation would look up municipality by name)
});
```

```
// Increase exhaustion
gameState.constraints.exhaustion += 0.5;
}
```

```
function processImminentPhase() {
  // Turn 3 of countdown: Last chance to prepare
  gameState.events.push({
    turn: gameState.currentTurn,
    type: "WAR_IMMINENT",
    title: "⚠ WAR IMMINENT ⚠",
    description: "Full-scale war will begin next turn. Make final preparations."
  });
}
```

```
// Final warning modal
showCriticalModal({
  title: "⚠ WAR BEGINS NEXT TURN ⚠",
  description: `
```

This is your final turn of preparation.

Next turn:

- Game phase changes to MAIN WAR
- Municipal flips enabled
- Combat operations begin
- Displacement accelerates
- International intervention begins

There is no turning back.

`,

```
actions: [
  "Review vulnerable municipalities",
  "Check stability scores",
  "Prepare defensive positions",
```

"Final mobilization orders"

```
]
});
}
```

5.2 War Initiation

javascript

```
function initiateWar() {  
  // Change game phase  
  gameState.gamePhase = "main_war";  
  gameState.warStatus.warStartDate = gameState.currentDate;  
  
  // Enable war mechanics  
  gameState.enabledActions.push(  
    'military_operation',  
    'artillery_bombardment',  
    'siege_operation',  
    'offensive_operation',  
    'defensive_operation'  
  );  
  
  // Trigger initial flips in very unstable areas  
  triggerInitialFlips();  
  
  // Major event  
  gameState.events.push({  
    turn: gameState.currentTurn,  
    date: gameState.currentDate,  
    type: "WAR_OUTBREAK",  
    title: "⚠ ⚠ ⚠ WAR IN BOSNIA-HERZEGOVINA ⚠ ⚠ ⚠",  
    description: `  
      Full-scale war has erupted across Bosnia-Herzegovina.  
  
      Multiple municipalities under attack.  
      Mass displacement beginning.  
      International community responds.  
  
      The war cannot be stopped through diplomacy.  
      Only exhaustion, collapse, or imposed settlement will end it.  
    `,  
  });  
  
  // Show war start modal  
  showCriticalModal({  
    title: "WAR HAS BEGUN",  
    description: `  
      April 1992. The war in Bosnia-Herzegovina has started.  
  
      Game phase: MAIN WAR  
    `,  
  });  
}
```

New mechanics enabled:

- ✓ Municipal control flips
- ✓ Military operations
- ✓ Combat resolution
- ✓ Siege warfare
- ✓ Displacement at scale
- ✓ International intervention

This is no longer about preventing war.

This is about managing an unwinnable conflict.

`

objectives: [

 "Secure core territories",

 "Manage exhaustion constraints",

 "Balance military action with IVP",

 "Survive until settlement possible"

],

style: "war_critical"

});

}

function triggerInitialFlips() {

// Municipalities with stability < 20 flip immediately

 Object.entries(gameState.municipalities).forEach(([id, mun]) => {

 if (mun.stability.score < 20 && !mun.hasFlipped) {

// Determine which faction should take control

 const targetFaction = determineFlipTarget(mun);

 if (targetFaction) {

// Execute flip

 mun.preFlipControl = mun.effectiveControl;

 mun.effectiveControl = targetFaction;

 mun.hasFlipped = true;

 mun.flipTurn = gameState.currentTurn;

 mun.controlStrength = {rbih: 0, rs: 0, hrhb: 0};

 mun.controlStrength[targetFaction] = 100;

// Log flip

 gameState.events.push({

 turn: gameState.currentTurn,

 type: "CONTROL_FLIP",

 municipality: mun.name,

 from: mun.preFlipControl,

 to: targetFaction,

```

        reason: "INITIAL_WAR_COLLAPSE"
    });
}
}
});
}

function determineFlipTarget(mun) {
    // Flip to faction with plurality + organizational advantage
    const ethnicPluralityFaction =
        mun.ethnicComposition.serb > mun.ethnicComposition.bosniak &&
        mun.ethnicComposition.serb > mun.ethnicComposition.croat ? 'rs' :
        mun.ethnicComposition.bosniak > mun.ethnicComposition.serb &&
        mun.ethnicComposition.bosniak > mun.ethnicComposition.croat ? 'rbih' :
        'hrhb';

    // Check organizational control
    if (mun.organization.sds === 'strong' && mun.organization.jna === 'present') {
        return 'rs';
    }

    return ethnicPluralityFaction;
}

```

Part 6: UI Implementation

6.1 Declaration Status Display

```

javascript

```

// Top bar addition showing declaration status

```
function renderDeclarationStatus() {
  const html = `
    <div class="declaration-status-bar">
      <div class="declaration-item ${gameState.declarations.hrhb.declared ? 'declared' : 'pending'}">
        <span class="decl-label">HR H-B:</span>
        <span class="decl-value">${gameState.declarations.hrhb.declared ?
          '✓ DECLARED' :
            gameState.declarations.hrhb.available ? '⌚ DECISION PENDING' : '- -'}</span>
      </div>

      <div class="declaration-item ${gameState.declarations.rs.declared ? 'declared' : 'pending'}">
        <span class="decl-label">RS:</span>
        <span class="decl-value">${gameState.declarations.rs.declared ?
          '✓ DECLARED' :
            gameState.declarations.rs.available ? '⌚ DECISION PENDING' : '- -'}</span>
      </div>

      <div class="declaration-item ${gameState.declarations.rbih.declared ? 'declared' : 'pending'}">
        <span class="decl-label">RBiH:</span>
        <span class="decl-value">${gameState.declarations.rbih.declared ?
          '✓ INDEPENDENT' :
            gameState.declarations.rbih.available ? '⌚ REFERENDUM PENDING' : '- -'}</span>
      </div>

      ${gameState.warStatus.countdownStarted ? `
        <div class="war-countdown critical">
          <span class="countdown-label">⚠ WAR IN:</span>
          <span class="countdown-value">${gameState.warStatus.turnsUntilWar} TURNS</span>
        </div>
      ` : ''}
    </div>
  `;

  return html;
}
```

6.2 CSS Styling

CSS

```
.declaration-status-bar {  
  display: flex;  
  gap: 20px;  
  padding: 10px 20px;  
  background: rgba(30, 41, 59, 0.9);  
  border-bottom: 1px solid var(--border-tactical);  
  font-family: 'JetBrains Mono', monospace;  
  font-size: 0.8em;  
}
```

```
.declaration-item {  
  display: flex;  
  gap: 8px;  
  align-items: center;  
}
```

```
.decl-label {  
  color: var(--text-secondary);  
  font-weight: 600;  
  text-transform: uppercase;  
  letter-spacing: 1px;  
}
```

```
.decl-value {  
  color: var(--text-muted);  
  font-weight: 700;  
}
```

```
.declaration-item.declared .decl-value {  
  color: var(--phosphor-amber);  
}
```

```
.declaration-item.pending .decl-value {  
  color: #f97316;  
  animation: urgentBlink 2s infinite;  
}
```

```
.war-countdown {  
  display: flex;  
  gap: 8px;  
  padding: 0 15px;  
  border-left: 2px solid var(--warning-red);  
  margin-left: 20px;
```

```
}

.countdown-label {
  color: var(--warning-red);
  font-weight: 700;
  text-transform: uppercase;
}

.countdown-value {
  color: var(--warning-red);
  font-weight: 900;
  font-size: 1.2em;
  animation: urgentBlink 1s infinite;
}
```

6.3 Decision Modal Component

javascript


```
function showDecisionModal(config) {
  const modal = document.createElement('div');
  modal.className = 'modal-overlay';
  modal.innerHTML = `
    <div class="decision-modal ${config.urgency}">
      <div class="modal-header">
        <div class="urgency-badge urgency-${config.urgency}">
          ${config.urgency}
        </div>
        <h2 class="modal-title">${config.title}</h2>
      </div>

      <div class="modal-body">
        <div class="context-section">
          ${config.context}
        </div>

        ${config.warning ? `
          <div class="warning-section">
            ⚠ ${config.warning}
          </div>
        ` : ""}

        <div class="options-section">
          ${config.options.map((opt, i) => `
            <div class="decision-option ${opt.style}" data-option="${i}">
              <div class="option-header">
                <span class="option-label">${opt.label}</span>
                ${opt.hotkey ? `<span class="hotkey">[${opt.hotkey}]</span>` : ""}
              </div>

              <div class="option-effects">
                <div class="effects-title">Immediate Effects:</div>
                <ul>
                  ${opt.effects.immediate.map(e => `<li>${e}</li>`).join("")}
                </ul>

                ${opt.effects.delayed ? `
                  <div class="effects-title">Future Consequences:</div>
                  <ul>
                    ${opt.effects.delayed.map(e => `<li>${e}</li>`).join("")}
                  </ul>
                ` : ""}
              </div>
            </div>
          `)}
        </div>
      </div>
    </div>`
}
```

```

        ${opt.effects.warning ? `
        <div class="option-warning">⚠️ ${opt.effects.warning}</div>
        ` : ""}
    </div>
</div>
    `).join("")}
</div>

    ${config.historicalNote ? `
    <div class="historical-note">
        <div class="note-title">📖 Historical Note</div>
        <div class="note-text">${config.historicalNote}</div>
    </div>
    ` : ""}
</div>
</div>
`;

```

// Add event listeners

```

modal.querySelectorAll('.decision-option').forEach((opt, i) => {
    opt.addEventListener('click', () => {
        handleDecisionChoice(config.id, i);
        document.body.removeChild(modal);
    });
});

```

// Keyboard shortcuts

```

document.addEventListener('keydown', function handler(e) {
    config.options.forEach((opt, i) => {
        if (opt.hotkey && e.key.toLowerCase() === opt.hotkey.toLowerCase()) {
            handleDecisionChoice(config.id, i);
            document.body.removeChild(modal);
            document.removeEventListener('keydown', handler);
        }
    });
});

```

```

document.body.appendChild(modal);
}

```

```

function handleDecisionChoice(decisionId, optionIndex) {
    switch(decisionId) {
        case 'hrhb_declaration':

```

```
        if (optionIndex === 0) executeHRHBDeclaration();
        break;
    case 'rs_declaration':
        if (optionIndex === 0) executeRSDeclaration();
        break;
    case 'rbih_referendum':
        if (optionIndex === 0) scheduleRBiHReferendum();
        break;
    }

    // Refresh UI
    gameState.updateUI();
}
```

Part 7: Testing & Validation

7.1 Test Scenarios

Scenario 1: Historical Path

Sept 1991: Player waits
Nov 1991: HR H-B declares (historical timing)
Jan 1992: RS declares (historical timing)
March 1992: RBiH referendum (historical timing)
April 1992: War begins
Expected: Matches historical timeline

Scenario 2: Early Declarations

Sept 1991: HR H-B declares immediately
Oct 1991: RS declares in response
Dec 1991: RBiH forced to referendum early
Feb 1992: War begins early
Expected: Higher IVP, faster escalation

Scenario 3: Delayed Declarations

Dec 1991: HR H-B declares (late)
Feb 1992: RS declares (near deadline)
March 1992: RBiH referendum (forced)
April 1992: War begins
Expected: Lost opportunities, weaker positions

7.2 Validation Checklist

- ☐ HR H-B can only declare Sept-Dec 1991
 - ☐ RS can only declare after HR H-B OR after Nov 1991
 - ☐ RS cannot declare after BiH independence
 - ☐ RBiH referendum triggered by RS OR Jan 1992
 - ☐ War countdown starts after RBiH independence
 - ☐ War cannot be prevented after referendum
 - ☐ Early declarations increase IVP correctly
 - ☐ Late declarations apply penalties
 - ☐ Cascade triggers work correctly
 - ☐ UI updates reflect declaration status
 - ☐ Historical dates produce optimal results
-

Part 8: Integration Points

8.1 With Flip System

javascript

```

// After war starts, use declaration status for flip calculations
function calculateFlipPressure(munID, targetController) {
  let pressure = 0;

  // Base pressure from war phase
  if (gameState.gamePhase === 'main_war') {
    pressure += 20;
  }

  // Bonus if faction declared entity
  if (targetController === 'rs' && gameState.declarations.rs.declared) {
    pressure += 10; // RS has institutional backing
  }

  // Timing bonus (earlier declaration = more prepared)
  const turnsSinceDeclared = gameState.currentTurn -
    gameState.declarations[targetController].declarationTurn;
  pressure += Math.min(20, turnsSinceDeclared * 2);

  return pressure;
}

```

8.2 With Constraint System

javascript

```
// Declarations affect IVP constraint
function updateIVPFromDeclarations() {
  let ivp = gameState.constraints.ivp;

  // Each declaration increases visibility
  if (gameState.declarations.hrhb.declared) {
    ivp += gameState.declarations.hrhb.effects.ivpIncrease;
  }
  if (gameState.declarations.rs.declared) {
    ivp += gameState.declarations.rs.effects.ivpIncrease;
  }
  if (gameState.declarations.rbih.declared) {
    ivp += gameState.declarations.rbih.effects.ivpIncrease;
  }

  gameState.constraints.ivp = ivp;
}
```

8.3 With Turn System

javascript

```
function processTurn() {  
    // 1. Check declaration windows  
    checkDeclarationWindows();  
  
    // 2. Increase pressure for delays  
    increasePressureForDelays();  
  
    // 3. Process referendum if scheduled  
    if (gameState.declarations.rbih.referendum.scheduled &&  
        gameState.currentTurn === gameState.declarations.rbih.referendum.scheduledTurn) {  
        conductRBIHReferendum();  
    }  
  
    // 4. Process war countdown  
    if (gameState.warStatus.countdownStarted) {  
        processWarCountdown();  
    }  
  
    // 5. Normal turn processing  
    // ... rest of turn logic  
}
```

Part 9: Implementation Roadmap

Phase 1: Core Framework (Week 1)

- ☐ Add declaration data structures to gameState
- ☐ Implement window checking functions
- ☐ Create basic decision modal component
- ☐ Add declaration status to UI

Phase 2: HR H-B Declaration (Week 2)

- ☐ Implement HR H-B availability checking
- ☐ Create HR H-B decision modal with full effects
- ☐ Implement execution function
- ☐ Test timing modifiers
- ☐ Validate against historical timeline

Phase 3: RS Declaration (Week 3)

- ☐ Implement RS availability (with HR H-B trigger)
- ☐ Create RS decision modal
- ☐ Implement execution with coercion enables
- ☐ Test cascade from HR H-B → RS
- ☐ Validate timing windows

Phase 4: RBiH Referendum (Week 4)

- ☐ Implement referendum availability (with RS trigger)
- ☐ Create referendum decision modal
- ☐ Implement 2-turn referendum process
- ☐ Test referendum results calculation
- ☐ Validate Serb boycott mechanics

Phase 5: War Countdown (Week 5)

- ☐ Implement countdown system
- ☐ Create escalation phase events
- ☐ Create incident phase events
- ☐ Test countdown progression
- ☐ Validate war initiation

Phase 6: Integration & Polish (Week 6)

- ☐ Integrate with flip system
 - ☐ Integrate with constraint system
 - ☐ Add comprehensive UI elements
 - ☐ Create event log entries
 - ☐ Polish modal styling
 - ☐ Add sound effects (optional)
 - ☐ Final testing
-

Part 10: Future Enhancements

Alternative History Mode

javascript


```
// What if declarations were different?
gameState.alternativeHistory = {
  enabled: false,
  allowPreventWar: false,    // Still cannot prevent
  allowedDelays: {
    hrhb: 6, // Can delay up to 6 turns
    rs: 4,
    rbiH: 3
  }
};
```

Diplomatic Interventions

```
javascript

// EC/UN attempts to delay declarations
gameState.diplomaticPressure = {
  ecMediation: false,
  unObservers: false,
  delayBonus: 0 // Adds turns to windows
};
```

Declaration Quality

```
javascript

// How declaration was done affects legitimacy
gameState.declarations.rs.quality = {
  parliamentaryVote: true, // Proper procedure
  internationalNotification: true,
  constitutionalBasis: "questionable",
  legitimacyScore: 60 // 0-100
};
```

Summary

This implementation creates a **hybrid system** where:

1. **Events are inevitable** - HR H-B, RS, RBiH, and war will all happen
2. **Timing is strategic** - Players choose WHEN (within windows)
3. **Consequences are real** - Early/late declarations have concrete effects

4. **Cascades create pressure** - Each declaration triggers the next
5. **War is unavoidable** - Once RBiH declares independence, war countdown begins

The system preserves historical plausibility while giving players meaningful agency over the MANNER and TIMING of escalation, not WHETHER it occurs.

This matches the game's core design: players are not preventing war, they are managing an impossible situation created by structural forces beyond any faction's control.