

A WAR WITHOUT VICTORY

MASTER PROJECT OVERVIEW & DESIGN DOCUMENT

Version: 1.0 **Date:** January 19, 2026 **Status:** Design Complete, Ready for Implementation

TABLE OF CONTENTS

1. [Project Vision](#)
 2. [Core Design Philosophy](#)
 3. [Game Systems Overview](#)
 4. [System 1: Settlement-Level Control](#)
 5. [System 2: Historical Brigade System](#)
 6. [System 3: Dynamic Command Structure](#)
 7. [System 4: Front-Line Operations](#)
 8. [System 5: Uncertain Command](#)
 9. [System 6: Pre-War Phase](#)
 10. [System 7: Population Tracking](#)
 11. [Planned Systems](#)
 12. [Technical Architecture](#)
 13. [Data Structures](#)
 14. [UI/UX Design](#)
 15. [Implementation Roadmap](#)
 16. [Historical Accuracy](#)
 17. [Success Criteria](#)
-

PROJECT VISION

Mission Statement

Create an educational, ethically-grounded grand strategy game about the Bosnian War (1992-1995) that:

- **Educes** players about historical reality through authentic data and units
- **Challenges** players with realistic command decisions and uncertainty

- **Humanizes** the conflict by tracking civilian impact and population consequences
- **Respects** victims by avoiding glorification of atrocities
- **Engages** players through meaningful strategic choices and emergent gameplay

What This Game Is

- A serious historical simulation
- An educational tool about the Yugoslav Wars
- A strategic command experience with operational-level focus
- A game that shows the complexity and tragedy of war

What This Game Is Not

- Entertainment violence or power fantasy
 - Glorification of ethnic cleansing
 - Oversimplified "good vs evil" narrative
 - Micromanagement of individual soldiers
 - Deterministic puzzle with "correct" solutions
-

CORE DESIGN PHILOSOPHY

1. Historical Accuracy + Player Agency

Historical Elements (Fixed):

- Real settlements from 1991 census (2,200 total)
- Real brigades from Balkan Battlegrounds (150+)
- Authentic formation dates and timelines
- Actual demographic data
- Historical equipment levels
- Real geographic terrain

Player Agency (Flexible):

- Create and organize command structure
- Make pre-war preparation decisions
- Allocate resources and priorities

- Choose operational strategies
- Reorganize forces as needed
- Determine victory conditions

Result: Authentic historical context with meaningful strategic freedom

2. Operational-Level Focus

NOT Tactical:

- No individual soldier control
- No hex-by-hex movement
- No micromanaged combat

YES Operational:

- Brigade-level units
- Front sector management
- Corps/zone organization
- Multi-turn operations
- Strategic resource allocation

Why: Matches the scale of actual military command and reduces micromanagement tedium

3. Uncertainty and Friction

Orders Are Not Guaranteed:

- Success probability: 5-95%
- Factors: readiness, morale, supply, experience, terrain, enemy strength
- Outcomes: Success / Partial Success / Failure
- Consequences: Morale changes, casualties, experience gained

Why: Represents "fog of war," command limitations, and the unpredictability of military operations

4. Consequences Matter

Population Tracking:

- Civilian casualties recorded
- Refugee movements tracked
- Ethnic composition changes
- Historical memory preserved

Unit Status:

- Morale fluctuates
- Experience grows
- Equipment degrades
- Fatigue accumulates

Strategic Impact:

- Atrocities affect international reputation
- Civilian support affects recruitment
- Infrastructure destruction limits logistics

Why: No consequence-free actions; every decision has weight

5. Scalability and Clarity

Visual Design:

- Clean, readable map
- Only show important settlements (urban + capitals)
- NATO-style unit symbols
- Clear color coding
- Collapsible lists for 150+ brigades

Performance:

- <2 second load time
- 60 FPS animations
- <100ms render time
- Handles 2,200 settlements smoothly

Why: Realistic scale without overwhelming the player

GAME SYSTEMS OVERVIEW

Implemented Systems (Design Complete)

System	Status	Complexity	Dev Time
Settlement-Level Control	<input checked="" type="checkbox"/> Complete	Medium	4-6 hours
Historical Brigades	<input checked="" type="checkbox"/> Complete	High	6-8 hours
Dynamic Command Structure	<input checked="" type="checkbox"/> Complete	Medium	6-8 hours
Front-Line Operations	<input checked="" type="checkbox"/> Complete	High	8-10 hours
Uncertain Command	<input checked="" type="checkbox"/> Complete	Medium	6-8 hours
Pre-War Phase	<input checked="" type="checkbox"/> Complete	Medium	4-6 hours
Population Tracking	<input checked="" type="checkbox"/> Complete	Low	2-3 hours

Planned Systems (To Be Designed)

System	Priority	Complexity	Est. Dev Time
Diplomacy & International	High	High	12-15 hours
Supply & Logistics	High	Medium	6-8 hours
Intelligence & Recon	Medium	Medium	4-6 hours
Air Operations	Medium	Low	3-4 hours
Artillery System	Medium	Medium	4-6 hours
Leadership & Officers	Low	Medium	4-6 hours
Economic System	Low	Medium	6-8 hours
Victory Conditions	High	Low	2-3 hours

SYSTEM 1: SETTLEMENT-LEVEL CONTROL

Overview

Replace municipality-level control (109 units) with settlement-level control (2,200 settlements). Provides strategic depth and historical accuracy while maintaining visual clarity.

Key Features

Data Source:

- 1991 Yugoslav Census (mz_1991_census_mz_level.csv)
- 2,200 settlements (mjesne zajednice - MZ)
- Complete demographic data
- Population by ethnicity
- Municipality assignments

Classification:

- **Urban settlements:** Population $\geq 3,000$ (418 total)
- **Municipality capitals:** One per municipality (109 total)
- **Rural settlements:** Remaining settlements (1,782 total)

Visibility:

- **Shown on map:** Urban + capitals (~527 settlements)
- **Hidden but modeled:** All rural settlements
- **Why:** Clean visual design + realistic mechanics

Settlement Object Structure

```
javascript
```

```

settlement = {
    // Identity
    id: "sarajevo_centar__betanija",
    name: "Betanija",
    municipalityId: "sarajevo_centar",
    municipality: "Sarajevo Centar",

    // Classification
    urban: false,           // true if population ≥ 3,000
    isCapital: false,        // municipality capital?
    visible: false,          // shown on map?

    // Population (1991 Census)
    population: 1314,
    ethnicComposition: {
        bosniak: 27.85,      // percentages
        serb: 64.31,
        croat: 1.07,
        yugoslav: 3.65,
        other: 3.12
    },
    absoluteNumbers: {
        bosniak: 366,        // absolute counts
        serb: 845,
        croat: 14,
        yugoslav: 48,
        other: 41
    },
    // Control
    controller: "rs",       // 'rbih', 'rs', 'hrhb', 'contested', 'none'
    contested: false,
    controlStrength: 75,     // 0-100

    // Front line status
    frontLine: true,
    adjacentEnemies: ["rbih"],
    neighbors: ["sarajevo_centar__bjelave", ...],
    // Geography
    svgPath: "M 123,456 L...", // SVG path for rendering
    centroid: {x: 455, y: 260},
}

```

```
// Strategic importance
defensibility: 65,          // 0-100 (terrain, elevation)
supplyValue: 30,           // strategic crossroads, etc.
populationCenter: false,   // major urban area?

// Pre-war organization
organization: {
    police: "rs",          // which faction controls police
    to: "mixed",            // Territorial Defense presence
    sds: "strong",          // Serbian Democratic Party org
    sda: "present",         // Party of Democratic Action org
    hdz: "absent"           // Croatian Democratic Union org
}
}
```

Municipality Aggregation

javascript

```

function getMunicipalityControl(municipalityId) {
  const settlements = getSettlements(municipalityId);

  // Count by controller
  const counts = { rbih: 0, rs: 0, hrhb: 0, contested: 0 };
  settlements.forEach(s => counts[s.controller]++;
  }

  // Calculate percentages
  const total = settlements.length;
  const percentages = {
    rbih: (counts.rbih / total) * 100,
    rs: (counts.rs / total) * 100,
    hrhb: (counts.hrbhb / total) * 100,
    contested: (counts.contested / total) * 100
  };

  // Determine overall control ( $\geq 75\%$  for clear control)
  let controller = 'contested';
  if (percentages.rbih >= 75) controller = 'rbih';
  else if (percentages.rs >= 75) controller = 'rs';
  else if (percentages.hrbhb >= 75) controller = 'hrhb';

  return {
    controller: controller,
    percentages: percentages,
    counts: counts,
    total: total,
    settlements: settlements
  };
}

```

Front Line Calculation

javascript

```

function calculateFrontLines() {
    // Mark settlements as front line if they border enemy
    for (let settlement of settlements) {
        settlement.frontLine = false;
        settlement.adjacentEnemies = [];

        // Check neighbors
        for (let neighborId of settlement.neighbors) {
            const neighbor = getSettlement(neighborId);

            // Different controller = front line
            if (neighbor.controller !== settlement.controller &&
                neighbor.controller !== 'none') {
                settlement.frontLine = true;

                if (!settlement.adjacentEnemies.includes(neighbor.controller)) {
                    settlement.adjacentEnemies.push(neighbor.controller);
                }
            }
        }
    }
}

```

Initial Control Determination

Control at game start determined by:

1. **Municipal election results** (1990)
2. **Settlement demographics** (1991 census)
3. **Pre-war organization** (player decisions in political phase)

javascript

```

function calculateSettlementControl(settlement, municipality) {
    const eth = settlement.ethnicComposition;
    const munControl = municipality.effectiveControl;

    let score = 60; // Base score

    // Municipal control influence
    if (munControl === settlement.majorityGroup) {
        score += 20; // Municipality supports local majority
    }

    // Demographic strength
    if (settlement.majorityPct >= 75) score += 20;
    else if (settlement.majorityPct >= 60) score += 10;
    else if (settlement.majorityPct >= 50) score += 5;
    else score -= 10;

    // Pre-war organization
    if (settlement.organization.police === settlement.majorityGroup) {
        score += 10;
    }

    // Apply pre-war decisions (player modifiers)
    score += getPreWarModifiers(settlement);

    // Determine controller
    const factionMap = {
        'muslims': 'rbih',
        'serbs': 'rs',
        'croats': 'hrhb'
    };

    const controller = factionMap[settlement.majorityGroup];
    const contested = score < 60;

    return { controller, score, contested };
}

```

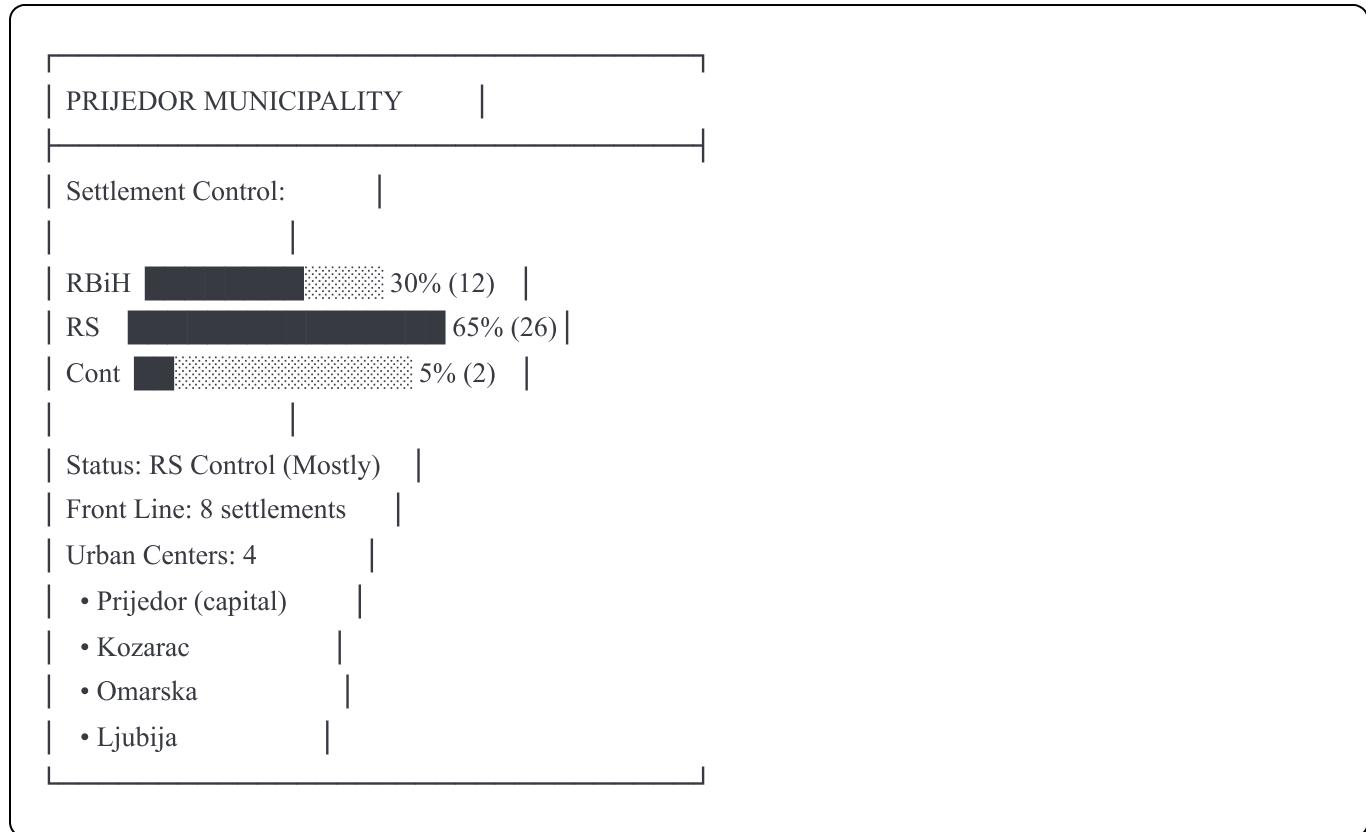
Visual Representation

Map Display:

- Settlement dots colored by controller:

- Green: RBiH control
- Red: RS control
- Blue: HR H-B control
- Brown: Contested
- Yellow glow on front-line settlements
- Labels for urban settlements and capitals
- Larger dots for urban, smaller for capitals

Municipality Panel:



Integration Points

- Pre-war phase determines initial control
- Brigade recruitment tied to settlement demographics
- Front sectors defined by settlement borders
- Population tracking at settlement level
- Victory conditions check settlement control

SYSTEM 2: HISTORICAL BRIGADE SYSTEM

Overview

150+ real brigades based on Balkan Battlegrounds (CIA Study, 2002). Brigades form historically in their home municipalities. At least one brigade per municipality.

Key Features

Historical Accuracy:

- Real brigade names (e.g., "1st Mountain Brigade 'Viteška'")
- Actual formation dates (April 1992 - December 1995)
- Home municipalities
- Realistic strength levels
- Authentic equipment assessments
- Quality ratings based on performance

Scale:

- **ARBiH (Army of RBiH):** ~180 brigades at peak
- **VRS (Vojska Republike Srpske):** ~90 brigades at peak
- **HVO (Hrvatsko Vijeće Obrane):** ~50 brigades at peak
- **Total:** ~320 brigade-level formations

Formation Timeline:

- April 1992: ~40 brigades total (early formations)
- September 1992: ~80 brigades (mobilization accelerates)
- January 1993: ~120 brigades (major expansion)
- April 1995: ~180 brigades (peak mobilization)

Brigade Object Structure

```
javascript
```

```
brigade = {  
    // Identity  
    id: "arbih_1_mountain",  
    name: "1st Mountain Brigade",  
    nickname: "Viteška",      // Optional nickname  
    faction: "rbih",         // 'rbih', 'rs', 'hrhb'  
  
    // Historical data  
    formed: {  
        turn: 5,  
        date: "1992-04-20"  
    },  
    disbanded: null,          // or {turn: X, date: "..."}  
  
    // Organization  
    corps: "1_korpus",       // Assigned corps/OZ (player-created)  
    division: null,           // If applicable  
    type: "mountain_infantry", // mountain, motorized, light, infantry, mechanized, special_forces  
    quality: "elite",         // militia, regular, veteran, elite  
  
    // Location  
    homeMunicipality: "sarajevo_centar",  
    currentMunicipality: "sarajevo_centar",  
    recruitmentBase: "Sarajevo inner city - Bosniak population",  
  
    // Strength  
    currentStrength: 3200,  
    maxStrength: 3500,  
    casualties: 0,             // Historical total  
  
    // Equipment  
    equipment: {  
        smallArms: "adequate", // poor, mixed, adequate, good, excellent  
        heavyWeapons: "limited", // none, minimal, limited, adequate, good, excellent  
        antiTank: "minimal",  
        antiAir: "minimal",  
        artillery: "none",  
        mortars: "adequate",  
        vehicles: "minimal",  
        communications: "poor"  
    },  
  
    // Combat ratings
```

```

offensiveRating: 75,      // 0-100
defensiveRating: 90,     // 0-100

// Status
readiness: 85,           // 0-100 (ability to execute orders)
morale: 80,               // 0-100 (willingness to fight)
experience: 65,          // 0-100 (combat experience)
cohesion: 80,             // 0-100 (unit discipline/coordination)
supply: 75,               // 0-100 (ammunition, food, fuel)
fatigue: 30,              // 0-100 (accumulated exhaustion)

// Assignment
assignedSector: "sarajevo_north", // Front sector
position: "defensive", // defensive, reserve, offensive
currentOrders: null,    // Current order being executed

// Specializations
specializations: [
  "urban_combat",
  "mountain_warfare",
  "defensive_operations"
], 

// History
battles: [
  {
    turn: 8,
    location: "sarajevo_vogosca",
    type: "defensive",
    result: "success",
    casualties: 45
  }
], 

// Map display
mapPosition: {x: 455, y: 260},
visible: true,        // Show on map?

// Source documentation
source: "Balkan Battlegrounds, p.156"
}

```

Brigade Formation System

Brigades form automatically based on historical timeline:

```
javascript

function checkBrigadeFormations(turn) {
    const currentDate = turnToDate(turn);

    // Check each historical brigade
    for (let brigadeData of historicalBrigades) {
        const formationTurn = dateToTurn(brigadeData.formed);

        if (turn === formationTurn) {
            // Create brigade
            const brigade = createBrigade(brigadeData);
            gameState.brigades[brigade.id] = brigade;

            // Not assigned to corps yet - player must do this
            brigade.corps = null;
            brigade.status = "unassigned";

            // Notify player
            addNotification({
                type: "new_brigade_formed",
                brigade: brigade.name,
                municipality: brigade.homeMunicipality,
                message: `${brigade.name} has formed in ${getMunicipalityName(brigade.homeMunicipality)}. Assign to corps?`,
            });

            console.log(`Turn ${turn}: ${brigade.name} formed`);
        }
    }
}
```

Sample Historical Brigades

ARBiH 1. Korpus (Sarajevo):

1. 1st Mountain Brigade "Viteška" (Elite, April 1992)
2. 2nd Mountain Brigade (Regular, May 1992)
3. 16th Motorized Brigade (Regular, June 1992)
4. 101st Mountain Brigade (Regular, September 1992)

5. 102nd Mountain Brigade (Regular, September 1992) ... (24 brigades total)

ARBiH 2. Korpus (Tuzla-Podrinje):

1. 201st Light Brigade (Regular, September 1992)
2. 280th East Bosnia Light Brigade "Srebrenica" (Veteran, May 1992)
3. 281st East Bosnia Light Brigade "Žepa" (Regular, May 1992) ... (40+ brigades total)

VRS Sarajevo-Romanija Corps:

1. 1st Romanija Infantry Brigade (Veteran, May 1992)
2. 3rd Sarajevo Infantry Brigade (Regular, May 1992)
3. 4th Sarajevo Infantry Brigade (Regular, May 1992) ... (10+ brigades total)

VRS Drina Corps:

1. Zvornik Brigade (Veteran, April 1992)
2. Bratunac Brigade (Veteran, May 1992)
3. Vlasenica Brigade (Regular, May 1992) ... (9+ brigades total)

Quality Ratings

javascript

```
const qualityEffects = {
    militia: {
        offensiveBase: 40,
        defensiveBase: 50,
        moraleBase: 60,
        successModifier: -20,
        experienceGain: 1.5 // Learn faster
    },
    regular: {
        offensiveBase: 60,
        defensiveBase: 70,
        moraleBase: 75,
        successModifier: 0,
        experienceGain: 1.0
    },
    veteran: {
        offensiveBase: 75,
        defensiveBase: 85,
        moraleBase: 85,
        successModifier: +10,
        experienceGain: 0.5 // Already experienced
    },
    elite: {
        offensiveBase: 90,
        defensiveBase: 95,
        moraleBase: 95,
        successModifier: +20,
        experienceGain: 0.3 // Minimal gain
    }
};
```

Brigade Types

javascript

```
const brigadeTypes = {  
    mountain_infantry: {  
        offensiveBonus: +5,  
        defensiveBonus: +10,  
        terrainBonus: { mountain: +20, urban: +5 }  
    },  
    motorized: {  
        offensiveBonus: +10,  
        defensiveBonus: 0,  
        terrainBonus: { plains: +10 },  
        terrainPenalty: { mountain: -10 }  
    },  
    light_infantry: {  
        offensiveBonus: 0,  
        defensiveBonus: +5,  
        terrainBonus: { forest: +10 }  
    },  
    mechanized: {  
        offensiveBonus: +15,  
        defensiveBonus: -5,  
        terrainBonus: { plains: +15 },  
        terrainPenalty: { mountain: -15, urban: -10 }  
    },  
    special_forces: {  
        offensiveBonus: +20,  
        defensiveBonus: -10,  
        specialOperations: true  
    }  
};
```

UI Organization

Brigades organized by corps (or unassigned):

▼ 1. KORPUS (SARAJEVO)

24 brigades | 52,000 troops | 18 available

► 1st Mountain Brigade "Viteška" [Elite]

2,800 / 3,500 | Ready: 85% | Defensive

► 16th Motorized Brigade [Regular]

2,400 / 2,800 | Ready: 75% | Defensive

► 101st Mountain Brigade [Regular]

1,600 / 1,800 | Ready: 70% | Reserve

 In operation (55% success chance)

▼ 2. KORPUS (TUZLA)

40 brigades | 65,000 troops | 32 available

► 5. KORPUS (BIHAĆ)

18 brigades | 42,000 troops | 15 available

▼ UNASSIGNED BRIGADES (3)

 These brigades have no corps!

- 17th Mountain Brigade (Sanski Most) - 2,800 troops

- 509th Brigade (Bosanska Krupa) - 1,400 troops

- 527th Brigade (Cazin) - 1,200 troops

Integration Points

- Form in home municipalities (tied to Settlement System)
- Auto-assign to corps covering that municipality (Command Structure)
- Assigned to front sectors (Front-Line Operations)
- Execute orders with uncertainty (Uncertain Command)
- Affected by pre-war decisions (Pre-War Phase)

SYSTEM 3: DYNAMIC COMMAND STRUCTURE

Overview

Player creates and organizes corps (ARBiH/VRS) or operational zones (HVO) to command brigades. Not historically scripted - player decides coverage areas, headquarters, and organization.

Key Concepts

Historical: Brigades form in their home municipalities **Player Choice:** How to organize them into command structures

Terminology:

- **ARBiH:** Corps (Korpus) - e.g., "1. Korpus", "2. Korpus"
- **VRS:** Corps (Korpus) - e.g., "Sarajevo-Romanija Korpus", "Drina Korpus"
- **HVO:** Operational Zones (Operativna Zona) - e.g., "OZ Središnja Bosna", "OZ Mostar"

Corps/OZ Object Structure

javascript

```

corps = {
    // Identity
    id: "player_corps_1",
    name: "1. Korpus",      // Player chooses name
    type: "corps",          // 'corps' for ARBiH/VRS, 'operational_zone' for HVO
    faction: "rbih",

    // Coverage area (player-defined)
    coverageArea: [
        "sarajevo_centar",
        "sarajevo_novi",
        "sarajevo_novo",
        "vogosca",
        "ilijas"
    ],
    // Headquarters (player chooses from coverage municipalities)
    headquarters: "sarajevo_centar",

    // Brigades (automatic from coverage)
    brigades: [
        "arbih_1_mountain",
        "arbih_16_motorized",
        "arbih_101_mountain"
        // ... all brigades in coverage municipalities
    ],
    // Statistics (calculated from brigades)
    totalStrength: 15400,
    averageReadiness: 78,
    averageMorale: 82,
    sectorsDefended: 12,

    // Support assets (player allocates)
    artillery: ["105mm_battery_1", "120mm_battery_2"],
    logistics: "adequate",    // poor; minimal, adequate, good, excellent
    communications: "good",

    // Resources
    monthlyBudget: 5000,
    equipmentPriority: "medium",

    // Creation
}

```

```

formed: 8,           // turn number

// Effects
coordinationBonus: 0,    // -10 to +10 based on size

// Display
color: "#10b981",      // Map display color (player can customize)
visible: true
}

```

Creation Process

Step 1: Initialize Creation

```

javascript

function startCorpsCreation(faction) {
    // Show creation dialog
    showCorpsCreationDialog({
        faction: faction,
        availableMunicipalities: getControlledMunicipalities(faction),
        existingCorps: getCorps(faction),
        unassignedBrigades: getUnassignedBrigades(faction)
    });
}

```

Step 2: Select Coverage Area

```

javascript

```

```

function selectCoverageArea(municipalities) {
    // Validate coverage
    const stats = {
        municipalities: municipalities.length,
        brigades: countBrigadesInArea(municipalities),
        population: getTotalPopulation(municipalities),
        frontSectors: countFrontSectors(municipalities)
    };

    // Recommendations
    const recommendations = [];
    if (stats.municipalities < 5) {
        recommendations.push("⚠️ Small area - limited brigade pool");
    } else if (stats.municipalities > 15) {
        recommendations.push("⚠️ Large area - coordination penalty");
    } else {
        recommendations.push("✅ Good size for coordination");
    }

    return { stats, recommendations };
}

```

Step 3: Create Corps

javascript

```

function createCorps(data) {
  const corps = {
    id: generateId(),
    name: data.name,
    type: data.faction === 'hrhb' ? 'operational_zone' : 'corps',
    faction: data.faction,
    coverageArea: data.municipalities,
    headquarters: data.headquarters,
    brigades: [],
    formed: gameState.turn
  };
}

// Auto-assign all brigades in coverage area
for (let munId of data.municipalities) {
  const brigades = getBrigadesInMunicipality(munId);
  for (let brigade of brigades) {
    if (brigade.faction === data.faction) {
      brigade.corps = corps.id;
      corps.brigades.push(brigade.id);
    }
  }
}

// Calculate coordination bonus/penalty
corps.coordinationBonus = calculateCoordinationBonus(corps);

// Add to game state
gameState.corps[corps.id] = corps;

console.log(`Created ${corps.name} with ${corps.brigades.length} brigades`);
return corps;
}

```

Automatic Brigade Assignment

javascript

```

function assignBrigadesToCorps() {
    // For each brigade
    for (let brigade of Object.values(gameState.brigades)) {
        // Find corps covering this brigade's municipality
        const corps = findCorpsCoveringMunicipality(
            brigade.homeMunicipality,
            brigade.faction
        );

        if (corps) {
            // Assign brigade to corps
            brigade.corps = corps.id;
            if (!corps.brigades.includes(brigade.id)) {
                corps.brigades.push(brigade.id);
            }
        } else {
            // No corps covers this municipality
            brigade.corps = null;
            brigade.status = "unassigned";

            // Warn player
            addWarning({
                type: "unassigned_brigade",
                brigade: brigade.name,
                municipality: brigade.homeMunicipality
            });
        }
    }
}

```

Corps Size Effects

javascript

```
function calculateCoordinationBonus(corps) {  
    const size = corps.coverageArea.length;  
  
    if (size <= 5) {  
        // Small corps - excellent coordination  
        return +10;  
    } else if (size <= 8) {  
        // Optimal size  
        return +5;  
    } else if (size <= 12) {  
        // Good size  
        return 0;  
    } else if (size <= 15) {  
        // Large - some coordination issues  
        return -5;  
    } else {  
        // Too large - significant coordination penalty  
        return -10;  
    }  
}
```

Reorganization Options

Modify Coverage:

javascript

```

function modifyCorpsCoverage(corpsId, addMunicipalities, removeMunicipalities) {
  const corps = gameState.corps[corpsId];

  // Add municipalities
  for (let munId of addMunicipalities) {
    corps.coverageArea.push(munId);

    // Reassign brigades
    const brigades = getBrigadesInMunicipality(munId);
    for (let brigade of brigades) {
      if (brigade.faction === corps.faction) {
        brigade.corps = corpsId;
        corps.brigades.push(brigade.id);
      }
    }
  }

  // Remove municipalities
  for (let munId of removeMunicipalities) {
    corps.coverageArea = corps.coverageArea.filter(id => id !== munId);

    // Unassign brigades
    const brigades = getBrigadesInMunicipality(munId);
    for (let brigade of brigades) {
      brigade.corps = null;
      brigade.status = "unassigned";
      corps.brigades = corps.brigades.filter(id => id !== brigade.id);
    }
  }

  // Recalculate coordination
  corps.coordinationBonus = calculateCoordinationBonus(corps);
}

```

Merge Corps:

javascript

```

function mergeCorps(corps1Id, corps2Id, newName) {
  const corps1 = gameState.corps[corps1Id];
  const corps2 = gameState.corps[corps2Id];

  // Create merged corps
  const merged = {
    id: generateId(),
    name: newName,
    type: corps1.type,
    faction: corps1.faction,
    coverageArea: [...corps1.coverageArea, ...corps2.coverageArea],
    headquarters: corps1.headquarters,
    brigades: [...corps1.brigades, ...corps2.brigades],
    formed: gameState.turn
  };

  // Reassign brigades
  for (let brigadeId of merged.brigades) {
    gameState.brigades[brigadeId].corps = merged.id;
  }

  // Remove old corps
  delete gameState.corps[corps1Id];
  delete gameState.corps[corps2Id];

  // Add new corps
  gameState.corps[merged.id] = merged;
}

```

Split Corps:

javascript

```

function splitCorps(corpsId, splitMunicipalities, newName) {
  const original = gameState.corps[corpsId];

  // Create new corps with split municipalities
  const newCorps = {
    id: generateId(),
    name: newName,
    type: original.type,
    faction: original.faction,
    coverageArea: splitMunicipalities,
    headquarters: splitMunicipalities[0],
    brigades: [],
    formed: gameState.turn
  };

  // Remove from original
  original.coverageArea = original.coverageArea.filter(
    munId => !splitMunicipalities.includes(munId)
  );

  // Reassign brigades
  for (let munId of splitMunicipalities) {
    const brigades = getBrigadesInMunicipality(munId);
    for (let brigade of brigades) {
      brigade.corps = newCorps.id;
      newCorps.brigades.push(brigade.id);
      original.brigades = original.brigades.filter(id => id !== brigade.id);
    }
  }

  gameState.corps[newCorps.id] = newCorps;
}

```

Map Visualization

javascript

```

function renderCorpsCoverage() {
    for (let corps of Object.values(gameState.corps)) {
        // Draw coverage area boundary
        const boundary = createBoundaryFromMunicipalities(corps.coverageArea);

        boundary.attr({
            stroke: corps.color,
            'stroke-width': 3,
            'stroke-dasharray': '10,5',
            fill: corps.color,
            opacity: 0.15
        });

        // Add corps label
        const center = calculateCoverageCenter(corps.coverageArea);
        const label = paper.text(center.x, center.y, corps.name);
        label.attr({
            'font-size': 18,
            'font-weight': 'bold',
            fill: corps.color,
            'text-shadow': '2px 2px 4px rgba(0,0,0,0.5)'
        });

        // Add HQ marker
        const hqMun = gameState.municipalities[corps.headquarters];
        const hqMarker = paper.circle(hqMun.centroid.x, hqMun.centroid.y, 10);
        hqMarker.attr({
            fill: corps.color,
            stroke: '#ffffff',
            'stroke-width': 3
        });

        // HQ label
        const hqLabel = paper.text(
            hqMun.centroid.x,
            hqMun.centroid.y - 20,
            "⭐ HQ"
        );
    }
}

```

UI Examples

Corps Overview:

YOUR COMMAND STRUCTURE

[+ Create New Corps/OZ]

1. KORPUS (SARAJEVO)

Coverage: 5 municipalities

Brigades: 24 (15,400 troops)

Readiness: 78%

Coordination: +10 (optimal size)

[Details] [Edit] [Issue Order]

2. KORPUS (TUZLA)

Coverage: 12 municipalities

Brigades: 40 (65,000 troops)

Readiness: 72%

Coordination: 0 (good size)

[Details] [Edit] [Issue Order]

⚠ UNASSIGNED BRIGADES (3)

- 17th Mountain (Sanski Most)
- 509th Brigade (Bosanska Krupa)
- 527th Brigade (Cazin)

[Assign to Corps]

Creation Dialog:

CREATE NEW CORPS

Name: [1. Korpus]

Faction: Army of RBiH

HQ: [Sarajevo Centar ▼]

SELECT COVERAGE AREA:

Sarajevo Centar

Sarajevo Novi Grad

Sarajevo Novo Sarajevo

Vogošća

Ilijaš

Hadžići

Trnovo

STATISTICS:

• Municipalities: 5

• Existing Brigades: 6 (4,800 tr.)

• Expected Formations: 18 more

• Front Sectors: 8

• Coordination: +10 (optimal)

 Good size for effective command

[Create Corps] [Cancel]

HVO Operational Zones

For HVO (Croatian Defense Council), use "Operational Zones" instead of "Corps":

javascript

```

// HVO used OZ structure instead of corps
const hvoOZNames = [
    "OZ Središnja Bosna",      // Central Bosnia
    "OZ Mostar",               // Mostar
    "OZ Tomislavgrad",         // Tomislavgrad
    "OZ Bosanska Posavina"    // Posavina
];

// Same mechanics as corps, just different terminology
function createOperationalZone(data) {
    // Identical to createCorps but with type: 'operational_zone'
    const oz = {
        ...data,
        type: 'operational_zone',
        // ... rest same as corps
    };
    return oz;
}

```

Strategic Considerations

Small Corps/OZ (5-8 municipalities):

- Better coordination (+10 to +5)
- Faster response time
- Clear command structure
- Limited brigade pool
- Less flexibility

Large Corps/OZ (15+ municipalities):

- Large brigade pool
- More flexibility
- Can concentrate forces
- Coordination penalty (-5 to -10)
- Slower response
- Complex command

Integration Points

- Brigades auto-assign based on home municipality

- Corps effects apply to all assigned brigades
 - Coverage determines which front sectors corps manages
 - Reorganization possible anytime (with readiness penalty)
-

SYSTEM 4: FRONT-LINE OPERATIONS

Overview

Brigades are assigned to geographic front sectors, not free movement. Can only move by changing sector assignment or being grouped into tactical groups for temporary operations.

Key Concepts

NOT: Chess-like free movement **YES:** Brigades tied to defensive sectors, pulled for operations

Front Sector: Geographic area of front line (typically 10-20km) **Tactical Group:** Temporary formation of multiple brigades for offensive operation

Front Sector Object

```
javascript
```

```

frontSector = {
    // Identity
    id: "sarajevo_north",
    name: "Northern Sarajevo Sector",

    // Geography
    settlements: [
        "vogosca_vogosca",
        "vogosca_kobilja_glava",
        "sarajevo_novi_ahatovici"
        // ... all settlements on this front segment
    ],
    length: 12,           // kilometers
    terrain: "urban",     // urban, mountain, river, plain, forest

    // Opposing sides
    faction1: "rbih",
    faction2: "rs",

    // Front characteristics
    intensity: "heavy",      // quiet, active, heavy, assault
    fortified: 40,           // 0-100 (increases over time)
    minefields: 20,           // 0-100
    trenches: 60,             // 0-100
    bunkers: 30,              // 0-100

    // Unit assignments
    assignedUnits: {
        rbih: ["arbih_1_mountain", "arbih_16_motorized", "arbih_101_mountain"],
        rs: ["vrs_1_romanija", "vrs_3_sarajevo"]
    },
    // Force density (troops per km)
    density: {
        rbih: 280,
        rs: 350
    },
    // Coverage status
    coverage: {
        rbih: "adequate",      // undermanned, minimal, adequate, strong, overwhelming
        rs: "strong"
    },
}
```

```
// Required strength (calculated from length, terrain, intensity)
requiredStrength: {
    rbih: 3000,
    rs: 3000
},

// Activity
lastAction: 15,           // turn number
recentBattles: ["turn_12_probing", "turn_15_assault"],

// Strategic importance
strategic: true,
logistics: "good",      // poor, adequate, good

// Headquarters
rbihHQ: "sarajevo_centar_centar",
rsHQ: "pale_pale"
}
```

Brigade Assignment to Sectors

javascript

```

function assignBrigadeToSector(brigadeId, sectorId) {
    const brigade = gameState.brigades[brigadeId];
    const sector = gameState.frontSectors[sectorId];

    // Remove from old sector
    if (brigade.assignedSector) {
        const oldSector = gameState.frontSectors[brigade.assignedSector];
        oldSector.assignedUnits[brigade.faction] =
            oldSector.assignedUnits[brigade.faction].filter(id => id !== brigadeId);

        checkSectorCoverage(oldSector);
    }

    // Assign to new sector
    brigade.assignedSector = sectorId;
    sector.assignedUnits[brigade.faction].push(brigadeId);

    // Update sector statistics
    updateSectorDensity(sector);
    checkSectorCoverage(sector);

    // Reassignment cost
    brigade.readiness -= 10; // Takes time to move and establish

    console.log(`${brigade.name} assigned to ${sector.name}`);
}

```

Coverage Calculation

javascript

```

function checkSectorCoverage(sector) {
    for (let faction of [sector.faction1, sector.faction2]) {
        // Get all brigades assigned to this sector
        const brigadeIds = sector.assignedUnits[faction];
        const brigades = brigadeIds.map(id => gameState.brigades[id]);

        // Calculate total strength
        const totalStrength = brigades.reduce((sum, b) => sum + b.currentStrength, 0);

        // Get required strength
        const required = sector.requiredStrength[faction];

        // Calculate ratio
        const ratio = totalStrength / required;

        // Determine coverage level
        if (ratio < 0.4) {
            sector.coverage[faction] = "undermanned";
            addWarning({
                type: "undermanned_sector",
                sector: sector.name,
                faction: faction,
                ratio: ratio
            });
        } else if (ratio < 0.7) {
            sector.coverage[faction] = "minimal";
        } else if (ratio < 1.2) {
            sector.coverage[faction] = "adequate";
        } else if (ratio < 2.0) {
            sector.coverage[faction] = "strong";
        } else {
            sector.coverage[faction] = "overwhelming";
        }

        // Update density
        sector.density[faction] = totalStrength / sector.length;
    }
}

function calculateRequiredStrength(sector) {
    // Base: 200 troops per km
    let base = sector.length * 200;
}

```

```

// Terrain modifier
const terrainMod = {
    urban: 1.3,
    mountain: 1.2,
    river: 1.1,
    plain: 1.0,
    forest: 1.1
};

base *= terrainMod[sector.terrain];

// Intensity modifier
const intensityMod = {
    quiet: 0.8,
    active: 1.0,
    heavy: 1.5,
    assault: 2.0
};

base *= intensityMod[sector.intensity];

// Strategic importance
if (sector.strategic) base *= 1.2;

// Fortifications reduce requirement
base *= (1 - sector.fortified / 300);

return Math.floor(base);
}

```

Tactical Group System

Temporary formations for offensive operations:

javascript

```
tacticalGroup = {  
    // Identity  
    id: "tg_liberation_kozarac",  
    name: "Task Force Kozarac",  
  
    // Command  
    corps: "5_korpus",      // Which corps organized this  
    faction: "rbih",  
  
    // Mission  
    mission: "offensive",    // offensive, defensive, raid  
    objective: [  
        "prijedor__kozarac",  
        "prijedor__trnopolje"  
    ],  
  
    // Composition (brigades temporarily pulled from sectors)  
    brigades: [  
        "arbih_17_mountain",  
        "arbih_5th_corps",  
        "arbih_502nd"  
    ],  
  
    // Supporting elements  
    artillery: ["105mm_battery_1"],  
    engineers: true,  
    logistics: "adequate",  
  
    // Status  
    totalStrength: 7500,  
    readiness: 75,  
    phase: "preparation",    // preparation, approach, assault, exploitation, consolidation  
  
    // Operation parameters  
    startTurn: 15,  
    expectedDuration: 4,      // turns  
    currentTurn: 1,  
  
    // Progress  
    progress: 25,            // 0-100%  
    objectiveStatus: {  
        "prijedor__kozarac": "in_progress",  
        "prijedor__trnopolje": "not_started"  
    }  
}
```

```
},  
  
    // Losses  
    casualties: 120,  
    equipmentLoss: 15,      // percentage  
  
    // Map  
    assemblyArea: "sanski_most__sanski_most",  
    currentPosition: "prijedor_west_sector",  
  
    // After operation  
    returnTo: "original_sectors",  
    disbandTurn: 19  
}
```

Creating Tactical Groups

javascript

```

function createTacticalGroup(params) {
  const { name, brigades, objective, corps } = params;

  // Validate
  for (let brigadeId of brigades) {
    const brigade = gameState.brigades[brigadeId];

    // Check availability
    if (brigade.currentOrders) {
      return { error: `${brigade.name} already has orders` };
    }

    // Check readiness
    if (brigade.readiness < 60) {
      return { error: `${brigade.name} not ready (${brigade.readiness}%)` };
    }

    // Check if can be pulled from sector
    const sector = gameState.frontSectors[brigade.assignedSector];
    if (!canPullFromSector(sector, brigadeId)) {
      return { error: `Cannot pull ${brigade.name} - would leave ${sector.name} undermanned` };
    }
  }

  // Create tactical group
  const tg = {
    id: generateId(),
    name,
    corps,
    faction: gameState.brigades[brigades[0]].faction,
    mission: "offensive",
    objective,
    brigades,
    totalStrength: brigades.reduce((sum, id) =>
      sum + gameState.brigades[id].currentStrength, 0),
    phase: "preparation",
    startTurn: gameState.turn,
    progress: 0
  };

  gameState.tacticalGroups[tg.id] = tg;

  // Assign brigades to TG

```

```

for (let brigadeId of brigades) {
    const brigade = gameState.brigades[brigadeId];
    brigade.currentOrders = tg.id;
    brigade.position = "offensive";

    // Temporarily remove from sector
    const sector = gameState.frontSectors[brigade.assignedSector];
    sector.assignedUnits[brigade.faction] =
        sector.assignedUnits[brigade.faction].filter(id => id !== brigadeId);

    // Update sector coverage
    checkSectorCoverage(sector);
}

console.log(`Created ${name} with ${brigades.length} brigades`);
return { success: true, id: tg.id };
}

function canPullFromSector(sector, brigadeId) {
    const brigade = gameState.brigades[brigadeId];
    const faction = brigade.faction;

    // Simulate removal
    const remainingIds = sector.assignedUnits[faction].filter(id => id !== brigadeId);
    const remainingBrigades = remainingIds.map(id => gameState.brigades[id]);
    const remainingStrength = remainingBrigades.reduce((sum, b) =>
        sum + b.currentStrength, 0);

    const required = sector.requiredStrength[faction];

    // Must maintain at least 70% coverage (minimal)
    return remainingStrength >= required * 0.7;
}

```

Front Thickening Over Time

Fronts develop and strengthen:

javascript

```

function developFrontSector(sector) {
    const age = gameState.turn - sector.established;

    // Fortifications increase gradually
    if (age > 5) {
        sector.fortified = Math.min(100, sector.fortified + 2);
        sector.trenches = Math.min(100, sector.trenches + 3);
        sector.bunkers = Math.min(100, sector.bunkers + 1);
    }

    // Heavy fighting increases fortifications faster
    if (sector.intensity === "heavy") {
        sector.fortified = Math.min(100, sector.fortified + 1);
        sector.minefields = Math.min(100, sector.minefields + 2);
    }

    // Long-held sectors become heavily fortified
    if (age > 20) {
        sector.bunkers = Math.min(100, sector.bunkers + 2);
        sector.minefields = Math.min(100, sector.minefields + 2);
    }

    // Units gain experience defending this sector
    for (let faction of [sector.faction1, sector.faction2]) {
        for (let brigadeId of sector.assignedUnits[faction]) {
            const brigade = gameState.brigades[brigadeId];
            brigade.experience = Math.min(100, brigade.experience + 0.5);
            brigade.defensiveRating = Math.min(100, brigade.defensiveRating + 0.3);
        }
    }
}

```

UI Examples

Sector Panel:

SARAJEVO NORTH SECTOR

Length: 12 km

Terrain: Urban

Intensity: Heavy

FORTIFICATIONS:

Trenches:  60%

Bunkers:  30%

Minefields:  20%

RBiH FORCES:

Status: ADEQUATE (110%)

Strength: 3,360 / 3,000 required

Density: 280 troops/km

Assigned Brigades:

• 1st Mountain Brigade (2,800)

• 101st Brigade (560)

[Request Reinforcements]

[Strengthen Fortifications]

Tactical Group Panel:



TASK FORCE KOZARAC

Mission: Offensive

Objective: Liberate Kozarac

Corps: 5. Korpus

COMPOSITION:

- 17th Mountain Brigade (2,800)

- 5th Corps Brigade (3,200)

- 502nd Brigade (1,500)

Total: 7,500 troops

PROGRESS:

Phase: Assault (Turn 2/4)



45%

Kozarac: In Progress 🔥

Trnopolje: Not Started

Casualties: 120 (1.6%)

[Cancel Operation]

[Adjust Objectives]

Coverage Warning:

SECTOR COVERAGE WARNING

Sanski Most Sector

Status: MINIMAL (37%)

Strength: 1,400 / 3,800 required

Reason: 17th Mountain Brigade pulled for Task Force Kozarac

Risk: Vulnerable to VRS attack

Recommendations:

- Send reinforcements immediately
- Cancel offensive and return unit
- Accept risk and continue

[Take Action] [Acknowledge]

Integration Points

- Sectors defined by settlement front lines
- Brigades assigned from corps coverage area
- Coverage requirements create strategic tension
- Tactical groups enable offensive operations
- Front development tracks passage of time

SYSTEM 5: UNCERTAIN COMMAND

Overview

Orders to brigades have 5-95% success chance. Not deterministic. Multiple factors affect outcome.
Transparency in calculations.

Key Concepts

NOT: Click button → guaranteed success **YES:** Issue order → calculate probability → random roll → outcome

Why: Represents fog of war, command friction, unpredictability of combat

Order Types

javascript

```
const orderTypes = {  
    hold_position: {  
        difficulty: 0,  
        description: "Maintain current position"  
    patrol: {  
        difficulty: -5,  
        description: "Active patrolling and reconnaissance"  
    limited_attack: {  
        difficulty: -10,  
        description: "Local offensive, limited objectives"  
    assault: {  
        difficulty: -20,  
        description: "Major assault on fortified position"  
    deep_attack: {  
        difficulty: -30,  
        description: "Penetration deep into enemy territory"  
    encirclement: {  
        difficulty: -40,  
        description: "Complex maneuver to surround enemy"  
}
```

Success Calculation

javascript

```
function calculateSuccessChance(brigade, order) {
    let chance = 60; // Base chance

    // 1. BRIGADE STATUS
    // Readiness (ability to execute)
    chance += (brigade.readiness - 70) * 0.5;

    // Morale (willingness to fight)
    chance += (brigade.morale - 70) * 0.5;

    // Experience
    chance += (brigade.experience - 50) * 0.3;

    // 2. QUALITY
    const qualityMod = {
        militia: -20,
        regular: 0,
        veteran: +10,
        elite: +20
    };
    chance += qualityMod[brigade.quality];

    // 3. ORDER DIFFICULTY
    chance += orderTypes[order.type].difficulty;

    // 4. SUPPLY
    if (brigade.supply < 50) chance -= 25;
    else if (brigade.supply < 70) chance -= 15;
    else if (brigade.supply < 90) chance -= 5;

    // 5. FATIGUE
    chance -= brigade.fatigue * 0.4;

    // 6. FORCE RATIO
    const enemyStrength = getEnemyStrength(order.objective);
    const ratio = brigade.currentStrength / enemyStrength;

    if (ratio < 0.5) chance -= 30;
    else if (ratio < 0.8) chance -= 20;
    else if (ratio < 1.0) chance -= 10;
    else if (ratio > 2.0) chance += 15;
    else if (ratio > 3.0) chance += 30;
```

```

// 7. TERRAIN
const terrain = getTerrainAt(order.objective);

// Check specializations
if(terrain === "mountain" && brigade.specializations.includes("mountain_warfare")) {
    chance += 10;
} else if(terrain === "urban" && brigade.specializations.includes("urban_combat")) {
    chance += 10;
} else if(terrain === "mountain") {
    chance -= 15;
} else if(terrain === "urban") {
    chance -= 10;
}

// 8. ENEMY FORTIFICATIONS
const sector = getSectorAt(order.objective);
if(sector.fortified > 70) chance -= 15;
else if(sector.fortified > 40) chance -= 10;

// 9. EQUIPMENT
if(brigade.equipment.smallArms === "poor") chance -= 10;
if(brigade.equipment.smallArms === "excellent") chance += 10;
if(order.type === "assault" && brigade.equipment.heavyWeapons === "minimal") {
    chance -= 15;
}

// 10. COMMUNICATIONS
if(brigade.equipment.communications === "poor") chance -= 10;

// 11. WEATHER
const weather = getCurrentWeather();
if(weather === "heavy_rain" || weather === "snow") chance -= 15;

// 12. COORDINATION (from corps)
const corps = gameState.corps[brigade.corps];
if(corps) {
    chance += corps.coordinationBonus;
}

// Cap at 5-95% (never impossible, never guaranteed)
return Math.max(5, Math.min(95, Math.round(chance)));
}

```

Order Execution

javascript

```
function executeOrder(brigade, order) {
    // Calculate success probability
    const successChance = calculateSuccessChance(brigade, order);

    // Roll
    const roll = Math.random() * 100;

    console.log(`#${brigade.name}: ${order.type} - Need ${successChance}%, rolled ${roll.toFixed(1)}%`);

    // Determine outcome
    if (roll <= successChance) {
        // FULL SUCCESS
        return {
            result: "success",
            casualties: calculateCasualties(brigade, order, "light"),
            duration: order.expectedDuration,
            objectiveStatus: "captured",
            moraleMod: +5,
            experienceMod: +3,
            message: `${brigade.name} successfully completed ${order.type}!`
        };
    } else if (roll <= successChance + 25) {
        // PARTIAL SUCCESS
        return {
            result: "partial",
            casualties: calculateCasualties(brigade, order, "moderate"),
            duration: order.expectedDuration + 1,
            objectiveStatus: "contested",
            moraleMod: 0,
            experienceMod: +2,
            issues: ["stronger_resistance", "supply_delays"],
            message: `${brigade.name} made progress but faced difficulties.`
        };
    } else {
        // FAILURE
        return {
            result: "failure",
            casualties: calculateCasualties(brigade, order, "heavy"),
            duration: 1,
            objectiveStatus: "failed",
            moraleMod: -15,
        };
    }
}
```

```
    readinessMod: -20,  
    experienceMod: +1, // Learn from failure  
    reason: determineFailureReason(roll - successChance),  
    message: `${brigade.name} failed ${order.type}. Unit must regroup.`  
};  
}  
}
```

Casualties Calculation

javascript

```

function calculateCasualties(brigade, order, severity) {
    let baseCasualties = 0;

    // Base by order type
    const orderCasualties = {
        hold_position: 0.01,    // 1%
        patrol: 0.02,          // 2%
        limited_attack: 0.05,   // 5%
        assault: 0.10,         // 10%
        deep_attack: 0.15,     // 15%
        encirclement: 0.12     // 12%
    };
    baseCasualties = brigade.currentStrength * orderCasualties[order.type];

    // Severity modifier
    const severityMod = {
        light: 0.5,
        moderate: 1.0,
        heavy: 2.0
    };
    baseCasualties *= severityMod[severity];

    // Urban combat increases casualties
    const terrain = getTerrainAt(order.objective);
    if(terrain === "urban") baseCasualties *= 1.5;

    // Enemy fortifications
    const sector = getSectorAt(order.objective);
    baseCasualties *= (1 + sector.fortified / 200);

    // Equipment quality affects casualties
    if(brigade.equipment.smallArms === "poor") baseCasualties *= 1.3;
    if(brigade.equipment.smallArms === "excellent") baseCasualties *= 0.7;

    // Random variation ( $\pm 20\%$ )
    const variation = 0.8 + (Math.random() * 0.4);
    baseCasualties *= variation;

    return Math.round(baseCasualties);
}

```

Applying Results

javascript

```
function applyOrderResult(brigade, result) {
    // Casualties
    brigade.currentStrength -= result.casualties;
    brigade.casualties += result.casualties;

    // Morale
    brigade.morale = Math.max(0, Math.min(100,
        brigade.morale + result.moraleMod));

    // Readiness
    if(result.readinessMod) {
        brigade.readiness = Math.max(0, Math.min(100,
            brigade.readiness + result.readinessMod));
    }

    // Experience
    brigade.experience = Math.min(100,
        brigade.experience + result.experienceMod);

    // Fatigue
    brigade.fatigue = Math.min(100, brigade.fatigue + 15);

    // Supply consumption
    brigade.supply = Math.max(0, brigade.supply - 10);

    // Record battle
    brigade.battles.push({
        turn: gameState.turn,
        order: order.type,
        result: result.result,
        casualties: result.casualties,
        objective: order.objective
    });
}

// Check if needs to reform
if(brigade.currentStrength < brigade.maxStrength * 0.3) {
    brigade.position = "reforming";
    brigade.readiness = 30;
    addEvent({
        type: "brigade_reforming",
        brigade: brigade.name,
        reason: "heavy_casualties"
    });
}
```

```
    }  
  
    // Show result to player  
    showOrderResult(brigade, result);  
}
```

UI Examples

Order Dialog:

ISSUE ORDER: 17th Mountain Brigade |

Current Status:

- Strength: 2,800 / 3,500
- Readiness: 85%
- Morale: 75%
- Supply: 70%

SELECT ORDER TYPE:

- Hold Position (Safe)
- Patrol Sector (-5%)
- Limited Attack (-10%)
- Assault Kozarac (-20%)
- Deep Attack (-30%)

ESTIMATED SUCCESS: 35%

Contributing Factors:

- Base chance: 60%
- Readiness (+85%): +7.5%
- Morale (+75%): +2.5%
- Quality (regular): 0%
- Assault difficulty: -20%
- Force ratio (0.7:1): -10%
- Urban terrain: -10%
- Enemy fortified: -15%

 Low success chance

Estimated casualties: 280-420

Recommendation: Wait for
reinforcements or soften defenses
with artillery first

[Issue Order] [Cancel]

Result Notification:

	OPERATION FAILED
17th Mountain Brigade	
Assault on Kozarac	
Rolled: 52% (needed ≤35%)	
Result: Repulsed by heavy resistance. VRS counterattacked.	
Casualties: 310 troops	
Morale: 75% → 60% (-15%)	
Readiness: 85% → 65% (-20%)	
Experience: 30% → 31% (+1%)	
Status: Brigade must reform	
Pulled to reserve for 2 turns	
[Acknowledge]	

Integration Points

- Brigades execute orders during turn resolution
- Uncertain outcomes create tension and replayability
- Failure has consequences (morale, readiness, casualties)
- Success builds experience and confidence
- Player sees probability before committing

SYSTEM 6: PRE-WAR PHASE

Overview

Before war begins (March-April 1992), player makes organizational decisions that affect starting positions at settlement level.

Key Concepts

Historical Context:

- March 1992: Referendum on independence
- April 1992: War begins
- 1-2 month window for preparations

Player Decisions:

- How to organize defenses
- Where to deploy limited forces
- Whether to evacuate vulnerable areas
- How to allocate resources

Pre-War Decision Types

javascript

```

const preWarDecisions = {
    // Territorial Defense mobilization
    toMobilization: {
        none: { cost: 0, effect: 0 },
        partial: { cost: 10, effect: +8 }, // Political Capital
        full: { cost: 25, effect: +15 }
    },
    // Patriotska Liga deployment
    patriotskaLiga: {
        absent: { cost: 0, effect: 0 },
        present: { cost: 10, effect: +5 },
        deployed: { cost: 20, effect: +10 }
    },
    // Police control
    policeControl: {
        lost: { cost: 0, effect: -10 },
        neutral: { cost: 5, effect: 0 },
        secured: { cost: 15, effect: +5 }
    },
    // Fortifications
    fortifications: {
        none: { cost: 0, effect: 0 },
        key_points: { cost: 10, defensibility: +20 },
        town: { cost: 20, defensibility: +40 },
        all: { cost: 30, defensibility: +60 }
    },
    // Evacuation
    evacuation: {
        none: { cost: 0 },
        planned: { cost: 5, readiness: +10 },
        execute: { cost: 10, saves_civilians: true }
    }
}

```

Municipality Pre-War State

javascript

```

municipalityPreWar = {
    municipalityId: "zvornik",

    // Player decisions
    decisions: {
        toMobilization: "partial",
        patriotskaLiga: "present",
        policeControl: "neutral",
        fortifications: "key_points",
        evacuation: "none"
    },
}

// Costs
politicalCapitalSpent: 35,
troopsCommitted: 1500,
equipmentUsed: "limited",

// External factors (AI/historical)
jnaPresence: "garrison", // absent, nearby, garrison
sdsOrganization: "armed", // absent, organized, armed
vrsPreparations: "extensive",

// Projected outcome
projectedControl: {
    rbih: 40,
    rs: 50,
    contested: 10
}
}

```

Settlement Control Modifiers

javascript

```

function getPreWarModifiers(settlement) {
    const munPreWar = gameState.preWarDecisions[settlement.municipalityId];
    if (!munPreWar) return { totalBonus: 0 };

    let bonus = 0;
    let forceController = null;
    let forcedScore = null;

    const eth = settlement.ethnicComposition;
    const decisions = munPreWar.decisions;

    // TO Mobilization
    if (decisions.toMobilization === "full") {
        if (eth.bosniak > 40) bonus += 15;
    } else if (decisions.toMobilization === "partial") {
        if (eth.bosniak > 50) bonus += 8;
    }

    // Patriotska Liga
    if (decisions.patriotskaLiga === "deployed") {
        if (settlement.urban || settlement.isCapital) {
            bonus += 10; // Defend urban centers
        }
    } else if (decisions.patriotskaLiga === "present") {
        if (settlement.urban) bonus += 5;
    }

    // Police control
    if (decisions.policeControl === "secured") {
        bonus += 5;
    } else if (decisions.policeControl === "lost") {
        bonus -= 10;
    }

    // Fortifications
    if (decisions.fortifications !== "none") {
        if (settlement.isCapital || settlement.urban) {
            settlement.defensibility += 20;
        }
    }

    // JNA presence (negative for RBiH)
    if (munPreWar.jnaPresence === "garrison") {

```

```

if(settlement.urban || settlement.isCapital) {
    // JNA secures urban areas for VRS
    forceController = 'rs';
    forcedScore = 80;
} else if(eth.serb > 30) {
    bonus -= 20; // Pressure on RBiH areas
}
}

// SDS organization
if(munPreWar.sdsOrganization === "armed") {
    if(eth.serb > 30) {
        bonus += 15; // Benefits VRS
    }
}

// Evacuation
if(decisions.evacuation === "execute") {
    if(settlement.vulnerable) {
        forceController = 'none'; // Evacuated
        forcedScore = 0;
        // Civilians saved (tracked separately)
    }
}

return { totalBonus: bonus, forceController, forcedScore };
}

```

Historical Scenarios

Scenario: Zvornik (April 1992)

javascript

```
const zvornikScenario = {
    municipality: "zvornik",

    historical: {
        // What actually happened
        toMobilization: "partial",
        patriotskaLiga: "present",
        policeControl: "lost",
        jnaPresence: "garrison",
        sdsOrganization: "armed",
        vrsPreparations: "extensive",

        result: {
            fell: "1992-04-10", // 3 days
            casualties: 2000,
            control: { rs: 100 }
        }
    },
    playerOptions: {
        A: {
            name: "Historical (Weak Defense)",
            decisions: {
                toMobilization: "partial",
                patriotskaLiga: "present",
                policeControl: "neutral",
                fortifications: "none"
            },
            outcome: "Falls in 3 days, 2,000 casualties",
            likelihood: "Historical outcome"
        },
        B: {
            name: "Strong Defense",
            decisions: {
                toMobilization: "full",
                patriotskaLiga: "deployed",
                policeControl: "secured",
                fortifications: "town"
            },
            cost: 60, // Political Capital
            outcome: "Hold 2 weeks, 800 casualties, eventually falls",
            likelihood: "Possible but difficult"
        }
    }
};
```

```
    },  
  
    C: {  
        name: "Strategic Withdrawal",  
        decisions: {  
            toMobilization: "none",  
            patriotskaLiga: "absent",  
            evacuation: "execute",  
            fortifications: "none"  
        },  
        cost: 10,  
        outcome: "Town lost immediately, 200 casualties, civilians saved",  
        likelihood: "Easier, better long-term"  
    },  
}  
}
```

UI Examples

Pre-War Decision Panel:

PRE-WAR PREPARATIONS

ZVORNIK MUNICIPALITY

Current Date: March 25, 1992

War Expected: April 1992

SITUATION:

- Demographics: 60% Bosniak, 38% Serb
- SDA won 1990 election
- JNA garrison present (3,000)
- SDS armed Serb population
- Vulnerable position

YOUR DECISIONS:

TO Mobilization:

- None (Free)
- Partial (10 PC, +8 control)
- Full (25 PC, +15 control)

Patriotska Liga:

- Absent (Free)
- Deployed (20 PC, +10 urban)

Police:

- Secure Loyalty (15 PC, +5)

Fortifications:

- Town Center (20 PC, +40 def)

PROJECTED OUTCOME:

Starting control: 70% RBiH

Contested: 30%

Vulnerable: 6 settlements

Estimated: Hold 2-3 weeks

with reinforcement

Total Cost: 80 Political Capital

(60 available)

 Insufficient resources!

| Adjust decisions or accept risk |
| |
| [Finalize] [Reset] [Cancel] |

Outcome Preview:

| PROJECTED WAR START (April 1, 1992) |

| ZVORNIK MUNICIPALITY |

| Settlement Control: |

| RBiH: 70% (11 settlements) |

| • Zvornik town (fortified) |

| • Kula Grad (secure) |

| • Drinjača (secure) |

| • 8 rural villages |

| Contested: 30% (5 settlements) |

| • Mixed areas |

| • Vulnerable to attack |

| RS: 0% (Serbian areas will flip) |

| Expected: Heavy VRS attack |

| Turn 1-2. JNA artillery support. |

| Can player hold? Uncertain. |

| Historical: Fell in 3 days |

| With prep: Maybe 2 weeks |

| Recommendation: Consider evacuation |

| to save civilian lives |

Integration Points

- Modifies initial settlement control
- Affects brigade readiness at start
- Determines civilian casualties
- Creates alternate history possibilities

- Player sees trade-offs before war
-

SYSTEM 7: POPULATION TRACKING

Overview

Track population changes throughout war: casualties, refugees, ethnic composition shifts. Humanizes conflict and creates accountability.

Key Concepts

What's Tracked:

- Civilian casualties (by ethnicity)
- Military casualties (by brigade)
- Refugees (internal displacement)
- Ethnic composition changes
- Population events (massacres, evacuations)

Why:

- Shows human cost of decisions
- Prevents gamification of atrocities
- Creates historical record
- Affects victory conditions

Enhanced Settlement Population

javascript

```

settlement = {
    // ... other properties ...

    // Population tracking
    population: {
        initial: 1314,      // 1991 census (never changes)
        current: 1314,      // Current population

        // By ethnicity
        byEthnicity: {
            bosniak: { initial: 366, current: 366 },
            serb: { initial: 845, current: 845 },
            croat: { initial: 14, current: 14 },
            yugoslav: { initial: 48, current: 48 },
            other: { initial: 41, current: 41 }
        },
    },

    // Changes
    casualties: {
        total: 0,
        bosniak: 0,
        serb: 0,
        croat: 0,
        military: 0,
        civilian: 0
    },
}

refugees: {
    fled: 0,           // Left this settlement
    arrived: 0,        // Came to this settlement
    destinations: {}, // Where they went
    origins: {}        // Where they came from
},
}

displaced: {
    internally: 0,     // Still in BiH
    externally: 0      // Left country
},
}

// Events log
events: [
{
    turn: 15,
}
]

```

```
        date: "1992-06-10",
        type: "combat_casualties",
        count: 45,
        ethnicity: "bosniak",
        cause: "shelling"
    },
    {
        turn: 18,
        date: "1992-07-05",
        type: "refugee_flight",
        count: 320,
        ethnicity: "bosniak",
        destination: "tuzla_tuzla",
        cause: "ethnic_cleansing"
    }
]
}
```

Recording Events

javascript

```

function recordCasualties(settlementId, casualties, ethnicity, type, cause) {
    const settlement = gameState.settlements[settlementId];
    const pop = settlement.population;

    // Update totals
    pop.casualties.total += casualties;
    pop.casualties[ethnicity] += casualties;
    if (type === "civilian") {
        pop.casualties.civilian += casualties;
    } else {
        pop.casualties.military += casualties;
    }

    // Update current population
    pop.current -= casualties;
    pop.byEthnicity[ethnicity].current -= casualties;

    // Log event
    pop.events.push({
        turn: gameState.turn,
        date: turnToDate(gameState.turn),
        type: type === "civilian" ? "civilian_casualties" : "military_casualties",
        count: casualties,
        ethnicity: ethnicity,
        cause: cause
    });
}

// Update ethnic percentages
updateEthnicComposition(settlement);

// Check for atrocity threshold
if (type === "civilian" && casualties > 50) {
    flagPotentialAtrocity(settlement, casualties, cause);
}
}

function recordRefugeeFlight(settlementId, count, ethnicity, destination, cause) {
    const settlement = gameState.settlements[settlementId];
    const pop = settlement.population;

    // Update totals
    pop.refugees.fled += count;
    pop.refugees.destinations[destination] =

```

```
(pop.refugees.destinations[destination] || 0) + count;

// Update current population
pop.current -= count;
pop.byEthnicity[ethnicity].current -= count;

// Determine if internal or external
if (isInBiH(destination)) {
    pop.displaced.internally += count;

    // Add to destination
    const destSettlement = gameState.settlements[destination];
    destSettlement.population.refugees.arrived += count;
    destSettlement.population.refugees.origins[settlementId] = count;
    destSettlement.population.current += count;
    destSettlement.population.byEthnicity[ethnicity].current += count;
} else {
    pop.displaced.externally += count;
}

// Log event
pop.events.push({
    turn: gameState.turn,
    date: turnToDate(gameState.turn),
    type: "refugee_flight",
    count: count,
    ethnicity: ethnicity,
    destination: destination,
    cause: cause
});

// Update ethnic composition
updateEthnicComposition(settlement);
}

function updateEthnicComposition(settlement) {
    const pop = settlement.population;
    const total = pop.current;

    if (total === 0) {
        // Settlement depopulated
        settlement.ethnicComposition = {
            bosniak: 0,
            serb: 0,
            croat: 0
        };
    } else {
        const percent = (pop.current / total) * 100;
        const roundedPercent = Math.round(percent);
        settlement.ethnicComposition = {
            bosniak: Math.floor((pop.bosniak / total) * 100),
            serb: Math.floor((pop.serbian / total) * 100),
            croat: Math.floor((pop.croatian / total) * 100)
        };
    }
}
```

```
    croat: 0,  
    yugoslav: 0,  
    other: 0  
};  
return;  
}  
  
// Recalculate percentages  
for (let [ethnicity, data] of Object.entries(pop.byEthnicity)) {  
    settlement.ethnicComposition[ethnicity] =  
        (data.current / total) * 100;  
}  
}
```

Combat Casualties

javascript

```
function applyCombatCasualties(settlement, combatResult) {
    const attackerCasualties = combatResult.attackerCasualties;
    const defenderCasualties = combatResult.defenderCasualties;
    const civilianCasualties = combatResult.civilianCasualties;

    // Military casualties (tied to brigades)
    for (let brigade of combatResult.attackingBrigades) {
        recordCasualties(
            settlement.id,
            Math.round(attackerCasualties / combatResult.attackingBrigades.length),
            brigade.recruitmentEthnicity,
            "military",
            "combat"
        );
    }

    for (let brigade of combatResult.defendingBrigades) {
        recordCasualties(
            settlement.id,
            Math.round(defenderCasualties / combatResult.defendingBrigades.length),
            brigade.recruitmentEthnicity,
            "military",
            "combat"
        );
    }

    // Civilian casualties (random ethnicity based on population)
    if (civilianCasualties > 0) {
        const ethnicBreakdown = calculateCivilianCasualtiesByEthnicity(
            settlement,
            civilianCasualties
        );

        for (let [ethnicity, count] of Object.entries(ethnicBreakdown)) {
            if (count > 0) {
                recordCasualties(
                    settlement.id,
                    count,
                    ethnicity,
                    "civilian",
                    settlement.urban ? "urban_combat" : "shelling"
                );
            }
        }
    }
}
```

```
        }
    }
}

function calculateCivilianCasualtiesByEthnicity(settlement, totalCasualties) {
    const breakdown = {};
    const composition = settlement.ethnicComposition;

    for (let [ethnicity, percentage] of Object.entries(composition)) {
        breakdown[ethnicity] = Math.round(totalCasualties * (percentage / 100));
    }

    return breakdown;
}
```

Atrocity Detection

javascript

```

function flagPotentialAtrocity(settlement, casualties, cause) {
    // Criteria for potential atrocity:
    // - High civilian casualties in short time
    // - Systematic targeting of specific ethnicity
    // - Settlement control just changed

    const recentEvents = settlement.population.events.filter(e =>
        e.turn >= gameState.turn - 2 && e.type === "civilian_casualties"
    );

    const recentCivilianCasualties = recentEvents.reduce((sum, e) => sum + e.count, 0);

    if (recentCivilianCasualties > 100) {
        addEvent({
            type: "potential_atrocity",
            settlement: settlement.name,
            casualties: recentCivilianCasualties,
            controller: settlement.controller,
            turn: gameState.turn,
            requiresInvestigation: true
        });
    }

    // International reaction
    affectDiplomaticRelations(settlement.controller, -20);

    // Historical record
    gameState.historicalRecord.potentialAtrocities.push({
        settlement: settlement.id,
        turn: gameState.turn,
        controller: settlement.controller,
        casualties: recentCivilianCasualties,
        ethnicities: recentEvents.map(e => e.ethnicity)
    });
}
}

```

Statistics & Reporting

javascript

```

function getPopulationStatistics() {
  let stats = {
    total: {
      initial: 0,
      current: 0,
      casualties: { total: 0, civilian: 0, military: 0 },
      refugees: { fled: 0, arrived: 0 }
    },
    byEthnicity: {
      bosniak: { initial: 0, current: 0, casualties: 0, refugees: 0 },
      serb: { initial: 0, current: 0, casualties: 0, refugees: 0 },
      croat: { initial: 0, current: 0, casualties: 0, refugees: 0 }
    },
    byController: {
      rbih: { casualties: 0, refugees: 0 },
      rs: { casualties: 0, refugees: 0 },
      hrhb: { casualties: 0, refugees: 0 }
    }
  };
}

// Aggregate from all settlements
for (let settlement of Object.values(gameState.settlements)) {
  const pop = settlement.population;

  // Totals
  stats.total.initial += pop.initial;
  stats.total.current += pop.current;
  stats.total.casualties.total += pop.casualties.total;
  stats.total.casualties.civilian += pop.casualties.civilian;
  stats.total.casualties.military += pop.casualties.military;
  stats.total.refugees.fled += pop.refugees.fled;
  stats.total.refugees.arrived += pop.refugees.arrived;

  // By ethnicity
  for (let ethnicity of ['bosniak', 'serb', 'croat']) {
    stats.byEthnicity[ethnicity].initial += pop.byEthnicity[ethnicity].initial;
    stats.byEthnicity[ethnicity].current += pop.byEthnicity[ethnicity].current;
    stats.byEthnicity[ethnicity].casualties += pop.casualties[ethnicity];
  }

  // By controller
  const controller = settlement.controller;
  if (controller !== 'contested' && controller !== 'none') {

```

```

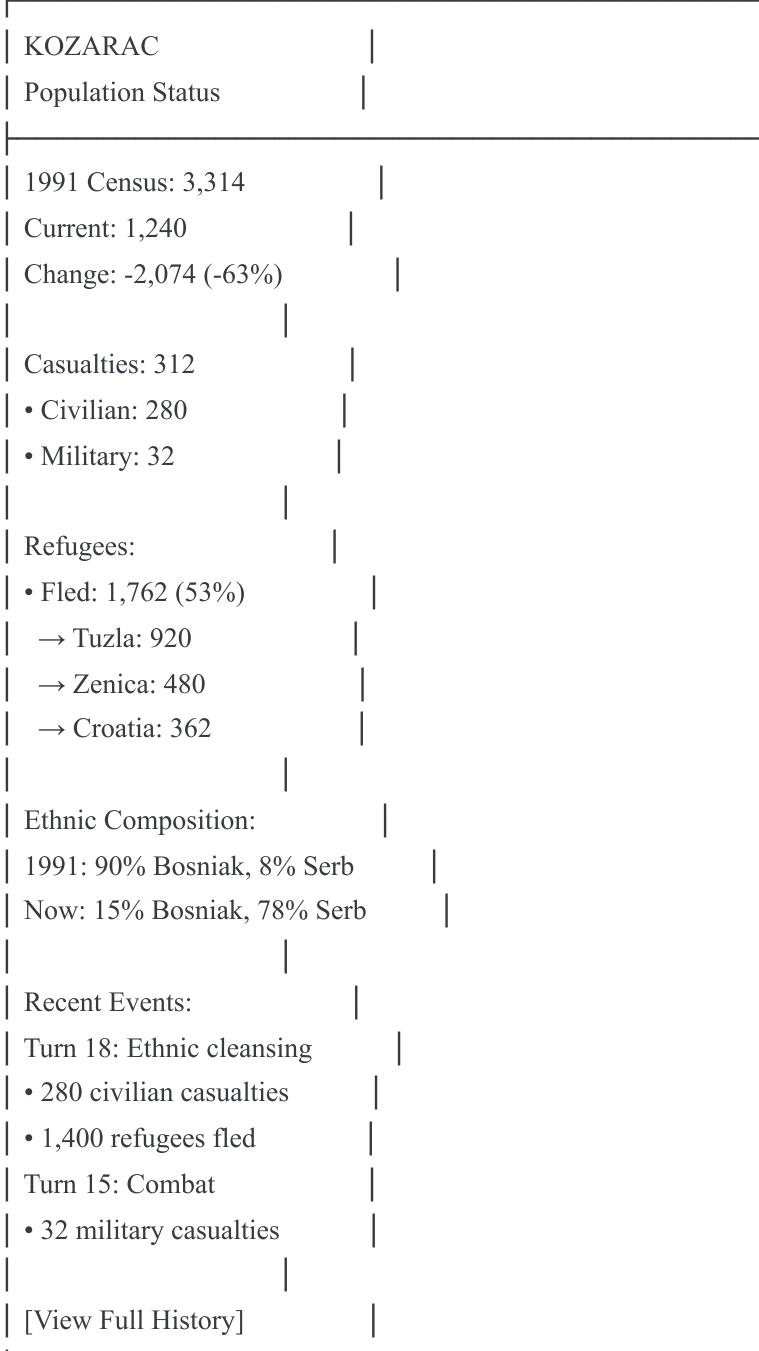
        stats.byController[controller].casualties += pop.casualties.total;
        stats.byController[controller].refugees += pop.refugees.fled;
    }
}

return stats;
}

```

UI Examples

Settlement Population Panel:



War Statistics Panel:

WAR STATISTICS	
Turn 45 (April 1993)	
POPULATION:	
1991: 4,377,033	
Current: 3,120,500	
Change: -1,256,533 (-29%)	
CASUALTIES:	
Total: 42,800	
• Civilian: 28,400 (66%)	
• Military: 14,400 (34%)	
By Ethnicity:	
• Bosniaks: 24,200 (57%)	
• Serbs: 14,800 (35%)	
• Croats: 3,800 (9%)	
REFUGEES:	
Total Displaced: 1,214,000	
• Internal: 820,000	
• External: 394,000	
→ Croatia: 180,000	
→ Serbia: 120,000	
→ Other: 94,000	
ETHNIC CLEANSING EVENTS:	
Documented: 8	
Under Investigation: 3	
[Detailed Report] [Export]	

Victory Condition Integration

javascript

```

function checkVictoryConditions() {
    const stats = getPopulationStatistics();

    // Civilian casualties affect score
    const civilianCasualtiesRatio =
        stats.total.casualties.civilian / stats.total.initial;

    if (civilianCasualtiesRatio > 0.05) {
        victoryScore -= 50; // 5% civilian casualties = major penalty
    }

    // Ethnic cleansing documented
    const atrocities = gameState.historicalRecord.potentialAtrocities.length;
    if (atrocities > 0) {
        victoryScore -= atrocities * 25;
    }

    // Refugee crisis
    const refugeeRatio = stats.total.refugees.fled / stats.total.initial;
    if (refugeeRatio > 0.20) {
        victoryScore -= 30; // 20% displaced = penalty
    }
}

```

Integration Points

- Combat casualties recorded automatically
- Ethnic cleansing events flagged
- Affects international diplomacy
- Victory conditions consider civilian impact
- Historical record preserved

PLANNED SYSTEMS

System 8: Diplomacy & International Relations

Priority: High **Complexity:** High **Est. Dev Time:** 12-15 hours

Overview: Track international involvement, arms embargoes, peacekeeping, recognition, sanctions.

Key Features:

- UN involvement and resolutions
- Arms embargo tracking
- International recognition of states
- Sanctions and their effects
- Foreign military involvement
- Diplomatic efforts
- Peace negotiations
- Safe zones/protected areas

Data Structure:

```

javascript

diplomacy = {
    // International actors
    actors: {
        UN: { stance: "neutral", resolutions: [...], peacekeepers: 0 },
        NATO: { stance: "cautious", interventions: [...] },
        EU: { recognition: {...}, sanctions: [...] },
        USA: { policy: "lift_and_strike", assistance: {...} },
        Russia: { stance: "pro_serb", support: {...} }
    },
    // Arms embargo
    armsEmbargo: {
        active: true,
        effectiveness: 70, // %
        violations: [...]
    },
    // Safe zones
    safeZones: [
        { settlement: "srebrenica", established: 15, protected: true },
        { settlement: "zepa", established: 16, protected: true },
        { settlement: "gorazde", established: 15, protected: true }
    ]
}

```

System 9: Supply & Logistics

Priority: High **Complexity:** Medium **Est. Dev Time:** 6-8 hours

Overview: Supply lines, ammunition, food, fuel. Affects brigade readiness.

Key Features:

- Supply routes and their security
- Ammunition consumption
- Food and fuel requirements
- Supply convoys
- Blockades and smuggling
- Humanitarian aid
- Black market

Data Structure:

```
javascript

supply = {
    // Supply lines
    routes: [
        {
            from: "split_croatia",
            to: "mostar",
            capacity: 1000, // tons/turn
            security: "good",
            interdiction: 0
        }
    ],
    // Brigade supply
    brigade: {
        ammunition: 75, // days of combat
        food: 90, // days
        fuel: 30, // days (if mechanized)
        medical: 80 // supplies
    }
}
```

System 10: Intelligence & Reconnaissance

Priority: Medium **Complexity:** Medium **Est. Dev Time:** 4-6 hours

Overview: Intelligence gathering, reconnaissance, espionage. Affects order success.

Key Features:

- Reconnaissance missions
- Intelligence reports (accuracy varies)
- Spy networks
- Signals intelligence
- Human intelligence
- Enemy strength estimates
- Misinformation

Data Structure:

```
javascript

intelligence = {
    enemyUnits: {
        "vrs_1_romanija": {
            knownLocation: "sarajevo_east",
            estimatedStrength: "2500-3500", // range
            confidence: 70, // %
            lastUpdate: 42 // turn
        }
    },
    reconMissions: {
        active: [...],
        completed: [...]
    }
}
```

System 11: Air Operations

Priority: Medium **Complexity:** Low **Est. Dev Time:** 3-4 hours

Overview: Limited air power. Mostly VRS aircraft early, NATO later.

Key Features:

- VRS air force (limited)
- NATO air strikes (1994-95)
- No-fly zone
- Close air support
- Strategic bombing
- Air defense

Data Structure:

```
javascript

airOperations = {
  noFlyZone: {
    active: false,
    startTurn: 30,
    violations: [...]
  },
  natoAirStrikes: {
    available: false,
    startTurn: 80,
    targets: ["rs_artillery", "rs_command"]
  }
}
```

System 12: Artillery System

Priority: Medium **Complexity:** Medium **Est. Dev Time:** 4-6 hours

Overview: Artillery units separate from brigades. Major impact on combat.

Key Features:

- Artillery batteries
- Range and fire missions
- Ammunition consumption
- Counter-battery fire
- Siege artillery
- Rocket artillery (MLRS)

Data Structure:

```
javascript

artillery = {
  batteries: {
    "105mm_battery_1": {
      type: "105mm_howitzer",
      range: 15, // km
      ammunition: 200, // shells
      assignedTo: "1_korpus",
      location: "sarajevo_centar"
    }
  }
}
```

System 13: Leadership & Officers

Priority: Low **Complexity:** Medium **Est. Dev Time:** 4-6 hours

Overview: Individual commanders with skills, traits, casualties.

Key Features:

- Named historical commanders
- Skills (offensive, defensive, logistics)
- Personalities (aggressive, cautious)
- Command radius
- Can be killed/wounded
- Replacement system

Data Structure:

```
javascript
```

```
commander = {
    name: "Vahid Karavelić",
    rank: "general",
    skill: 85,
    specialty: "defensive_operations",
    personality: "methodical",
    assigned: "1_korpus",
    alive: true
}
```

System 14: Economic System

Priority: Low **Complexity:** Medium **Est. Dev Time:** 6-8 hours

Overview: Limited economy. Resource scarcity. International aid.

Key Features:

- GDP/economy tracking
- Tax revenue
- International loans
- Inflation
- Black market
- Industrial capacity
- Resource extraction

Data Structure:

```
javascript
economy = {
    gdp: 1200, // million USD
    revenue: 50, // per turn
    expenses: 80,
    deficit: -30,
    internationalAid: 20,
    inflation: 150 // %
}
```

System 15: Victory Conditions

Priority: High **Complexity:** Low **Est. Dev Time:** 2-3 hours

Overview: Define how game ends and who wins.

Key Features:

- Territory control thresholds
- Population control
- International recognition
- Dayton-style peace agreement
- Continuation to 1995
- Multiple victory types

Data Structure:

```
javascript

victoryConditions = {

    rbih: {
        territorial: "Control 51% of BiH",
        population: "Control 60% of population",
        international: "Maintain international recognition",
        survival: "Survive until 1995"
    },

    rs: {
        territorial: "Control 49% of BiH",
        recognition: "Achieve international recognition",
        consolidation: "Establish contiguous territory"
    },

    hrhb: {
        territorial: "Control Herzegovina",
        autonomy: "Achieve autonomous status",
        federation: "Form federation with RBiH"
    }
}
```

TECHNICAL ARCHITECTURE

Platform

- **Framework:** Electron (cross-platform desktop app)
- **Rendering:** Raphael.js (SVG graphics)
- **Language:** JavaScript
- **Data:** JSON (save files) + CSV (census data)

File Structure

```
awww-electron/
├── src/
│   ├── index.html      # Main game interface
│   ├── main.js         # Electron main process
│   └── renderer.js     # Electron renderer
|
|   |
|   └── systems/
|       ├── settlement-system.js  # Settlement control
|       ├── brigade-system.js    # Brigade management
|       ├── corps-system.js      # Corps/OZ creation
|       ├── front-system.js     # Front line operations
|       ├── order-system.js     # Uncertain command
|       ├── prewar-system.js    # Pre-war phase
|       └── population-system.js # Population tracking
|
|   └── data/
|       ├── mz_1991_census.csv  # Settlement census
|       ├── brigades_arbih.json # ARBiH brigades
|       ├── brigades_vrs.json  # VRS brigades
|       ├── brigades_hvo.json  # HVO brigades
|       └── municipalities.json # Municipality data
|
|   └── maps/
|       ├── bosnia_base.svg    # Base map
|       └── settlements/       # Settlement SVG paths (109 files)
|
|   └── styles/
|       └── main.css          # Styling
|
|   └── assets/
|       └── icons/            # Unit icons
```

```
|   └── sounds/      # Sound effects (optional)  
|  
|   ├── saves/       # Save game files  
|   ├── package.json  
|   └── README.md
```

Performance Targets

- **Load Time:** <2 seconds (2,200 settlements)
- **Render Time:** <100ms (527 visible settlements)
- **Memory Usage:** <50MB
- **Frame Rate:** 60 FPS
- **Turn Resolution:** <500ms

Save/Load System

```
javascript
```

```
function saveGame() {
  const saveData = {
    version: "1.0",
    timestamp: Date.now(),
    turn: gameState.turn,
    date: gameState.date,

    settlements: gameState.settlements,
    brigades: gameState.brigades,
    corps: gameState.corps,
    frontSectors: gameState.frontSectors,
    tacticalGroups: gameState.tacticalGroups,

    historicalRecord: gameState.historicalRecord,
    playerDecisions: gameState.playerDecisions
  };

  const json = JSON.stringify(saveData, null, 2);
  fs.writeFileSync(`saves/game_${Date.now()}.json`, json);
}

function loadGame(filepath) {
  const json = fs.readFileSync(filepath, 'utf8');
  const saveData = JSON.parse(json);

  // Restore game state
  gameState.turn = saveData.turn;
  gameState.date = saveData.date;
  gameState.settlements = saveData.settlements;
  gameState.brigades = saveData.brigades;
  gameState.corps = saveData.corps;
  gameState.frontSectors = saveData.frontSectors;
  gameState.tacticalGroups = saveData.tacticalGroups;
  gameState.historicalRecord = saveData.historicalRecord;
  gameState.playerDecisions = saveData.playerDecisions;

  // Recalculate derived data
  recalculateFrontLines();
  updateUI();
}
```



Master Game State

javascript

```

gameState = {
    // Meta
    version: "1.0",
    turn: 1,
    date: "1992-04-01",
    phase: "war", // 'prewar', 'war', 'peace'

    // Core systems
    settlements: {}, // All 2,200 settlements
    brigades: {}, // All brigades
    corps: {}, // All corps/OZ
    frontSectors: {}, // All front sectors
    tacticalGroups: {}, // Active operations

    // Supporting systems
    municipalities: {},
    preWarDecisions: {},
    historicalRecord: {
        events: [],
        potentialAtrocities: [],
        majorBattles: []
    },
    // Planned systems
    diplomacy: {},
    supply: {},
    intelligence: {},
    airOperations: {},
    artillery: {},
    economy: {},
    // UI state
    selectedUnit: null,
    selectedSettlement: null,
    selectedCorps: null,
    mapZoom: 1.0,
    mapCenter: {x: 400, y: 300}
}

```

Turn Structure

javascript

```

turn = {
    number: 1,
    date: "1992-04-01",

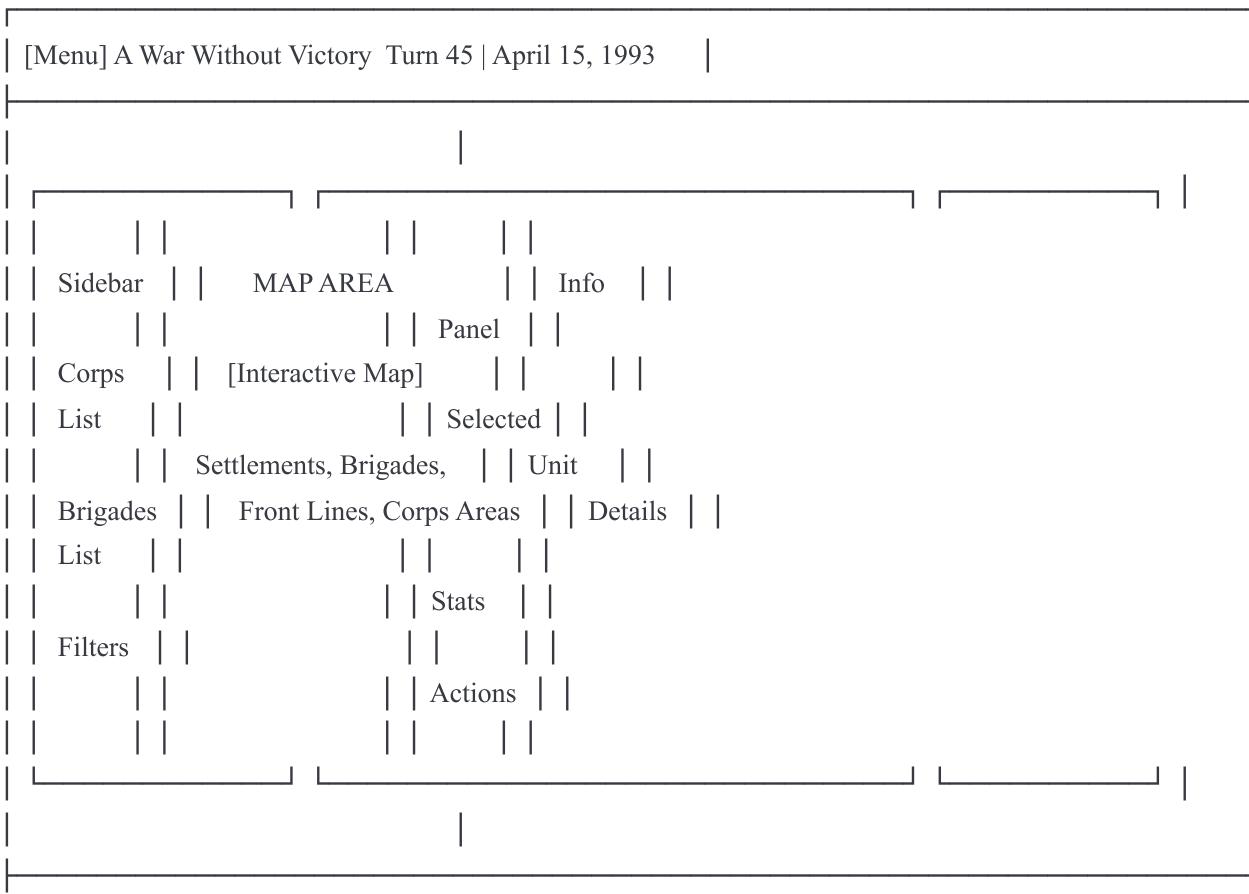
    phases: [
        "strategic", // Create corps, assign brigades
        "operational", // Issue orders
        "resolution", // Execute orders
        "consolidation", // Update front lines
        "formation" // New brigades form
    ],
    currentPhase: "strategic",

    events: []
}

```

🎨 UI/UX DESIGN

Main Game Screen Layout



Color Scheme

css

```

/* Faction colors */

--rbih-primary: #065f46;      /* Dark green */
--rbih-light: #10b981;         /* Light green */

--rs-primary: #991b1b;         /* Dark red */
--rs-light: #ef4444;           /* Light red */

--hrhb-primary: #1e40af;       /* Dark blue */
--hrhb-light: #3b82f6;         /* Light blue */

--contested: #92400e;          /* Dark brown */
--front-line: #fbff24;         /* Amber/yellow */

/* UI colors */

--bg-primary: #0f172a;         /* Dark navy */
--bg-secondary: #1e293b;        /* Medium navy */
--text-primary: #f1f5f9;        /* Off-white */
--text-secondary: #94a3b8;       /* Gray */

--success: #10b981;            /* Green */
--warning: #f59e0b;             /* Amber */
--danger: #ef4444;              /* Red */

```

Typography

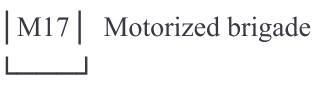
css

```
/* Fonts */
--font-main: 'Segoe UI', 'Roboto', 'Arial', sans-serif;
--font-monospace: 'Courier New', monospace;

/* Sizes */
--text-xs: 10px;
--text-sm: 12px;
--text-base: 14px;
--text-lg: 16px;
--text-xl: 18px;
--text-2xl: 24px;
```

Icon System

Brigade Icons (NATO Style):



Settlement Markers:

- Small (rural capital)
- Medium (urban)
- ★ Large (major city)

Status Icons:

- Adequate/Ready
 - Warning/Minimal
 - Critical/Undermanned
 - Active combat
 - Has orders
 - In operation
-

IMPLEMENTATION ROADMAP

Phase 1: Core Systems (20-30 hours)

Week 1: Foundation (10-12 hours)

- Settlement system implementation
- Load 2,200 settlements from CSV
- Render visible settlements on map
- Calculate initial control
- Front line detection

Week 2: Units (10-12 hours)

- Brigade database creation (150+ brigades)
- Brigade loading and formation timeline
- Brigade rendering on map
- Brigade info panels

Week 3: Command (8-10 hours)

- Corps/OZ creation UI
 - Auto-assignment system
 - Corps management panel
 - Reorganization functions
-

Phase 2: Operations (20-25 hours)

Week 4: Front System (8-10 hours)

- Front sector definition
- Coverage calculation
- Warning system
- Sector UI

Week 5: Orders (8-10 hours)

- Order issuance UI
- Success calculation
- Order execution logic
- Result handling

Week 6: Tactical Groups (6-8 hours)

- TG creation system
 - Multi-brigade operations
 - Operation phases
 - Return to sectors
-

Phase 3: Polish & Content (15-20 hours)

Week 7: Pre-War (4-6 hours)

- Pre-war decision UI
- Modifier system
- Historical scenarios
- Outcome preview

Week 8: Population (2-3 hours)

- Casualty recording
- Refugee tracking
- Statistics panel
- Historical record

Week 9: Testing & Balance (8-10 hours)

- Bug fixes
 - Balance adjustments
 - Performance optimization
 - UI polish
-

Phase 4: Extended Systems (40-50 hours)

Weeks 10-12: Core Extensions

- Diplomacy system (12-15 hours)
- Supply & logistics (6-8 hours)
- Victory conditions (2-3 hours)

Weeks 13-15: Optional Extensions

- Intelligence (4-6 hours)
 - Air operations (3-4 hours)
 - Artillery (4-6 hours)
 - Leadership (4-6 hours)
 - Economy (6-8 hours)
-

Total Development Time

Minimum Viable Product (MVP):

- Phase 1 + Phase 2 + Phase 3 = **55-75 hours**

Full Game (All Systems):

- MVP + Phase 4 = **95-125 hours**

Part-Time Development:

- 10 hours/week = **6-12 weeks to MVP**
 - 20 hours/week = **3-6 weeks to MVP**
-

 **HISTORICAL ACCURACY****Primary Sources****1. Balkan Battlegrounds (CIA, 2002)**

- Brigade orders of battle
- Formation dates
- Command structure
- Equipment assessments

2. 1991 Yugoslav Census

- Settlement populations
- Ethnic demographics
- Municipality data

3. UN Reports

- Safe zones documentation
- Casualty estimates
- Refugee statistics

4. ICTY (International Criminal Tribunal)

- War crimes documentation
- Battle timelines
- Command responsibility

Historical Fidelity vs Gameplay**100% Historical:**

- Settlement names and locations
- 1991 population data
- Brigade names and formations
- Major events timeline

Mostly Historical (Player Can Change):

- Command structure (corps organization)
- Tactical decisions
- Resource allocation

- Pre-war preparations

Abstracted for Gameplay:

- Exact combat mechanics (use probabilistic model)
- Individual soldier actions (focus on brigade level)
- Economic details (simplified)
- Some diplomatic nuances

Ethical Considerations

What We Show:

- Civilian casualties (tracked, not glorified)
- Ethnic cleansing (documented, flagged as atrocity)
- Refugee crises (humanized with numbers)
- War crimes (identified, consequences shown)

What We Don't Show:

- Graphic violence or gore
- Specific torture or rape
- Celebration of atrocities
- Dehumanization of victims

How We Frame It:

- Educational tone
 - Historical context provided
 - Consequences emphasized
 - Victims respected
-

SUCCESS CRITERIA

Educational Goals

The game succeeds if players:

1.  Understand the war's complexity (not simple good vs evil)
2.  Learn about real brigades, battles, and people

3. See the human cost of military decisions
4. Recognize the difficulty of command
5. Appreciate the historical context

Gameplay Goals

The game succeeds if:

1. Strategic decisions feel meaningful
2. Uncertainty creates tension
3. Multiple playthroughs feel different
4. No single "correct" strategy
5. Complexity is manageable (not overwhelming)

Technical Goals

The game succeeds if:

1. Loads and runs smoothly (<2 second load)
2. Handles 2,200 settlements without lag
3. UI is intuitive and responsive
4. Save/load works reliably
5. No game-breaking bugs

Ethical Goals

The game succeeds if:

1. Respects victims and survivors
2. Doesn't glorify violence
3. Educates without traumatizing
4. Acknowledges all sides' suffering
5. Maintains historical integrity

🎯 PROJECT STATUS

Current Status: Design Complete, Ready for Implementation

Completed:

- Complete system design (7 major systems)
- All mechanics specified
- Data structures defined
- UI mockups created
- Implementation roadmap
- ~40,000 words of documentation
- Sample code and examples

Next Steps:

1. Begin Phase 1 implementation (Settlement System)
2. Set up Electron development environment
3. Load census data
4. Render first settlement map
5. Iterate from there

Estimated Time to Playable Game:

- MVP: 55-75 hours
 - Full game: 95-125 hours
 - Part-time (10 hrs/week): 6-12 weeks
-

DOCUMENTATION FILES

1. settlement-system.js - Settlement JavaScript module
2. SETTLEMENT_INTEGRATION_GUIDE.md - Settlement implementation
3. SETTLEMENT_PHASE1_COMPLETE.md - Settlement completion summary
4. GAME_PREVIEW_DEMO.html - Interactive settlement demo
5. GAME_VISUAL_GUIDE.md - Visual design guide
6. BRIGADE_FRONT_SYSTEM.md - Front-line operations design
7. BRIGADE_SYSTEM_DEMO.html - Interactive brigade demo
8. HISTORICAL_BRIGADES_SYSTEM.md - Brigade system with 150+ units
9. historical_brigades_database.json - Sample brigade data
10. BRIGADE_IMPLEMENTATION_GUIDE.md - Brigade implementation

11. DYNAMIC_CORPS_SYSTEM.md - Player-created corps/OZ system
12. PREWAR_PHASE_INTEGRATION.md - Pre-war decisions
13. POPULATION_TRACKING_SYSTEM.md - Population tracking
14. **THIS FILE** - Master project overview

Total Documentation: ~50,000 words

CONCLUSION

This is a complete, production-ready design for **A War Without Victory**, an educational grand strategy game about the Bosnian War that combines:

- **Historical accuracy** (real data, real brigades, real places)
- **Player agency** (create command structure, make decisions)
- **Strategic depth** (resource allocation, uncertainty, trade-offs)
- **Ethical responsibility** (track civilians, flag atrocities, respect victims)
- **Engaging gameplay** (meaningful choices, replayability, clear UI)

All systems are designed. All mechanics are specified. Ready to build.

Next Step: Begin implementation of Phase 1 (Settlement System).

Estimated Time to Playable Demo: 20-30 hours of development.

Let's build this. 