

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
  <title>GRIDIRON 26: PRO MOBILE</title>
  <script src="https://cdnjs.cloudflare.com"></script>
  <style>
    :root { --neon: #00ff88; --ui-bg: rgba(0,0,0,0.8); }
    body { margin: 0; background: #000; color: #fff; font-family: 'Orbitron', sans-serif; overflow:
hidden; touch-action: none; }

    /* ANALOG STICK SYSTEM */
    #joystick-container { position: absolute; bottom: 40px; left: 40px; width: 120px; height:
120px; background: var(--ui-bg); border-radius: 50%; border: 2px solid #333; z-index: 100; }
    #joystick-knob { position: absolute; top: 50%; left: 50%; width: 50px; height: 50px;
background: var(--neon); border-radius: 50%; transform: translate(-50%, -50%); box-shadow: 0
0 15px var(--neon); }

    /* SKILL ACTION BUTTONS */
    #action-bar { position: absolute; bottom: 40px; right: 40px; display: flex; gap: 15px; z-index:
100; }
    .skill-btn { width: 70px; height: 70px; background: var(--ui-bg); border: 2px solid var(--neon);
border-radius: 50%; color: #fff; font-size: 10px; font-weight: 900; display: flex; align-items:
center; justify-content: center; text-transform: uppercase; cursor: pointer; }
    .skill-btn:active { background: var(--neon); color: #000; }

    #hud { position: absolute; top: 20px; width: 100%; display: flex; justify-content:
space-between; padding: 0 30px; box-sizing: border-box; pointer-events: none; }
    #overlay { position: absolute; inset: 0; background: rgba(0,0,0,0.9); display: flex;
align-items: center; justify-content: center; z-index: 1000; }
  </style>
</head>
<body>

<div id="hud">
  <div style="border-left: 3px solid var(--neon); padding-left: 10px;">
    <small>2026 SEASON</small><br><b id="ui-score">COL 0 - 0 TEX</b>
  </div>
  <div style="text-align: right;">
    <small>MOCK DRAFT</small><br><b style="color:var(--neon)">1ST ROUND</b>
  </div>
</div>

```

```

<!-- TOUCH CONTROLS -->
<div id="joystick-container"><div id="joystick-knob"></div></div>
<div id="action-bar">
  <div class="skill-btn" onclick="Engine.skillMove('SPIN')">Spin</div>
  <div class="skill-btn" onclick="Engine.skillMove('JUMP')">Jump</div>
  <div class="skill-btn" onclick="Engine.skillMove('JUKE')">Juke</div>
</div>

<div id="overlay">
  <div style="text-align:center;">
    <h1 style="font-size: 48px; color:var(--neon); margin:0;">GRIDIRON 26</h1>
    <button onclick="Engine.launch()" style="background:var(--neon); border:none;
padding:15px 40px; font-weight:900; margin-top:20px; clip-path: polygon(10% 0, 100% 0, 90%
100%, 0 100%);">KICKOFF</button>
  </div>
</div>

<script>
const Engine = {
  scene: null, camera: null, renderer: null, player: null,
  joystick: { active: false, x: 0, z: 0, startPos: { x: 0, y: 0 } },
  active: false,

  launch() {
    document.getElementById('overlay').style.display = 'none';
    this.init3D();
    this.initJoystick();
    this.active = true;
  },

  init3D() {
    this.scene = new THREE.Scene();
    this.camera = new THREE.PerspectiveCamera(60,
window.innerWidth/window.innerHeight, 0.1, 1000);
    this.renderer = new THREE.WebGLRenderer({ antialias: true });
    this.renderer.setSize(window.innerWidth, window.innerHeight);
    this.renderer.setPixelRatio(window.devicePixelRatio);
    document.body.appendChild(this.renderer.domElement);

    this.scene.add(new THREE.AmbientLight(0xffffff, 0.7));
    const field = new THREE.Mesh(new THREE.PlaneGeometry(200, 100), new
THREE.MeshPhongMaterial({color: 0x1a3d16}));
    field.rotation.x = -Math.PI/2;
    this.scene.add(field);
  }
};

```

```

    const pGeo = new THREE.CapsuleGeometry(0.6, 1.5, 4, 8);
    this.player = new THREE.Mesh(pGeo, new THREE.MeshPhongMaterial({color:
0x00ff88}));
    this.player.position.set(-60, 1.5, 0);
    this.scene.add(this.player);

    this.animate();
  },

  initJoystick() {
    const container = document.getElementById('joystick-container');
    const knob = document.getElementById('joystick-knob');
    const rect = container.getBoundingClientRect();
    this.joystick.startPos = { x: rect.left + rect.width/2, y: rect.top + rect.height/2 };

    window.addEventListener('touchstart', e => {
      if(e.touches[0].clientX < window.innerWidth/2) this.joystick.active = true;
    });

    window.addEventListener('touchmove', e => {
      if(!this.joystick.active) return;
      const touch = e.touches[0];
      let dx = touch.clientX - this.joystick.startPos.x;
      let dy = touch.clientY - this.joystick.startPos.y;
      const dist = Math.min(60, Math.hypot(dx, dy));
      const angle = Math.atan2(dy, dx);

      this.joystick.x = Math.cos(angle) * (dist / 60);
      this.joystick.z = Math.sin(angle) * (dist / 60);

      knob.style.left = `calc(50% + ${Math.cos(angle)*dist}px)`;
      knob.style.top = `calc(50% + ${Math.sin(angle)*dist}px)`;
    });

    window.addEventListener('touchend', () => {
      this.joystick.active = false;
      this.joystick.x = 0; this.joystick.z = 0;
      knob.style.left = '50%'; knob.style.top = '50%';
    });
  },

  skillMove(type) {
    if(!this.active) return;

```

```

        if(window.navigator.vibrate) window.navigator.vibrate(20); // iPhone 16e Haptics

        if(type === 'SPIN') this.player.rotation.y += Math.PI * 2;
        if(type === 'JUMP' && this.player.position.y <= 1.6) this.player.position.y += 3;
        if(type === 'JUKE') this.player.position.z += this.player.position.z > 0 ? -4 : 4;
    },

    update() {
        if(!this.active) return;
        // Apply Joystick Movement to 3D Player
        const speed = 0.25;
        this.player.position.x += this.joystick.x * speed + 0.1; // Forward bias
        this.player.position.z += this.joystick.z * speed;

        // Gravity for Jump
        if(this.player.position.y > 1.5) this.player.position.y -= 0.15;

        this.camera.position.lerp(new THREE.Vector3(this.player.position.x - 15, 12,
this.player.position.z), 0.1);
        this.camera.lookAt(this.player.position);
    },

    animate() {
        requestAnimationFrame(() => this.animate());
        this.update();
        this.renderer.render(this.scene, this.camera);
    }
};
window.onload = () => console.log("GRIDIRON 26: TOUCH ENGINE ACTIVE");
</script>
</body>
</html>

```