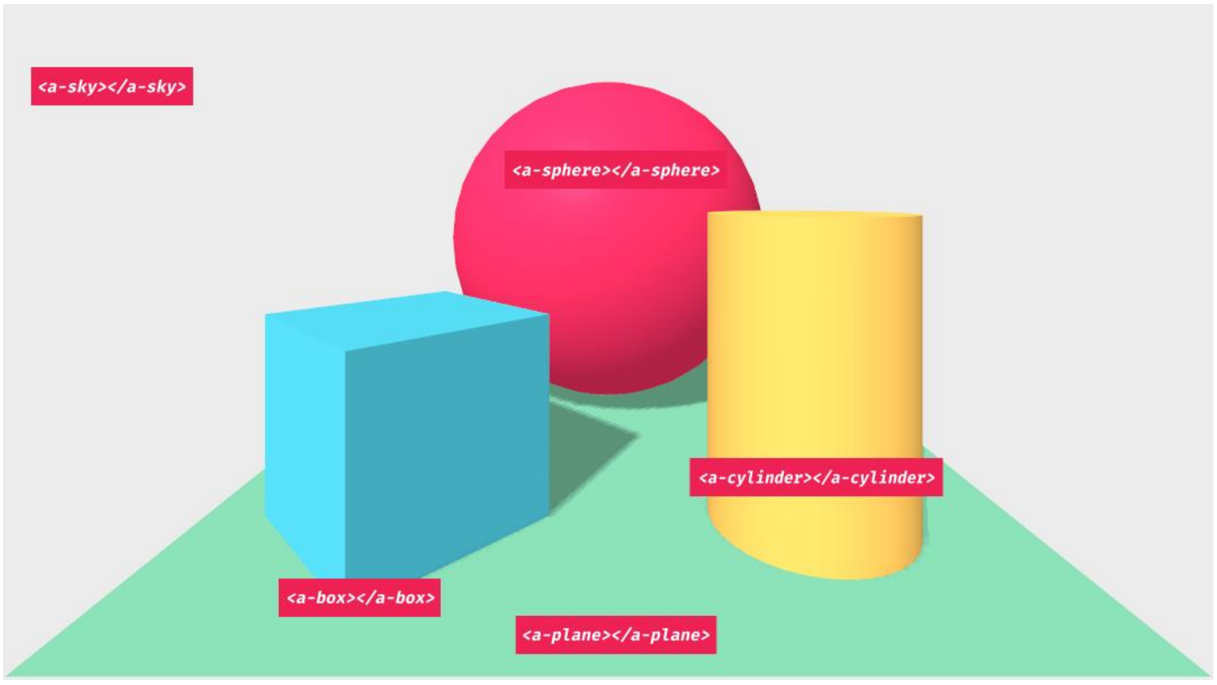


Střelnice

(2 body) Připravte střelnici. Rozmístěte do prostoru podklad (plochu s texturou trávy) a několik terčů (červené koule s hnědým válcem jako podstavou).

(2 body) Před kamerou bude zbraň, např. https://poly.pizza/m/BoZWhFdsj4.

```
  <a-entity camera look-controls="enabled: false" player position="0 1.6 0">
  <a-entity gltf-model="#pistol" log-gltf-animations animation-mixer="clip: idle;" position="0.2 -0.2 -0.3"
rotation="0 -90 0" scale="0.2 0.2 0.2" animation="property: rotation; to: 0 -90 0; dur: 200; easing: linear"></a-
entity>
  <a-entity raycaster="direction: 1 0 0; far: 2;" position="0 0.5 0" rotation="0 0 0" collider-check></a-entity>
</a-entity>
```

(3 body) Klávesami WASD dojde k naklopení pohledu kamery do stran pomocí animace. Zbraň je vždy ve středu zorného pole před kamerou.

```
AFRAME.registerComponent('player', {
    init: function () {
        this.moveSpeed = 0.1;
        this.tiltSpeed = 2;
        this.keys = {
            w: false,
            a: false,
            s: false,
            d: false
        };

        document.addEventListener('keydown', (e) => {
            if (this.keys.hasOwnProperty(e.key.toLowerCase())) {
                this.keys[e.key.toLowerCase()] = true;
            }
        });

        document.addEventListener('keyup', (e) => {
            if (this.keys.hasOwnProperty(e.key.toLowerCase())) {
                this.keys[e.key.toLowerCase()] = false;
            }
        });
    },

    tick: function () {
        const position = this.el.getAttribute('position');
        const rotation = this.el.getAttribute('rotation');

        // Pohyb dopředu/dozadu
        if (this.keys.w) {
            position.z -= Math.cos(rotation.y * Math.PI / 180) *
this.moveSpeed;
            position.x -= Math.sin(rotation.y * Math.PI / 180) *
this.moveSpeed;
        }
        if (this.keys.s) {
            position.z += Math.cos(rotation.y * Math.PI / 180) *
this.moveSpeed;
            position.x += Math.sin(rotation.y * Math.PI / 180) *
this.moveSpeed;
        }

        // Naklánění do stran
        if (this.keys.a) {
            rotation.y += this.tiltSpeed;
        }
        if (this.keys.d) {
            rotation.y -= this.tiltSpeed;
        }

        this.el.setAttribute('position', position);
        this.el.setAttribute('rotation', rotation);
    }
});
```

//může být i rotace nahoru a dolů

```
if (this.keys.w) {
            rotation.x += this.tiltSpeed;
        }
        if (this.keys.s) {
            rotation.x -= this.tiltSpeed;
        }
```

(3 body) Stisknutím mezery dojde k vystřelení. Využijte raycaster k zjištění, jestli došlo k trefení některého terče, a vypište, který terč jsem trefil, do konzole. Není třeba zobrazovat vystřelený náboj.

```
// Přidání raycasteru pro střelbu
this.raycaster = this.el.querySelector('[raycaster]');
```

keydown

```
if (e.code === 'Space') {
   this.keys.space = true;
   this.shoot();
}
```

keyup

```
if (e.code === 'Space') {
   this.keys.space = false;
}
```

funkce střílení

```
shoot: function() {
   // Získání průsečíků raycasteru
   const intersections = this.raycaster.components.raycaster.intersectedEls;

   // Kontrola, zda byl zasažen nějaký terč
   for (let i = 0; i < intersections.length; i++) {
      const hitObject = intersections[i];
      if (hitObject.hasAttribute('obstacle')) {
         console.log('Trefil jsem terč na pozici:', hitObject.getAttribute('position'));

         // Animace pádu terče
         const currentRotation = hitObject.getAttribute('rotation');
         const currentPosition = hitObject.getAttribute('position');

         // Nastavení animace pádu
         hitObject.setAttribute('animation', {
            property: 'rotation',
            to: `${currentRotation.x + 90} ${currentRotation.y} ${currentRotation.z}`,
            dur: 1000,
            easing: 'easeOutQuad'
         });

         // Nastavení animace posunu dolů
         hitObject.setAttribute('animation__position', {
            property: 'position',
            to: `${currentPosition.x} ${currentPosition.y - 1} ${currentPosition.z}`,
            dur: 1000,
            easing: 'easeOutQuad'
```

```
        });

        break;
      }
    }
},
```

(3 body) Pokud jsem terč zasáhnul, daný terč se pomocí animace skácí k zemi.

```
// Animace pádu terče
const currentRotation = hitObject.getAttribute('rotation');
const currentPosition = hitObject.getAttribute('position');

// Nastavení animace pádu
hitObject.setAttribute('animation', {
   property: 'rotation',
   to: `${currentRotation.x + 90} ${currentRotation.y} ${currentRotation.z}`,
   dur: 1000,
   easing: 'easeOutQuad'
});

// Nastavení animace posunu dolů
hitObject.setAttribute('animation__position', {
   property: 'position',
   to: `${currentPosition.x} ${currentPosition.y - 1} ${currentPosition.z}`,
   dur: 1000,
   easing: 'easeOutQuad'
});
```

(2 body) Jakmile trefím všechny terče, vypíše se zeleně výhra.

Přidám do style.css

```
#game-win {
  display: none;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  color: green;
  font-size: 48px;
  font-weight: bold;
  z-index: 999;
}
```

Do main.css

```
 <div id="game-win">You won!</div>
```

Vložím raycaster:

```
 <!-- Character -->


            <a-entity camera look-controls="enabled: false" player position="0 1.6 0">
            <a-entity gltf-model="#pistol" log-gltf-animations animation-mixer="clip: idle;" position="0.2 -0.2 -0.3" rotation="0 -90 0" scale="0.2 0.2 0.2" animation="property: rotation; to: 0 -90 0; dur: 200; easing: linear"></a-entity>
            <a-entity raycaster="objects: [obstacle]; direction: 0 0 -1; far: 100;" position="0 0 0"></a-entity>
        </a-entity>
```

Minecraft

(3b) Vytvořte scénu tvořenou různobarevnými kostkami. Scéna bude mít světle zelené kostky jako podklad, stromy budou tvořeny hnědými kostkami jako kmen a tmavě zelenými jako koruna, dále bude ve scéně modrá voda

```
<!--  ground --> <a-box static-body="friction: 0;" position="0 0 0" width="20" depth="20" height="0.2"
material="color: #90EE90; repeat: 20 20;"></a-box>
<!--  water  --> <a-box static-body="friction: 0;" position="0 0.1 0" width="20" depth="2" height="0.2"
material="color: #4169E1; repeat: 1 1;"></a-box>

<!-- Camera -->

<!-- Trees -->
<a-box obstacle="strength: 9999" static-body position="1.5 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
<a-box obstacle="strength: 9999" static-body position="1.5 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

<a-box obstacle="strength: 9999" static-body position="-5.0 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
<a-box obstacle="strength: 9999" static-body position="-5.0 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

<a-box obstacle="strength: 9999" static-body position="-1.0 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
<a-box obstacle="strength: 9999" static-body position="-1.0 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

<!-- Nové stromy -->
<a-box obstacle="strength: 9999" static-body position="3.0 1.5 -8" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
<a-box obstacle="strength: 9999" static-body position="3.0 4.5 -8" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

<a-box obstacle="strength: 9999" static-body position="-3.0 1.5 -8" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
<a-box obstacle="strength: 9999" static-body position="-3.0 4.5 -8" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>
```

(2b) Hráč bude chodit s postavou pomocí wasd, samotná postava nebude vidět, pouze bude vidět před kamerou aktuální nástroj, a to jeden z

https://poly.pizza/m/*6oI2NlNjsG*

https://poly.pizza/m/ziKFCn82Zz

```
<!-- Character -->

  <a-entity camera look-controls="enabled: false" player position="0 1.6 0">
  <a-entity gltf-model="#axe" log-gltf-animations animation-mixer="clip: idle;" position="0.2 -0.2 -0.3"
rotation="0 -45 0" scale="0.2 0.2 0.2"></a-entity>
  <a-entity raycaster="objects: [obstacle]; direction: 0 0 -1; far: 3;" position="0 0 0"></a-entity>
</a-entity>
```

```
document.addEventListener('keydown', (e) => {
    if (this.keys.hasOwnProperty(e.key.toLowerCase())) {
        this.keys[e.key.toLowerCase()] = true;
    }
    if (e.code === 'Space') {
        this.keys.space = true;
        this.startBreaking();
    }
    if (e.code === 'KeyM') {
        this.switchTool();
    }
});
```

Bloky ve hře bude možné rozbíjet. Rozbíjí se zmáčknutím mezerníku. Daný nástroj pomocí animace několikrát sekne do bloku před hráčem a ten je následně odebrán ze scény. Pokud jsou další bloky nad rozbitým blokem, propadnou se dolů

```
startBreaking: function() {
    if (this.isBreaking) return;

    const intersections = this.raycaster.components.raycaster.intersectedEls;
    for (let i = 0; i < intersections.length; i++) {
        const block = intersections[i];
        if (block.hasAttribute('obstacle') && !this.hitTargets.has(block)) {
            this.isBreaking = true;
            this.breakingBlock = block;

            // Animace sekání - pouze rotace
            const tool = this.el.querySelector('[gltf-model]');

            // Animace rotace
            tool.setAttribute('animation__rotation', {
                property: 'rotation',
                to: '-90 45 0',
                dur: 200,
                easing: 'linear',
                loop: true,
                dir: 'alternate'
            });

            // Počkáme 2 sekundy a pak blok odstraníme
            setTimeout(() => {
                if (this.breakingBlock) {
                    this.hitTargets.add(this.breakingBlock);
                    this.blocksMined++;

                    // Aktualizace konzole
                    console.log(`Vytěžené bloky: ${this.blocksMined}`);
                    if (this.blocksMined === 5) {
                        console.log('Krumpáč odemčen! Stiskněte M pro přepnutí nástroje.');
                    }

                    // Kontrola bloků nad rozbitým blokem
                    const blockPosition = this.breakingBlock.getAttribute('position');
                    const blocksAbove = document.querySelectorAll('[obstacle]');
```

```
          blocksAbove.forEach(block => {
            const pos = block.getAttribute('position');
            if (pos.x === blockPosition.x &&
              pos.z === blockPosition.z &&
              pos.y > blockPosition.y) {
              // Propadnutí bloku dolů
              block.setAttribute('position', {
                x: pos.x,
                y: pos.y - 1,
                z: pos.z
              });
            }
          });

          // Odstranění rozbitého bloku
          this.breakingBlock.parentNode.removeChild(this.breakingBlock);
          this.isBreaking = false;
          this.breakingBlock = null;

          // Zastavení animace
          const tool = this.el.querySelector('[gltf-model]');
          tool.removeAttribute('animation__rotation');

          // Vrácení do výchozí pozice
          tool.setAttribute('position', '0.2 -0.2 -0.3');
          tool.setAttribute('rotation', '0 -45 0');

          // Kontrola výhry
          this.checkWinCondition();
        }
      }, 2000);

      break;
    }
  }
},
```

Na začátku má hráč pouze sekeru. Po vytěžení 5 bloků dřeva obdrží krumpáč. Nástroj je možné měnit stisknutím klávesy M.

```
switchTool: function() {
  if (this.blocksMined >= 5) {
    this.currentTool = this.currentTool === 'axe' ? 'pickaxe' : 'axe';
    const tool = this.el.querySelector('[gltf-model]');

    // Změna modelu nástroje
    if (this.currentTool === 'pickaxe') {
      tool.setAttribute('gltf-model', '#pickaxe');
      tool.setAttribute('position', '0.2 -0.2 -0.3');
      tool.setAttribute('rotation', '0 -45 0');
      console.log('Nástroj změněn na: Krumpáč');
    } else {
      tool.setAttribute('gltf-model', '#axe');
      tool.setAttribute('position', '0.2 -0.2 -0.3');
      tool.setAttribute('rotation', '0 -45 0');
```

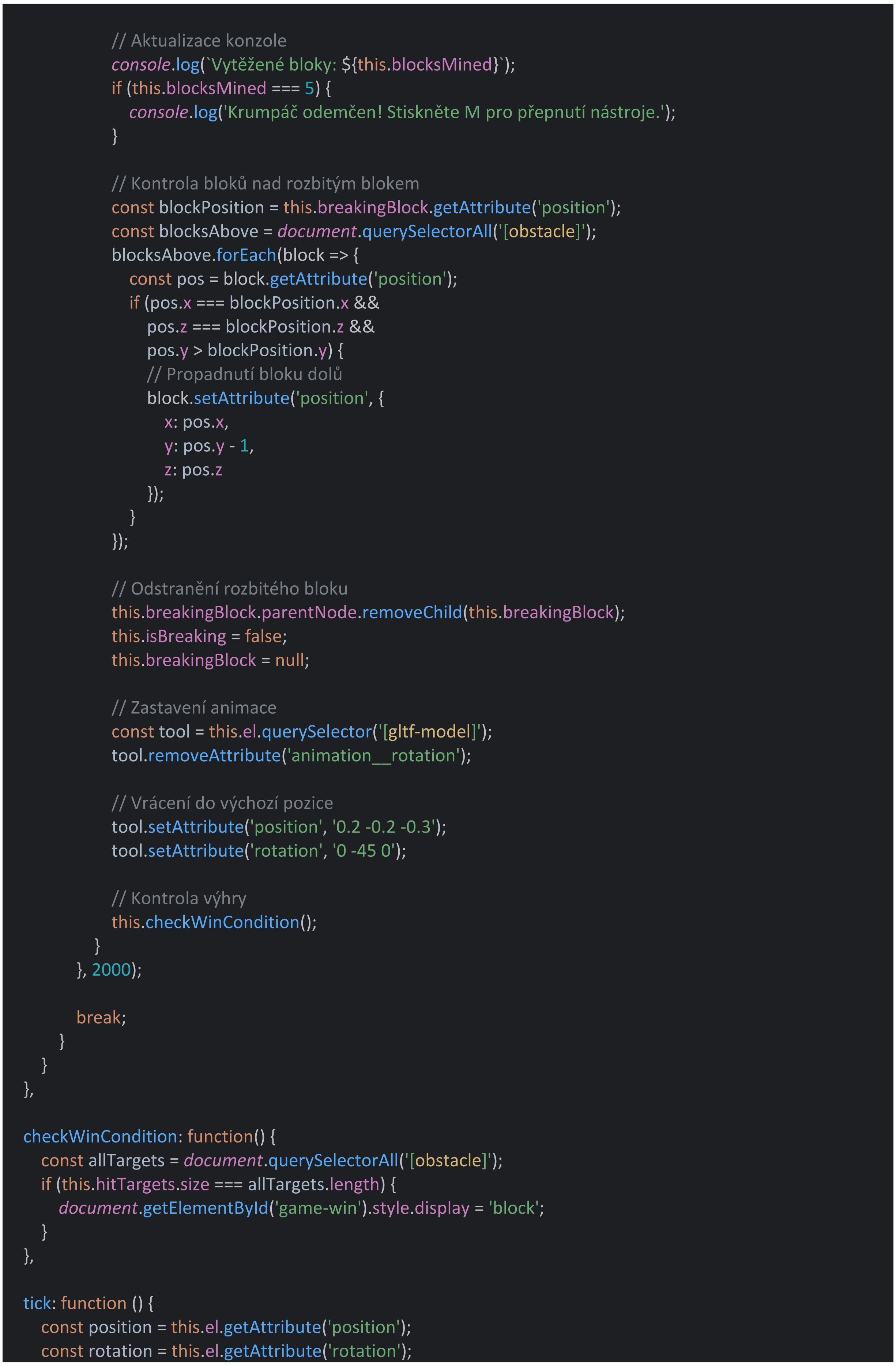
```
            console.log('Nástroj změněn na: Sekera');
        }
    } else {
        console.log(`Potřebujete vytěžit ještě ${5 - this.blocksMined} bloků pro odemknutí krumpáče`);
    }
},
```

V konzoli se ukazuje počet vytěžených bloků dřeva a zeminy

```
// Inicializace konzole
console.log('=== Stavy vytěžených bloků ===');
console.log('Vytěžené bloky: 0');
```

```
Character.js
AFRAME.registerComponent('player', {
    init: function () {
        this.moveSpeed = 0.1;
        this.tiltSpeed = 2;
        this.keys = {
            w: false,
            a: false,
            s: false,
            d: false,
            space: false,
            m: false
        };
        this.hitTargets = new Set();
        this.isBreaking = false;
        this.breakingBlock = null;
        this.blocksMined = 0;
        this.currentTool = 'axe'; // 'axe' nebo 'pickaxe'

        // Přidání raycasteru pro těžbu
        this.raycaster = this.el.querySelector('[raycaster]');

        // Inicializace konzole
        console.log('=== Stavy vytěžených bloků ===');
        console.log('Vytěžené bloky: 0');

        document.addEventListener('keydown', (e) => {
            if (this.keys.hasOwnProperty(e.key.toLowerCase())) {
                this.keys[e.key.toLowerCase()] = true;
            }
            if (e.code === 'Space') {
                this.keys.space = true;
                this.startBreaking();
            }
            if (e.code === 'KeyM') {
                this.switchTool();
            }
        });

        document.addEventListener('keyup', (e) => {
            if (this.keys.hasOwnProperty(e.key.toLowerCase())) {
                this.keys[e.key.toLowerCase()] = false;
```

```
      }
      if (e.code === 'Space') {
        this.keys.space = false;
      }
    });
  },

  switchTool: function() {
    if (this.blocksMined >= 5) {
      this.currentTool = this.currentTool === 'axe' ? 'pickaxe' : 'axe';
      const tool = this.el.querySelector('[gltf-model]');

      // Změna modelu nástroje
      if (this.currentTool === 'pickaxe') {
        tool.setAttribute('gltf-model', '#pickaxe');
        tool.setAttribute('position', '0.2 -0.2 -0.3');
        tool.setAttribute('rotation', '0 -45 0');
        console.log('Nástroj změněn na: Krumpáč');
      } else {
        tool.setAttribute('gltf-model', '#axe');
        tool.setAttribute('position', '0.2 -0.2 -0.3');
        tool.setAttribute('rotation', '0 -45 0');
        console.log('Nástroj změněn na: Sekera');
      }
    } else {
      console.log(`Potřebujete vytěžit ještě ${5 - this.blocksMined} bloků pro odemknutí krumpáče`);
    }
  },

  startBreaking: function() {
    if (this.isBreaking) return;

    const intersections = this.raycaster.components.raycaster.intersectedEls;
    for (let i = 0; i < intersections.length; i++) {
      const block = intersections[i];
      if (block.hasAttribute('obstacle') && !this.hitTargets.has(block)) {
        this.isBreaking = true;
        this.breakingBlock = block;

        // Animace sekání - pouze rotace
        const tool = this.el.querySelector('[gltf-model]');

        // Animace rotace
        tool.setAttribute('animation__rotation', {
          property: 'rotation',
          to: '-90 45 0',
          dur: 200,
          easing: 'linear',
          loop: true,
          dir: 'alternate'
        });

        // Počkáme 2 sekundy a pak blok odstraníme
        setTimeout(() => {
          if (this.breakingBlock) {
            this.hitTargets.add(this.breakingBlock);
            this.blocksMined++;
```



```
                    // Aktualizace konzole
                    console.log(`Vytěžené bloky: ${this.blocksMined}`);
                    if (this.blocksMined === 5) {
                        console.log('Krumpáč odemčen! Stiskněte M pro přepnutí nástroje.');
                    }

                    // Kontrola bloků nad rozbitým blokem
                    const blockPosition = this.breakingBlock.getAttribute('position');
                    const blocksAbove = document.querySelectorAll('[obstacle]');
                    blocksAbove.forEach(block => {
                        const pos = block.getAttribute('position');
                        if (pos.x === blockPosition.x &&
                            pos.z === blockPosition.z &&
                            pos.y > blockPosition.y) {
                            // Propadnutí bloku dolů
                            block.setAttribute('position', {
                                x: pos.x,
                                y: pos.y - 1,
                                z: pos.z
                            });
                        }
                    });

                    // Odstranění rozbitého bloku
                    this.breakingBlock.parentNode.removeChild(this.breakingBlock);
                    this.isBreaking = false;
                    this.breakingBlock = null;

                    // Zastavení animace
                    const tool = this.el.querySelector('[gltf-model]');
                    tool.removeAttribute('animation__rotation');

                    // Vrácení do výchozí pozice
                    tool.setAttribute('position', '0.2 -0.2 -0.3');
                    tool.setAttribute('rotation', '0 -45 0');

                    // Kontrola výhry
                    this.checkWinCondition();
                }
            }, 2000);

            break;
        }
    }
},

checkWinCondition: function() {
    const allTargets = document.querySelectorAll('[obstacle]');
    if (this.hitTargets.size === allTargets.length) {
        document.getElementById('game-win').style.display = 'block';
    }
},

tick: function () {
    const position = this.el.getAttribute('position');
    const rotation = this.el.getAttribute('rotation');
```

```
      if (this.keys.w) {
        position.z -= Math.cos(rotation.y * Math.PI / 180) * this.moveSpeed;
        position.x -= Math.sin(rotation.y * Math.PI / 180) * this.moveSpeed;
      }
      if (this.keys.s) {
        position.z += Math.cos(rotation.y * Math.PI / 180) * this.moveSpeed;
        position.x += Math.sin(rotation.y * Math.PI / 180) * this.moveSpeed;
      }

      if (this.keys.a) {
        rotation.y += this.tiltSpeed;
      }
      if (this.keys.d) {
        rotation.y -= this.tiltSpeed;
      }

      this.el.setAttribute('position', position);
      this.el.setAttribute('rotation', rotation);
    }
});
```

Main.js

```
import './style.css'
import 'aframe'
import 'aframe-extras'
import 'aframe-physics-system'
import './components/character'
import './components/obstacle'
import './components/collider-check'
import './components/log-gltf-animations'

document.querySelector('#app').innerHTML = `
  <div id="game-over">You lost!</div>
  <div id="game-win">You won!</div>
  <a-scene>
    <!-- External files -->
    <a-assets>
        <a-asset-item id="axe" src="/models/Diamond Axe.glb"></a-asset-item>
        <a-asset-item id="pickaxe" src="/models/Diamond Pickaxe.glb"></a-asset-item>
      <img src="/models/grass.jpg" id="grass">
    </a-assets>

    <!-- Environment -->
    <!--  sky    --> <a-sky color="#eeeeee"></a-sky>
    <!--  ground --> <a-box static-body="friction: 0;" position="0 0 0" width="20" depth="20" height="0.2"
material="color: #90EE90; repeat: 20 20;"></a-box>
    <!--  water  --> <a-box static-body="friction: 0;" position="0 0.1 0" width="20" depth="2" height="0.2"
material="color: #4169E1; repeat: 1 1;"></a-box>

    <!-- Camera -->

    <!-- Trees -->
    <a-box obstacle="strength: 9999" static-body position="1.5 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
```

```
        <a-box obstacle="strength: 9999" static-body position="1.5 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

        <a-box obstacle="strength: 9999" static-body position="-5.0 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
        <a-box obstacle="strength: 9999" static-body position="-5.0 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

        <a-box obstacle="strength: 9999" static-body position="-1.0 1.5 -6" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
        <a-box obstacle="strength: 9999" static-body position="-1.0 4.5 -6" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

        <!-- Nové stromy -->
        <a-box obstacle="strength: 9999" static-body position="3.0 1.5 -8" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
        <a-box obstacle="strength: 9999" static-body position="3.0 4.5 -8" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

        <a-box obstacle="strength: 9999" static-body position="-3.0 1.5 -8" width="1" height="3" depth="1"
color="#8B4513" material="opacity: 1"></a-box>
        <a-box obstacle="strength: 9999" static-body position="-3.0 4.5 -8" width="4" height="4" depth="4"
color="#006400" material="opacity: 1"></a-box>

        <!-- Character -->

          <a-entity camera look-controls="enabled: false" player position="0 1.6 0">
          <a-entity gltf-model="#axe" log-gltf-animations animation-mixer="clip: idle;" position="0.2 -0.2 -0.3"
rotation="0 -45 0" scale="0.2 0.2 0.2"></a-entity>
          <a-entity raycaster="objects: [obstacle]; direction: 0 0 -1; far: 3;" position="0 0 0"></a-entity>
        </a-entity>



    </a-scene>
`
```