# p5.js: ukázková hra ASTEROIDS

## Grafický framework p5.js

Představení frameworku

## Základní kostra aplikace

index.html

<!doctype html>

<html lang="cs">

  <head>

    <title>P5JS: asteroids</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  </head>

  <body>

    <script src="js/p5.js"></script>

<script src="js/fullscreen.js"></script>

    <script src="js/main.js"></script>

  </body>

</html>

fullscreen.js

/\* Skript pro přepínání okna do režimu celé obrazovky \*/

/\* Zdroj: https://developer.mozilla.org/en-US/docs/Web/API/Fullscreen\_API \*/

/\* Po načtení okna vyvolat funkci startup \*/

window.addEventListener("load", startup, false);

/\* Funkce nastavující zobrazení těla stránky v režimu celé obrazovky po stisku klávesy Enter \*/

function startup() {

  // Konstanta body odkazuje na element body (tělo stránky)

  const body = document.querySelector("body");

  // Po stisku klávesy Enter bude vyvolána funkce toggleFullScreen

  document.addEventListener("keypress", function(e) {

    if (e.key === 'Enter') {

      toggleFullScreen(body);

    }

  }, false);

}

/\* Funkce pro přepínání zobrazení elementu v režimu celé obrazovky \*/

function toggleFullScreen(element) {

    // Jestliže dokument není v režimu celé obrazovky

    if (!document.fullscreenElement) {

      // přepne zadaný element do režimu celé obrazovky

      element.requestFullscreen();

    } else {

    // v opačném případě dojde k opuštění režimu celé obrazovky

    if (document.exitFullscreen) {

      document.exitFullscreen();

    }

  }

}

main.js

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Vytvoření objektu kosmické lodi (main-1.js)

main.js

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    rect(this.x, this.y, this.w, this.h, 10, 10);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Ovládání pohybu kosmické lodi (main-2.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

/\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    rect(this.x, this.y, this.w, this.h, 10, 10);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Otáčení kosmické lodi (main-3.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

  /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Načtení a zobrazení rastrových obrázků v objektu kosmické lodi (main-4.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků spojených s objektem kosmické lodi \*/

let shipImage, blastImage;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

  /\* https://p5js.org/reference/#/p5/image \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

/\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Vypouštění střel z kosmické lodi (main-5.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků spojených s objektem kosmické lodi \*/

let shipImage, blastImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

  /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Padající asteroidy (main-6.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků \*/

let shipImage, blastImage, asteroidImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\* Proměnná typu pole pro uložení objektů asteroidů \*/

let asteroids = [];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Asteroid - návrh šutrů padajících z nebes \*/

class Asteroid {

  constructor() {

    /\* Náhodná velikost v rozmezí 20 až 50 \*/

    this.size = random(20, 50);

    /\* Počáteční souřadnice y mimo horní okraj \*/

    this.y = -100;

    /\* Počáteční souřadnice x - náhodná v rozmezí od 0 po celou šířku plátna - velikost asteroidu\*/

    this.x = random(0, width - this.size);

    this.speed = 3;

    /\* Náhodný úhel otočení v rozmezí 0 až 359 \*/

    this.angle = random(0, 359);

  }

  /\* Metoda vypočítá pohyb asteroidu v ose y \*/

  move() {

    this.y += this.speed;

  }

  /\* Metoda vypočítá změnu úhlu vlivem rotace \*/

  rotate() {

    this.angle += this.speed;

  }

  /\* Metoda provádí vykreslení asteroidu \*/

  draw() {

    this.move();

    this.rotate();

    push();

    translate(this.x + this.size/2, this.y + this.size/2);

    rotate(((2 \* PI) / 360) \* this.angle);

    image(asteroidImage, -this.size/2, -this.size/2, this.size, this.size);

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

  /\* Načtení obrázku asteroidu \*/

  asteroidImage = loadImage("img/asteroid.png");

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

  /\* Každý šedesátý snímek  se vytvoří nový objekt asteroidu \*/

  if (frameCount % 60 == 0) {

    /\* Nový objekt asteroidu bude přidán do pole asteroids \*/

    asteroids.push(new Asteroid());

  }

  /\* Vykresluje postupně všechny objekty z pole asteroids na plátno \*/

  asteroids.forEach(function (asteroid, idx, arr) {

    asteroid.draw();

    /\* Jestliže je y-pozice objektu asteroidu větší než výška plátna, je objekt z pole odstraněn \*/

    if (asteroid.y > height) {

      arr.splice(idx, 1);

    }

  });

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

}

## Vytvoření simulace města z objektů budov (main-7.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků \*/

let shipImage, blastImage, asteroidImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\* Proměnná typu pole pro uložení objektů asteroidů \*/

let asteroids = [];

/\* Proměnná typu pole pro uložení objektů budov \*/

let buildings = [];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Asteroid - návrh šutrů padajících z nebes \*/

class Asteroid {

  constructor() {

    /\* Náhodná velikost v rozmezí 20 až 50 \*/

    this.size = random(20, 50);

    /\* Počáteční souřadnice y mimo horní okraj \*/

    this.y = -100;

    /\* Počáteční souřadnice x - náhodná v rozmezí od 0 po celou šířku plátna - velikost asteroidu\*/

    this.x = random(0, width - this.size);

    this.speed = 3;

    /\* Náhodný úhel otočení v rozmezí 0 až 359 \*/

    this.angle = random(0, 359);

  }

  /\* Metoda vypočítá pohyb asteroidu v ose y \*/

  move() {

    this.y += this.speed;

  }

  /\* Metoda vypočítá změnu úhlu vlivem rotace \*/

  rotate() {

    this.angle += this.speed;

  }

  /\* Metoda provádí vykreslení asteroidu \*/

  draw() {

    this.move();

    this.rotate();

    push();

    translate(this.x + this.size/2, this.y + this.size/2);

    rotate(((2 \* PI) / 360) \* this.angle);

    image(asteroidImage, -this.size/2, -this.size/2, this.size, this.size);

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Building - návrh symbolického objektu budovy \*/

class Building {

  constructor() {

    /\* Náhodná šířka a výška budovy v rozsahu 20 až 100 \*/

    /\* https://p5js.org/reference/#/p5/random \*/

    this.w = random(20, 100);

    this.h = random(20, 100);

    /\* Náhodný výběr umístění objektu budovy v ose x v rozsahu celé šířky obrazovky \*/

    this.x = random(0, width - this.w);

    /\* Nastavení y-souřadnice budovy tak, aby její spodní okraj ležel na spodním okraji obrazovky \*/

    this.y = height - this.h;

  }

  /\* Vykreslení objektu budovy \*/

  draw() {

    stroke(200);

    fill(color(30));

    rectMode(CORNER);

    rect(this.x, this.y, this.w, this.h);

    /\* Vykreslení rastru čtverců v ploše budovy \*/

    for (let xb = 0; xb < this.w; xb+=10) {

      for (let yb = 0; yb < this.h; yb+=10) {

        /\* Náhodné střídání tmavých a žlutých čtverců \*/

        if (random(5) > 2)

          fill(color(50));

        else

          fill(color(150, 150, 0));

        square(this.x + xb, this.y + yb, 5);

      }

    }

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

  /\* Načtení obrázku asteroidu \*/

  asteroidImage = loadImage("img/asteroid.png");

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

  /\* Vygenerování 50 objektů budov \*/

  for (i = 0; i < 50; i++) {

    /\* Vložení nového objektu budovy do pole buildings \*/

    buildings.push(new Building());

  }

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

  /\* Každý šedesátý snímek  se vytvoří nový objekt asteroidu \*/

  if (frameCount % 60 == 0) {

    /\* Nový objekt asteroidu bude přidán do pole asteroids \*/

    asteroids.push(new Asteroid());

  }

  /\* Vykresluje postupně všechny objekty z pole asteroids na plátno \*/

  asteroids.forEach(function (asteroid, idx, arr) {

    asteroid.draw();

    /\* Jestliže je y-pozice objektu asteroidu větší než výška plátna, je objekt z pole odstraněn \*/

    if (asteroid.y > height) {

      arr.splice(idx, 1);

    }

  });

  /\* Vykreslení všech budov \*/

  buildings.forEach(function(building, idx, arr) {

    building.draw();

  });

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

  /\* Aktualizace pozic budov při změně velikosti obrazovky \*/

  buildings.forEach(function(building) {

    building.y = height - building.h;

  });

}

## Detekce kolizí asteroidů s ostatními objekty (main-8.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků \*/

let shipImage, blastImage, asteroidImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\* Proměnná typu pole pro uložení objektů asteroidů \*/

let asteroids = [];

/\* Proměnná typu pole pro uložení objektů budov \*/

let buildings = [];

/\* Proměnná hits - počet sestřelených asteroidů \*/

let hits = 0;

/\* Proměnná damage - míra poškození lodi \*/

let damage = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

    }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Asteroid - návrh šutrů padajících z nebes \*/

class Asteroid {

  constructor() {

    /\* Náhodná velikost v rozmezí 20 až 50 \*/

    this.size = random(20, 50);

    /\* Počáteční souřadnice y mimo horní okraj \*/

    this.y = -100;

    /\* Počáteční souřadnice x - náhodná v rozmezí od 0 po celou šířku plátna - velikost asteroidu\*/

    this.x = random(0, width - this.size);

    this.speed = 3;

    /\* Náhodný úhel otočení v rozmezí 0 až 359 \*/

    this.angle = random(0, 359);

  }

  /\* Metoda vypočítá pohyb asteroidu v ose y \*/

  move() {

    this.y += this.speed;

  }

  /\* Metoda vypočítá změnu úhlu vlivem rotace \*/

  rotate() {

    this.angle += this.speed;

  }

  /\* Metoda provádí vykreslení asteroidu \*/

  draw() {

    this.move();

    this.rotate();

    push();

    translate(this.x + this.size/2, this.y + this.size/2);

    rotate(((2 \* PI) / 360) \* this.angle);

    image(asteroidImage, -this.size/2, -this.size/2, this.size, this.size);

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Building - návrh symbolického objektu budovy \*/

class Building {

  constructor() {

    /\* Náhodná šířka a výška budovy v rozsahu 20 až 100 \*/

    /\* https://p5js.org/reference/#/p5/random \*/

    this.w = random(20, 100);

    this.h = random(20, 100);

    /\* Náhodný výběr umístění objektu budovy v ose x v rozsahu celé šířky obrazovky \*/

    this.x = random(0, width - this.w);

    /\* Nastavení y-souřadnice budovy tak, aby její spodní okraj ležel na spodním okraji obrazovky \*/

    this.y = height - this.h;

  }

  /\* Vykreslení objektu budovy \*/

  draw() {

    stroke(200);

    fill(color(30));

    rectMode(CORNER);

    rect(this.x, this.y, this.w, this.h);

    /\* Vykreslení rastru čtverců v ploše budovy \*/

    for (let xb = 0; xb < this.w; xb+=10) {

      for (let yb = 0; yb < this.h; yb+=10) {

        /\* Náhodné střídání tmavých a žlutých čtverců \*/

        if (random(5) > 2)

          fill(color(50));

        else

          fill(color(150, 150, 0));

        square(this.x + xb, this.y + yb, 5);

      }

    }

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

  /\* Načtení obrázku asteroidu \*/

  asteroidImage = loadImage("img/asteroid.png");

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

  /\* Vygenerování 50 objektů budov \*/

  for (i = 0; i < 50; i++) {

    /\* Vložení nového objektu budovy do pole buildings \*/

    buildings.push(new Building());

  }

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

  /\* Každý šedesátý snímek  se vytvoří nový objekt asteroidu \*/

  if (frameCount % 60 == 0) {

    /\* Nový objekt asteroidu bude přidán do pole asteroids \*/

    asteroids.push(new Asteroid());

  }

  /\* Vykresluje postupně všechny objekty z pole asteroids na plátno \*/

  asteroids.forEach(function (asteroid, idx, arr) {

    asteroid.draw();

    /\* Jestliže je y-pozice objektu asteroidu větší než výška plátna, je objekt z pole odstraněn \*/

    if (asteroid.y > height) {

      arr.splice(idx, 1);

    }

    /\* Detekce kolize vykreslovaného asteroidu se střelami (missiles) \*/

    /\* Pro detekci kolizí využijeme externí knihovnu p5.collide2D (soubor p5.collide2d.min.js): \*/

    /\* https://github.com/bmoren/p5.collide2D \*/

    missiles.forEach(function(missile, index, array) {

      /\* Funkce collideRectCircle() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) se střelou  (kruh - circle) \*/

      if (collideRectCircle(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        missile.x,

        missile.y,

        missile.size / 2

      ))

      {

        /\* Když funkce vrátí hodnotu true: \*/

        /\* Navýší se počet zásahů (hits) \*/

        hits++;

        /\* Odstraní se z pole missiles kontrolovaná střela \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids zasažený asteroid \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objekty budov (buildings) \*/

    buildings.forEach(function(building, index, array) {

      /\* Funkce collideRectRect() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) s budovou (obdélník - rectangle) \*/

      if (collideRectRect(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        building.x,

        building.y,

        building.w,

        building.h

      ))

      {

        /\* Když došlo ke kolizi: \*/

        /\* Odstraní se z pole buildings kontrolovaný objekt \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids kontrolovaný objekt \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objektem kosmické lodi (spaceShip) \*/

    if (collideRectRect(

      asteroid.x,

      asteroid.y,

      asteroid.size,

      asteroid.size,

      spaceShip.x,

      spaceShip.y,

      spaceShip.w,

      spaceShip.h

    ))

    {

      /\* Když došlo ke kolizi, navýší se hodnota poškození lodi o polovinu velikosti asteroidu \*/

      damage += asteroid.size / 2;

arr.splice(idx, 1);

    }

  });

  /\* Vykreslení všech budov \*/

  buildings.forEach(function(building, idx, arr) {

    building.draw();

  });

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

  /\* Aktualizace pozic budov při změně velikosti obrazovky \*/

  buildings.forEach(function(building) {

    building.y = height - building.h;

  });

}

## Zvukové a grafické efekty (main-9.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků \*/

let shipImage, blastImage, asteroidImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\* Proměnná typu pole pro uložení objektů asteroidů \*/

let asteroids = [];

/\* Proměnná typu pole pro uložení objektů budov \*/

let buildings = [];

/\* Ohnivá místa \*/

let fires = [];

/\* Proměnná hits - počet sestřelených asteroidů \*/

let hits = 0;

/\* Proměnná damage - míra poškození lodi \*/

let damage = 0;

/\* Nastavení proměnných pro zvukové soubory \*/

/\* https://p5js.org/reference/#/libraries/p5.sound \*/

let impactAudio;

let hitAudio;

let damageAudio;

/\* Vytvoření objektu pro zvukovou syntézu \*/

/\* https://p5js.org/reference/#/p5.MonoSynth \*/

let synth = new p5.MonoSynth();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

        synth.play('C4', 0.6, 0, 0.1);

      }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Asteroid - návrh šutrů padajících z nebes \*/

class Asteroid {

  constructor() {

    /\* Náhodná velikost v rozmezí 20 až 50 \*/

    this.size = random(20, 50);

    /\* Počáteční souřadnice y mimo horní okraj \*/

    this.y = -100;

    /\* Počáteční souřadnice x - náhodná v rozmezí od 0 po celou šířku plátna - velikost asteroidu\*/

    this.x = random(0, width - this.size);

    this.speed = 3;

    /\* Náhodný úhel otočení v rozmezí 0 až 359 \*/

    this.angle = random(0, 359);

  }

  /\* Metoda vypočítá pohyb asteroidu v ose y \*/

  move() {

    this.y += this.speed;

  }

  /\* Metoda vypočítá změnu úhlu vlivem rotace \*/

  rotate() {

    this.angle += this.speed;

  }

  /\* Metoda provádí vykreslení asteroidu \*/

  draw() {

    this.move();

    this.rotate();

    push();

    translate(this.x + this.size/2, this.y + this.size/2);

    rotate(((2 \* PI) / 360) \* this.angle);

    image(asteroidImage, -this.size/2, -this.size/2, this.size, this.size);

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Building - návrh symbolického objektu budovy \*/

class Building {

  constructor() {

    /\* Náhodná šířka a výška budovy v rozsahu 20 až 100 \*/

    /\* https://p5js.org/reference/#/p5/random \*/

    this.w = random(20, 100);

    this.h = random(20, 200);

    /\* Náhodný výběr umístění objektu budovy v ose x v rozsahu celé šířky obrazovky \*/

    this.x = random(0, width - this.w);

    /\* Nastavení y-souřadnice budovy tak, aby její spodní okraj ležel na spodním okraji obrazovky \*/

    this.y = height - this.h;

  }

  /\* Vykreslení objektu budovy \*/

  draw() {

    stroke(200);

    fill(color(30));

    rectMode(CORNER);

    rect(this.x, this.y, this.w, this.h);

    /\* Vykreslení rastru čtverců v ploše budovy \*/

    for (let xb = 0; xb < this.w; xb+=10) {

      for (let yb = 0; yb < this.h; yb+=10) {

        /\* Náhodné střídání tmavých a žlutých čtverců \*/

        if (random(5) > 2)

          fill(color(50));

        else

          fill(color(150, 150, 0));

        square(this.x + xb, this.y + yb, 5);

      }

    }

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

  /\* Načtení obrázku asteroidu \*/

  asteroidImage = loadImage("img/asteroid.png");

  /\* Výběr používaných zvukových formátů \*/

  soundFormats('mp3', 'ogg');

  /\* Načtení zvukových souborů \*/

  /\* https://p5js.org/reference/#/p5.SoundFile \*/

  impactAudio = loadSound('sounds/big-impact');

  hitAudio = loadSound('sounds/hit-sound');

  damageAudio = loadSound('sounds/electronic-impact');

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

  /\* Vygenerování 50 objektů budov \*/

  for (i = 0; i < 50; i++) {

    /\* Vložení nového objektu budovy do pole buildings \*/

    buildings.push(new Building());

  }

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vykreslení ohňů na místě kolizí objektů \*/

  fires.forEach(function(fire, idx, arr) {

    image(blastImage, fire.x, fire.y, fire.w, fire.h);

    /\* Po 30 snímcích bude informace o hořícím místě odstraněna z pole \*/

    if (frameCount - fire.frame > 30) {

      arr.splice(idx, 1);

    }

  });

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

  /\* Každý šedesátý snímek  se vytvoří nový objekt asteroidu \*/

  if (frameCount % 60 == 0) {

    /\* Nový objekt asteroidu bude přidán do pole asteroids \*/

    asteroids.push(new Asteroid());

  }

  /\* Vykresluje postupně všechny objekty z pole asteroids na plátno \*/

  asteroids.forEach(function (asteroid, idx, arr) {

    asteroid.draw();

    /\* Jestliže je y-pozice objektu asteroidu větší než výška plátna, je objekt z pole odstraněn \*/

    if (asteroid.y > height) {

      arr.splice(idx, 1);

    }

    /\* Detekce kolize vykreslovaného asteroidu se střelami (missiles) \*/

    /\* Pro detekci kolizí využijeme externí knihovnu p5.collide2D (soubor p5.collide2d.min.js): \*/

    /\* https://github.com/bmoren/p5.collide2D \*/

    missiles.forEach(function(missile, index, array) {

      /\* Funkce collideRectCircle() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) se střelou  (kruh - circle) \*/

      if (collideRectCircle(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        missile.x,

        missile.y,

        missile.size / 2

      ))

      {

        /\* Když funkce vrátí hodnotu true: \*/

        /\* Nastavení síly zvuku - rozmezí 0 až 1 \*/

        hitAudio.setVolume(0.5);

        /\* Přehrání zvuku \*/

        hitAudio.play();

        /\* Navýší se počet zásahů (hits) \*/

        hits++;

        /\* Do pole fires přidá souřadnice a velikost trefeného asteroidu a aktuální číslo snímku (frameCount) \*/

        fires.push({x: asteroid.x, y: asteroid.y, w: asteroid.size, h: asteroid.size, frame: frameCount});

        /\* Odstraní se z pole missiles kontrolovaná střela \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids zasažený asteroid \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objekty budov (buildings) \*/

    buildings.forEach(function(building, index, array) {

      /\* Funkce collideRectRect() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) s budovou (obdélník - rectangle) \*/

      if (collideRectRect(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        building.x,

        building.y,

        building.w,

        building.h

      ))

      {

        /\* Když došlo ke kolizi: \*/

        /\* Nastavení síly zvuku pro efekt dopadu na budovu - síla zvuku je nastavena podle velikosti asteroidu \*/

        /\* https://p5js.org/reference/#/p5.SoundFile/setVolume \*/

        impactAudio.setVolume((1 - 10 / asteroid.size) \* 0.5, 0);

        impactAudio.play();

        /\* Nastavení rychlosti přehrávání zvuku - v tomto případě pětinásobné zpomalení (0.2) \*/

        /\* https://p5js.org/reference/#/p5.SoundFile/rate \*/

        impactAudio.rate(0.2);

      /\* Do pole fires přidá souřadnice a velikost trefené budovy a aktuální číslo snímku (frameCount) \*/

        fires.push({x: building.x, y: building.y, w: building.w, h: building.h, frame: frameCount});

        /\* Odstraní se z pole buildings kontrolovaný objekt \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids kontrolovaný objekt \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objektem kosmické lodi (spaceShip) \*/

    if (collideRectRect(

      asteroid.x,

      asteroid.y,

      asteroid.size,

      asteroid.size,

      spaceShip.x - spaceShip.w / 2,

      spaceShip.y - spaceShip.h / 2,

      spaceShip.w,

      spaceShip.h

    ))

    {

      /\* Když došlo ke kolizi, navýší se hodnota poškození lodi o polovinu velikosti asteroidu \*/

      /\* Nastavení síly zvuku v závisloti na velikosti asteroidu \*/

      damageAudio.setVolume((1 - 10 / asteroid.size) \* 0.5, 0);

      /\* Přehrání zvuku \*/

      damageAudio.play();

      /\* Zpomalení přehrávání zvuku na polovinu \*/

      damageAudio.rate(0.5);

      /\* proměnná damage bude navýšena o velikost asteroidu \*/

      damage += asteroid.size;

      /\* Odstranění asteroidu, který zasáhl kosmickou loď \*/

      arr.splice(idx, 1);

    }

  });

  /\* Vykreslení všech budov \*/

  buildings.forEach(function(building, idx, arr) {

    building.draw();

  });

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

  /\* Aktualizace pozic budov při změně velikosti obrazovky \*/

  buildings.forEach(function(building) {

    building.y = height - building.h;

  });

}

## Stavový řádek a vyhodnocení hry (main-10.js)

/\* Globální proměnné a konstanty \*/

/\* Proměnná pro uložení objektu kosmické lodi \*/

let spaceShip;

/\* Proměnné pro uložení rastrových obrázků \*/

let shipImage, blastImage, asteroidImage;

/\* Proměnná typu pole pro uložení objektů střel vypouštěných z kosmické lodi \*/

let missiles = [];

/\* Proměnná typu pole pro uložení objektů asteroidů \*/

let asteroids = [];

/\* Proměnná typu pole pro uložení objektů budov \*/

let buildings = [];

/\* Ohnivá místa \*/

let fires = [];

/\* Proměnná hits - počet sestřelených asteroidů \*/

let hits = 0;

/\* Proměnná damage - míra poškození lodi \*/

let damage = 0;

/\* Nastavení proměnných pro zvukové soubory \*/

/\* https://p5js.org/reference/#/libraries/p5.sound \*/

let impactAudio;

let hitAudio;

let damageAudio;

/\* Vytvoření objektu pro zvukovou syntézu \*/

/\* https://p5js.org/reference/#/p5.MonoSynth \*/

let synth = new p5.MonoSynth();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída SpaceShip - návrh kosmické lodi \*/

class SpaceShip {

  /\* Konstruktor - speciální metoda, která slouží k inicializaci objektu (pomocí klíčového slova new). \*/

  constructor(posX, posY) {

    /\* Souřadnice pozice kosmické lodi - střed obdélníku \*/

    this.x = posX;

    this.y = posY;

    /\* Šířka lodi \*/

    this.w = 40;

    /\* Výška lodi \*/

    this.h = 80;

    /\* Rychlost pohybu lodi (v pixelech) \*/

    this.v = 5;

    /\* Úhel rotace lodi (ve stupních) \*/

    this.angle = 0;

  }

  /\* Metoda zjišťuje změnu polohy lodi podle stisku kláves \*/

  /\* https://p5js.org/reference/#/p5/keyCode \*/

  move() {

    /\* Reakce na stisk kurzorových kláves \*/

    /\* Podmínky ověřují, zda je poloha obdélníku lodi uvnitř hranic obrazovky; \*/

    /\* jestliže ano, může být loď posunuta o počet pixelů nastavených v atributu this.v \*/

    if (keyIsDown(LEFT\_ARROW)) {

      if (this.x > this.w / 2) this.x -= this.v;

    }

    if (keyIsDown(RIGHT\_ARROW)) {

      if (this.x < width - this.w / 2) this.x += this.v;

    }

    if (keyIsDown(UP\_ARROW)) {

      if (this.y > this.h / 2) this.y -= this.v;

    }

    if (keyIsDown(DOWN\_ARROW)) {

      if (this.y < height - this.h / 2) this.y += this.v;

    }

  }

  /\* Metoda zjišťuje změnu úhlu natočení lodi podle stisku kláves PageDown a PageUp \*/

  rotate() {

    /\* Byla-li stisknuta klávesa PageDown (kód 33), loď se pootočí o 10° proti směru hodinových ručiček \*/

    if (keyIsDown(33)) {

      this.angle -= 10;

    }

    /\* Byla-li stisknuta klávesa PageUp (kód 34), loď se pootočí o 10° ve směru hodinových ručiček \*/

    if (keyIsDown(34)) {

      this.angle += 10;

    }

  }

  /\* Metoda provede vypuštění střel po stisku klávesy Space (mezerník) - kód 32 \*/

  fire() {

    /\* Jestliže hráč stiskl klávesu mezerník (kód 32) \*/

    if (keyIsDown(32)) {

      /\* Nastavuje kadenci vypouštění střel - každý pátý snímek (frame) \*/

      if (frameCount % 5 == 0)

        /\* Do pole missiles se přidá nový prvek - objekt střely \*/

        /\* Počáteční bod střely je nastaven na střed kosmické lodi \*/

        /\* Úhel výstřelu vychází z úhlu, pod kterým je natočena kosmická loď - 90° \*/

        missiles.push(new Missile(this.x, this.y, this.angle - 90));

        synth.play('C4', 0.6, 0, 0.1);

      }

  }

  /\* Metoda provádí vykreslení kosmické lodi \*/

  draw() {

    /\* Nejprve je vyvolána metoda move(), aby se ověřil možný pohyb lodi \*/

    this.move();

    /\* Vyvolána metoda rotate() kvůli ověření rotace lodi \*/

    this.rotate();

    /\* Kontrola, zda došlo k vypuštění nějaké střely z lodi \*/

    this.fire();

    /\* Nastavení barvy výplně: https://p5js.org/reference/#/p5/fill \*/

    fill('#ff0000');

    /\* Nastavení pozicování obdélníku: https://p5js.org/reference/#/p5/rectMode \*/

    /\* V tomto případě místo standardní předvolby CORNER (levý horní roh) použijeme CENTER (střed) \*/

    rectMode(CENTER);

    /\* Metodou push() se zahajuje transformace objektu: uloží se aktuální stav kreslicího režimu \*/

    /\* https://p5js.org/reference/#/p5/push \*/

    push();

    /\* Metodou translate() se přemístí střed transformace do určitého bodu na plátně. \*/

    /\* V našem případě potřebujeme otáčet lodí přesně v jejím středu. \*/

    /\* viz https://p5js.org/reference/#/p5/translate \*/

    translate(this.x, this.y);

    /\* Provedeme rotaci o určený úhel, který musíme převést na radiány \*/

    /\* viz https://p5js.org/reference/#/p5/rotate \*/

    rotate(((2 \* PI) / 360) \* this.angle);

    /\* Vykreslení obdélníku: https://p5js.org/reference/#/p5/rect \*/

    /\* Kvůli transformaci vykreslujeme obdélník na souřadnicích 0,0 \*/

    rect(0, 0, this.w, this.h, 10, 10);

    /\* Vykreslí animovaný blast.gif ve spodní části kosmické lodi \*/

    image(blastImage, -20, 20, 40, 40);

    /\* Vykreslí obrázek kosmické lodi od bodu -20, -40 vůči středu obdélníku \*/

    image(shipImage, -20, -40);

    /\* Metoda pop() vrací kreslicí režim do původního stavu - uloženého metodou push() \*/

    /\* https://p5js.org/reference/#/p5/pop \*/

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Missile - návrh střely \*/

class Missile {

  /\* Při vytváření objektu střely předáme souřadnice jejího středu a úhel \*/

  constructor(x, y, angle) {

    /\* Pozicové souřadnice - střed střely \*/

    this.y = y;

    this.x = x;

    /\* Úhel střely \*/

    this.angle = angle;

    /\* Velikost střely \*/

    this.size = 6;

    /\* Rychlost střely \*/

    this.speed = 10;

  }

  /\* Metoda move() vypočítává posun pozice střely při zadané rychlosti a pod daným úhlem \*/

  move() {

    /\* https://www.matweb.cz/goniometrie/ \*/

    this.x += this.speed \* Math.cos((this.angle \* Math.PI) / 180);

    this.y += this.speed \* Math.sin((this.angle \* Math.PI) / 180);

  }

  /\* Metoda draw() vykresluje střelu jako kruh \*/

  draw() {

    /\* Nejprve se provede aktualizace pozice \*/

    this.move();

    /\* Nastaví bílý obrys a žlutou výplň \*/

    stroke(255);

    fill('yellow');

    /\* Vykreslí se kruh se středem v bodě this.x, this.y a o průměru this.size \*/

    /\* https://p5js.org/reference/#/p5/circle \*/

    circle(this.x, this.y, this.size);

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Asteroid - návrh šutrů padajících z nebes \*/

class Asteroid {

  constructor() {

    /\* Náhodná velikost v rozmezí 20 až 50 \*/

    this.size = random(20, 50);

    /\* Počáteční souřadnice y mimo horní okraj \*/

    this.y = -100;

    /\* Počáteční souřadnice x - náhodná v rozmezí od 0 po celou šířku plátna - velikost asteroidu\*/

    this.x = random(0, width - this.size);

    this.speed = 3;

    /\* Náhodný úhel otočení v rozmezí 0 až 359 \*/

    this.angle = random(0, 359);

  }

  /\* Metoda vypočítá pohyb asteroidu v ose y \*/

  move() {

    this.y += this.speed;

  }

  /\* Metoda vypočítá změnu úhlu vlivem rotace \*/

  rotate() {

    this.angle += this.speed;

  }

  /\* Metoda provádí vykreslení asteroidu \*/

  draw() {

    this.move();

    this.rotate();

    push();

    translate(this.x + this.size/2, this.y + this.size/2);

    rotate(((2 \* PI) / 360) \* this.angle);

    image(asteroidImage, -this.size/2, -this.size/2, this.size, this.size);

    pop();

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Třída Building - návrh symbolického objektu budovy \*/

class Building {

  constructor() {

    /\* Náhodná šířka a výška budovy v rozsahu 20 až 100 \*/

    /\* https://p5js.org/reference/#/p5/random \*/

    this.w = random(20, 100);

    this.h = random(20, 200);

    /\* Náhodný výběr umístění objektu budovy v ose x v rozsahu celé šířky obrazovky \*/

    this.x = random(0, width - this.w);

    /\* Nastavení y-souřadnice budovy tak, aby její spodní okraj ležel na spodním okraji obrazovky \*/

    this.y = height - this.h;

  }

  /\* Vykreslení objektu budovy \*/

  draw() {

    stroke(200);

    fill(color(30));

    rectMode(CORNER);

    rect(this.x, this.y, this.w, this.h);

    /\* Vykreslení rastru čtverců v ploše budovy \*/

    for (let xb = 0; xb < this.w; xb+=10) {

      for (let yb = 0; yb < this.h; yb+=10) {

        /\* Náhodné střídání tmavých a žlutých čtverců \*/

        if (random(5) > 2)

          fill(color(50));

        else

          fill(color(150, 150, 0));

        square(this.x + xb, this.y + yb, 5);

      }

    }

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Vykresluje stavový řádek s průběžnými informacemi o stavu hry \*/

function statusBar() {

  /\* Nastaví se poloprůhledná černá výplň \*/

  fill(color(0, 0, 0, 127));

  /\* Vykreslí se obdélník o výšce 40 pixelů nad spodním okrajem obrazovky \*/

  rect(0, height - 40, width, 40);

  /\* Tloušťka obrysu \*/

  strokeWeight(0);

  /\* Výška písma \*/

  textSize(20);

  /\* Řez písma - tučné \*/

  textStyle(BOLD);

  /\* Barva výplně - světle šedá \*/

  fill(color(200));

  /\* Výpis počtu zasažených asteroidů \*/

  text(`Hits: ${hits}`, 100, height - 15);

  /\* Výpis úrovně poškození kosmické lodi \*/

  text(`Spaceship damage: ${round(damage) < 100 ? round(damage) : 100} %`, 250, height - 15);

  /\* Procentuální vyjádření stojících (nezasažených) budov \*/

  text(`Buildings: ${round(buildings.length/50 \* 100)} %`, 600, height - 15);

  /\* Čas hry v sekundách a setinách sekundy \*/

  text(`Time: ${round(millis()/1000)}.${round(millis() % 100)}`, 850, height - 15);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Funkce preload() je provedena před načtením aplikace. \*/

/\* Jde o tzv. asynchronní funkci; až po jejím dokončení je provedena funkce setup(). \*/

/\* Měla by sloužit výhradně k načítání externích zdrojů - tj. obrázků, fontů nebo datových souborů. \*/

/\* https://p5js.org/reference/#/p5/preload. \*/

function preload() {

  /\* Načtení grafických souborů do proměnných \*/

  /\* https://p5js.org/reference/#/p5/loadImage \*/

  shipImage = loadImage("img/spaceship.png");

  blastImage = loadImage("img/blast.gif");

  /\* Načtení obrázku asteroidu \*/

  asteroidImage = loadImage("img/asteroid.png");

  /\* Výběr používaných zvukových formátů \*/

  soundFormats('mp3', 'ogg');

  /\* Načtení zvukových souborů \*/

  /\* https://p5js.org/reference/#/p5.SoundFile \*/

  impactAudio = loadSound('sounds/big-impact');

  hitAudio = loadSound('sounds/hit-sound');

  damageAudio = loadSound('sounds/electronic-impact');

}

/\* Funkce pro základní nastavení aplikace v P5 JS \*/

function setup() {

  /\* Vytvoří plátno podle velikosti displeje: https://p5js.org/reference/#/p5/createCanvas \*/

  canvas = createCanvas(displayWidth, displayHeight);

  /\* Vytvoření objektu kosmické lodi pomocí konstruktoru třídy SpaceShip \*/

  /\* Souřadnice středu kosmické lodi budou nastaveny na střed obrazovky \*/

  spaceShip = new SpaceShip(width / 2, height / 2);

  /\* Vygenerování 50 objektů budov \*/

  for (i = 0; i < 50; i++) {

    /\* Vložení nového objektu budovy do pole buildings \*/

    buildings.push(new Building());

  }

}

/\* Funkce pro vykreslení plátna \*/

function draw() {

  /\* Nastaví černou barvu pozadí: https://p5js.org/reference/#/p5/background \*/

  background(0);

  /\* Vykreslení ohňů na místě kolizí objektů \*/

  fires.forEach(function(fire, idx, arr) {

    image(blastImage, fire.x, fire.y, fire.w, fire.h);

    /\* Po 30 snímcích bude informace o hořícím místě odstraněna z pole \*/

    if (frameCount - fire.frame > 30) {

      arr.splice(idx, 1);

    }

  });

  /\* Vyvolání metody draw() provede vykreslení objektu kosmické lodi \*/

  spaceShip.draw();

  /\* Vykreslí všechny objekty střel uložené v poli missiles \*/

  /\* missile = aktuálně načtený prvek-objekt pole \*/

  /\* idx = index aktuálně načteného prvku \*/

  /\* arr = proměnná zastupující celé pole - v tomto případě missiles \*/

  /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach \*/

  missiles.forEach((missile, idx, arr) => {

    /\* Střela se vykreslí s použitím metody draw() \*/

    missile.draw();

    /\* Ověření, zda souřadnice středu střely nejsou mimo zobrazovanou oblast (0, 0, canvas.width, canvas.height) \*/

    if (

      missile.y > height ||

      missile.y < 0 ||

      missile.x < 0 ||

      missile.x > width

    ) {

      /\* Pokud je střela mimo oblast plátna, provede se její odstranění z pole \*/

      /\* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/splice \*/

      arr.splice(idx, 1);

    }

  });

  /\* Každý šedesátý snímek  se vytvoří nový objekt asteroidu \*/

  if (frameCount % 60 == 0) {

    /\* Nový objekt asteroidu bude přidán do pole asteroids \*/

    asteroids.push(new Asteroid());

  }

  /\* Vykresluje postupně všechny objekty z pole asteroids na plátno \*/

  asteroids.forEach(function (asteroid, idx, arr) {

    asteroid.draw();

    /\* Jestliže je y-pozice objektu asteroidu větší než výška plátna, je objekt z pole odstraněn \*/

    if (asteroid.y > height) {

      arr.splice(idx, 1);

    }

    /\* Detekce kolize vykreslovaného asteroidu se střelami (missiles) \*/

    /\* Pro detekci kolizí využijeme externí knihovnu p5.collide2D (soubor p5.collide2d.min.js): \*/

    /\* https://github.com/bmoren/p5.collide2D \*/

    missiles.forEach(function(missile, index, array) {

      /\* Funkce collideRectCircle() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) se střelou  (kruh - circle) \*/

      if (collideRectCircle(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        missile.x,

        missile.y,

        missile.size / 2

      ))

      {

        /\* Když funkce vrátí hodnotu true: \*/

        /\* Nastavení síly zvuku - rozmezí 0 až 1 \*/

        hitAudio.setVolume(0.5);

        /\* Přehrání zvuku \*/

        hitAudio.play();

        /\* Navýší se počet zásahů (hits) \*/

        hits++;

        /\* Do pole fires přidá souřadnice a velikost trefeného asteroidu a aktuální číslo snímku (frameCount) \*/

        fires.push({x: asteroid.x, y: asteroid.y, w: asteroid.size, h: asteroid.size, frame: frameCount});

        /\* Odstraní se z pole missiles kontrolovaná střela \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids zasažený asteroid \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objekty budov (buildings) \*/

    buildings.forEach(function(building, index, array) {

      /\* Funkce collideRectRect() zjišťuje, zda došlo ke kolizi asteroidu \*/

      /\* (obdélník - rectangle) s budovou (obdélník - rectangle) \*/

      if (collideRectRect(

        asteroid.x,

        asteroid.y,

        asteroid.size,

        asteroid.size,

        building.x,

        building.y,

        building.w,

        building.h

      ))

      {

        /\* Když došlo ke kolizi: \*/

        /\* Nastavení síly zvuku pro efekt dopadu na budovu - síla zvuku je nastavena podle velikosti asteroidu \*/

        /\* https://p5js.org/reference/#/p5.SoundFile/setVolume \*/

        impactAudio.setVolume((1 - 10 / asteroid.size) \* 0.5, 0);

        impactAudio.play();

        /\* Nastavení rychlosti přehrávání zvuku - v tomto případě pětinásobné zpomalení (0.2) \*/

        /\* https://p5js.org/reference/#/p5.SoundFile/rate \*/

        impactAudio.rate(0.2);

        /\* Do pole fires přidá souřadnice a velikost trefené budovy a aktuální číslo snímku (frameCount) \*/

        fires.push({x: building.x, y: building.y, w: building.w, h: building.h, frame: frameCount});

        /\* Odstraní se z pole buildings kontrolovaný objekt \*/

        array.splice(index, 1);

        /\* Odstraní se z pole asteroids kontrolovaný objekt \*/

        arr.splice(idx, 1);

      }

    });

    /\* Detekce kolize kontrolovaného asteroidu s objektem kosmické lodi (spaceShip) \*/

    if (collideRectRect(

      asteroid.x,

      asteroid.y,

      asteroid.size,

      asteroid.size,

      spaceShip.x - spaceShip.w / 2,

      spaceShip.y - spaceShip.h / 2,

      spaceShip.w,

      spaceShip.h

    ))

    {

      /\* Když došlo ke kolizi, navýší se hodnota poškození lodi o polovinu velikosti asteroidu \*/

      /\* Nastavení síly zvuku v závisloti na velikosti asteroidu \*/

      damageAudio.setVolume((1 - 10 / asteroid.size) \* 0.5, 0);

      /\* Přehrání zvuku \*/

      damageAudio.play();

      /\* Zpomalení přehrávání zvuku na polovinu \*/

      damageAudio.rate(0.5);

      /\* proměnná damage bude navýšena o velikost asteroidu \*/

      damage += asteroid.size;

      /\* Odstranění asteroidu, který zasáhl kosmickou loď \*/

      arr.splice(idx, 1);

    }

  });

  /\* Vykreslení všech budov \*/

  buildings.forEach(function(building, idx, arr) {

    building.draw();

  });

  /\* Kontroluje poškození lodi a procento zbylých budov \*/

  /\* Když je loď poškozena nad hodnotu 99 nebo zbylo méně než 30 % budov, bude hra ukončena \*/

  if (round(damage) > 99 || round(buildings.length/50 \* 100) < 30) {

    /\* Ukončení smyčky hry \*/

    noLoop();

    /\* Nastavení tmavě červeného pozadí s průhledností (200) \*/

    background(100, 0, 0, 200);

    textSize(50);

    fill(255, 0, 0, 200);

    text('GAME IS OVER', width/2 - 200, height/2);

    /\* Vykreslí stavový řádek \*/

    statusBar();

  } else {

    statusBar();

  }

}

/\* Funkce pro změnu velikosti plátna podle velikosti okna \*/

function windowResized() {

  /\* Změní velikost plátna podle rozměrů okna: https://p5js.org/reference/#/p5/resizeCanvas \*/

  resizeCanvas(windowWidth, windowHeight);

  /\* Aktualizace pozic budov při změně velikosti obrazovky \*/

  buildings.forEach(function(building) {

    building.y = height - building.h;

  });

}