

Pixel Pets

```
// Global Functions

function updateScreen() {
    setText("coinAmount", coins);
    setText("levelLabel", level);
    setText("petNicknameLabel", petNickname);
    if (coinToLevel != 1) {
        setText("feedCostAmountText", "Cost: " + coinToLevel + " coins");
    }
    setText("shopCoinsAmount", coins);
}

function playMusic(status, name) {
    if (status == "on") {
        playSound(name, true);
    } else if (status == "off") {
        stopSound();
    }
}

// Home Screen

playSound("Wondrous-Waters.mp3", true);

onEvent("homeNextButton", "click", function() {
    setScreen("infoScreen");
});

// Info Screen

onEvent("infoNextButton", "click", function() {
    setScreen("petSelectionScreen");
});

// Pet Selection Screen

var selectedPet;
var allPetNames = getColumn("Pet Names", "petNames");

onEvent("dogImage", "click", function() {
   setProperty("dogImage", "border-width", 3);
   setProperty("catImage", "border-width", 0);
    selectedPet = "dog";
});

onEvent("catImage", "click", function() {
   setProperty("catImage", "border-width", 3);
   setProperty("dogImage", "border-width", 0);
    selectedPet = "cat";
});
```

```
onEvent("petSelectionNextButton", "click", function() {
  if (selectedPet == "dog" || selectedPet == "cat") {
    setScreen("nicknameScreen");
  }
});

// Pet Nickname Screen

var petNickname;

onEvent("nicknameRandomButton", "click", function() {
  var randomName = allPetNames[randomNumber(0, 24)];
  setText("nicknameInput", randomName);
});

onEvent("startButton", "click", function(){
  petNickname = getText("nicknameInput");
  if (petNickname != "") {
    setScreen("mainGameScreen");
    updateScreen();
    setText("petNicknameLabel", petNickname);
    if (selectedPet == "dog") {
      setProperty("petImage", "image", "dog.png");
    } else if (selectedPet == "cat") {
      setProperty("petImage", "image", "cat.png");
    }
  }
});

// Main Game Screen

var coins = 0;
var level = 1;
var coinToLevel = 1;

onEvent("trainButton", "click", function() {
  setScreen("trainingSelectionScreen");
  resetTrainingSelectionScreen();
  randomGameGenerator();
});

onEvent("feedButton", "click", function() {
  if (coins >= coinToLevel) {
    level += 1;
    coins -= coinToLevel;
    coinToLevel += 1;
    updateScreen();
  }

  if (level >= 20) {
    setText("feedButton", "MAX");
    setProperty("feedButton", "text-color", "green");

    victory();
  }
});

if (level >= 20) {
```

```
    coinToLevel = 0;
    level = 20;
    updateScreen();
}
}

});

onEvent("settingsIcon", "click", function() {
    setScreen("settingsScreen");
});

// Victory Screen

function victory() {
    if (selectedPet == "cat") {
        setProperty("victoryPetImage", "image", "cat.png");
    } else {
        setProperty("victoryPetImage", "image", "dog.png");
    }

    setScreen("victoryScreen");
}

onEvent("congratsBackButton", "click", function() {
    setScreen("mainGameScreen");
});

// Shop Screen

onEvent("shopButton", "click", function() {
    setScreen("shopScreen");
    updateScreen();
});

onEvent("shopBackButton", "click", function() {
    setScreen("mainGameScreen");
});

onEvent("crownPurchaseButton", "click", function() {
    if (coins >= 150) {
        setProperty("crownPurchaseButton", "hidden", true);
        setProperty("crownEquipButton", "hidden", false);

        coins -= 150;
        updateScreen();
    }
});

onEvent("crownEquipButton", "click", function() {
    if (getText("crownEquipButton") == "Equip") {
        setText("crownEquipButton", "Unequip");
        if (selectedPet == "cat") {
            setPosition("crownOnPetImage", 135, 250);
        } else {
            setPosition("crownOnPetImage", 120, 230);
        }
    }
});
```

```
        setProperty("crownOnPetImage", "hidden", false);
    } else {
        setText("crownEquipButton", "Equip");
        setProperty("crownOnPetImage", "hidden", true);
    }
});

onEvent("shadesPurchaseButton", "click", function() {
    if (coins >= 200) {
        setProperty("shadesPurchaseButton", "hidden", true);
        setProperty("shadesEquipButton", "hidden", false);

        coins -= 200;
        updateScreen();
    }
});

onEvent("shadesEquipButton", "click", function() {
    if (getText("shadesEquipButton") == "Equip") {
        setText("shadesEquipButton", "Unequip");
        if (selectedPet == "cat") {
            setPosition("shadesOnPetImage", 145, 305);
        } else {
            setPosition("shadesOnPetImage", 125, 290);
        }
        setProperty("shadesOnPetImage", "hidden", false);
    } else {
        setText("crownEquipButton", "Equip");
        setProperty("shadesOnPetImage", "hidden", true);
    }
});
});

// Settings Screen

var currentAudio = "Wondrous-Waters.mp3";

var secretGameCode = "debug";

onEvent("muteButton", "click", function() {
    var muteButtonTemp = getText("muteButton");
    if (muteButtonTemp == "Mute") {
        playMusic("off");
        setText("muteButton", "Unmute");
    } else if (muteButtonTemp == "Unmute") {
        playMusic("on", currentAudio);
        setText("muteButton", "Mute");
    }
});

onEvent("gameMusic1", "click", function() {
    stopSound();
    playMusic("on", "Wondrous-Waters.mp3");
    currentAudio = "Wondrous-Waters.mp3";
});
onEvent("gameMusic2", "click", function() {
    stopSound();
```

```

    playMusic("on", "8-Bit-Dream-Land.mp3");
    currentAudio = "8-Bit-Dream-Land.mp3";
});
onEvent("gameMusic3", "click", function() {
    stopSound();
    playMusic("on", "Beam.mp3");
    currentAudio = "Beam.mp3";
});

function resetNicknameChangeInput() {
    setText("applyButton", "Apply");
    setProperty("applyButton", "text-color", "white");
}

onEvent("changeNicknameInput", "input", function() {
    resetNicknameChangeInput();
});

onEvent("applyButton", "click", function() {
    var nicknameChange = getText("changeNicknameInput");
    if (nicknameChange != "") {
        if (nicknameChange == secretGameCode) {
            coins += 10000;
            setProperty("applyButton", "font-size", 20);
            setText("applyButton", "Secret Code Activated");
            setProperty("applyButton", "text-color", "#adf1ab");
            setProperty("changeNicknameInput", "text", "");
        } else {
            petNickname = nicknameChange;
            setText("applyButton", "Applied!");
            setProperty("applyButton", "text-color", "#adf1ab");
            setProperty("changeNicknameInput", "text", "");
        }
    }
});
};

onEvent("settingsGoBackButton", "click", function() {
    setScreen("mainGameScreen");
    resetNicknameChangeInput();

    updateScreen();
});

// Training Selection Screen

var allGameNames = getColumn("MiniGames", "gameNames");
var allGameIcons = getColumn("MiniGames", "gameIcons");

onEvent("randomGameButton", "click", function() {
    var selectedMinigame = getText("randomGameButton");
    if (selectedMinigame == "Balloon Burst") {
        setScreen("balloonBurstGameHTP");
    } else if (selectedMinigame == "Memory Mania") {
        setScreen("memoryManiaGameHTP");
    } else if (selectedMinigame == "Digit Dash") {
        setScreen("digitDashGameHTP");
    }
});

```

```

        }

    });

    function randomGameGenerator() {
        var randomGame = randomNumber(0, 2);
        var randomTempVal = 0;
        var selectedRandomGame = randomNumber(15, 17);
        timedLoop(300, function() {
            randomTempVal += 1;
            randomGame += 1;
            if (randomGame == 3) {
                randomGame = 0;
            }
           setProperty("randomGameImage", "hidden", false);
           setProperty("randomGameImage", "image", allGameIcons[randomGame]);
           setText("randomGameButton", allGameNames[randomGame]);
            if (randomGame == 0) {
                setText("generatingGameText", "Generating Random Game.");
            } else if (randomGame == 1) {
                setText("generatingGameText", "Generating Random Game..");
            } else if (randomGame == 2) {
                setText("generatingGameText", "Generating Random Game...");
            }
        });

        if (randomTempVal == selectedRandomGame) {
            stopTimedLoop();
            setText("generatingGameText", "A Game has been Selected!");
           setProperty("backgroundText1", "background-color", "#a4e6a2");
           setProperty("imagesBackground", "background-color", "#a4e6a2");
           setProperty("randomGameButton", "background-color", "#a4e6a2");
           setProperty("randomGameButton", "hidden", false);
        }
    });
}

function resetTrainingSelectionScreen() {
    setText("generatingGameText", "Generating Random Game...");
   setProperty("backgroundText1", "background-color", "rgba(234, 148, 148, 0.86)");
   setProperty("imagesBackground", "background-color", "rgb(222, 129, 129)");
   setProperty("randomGameButton", "hidden", true);
}

// Balloon Burst Game HTP Screen

onEvent("balloonGameHTPStartButton", "click", function() {
    setScreen("balloonBurstGame");
    resetBalloonGame();
    balloonGameStart();
});

// Balloon Burst Game

var tempScore = 0;
var listenersAdded = false;
var gameSession = 0;
var timeOnTimer;

```

```
var balloonSize = 80;
var bbGameTotalEarnings;

onEvent("balloonGameNextButton", "click", function() {
    balloonGameStart();
});

function countdownTimerBBGame(id, speed) {
    timedLoop(speed, function() {
        var timer = getText(id);
        if (timer != 0) {
            timer = timer - 1;
            setText(id, timer);
        } else {
            balloonGameEnd(4);
            stopTimedLoop();
        }
    });
}

function balloonGameRandomizer() {
    balloonSize -= gameSession * 2;
    for (var i = 1; i < 21; i++) {
        setProperty("balloon" + i, "hidden", false);
        setProperty("balloon" + i, "width", balloonSize);
        setProperty("balloon" + i, "height", balloonSize);
        setProperty("balloon" + i, "x", randomNumber(10, 230));
        setProperty("balloon" + i, "y", randomNumber(90, 350));
    }
}

function balloonGameScoreTracker() {
    if (listenersAdded == false) {
        for (var i = 1; i < 21; i++) {
            onEvent("balloon" + i, "click", balloonGameEventHandler(i));
        }
        listenersAdded = true;
    }
}

function balloonGameEventHandler(index) {
    return function() {
        setProperty("balloon" + index, "hidden", true);
        tempScore += 1;
        if (tempScore == 20) {
            balloonGamePause();
        }
    };
}

function balloonGamePause() {
    stopTimedLoop();
    setProperty("balloonGameCongratsLabel", "hidden", false);
    setProperty("balloonGameBackgroundText", "hidden", false);
    setProperty("balloonGameNextButton", "hidden", false);
}
```

```
function balloonGameStart() {
    gameSession += 1;
    timeOnTimer = 10;
    tempScore = 0;
    setProperty("balloonGameNextButton", "hidden", true);
    setProperty("balloonGameCongratsLabel", "hidden", true);
    setProperty("balloonGameBackgroundText", "hidden", true);
    balloonGameRandomizer(80);
    setText("timer", timeOnTimer);
    countdownTimerBBGame("timer", 1000);
    balloonGameScoreTracker();
}

function balloonGameEnd(earningsMultiplier) {
    for (var i = 1; i < 21; i++) {
        setProperty("balloon" + i, "hidden", true);
    }
    setProperty("balloonGameFailLabel", "hidden", false);
    bbGameTotalEarnings = gameSession * earningsMultiplier;

    if (bbGameTotalEarnings == 1) {
        setText("balloonGameEarningsLabel", "You gained " + bbGameTotalEarnings + " coin!");
    } else {
        setText("balloonGameEarningsLabel", "You gained " + bbGameTotalEarnings + " coins!");
    }

    coins += bbGameTotalEarnings;
    setProperty("balloonGameEarningsLabel", "hidden", false);
    setProperty("balloonGameBackgroundText", "hidden", false);
    setProperty("balloonGameBackButton", "hidden", false);
    onEvent("balloonGameBackButton", "click", function() {
        setScreen("mainGameScreen");
        updateScreen();
    });
}

function resetBalloonGame() {
    gameSession = 0;
    tempScore = 0;
    bbGameTotalEarnings = 0;

    setProperty("balloonGameFailLabel", "hidden", true);
    setProperty("balloonGameEarningsLabel", "hidden", true);
    setProperty("balloonGameBackButton", "hidden", true);
    setProperty("balloonGameBackgroundText", "hidden", true);
}

// Memory Mania Game HTP Screen

onEvent("memoryManiaHTPStartButton", "click", function() {
    setScreen("memoryManiaGame");
    mmGameStart();
    countdownTimerMMGame("memoryManiaGameCountdown", 750);
});
```

```
// Memory Mania Game

function countdownTimerMMGame(id, speed) {

    stopTimedLoop();

    setProperty("memoryManiaGameCountdown", "hidden", false);
    setText("memoryManiaGameCountdown", "3");
    timedLoop(speed, function() {
        var timer = getText(id);
        if (timer != 0) {
            timer = timer - 1;
            setText(id, timer);
        } else {
            stopTimedLoop();
            mmGameSequenceGenerator();
        }
    });
}

//var mmGameFruits = getColumn("Memory Mania Icons", "fruitName");
var mmGameFruits = ["Apple", "Orange", "Banana"];
//var mmGameIcons = getColumn("Memory Mania Icons", "fruitIcon");
var mmGameIcons = ["apple.png", "orange.png", "banana.png"];
var randomFruitIndex;
var randomFruit;
var randomFruitIcon;
var sequenceGeneratorTempValue;
var sequenceAmount;
var sequenceAmountTotal;
var correctAnswer;

function mmGameStart() {
    setText("memoryManiaGameTitle", "Memorize the following sequence:");
    setProperty("memoryManiaGameTitle", "font-size", 26);

    setProperty("memoryManiaGameCountdown", "hidden", false);
    setProperty("memoryManiaGameBGIcons", "hidden", false);

    setProperty("mmGameEarningsLabel", "hidden", true);
    setProperty("mmGameEarningsLabelBG", "hidden", true);
    setProperty("mmGameExitButton", "hidden", true);

    sequenceAmount = 2;
    sequenceAmountTotal = 4;
}

function mmGameSequenceGenerator() {
    setProperty("memoryManiaGameCountdown", "hidden", true);
    setProperty("memoryManiaGameIcons", "hidden", false);

    sequenceAmountTotal += 2;
    sequenceAmount += 1;
    sequenceGeneratorTempValue = 0;

    correctAnswer = [];
}
```

```
timedLoop(500, function() {
    randomFruitIndex = randomNumber(0, 2);
    randomFruit = mmGameFruits[randomFruitIndex];
    randomFruitIcon = mmGameIcons[randomFruitIndex];

    sequenceGeneratorTempValue += 1;

    if (sequenceGeneratorTempValue % 2 != 0) {
        setProperty("memoryManiaGameIcons", "image", randomFruitIcon);
        appendItem(correctAnswer, randomFruit);
    } else {
        setProperty("memoryManiaGameIcons", "image", "");
    }

    if (sequenceGeneratorTempValue == sequenceAmountTotal) {
        stopTimedLoop();
        setProperty("mmGameInputButton", "hidden", false);

        onEvent("mmGameInputButton", "click", function() {
            mmGameGuessStage();
        });
    }
});

function setUpGuessStage() {
    setProperty("memoryManiaGameIcons", "hidden", true);
    setProperty("memoryManiaGameBGIcons", "hidden", true);
    setProperty("mmGameInputButton", "hidden", true);
    setProperty("memoryManiaGameTitle", "text", "Input the Sequence Below:");
    setProperty("appleIcon", "hidden", false);
    setProperty("appleIconBGColor", "hidden", false);
    setProperty("bananaIcon", "hidden", false);
    setProperty("bananaIconBGColor", "hidden", false);
    setProperty("orangeIcon", "hidden", false);
    setProperty("orangeIconBGColor", "hidden", false);
}

var userGuess;
var userGuessIndexValue;
var eventListenersAdded = false;
var isGuessingAllowed;

function mmGameGuessStage() {

    userGuess = [];
    userGuessIndexValue = -1;
    isGuessingAllowed = true;

    setUpGuessStage();

    if (!eventListenersAdded) {
        iconEventListeners();
    }
}
```

```
eventListenersAdded = true;
}

var currentGuessedFruitBG;

function iconEventListeners() {
    onEvent("appleIcon", "click", function() {
        if (isGuessingAllowed == true) {
            appendItem(userGuess, "Apple");
            userGuessIndexValue += 1;
            currentGuessedFruitBG = "appleIconBGColor";
            checkUserGuess();
        }
    });
    onEvent("bananaIcon", "click", function() {
        if (isGuessingAllowed == true) {
            appendItem(userGuess, "Banana");
            userGuessIndexValue += 1;
            currentGuessedFruitBG = "bananaIconBGColor";
            checkUserGuess();
        }
    });
    onEvent("orangeIcon", "click", function() {
        if (isGuessingAllowed == true) {
            appendItem(userGuess, "Orange");
            userGuessIndexValue += 1;
            currentGuessedFruitBG = "orangeIconBGColor";
            checkUserGuess();
        }
    });
}

function checkUserGuess() {
    if (userGuess[userGuessIndexValue] != correctAnswer[userGuessIndexValue]) {
        mmGameEnd();
    } else {
        setProperty(currentGuessedFruitBG, "background-color", rgb(173, 236, 118));
        timedLoop(250, function() {
            if (userGuessIndexValue != sequenceAmount - 1) {
                setProperty(currentGuessedFruitBG, "background-color", rgb(238, 132, 132));
                stopTimedLoop();
            }
        });
    }
    if (userGuessIndexValue == sequenceAmount - 1) {
        if (userGuess[userGuessIndexValue] != correctAnswer[userGuessIndexValue]) {
            mmGameEnd();
        } else {
            mmGameCorrectPhase();

            onEvent("mmGameNextLvlButton", "click", function() {
                mmGameNextLevel();
            });
        }
    }
}
```

```
function mmGameCorrectPhase() {
    setText("memoryManiaGameTitle", "Correct!");
    setProperty("memoryManiaGameTitle", "font-size", 45);
    setProperty("mmGameNextLvlButton", "hidden", false);

    isGuessingAllowed = false;

    setProperty("appleIconBGColor", "background-color", rgb(173, 236, 118));
    setProperty("bananaIconBGColor", "background-color", rgb(173, 236, 118));
    setProperty("orangeIconBGColor", "background-color", rgb(173, 236, 118));
}

function mmGameEnd() {
    setText("memoryManiaGameTitle", "Game Over");
    setProperty("memoryManiaGameTitle", "font-size", 46);
    setProperty("appleIcon", "hidden", true);
    setProperty("appleIconBGColor", "hidden", true);
    setProperty("bananaIcon", "hidden", true);
    setProperty("bananaIconBGColor", "hidden", true);
    setProperty("orangeIcon", "hidden", true);
    setProperty("orangeIconBGColor", "hidden", true);

    var mmGameEarningsAmount = sequenceAmount * 4;

    if (sequenceAmount == 3) {
        mmGameEarningsAmount = 0;
        setText("mmGameEarningsLabel", "You gained " + mmGameEarningsAmount + " coins for not guessing any sequence");
    } else {
        setText("mmGameEarningsLabel", "You gained " + mmGameEarningsAmount + " coins for memorizing your last sequence");
    }

    coins += mmGameEarningsAmount;

    setProperty("mmGameEarningsLabel", "hidden", false);
    setProperty("mmGameEarningsLabelBG", "hidden", false);
    setProperty("mmGameExitButton", "hidden", false);

    onEvent("mmGameExitButton", "click", function() {
        setScreen("mainGameScreen");
        updateScreen();
    });
}

function mmGameNextLevel() {
    setProperty("appleIcon", "hidden", true);
    setProperty("appleIconBGColor", "hidden", true);
    setProperty("appleIconBGColor", "background-color", rgb(238, 132, 132));

    setProperty("bananaIcon", "hidden", true);
    setProperty("bananaIconBGColor", "hidden", true);
    setProperty("bananaIconBGColor", "background-color", rgb(238, 132, 132));

    setProperty("orangeIcon", "hidden", true);
    setProperty("orangeIconBGColor", "hidden", true);
    setProperty("orangeIconBGColor", "background-color", rgb(238, 132, 132));
}
```

```
setProperty("mmGameNextLvlButton", "hidden", true);
setText("memoryManiaGameTitle", "Memorize the following sequence:");
setProperty("memoryManiaGameTitle", "font-size", 26);
setProperty("memoryManiaGameCountdown", "hidden", false);
setProperty("memoryManiaGameBGIcons", "hidden", false);

countdownTimerMMGame("memoryManiaGameCountdown", 750);
}

// Digit Dash HTP Screen

onEvent("digitDashHTPButton", "click", function() {
    digitDashGameReset();
    digitDashGameGenerateProblem();
    digitDashGameSetText();
    ddGameCountdown("digitDashGameCountdownText", 1000);
    setScreen("digitDashGame");
});

// Digit Dash Game

var operators = ["+", "-", "x"];

var digitDashGameNum1;
var digitDashGameNum2;
var digitDashGameOperator;
var digitDashGameAnswer;

var tempOffsetValue;
var digitDashGameWrongAnswer;

var correctAnswerButton;
var incorrectAnswerButton;
var tempRandomNum;

var ddGameScore = 0;

var ddGameEventListenersActive = false;

function ddGameCountdown(id, speed) {
    timedLoop(speed, function() {
        var timer = getText(id);
        if (timer != 0) {
            timer = timer - 1;
            setText(id, timer);
        } else {
            digitDashGameEnd();
            stopTimedLoop();
        }
    });
}

function digitDashGameReset() {
    setProperty("digitDashGameNum1", "hidden", false);
```

```
setProperty("digitDashGameNum2", "hidden", false);
setProperty("digitDashGameOperator", "hidden", false);
setProperty("digitDashGameProblemBGColor", "hidden", false);
setProperty("digitDashGameAnswer1", "hidden", false);
setProperty("digitDashGameAnswer2", "hidden", false);

setProperty("ddGameOverText", "hidden", true);
setProperty("ddGameOverTextBG", "hidden", true);
setProperty("ddGameOverEarningsText", "hidden", true);
setProperty("ddGameOverEarningsBG", "hidden", true);
setProperty("ddGameBackButton", "hidden", true);

setText("digitDashGameCountdownText", "30");

ddGameScore = 0;
ddGameEarnings = 0;
}

function digitDashGameStart() {

if (!ddGameEventListenersActive) {
    onEvent("digitDashGameAnswer1", "click", function() {
        ddGameCheckAnswer("digitDashGameAnswer1");
    });
    onEvent("digitDashGameAnswer2", "click", function() {
        ddGameCheckAnswer("digitDashGameAnswer2");
    });
    ddGameEventListenersActive = true;

    digitDashGameGenerateProblem();
    digitDashGameSetText();
}
}

digitDashGameStart();

function digitDashGameGenerateProblem() {
    digitDashGameNum1 = randomNumber(0, 10);
    digitDashGameNum2 = randomNumber(0, 10);
    digitDashGameOperator = operators[randomNumber(0, 2)];

    if (digitDashGameOperator == "+") {
        digitDashGameAnswer = digitDashGameNum1 + digitDashGameNum2;
    } else if (digitDashGameOperator == "-") {
        digitDashGameAnswer = digitDashGameNum1 - digitDashGameNum2;
    } else {
        digitDashGameAnswer = digitDashGameNum1 * digitDashGameNum2;
    }

    tempRandomNum = randomNumber(0, 1);

    if (tempRandomNum == 0) {
        correctAnswerButton = "digitDashGameAnswer1";
        incorrectAnswerButton = "digitDashGameAnswer2";
    } else {
        correctAnswerButton = "digitDashGameAnswer2";
        incorrectAnswerButton = "digitDashGameAnswer1";
    }
}
```

```
incorrectAnswerButton = "digitDashGameAnswer1";
}

tempOffsetValue = randomNumber(3, 10);
tempRandomNum = randomNumber(0, 1);

if (tempRandomNum == 0) {
    digitDashGameWrongAnswer = digitDashGameAnswer + tempOffsetValue;
} else {
    digitDashGameWrongAnswer = digitDashGameAnswer - tempOffsetValue;
}
}

function digitDashGameSetText() {
    setText("digitDashGameNum1", digitDashGameNum1);
    setText("digitDashGameNum2", digitDashGameNum2);
    setText("digitDashGameOperator", digitDashGameOperator);

    setText(correctAnswerButton, digitDashGameAnswer);
    setText(incorrectAnswerButton, digitDashGameWrongAnswer);

    setText("digitDashGameScore", ddGameScore);
}

function ddGameCheckAnswer(buttonId) {
    if (getText(buttonId) == digitDashGameAnswer) {
        ddGameScore += 1;
        digitDashGameGenerateProblem();
        digitDashGameSetText();
    } else {
        stopTimedLoop();
        digitDashGameEnd();
    }
}

var ddGameEarnings = 0;

function digitDashGameEnd() {
    setProperty("digitDashGameNum1", "hidden", true);
    setProperty("digitDashGameNum2", "hidden", true);
    setProperty("digitDashGameOperator", "hidden", true);
    setProperty("digitDashGameProblemBGColor", "hidden", true);
    setProperty("digitDashGameAnswer1", "hidden", true);
    setProperty("digitDashGameAnswer2", "hidden", true);

    ddGameEarnings = ddGameScore;
    coins += ddGameEarnings;

    setText("ddGameOverEarningsText", "You earned " + ddGameEarnings + " coins for answering " + ddGameScore);

    setProperty("ddGameOverText", "hidden", false);
    setProperty("ddGameOverTextBG", "hidden", false);
    setProperty("ddGameOverEarningsText", "hidden", false);
    setProperty("ddGameOverEarningsBG", "hidden", false);
    setProperty("ddGameBackButton", "hidden", false);
}
```

```
onEvent("ddGameBackButton", "click", function() {
    setScreen("mainGameScreen");
    updateScreen();
});

//-----
//-----Citations-----
//-----//


// Game Background Citations:

// BG 1: "Hình nền game background portrait" by Cho Máy Tính, https://xaydungso.vn/bai-viet-khac/hinh-ne
// BG 2: "Background Vector Landscape Background Stock Vector" by Design Adelsa, stock.adobe.com/7601000
// BG 3: "Decorative Summer Landscape Children Illustration Aautism Day Vector Art Wallpaper, Generative
// BG 4: "Cartoon Meadow Spring Country Meadow Landscape Background of a Springtime Green Pasture Field \n" by
// BG 5: "Green Plant Landscape with Grass and Clouds Created with Generative Ai Stock Illustration" by
// BG 6: "Summer Style Vector Illustration Border Banner Art Background with Empty Copy Space Created wi
// BG 7: "Background Vector Landscape Background Stock Vector" by Adelsa Design, stock.adobe.com/images/
// BG 8: "Nature Background Portrait 1. Realistic Fantastic Cartoon Style Artwork Scene" by info@nextmar
// BG 9: "Mountains and bright sky in the morning" by jamesjoong, https://stock.adobe.com/images/mountai

// Game Images and Icons Citations:

// Dog Image Citation:
// "Angry cute dog sticker, Dog Clipart, dog sticker, sticker png and vector with transparent background
// "Dogs stung by bees" by liljupiterr, https://www.instagram.com/liljupiterr/p/Cd7wfLWOR1c/?img\_index=1

// Cat Image Citation:
// "ReferencesPngtree" by unknown, https://pngtree.com/element/down?id=ODE4ODY3Mg==&type=1&time=17122440
// "cat with big chin" by reactjpg, https://twitter.com/reactjpg/status/1471415186766450692.

// Coin Icon Citation:
// "Stock Photos, Royalty-Free Images, Graphics, Vectors & Videos." by Adobe Stock, https://stock.adobe.com/

// Math Symbols Icon Citation:
// "RED ICON WITH MATHS SYMBOLS" by Merry Steward, https://pixy.org/4768416/.

// Gold Crown Image Citation:
// "Simple gold crown", by Staszek Szybki Jest, https://en.m.wikipedia.org/wiki/File:Simple\_gold\_crown.s

// Sunglasses Image Citation:
// "Sunglasses clipart design" by emiltimplaru, https://www.vecteezy.com/png/9303640-sunglasses-clipart-

// Balloon Images Citations:
// "New Red Balloon Clip Art" by Chuck, http://www.clker.com/clipart-new-red-balloon.html.
// "New Green Balloon Clip Art" by Ronielan Merciales, http://www.clker.com/clipart-new-green-balloon.html
// "New Blue Balloon - Light Lft Clip Art" by Chuck, http://www.clker.com/clipart-new-blue-balloon-light.html
// "Balloon Clip Art" by Christina Mountford, http://www.clker.com/clipart-balloon-7.html.
// "Light Pink Balloon Clip Art" by Melissa Sherwin, http://www.clker.com/clipart-light-pink-balloon-1.html

// Fruits Images Citations:
// Apple: "Free Vector | Sticker Design with an Apple Isolated" by brgfx, www.freepik.com/free-vector/st
// Orange: "Orange Fruit Stock Vectors and Vector Art" by AufLager, https://brentwoodental.com/Stock-Il
// Banana: "Summer Fruit Cartoon Banana" by 588ku, https://vecteezy.com/png/9303640-sunglasses-clipart-
```

```
// Music Citations:  
// "Wondrous Waters" by 29811401, pixabay.com/music/ambient-wondrous-waters-119518/.  
// "8-Bit Dream Land" by moodmode, pixabay.com/music/video-games-8-bit-dream-land-142093/.  
// "Beam" by sinneschlösen, pixabay.com/music/video-games-sinnesloschen-beam-117362/.  
  
// Generative AI:  
// I did not use any exact code generated from AI, however I did use some help with understanding my eve  
// As code.org's AppLab does not have an implemented function to remove these event handlers, I asked Ch
```

PDF document made with CodePrint using [Prism](#)