```
script Two game controller (functions)

function main ()
basic → show string("A or B or AB", 75)

// AB used to select Dodge Game
input → on button pressed(A+B) do
◳ crashed := false
◳ delay := 300
◳ car := game → create sprite(2, 4)
◳ car → set blink(200)
◳ m := game → create sprite(math → random(5), math → random(2))
◳ m2 := game → create sprite(math → random(5), math → random(2))
◳ m3 := game → create sprite(math → random(5), math
→ random(2))
game → start stopwatch
basic → forever do
        ▷ readButtons
        ▷ crashed 1
end
control → in background do
        while ◳ crashed → equals(false) do
                basic → pause(◳ delay)
                ◳ m → change y by(1)
                ▷ meteor 1
                ◳ m2 → change y by(1)
                ▷ meteor 2
                ◳ m3 → change y by(1)
                ▷ meteor 3
                ◳ delay := ◳ delay − 5
                if ◳ delay ≤ 100 then
                        ◳ delay := 100
                else add code here end if
        end while
end

end

// Invaders Game
input → on button pressed(A) do
game → set score(0)
◳ InvadersDelay := 100
◳ xInvaders := 2
◳ yInvaders := 4
◳ alienX := math → random(5)
◳ alienY := math → random(2)
game → start countdown(20000)
led → plot(◳ xInvaders, ◳ yInvaders)
led → plot(◳ alienX, ◳ alienY)
while game → current time > 0 do
        ▷ InvadersFire
        ▷ InvadersLeft
        ▷ InvadersRight
end while
```

```
basic → forever do
        if game → current time = 0 then
                var valueUp := pins → digital read pin(P12)
                if valueUp > 0 then
                        control → reset
                else add code here end if
        else add code here end if
end
end
// End of the Invaders game

// Snake Game
input → on button pressed(B) do
game → set score(0)
◳ SnakeDelay := 200
// Plot snake
◳ xSnake := 2
◳ ySnake := 2
// Snake's Tail
◳ Ox := 1
◳ Oy := 2
// food x and y Co ordinates
◳ fx := math → random(5)
◳ fy := math → random(5)
game → start countdown(30000)
led → plot(◳ Ox, ◳ Oy)
led → plot(◳ xSnake, ◳ ySnake)
led → plot(◳ fx, ◳ fy)
// Start of programme
while game → current time > 0 do
        ▷ snakeUp
        ▷ snakeRight
        ▷ snakeLeft
        ▷ snakeDown
        ▷ snakeEats
        if game → score > 10 and game → score < 20 then
                ◳ SnakeDelay := 100
        else if game → score ≥ 20 then
                ◳ SnakeDelay := 50
        else add code here end if
end while

//Waits for the player to press FIRE to reset the game
basic → forever do
        var valueReset := pins → digital read pin(P15)
        if valueReset > 0 then
                control → reset
        else add code here end if
end
end
end function

_____________________________
function InvadersFire()
```

```
var valueFire := pins → digital read pin(P15)
if valueFire > 0 then
        for 0 ≤ i < 4 do
        led → plot(◳ xInvaders, ◳ yInvaders - (i + 1))
        if ◳ alienY = i and ◳ alienX = ◳ xInvaders then
                game → set score(game → score + 1)
                if ◳ buzzOff → equals(false) then
                        music → play tone(440, 100)
                else add code here end if
        else add code here end if
        basic → pause(◳ InvadersDelay)
        led → unplot(◳ xInvaders, ◳ yInvaders - (i + 1))
        end for
◳ alienX := math → random(5)
◳ alienY := math → random(2)
led → plot(◳ alienX, ◳ alienY)
led → plot(◳ xInvaders, ◳ yInvaders)
else add code here end if
end function

_____________________________
function InvadersLeft()
var valueLeft := pins → digital read pin(P8)
if valueLeft > 0 then
        led → unplot(◳ xInvaders, ◳ yInvaders)
        ◳ xInvaders := ◳ xInvaders - 1
        if ◳ xInvaders < 0 then
                ◳ xInvaders := 0
        else add code here end if
        led → plot(◳ xInvaders, ◳ yInvaders)
        basic → pause(◳ InvadersDelay)
else add code here end if
end function

_____________________________
function InvadersRight()
var valueRight := pins → digital read pin(P16)
if valueRight > 0 then
        led → unplot(◳ xInvaders, ◳ yInvaders)
        ◳ xInvaders := ◳ xInvaders + 1
        if ◳ xInvaders > 4 then
                ◳ xInvaders := 4
        else add code here end if
        led → plot(◳ xInvaders, ◳ yInvaders)
        basic → pause(◳ InvadersDelay)
else add code here end if
end function

_____________________________
function snakeDown()
var valueDown := pins → digital read pin(P2)
if valueDown > 0 then
        led → unplot(◳ 0x, ◳ 0y)
        led → unplot(◳ xSnake, ◳ ySnake)
```

```
        ◳ Ox := ◳ xSnake
        ◳ Oy := ◳ ySnake
        ◳ ySnake := ◳ ySnake + 1
        if ◳ ySnake > 4 then
                ◳ ySnake := 0
        else add code here end if
        led → plot(◳ Ox, ◳ Oy)
        led → plot(◳ xSnake, ◳ ySnake)
        basic → pause(◳ SnakeDelay)
else add code here end if
end function

_____________________________________
function snakeUp()
var valueUp := pins → digital read pin(P12)
if valueUp > 0 then
        led → unplot(◳ Ox, ◳ Oy)
        led → unplot(◳ xSnake, ◳ ySnake)
        ◳ Ox := ◳ xSnake
        ◳ Oy := ◳ ySnake
        ◳ ySnake := ◳ ySnake - 1
        if ◳ ySnake < 0 then
                ◳ ySnake := 4
        else add code here end if
        led → plot(◳ Ox, ◳ Oy)
        led → plot(◳ xSnake, ◳ ySnake)
        basic → pause(◳ SnakeDelay)
else add code here end if
end function

_____________________________________
function snakeLeft()
var valueLeft := pins → digital read pin(P8)
if valueLeft > 0 then
        led → unplot(◳ Ox, ◳ Oy)
        led → unplot(◳ xSnake, ◳ ySnake)
        ◳ Ox := ◳ xSnake
        ◳ Oy := ◳ ySnake
        ◳ xSnake := ◳ xSnake - 1
        if ◳ xSnake < 0 then
                ◳ xSnake := 4
        else add code here end if
        led → plot(◳ Ox, ◳ Oy)
        led → plot(◳ xSnake, ◳ ySnake)
        basic → pause(◳ SnakeDelay)
else add code here end if
end function

_____________________________________
function snakeRight()
var valueRight := pins → digital read pin(P16)
if valueRight > 0 then
        led → unplot(◳ Ox, ◳ Oy)
        led → unplot(◳ xSnake, ◳ ySnake)
```

```
        ◳ Ox := ◳ xSnake
        ◳ Oy := ◳ ySnake
        ◳ xSnake := ◳ xSnake + 1
        if ◳ xSnake > 4 then
                ◳ xSnake := 0
        else add code here end if
        led → plot(◳ Ox, ◳ Oy)
        led → plot(◳ xSnake, ◳ ySnake)
        basic → pause(◳ SnakeDelay)
else add code here end if
end function
```

```
function snakeEats()
if ◳ xSnake = ◳ fx and ◳ ySnake = ◳ fy then
        game → set score(game → score + 1)
        if ◳ buzzOff → equals(false) then
                music → play tone(440, 100)
        else add code here end if
        ◳ fx := math → random(5)
        ◳ fy := math → random(5)
        led → plot(◳ fx, ◳ fy)
else
        led → plot(◳ fx, ◳ fy)
end if
end function
```

```
function readButtons()
var leftButton := pins → digital read pin(P8)
var rightButton := pins → digital read pin(P16)
if leftButton > 0 then
        ◳ car → change x by( - 1)
else add code here end if
if rightButton > 0 then
        ◳ car → change x by(1)
else add code here end if
basic → pause(50)
end function
```

```
function crashed 1()
if ◳ crashed → equals(true) then
        var ms := game → current time
        ms := ms / 1000
        game → set score(ms)
        game → game over
        var valueReset := pins → digital read pin(P15)
        if valueReset > 0 then
                control → reset
        else add code here end if
else add code here end if
end function
```

```
_____________________
function meteor 1()
if ◳ m → is touching(◳ car) then
        ◳ crashed := true
else if ◳ m → y ≥ 4 then
        basic → pause(100)
        ◳ m → delete
        ◳ m := game → create sprite(math → random(5), math
→ random(2))
else add code here end if
end function

_____________________
function meteor 2()
if ◳ m2 → is touching(◳ car) then
        ◳ crashed := true
else if ◳ m2 → y ≥ 4 then
        basic → pause(100)
        ◳ m2 → delete
        ◳ m2 := game → create sprite(math → random(5), math
→ random(2))
else add code here end if
end function

_____________________
function meteor 3()
if ◳ m → is touching(◳ car) then
        ◳ crashed := true
else if ◳ m3 → y ≥ 4 then
        basic → pause(100)
        ◳ m3  → delete
        ◳ m3 := game → create sprite(math → random(5), math
→ random(2))
else add code here end if
end function
_____________________
```