%------------------------------------Sagar Patel----------------------------------------------------------------------------------------------------------------------------%|Name

%------------------------------------April 28, 2019-------------------------------------------------------------------------------------------------------------------------%|Date

%------------------------------------Survival Adventure---------------------------------------------------------------------------------------------------------------------%|Game

%------------------------------------Darren Moore---------------------------------------------------------------------------------------------------------------------------%|Teacher

var winID : int := Window.Open ("position:100;100,graphics:800;800") %|This sets the screen size of the game

%------------------------------------Variables------------------------------------------------------------------------------------------------------------------------------%|

var font1, font2, font3, font4, font5, font6, font7, font8, font9, x, y, b, confettix, confettiy, healthcolour, onscreen : int %|This is for fonts, mouse,confetti, and health

var keys : array char of boolean %|This is for movement with keys

var circlexposition := maxx div 2 %|The player begins in the middle

var circleyposition := 70 %|The player begins 70 up (y)

var t, distancebetweencenter1, distancebetweencenter2, distancebetweencenter3, distancebetweencenter4, distancebetweencenter5, distancebetweencenter6 : real %|This is for collision by measuring distance

var distancebetweencenter7, distancebetweencenter8, distancebetweencenter9, distancebetweencenter10, distancebetweencenter11, distancebetweencenter12 : real %|This is for collision by measuring distance

var distancebetweencenter13, distancebetweencenter14, distancebetweencenter15, distancebetweencenter16, distancebetweencenter17, distancebetweencenter18 : real %|This is for collision by measuring distance

var distancebetweencenter19, distancebetweencenter20, distancebetweencenter21, distancebetweencenter22, distancebetweencenter23, distancebetweencenter24 : real %|This is for collision by measuring distance

var distancebetweencenter25, distancebetweencenter26, distancebetweencenter27, distancebetweencenter28, distancebetweencenter29, distancebetweencenter30 : real %|This is for collision by measuring distance

var firing, pausedebug, upgradedebug100, upgradedebug200, upgradedebug300, upgradedebug400 := false %|This is to exit the upgrade loop

var ammo := 20 %|This determines how many bullets you have

var bulletanimationx, healthbar := 680 %|This is the size of the health and bullet box

var reloaddebug : array char of boolean %|This is an array for the bullets begin created

var bulletx : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var bullety : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var xdestination : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var ydestination : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var xsnapshot : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var ysnapshot : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var timeonscreen : flexible array 1 .. 1 of int %|This is an array for the bullets begin created

var shotdelay, killcounter := 0 %|This is the time the bullet is on screen/kill count

timeonscreen (upper (timeonscreen)) := 0 %|This sets the screen time to 0

bulletx (1) := 1 %|This sets the value of the bullets beginning at 1

bullety (1) := 1 %|This sets the value of the bullets beginning at 1

xdestination (1) := 1 %|This sets the value of the bullets beginning at 1

ydestination (1) := 1 %|This sets the value of the bullets beginning at 1

xsnapshot (1) := 1 %|This sets the value of the bullets beginning at 1

ysnapshot (1) := 1 %|This sets the value of the bullets beginning at 1

var zombiehealth1, zombiehealth2, zombiehealth3, zombiehealth4, zombiehealth5, zombiehealth6, zombiehealth7, zombiehealth8, zombiehealth9, zombiehealth10 := 60 %|This is the zombie health for the first 2 waves

var zombiecolour1, zombiecolour2, zombiecolour3, zombiecolour4, zombiecolour5, zombiecolour6, zombiecolour7, zombiecolour8, zombiecolour9, zombiecolour10, zombiecolour11, %|This is the colour of the zombie healthbar

zombiecolour12, zombiecolour13, zombiecolour14, zombiecolour15 := 47 %|This is the colour of the zombie healthbar

var zombiedamage1, zombiedamage2, zombiedamage3, zombiedamage4, zombiedamage5, zombiedamage6, zombiedamage7, zombiedamage8, zombiedamage9, zombiedamage10, zombiedamage11, %|This is the health of the zombie

zombiedamage12, zombiedamage13, zombiedamage14, zombiedamage15 := 0 %|This is the health of the zombie

var killtandf1, killtandf2, killtandf3, killtandf4, killtandf5, killtandf6, killtandf7, killtandf8, killtandf9, killtandf10, %|This is to count the death of zombies

killtandf11, killtandf12, killtandf13, killtandf14, killtandf15 := true %|This is to count the death of zombies

var zombiex1 := Rand.Int (-300, -200) %|This sets the zombie position randomly

var zombiey1 := Rand.Int (800, 1000) %|This sets the zombie position randomly

var zombiex2 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiey2 := Rand.Int (800, 1000) %|This sets the zombie position randomly

var zombiex3 := Rand.Int (800, 900) %|This sets the zombie position randomly

var zombiey3 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiex4 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiey4 := Rand.Int (800, 1000) %|This sets the zombie position randomly

var zombiex5 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiey5 := Rand.Int (-30, -20) %|This sets the zombie position randomly

var zombiex6 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiey6 := Rand.Int (-100, -30) %|This sets the zombie position randomly

var zombiex7 := Rand.Int (700, 800) %|This sets the zombie position randomly

var zombiey7 := Rand.Int (900, 1200) %|This sets the zombie position randomly

var zombiex8 := Rand.Int (200, 400) %|This sets the zombie position randomly

var zombiey8 := Rand.Int (900, 1000) %|This sets the zombie position randomly

var zombiex9 := Rand.Int (-30, 0) %|This sets the zombie position randomly

var zombiey9 := Rand.Int (0, 900) %|This sets the zombie position randomly

var zombiex10 := Rand.Int (0, 800) %|This sets the zombie position randomly

var zombiey10 := Rand.Int (-200, -70) %|This sets the zombie position randomly

var zombiehealth11 := 2000 %|This sets the health of the zombie

var zombiex11 := Rand.Int (-200, -60) %|This sets the zombie position randomly

var zombiey11 := Rand.Int (0, 800) %|This sets the zombie position randomly

var bulletsize := 10 %|This presets the zombie upgrade values

var bulletspeed := 3 %|This presets the zombie upgrade values

var playermovement := 5 %|This presets the zombie upgrade values

var bulletdamage := 1 %|This presets the zombie upgrade values

var upgradedebug1, upgradedebug2, upgradedebug3, upgradedebug4, upgradedebug10, upgradedebug20, upgradedebug30, upgradedebug40 := false %|This is to be able to exit the upgrade

var zombiex12 := Rand.Int (-50, -30) %|This sets the zombie position randomly

var zombiey12 := Rand.Int (820, 850) %|This sets the zombie position randomly

var zombiex13 := Rand.Int (-50, -30) %|This sets the zombie position randomly

var zombiey13 := Rand.Int (-50, -30) %|This sets the zombie position randomly

var zombiex14 := Rand.Int (830, 850) %|This sets the zombie position randomly

var zombiey14 := Rand.Int (820, 850) %|This sets the zombie position randomly

var zombiex15 := Rand.Int (830, 850) %|This sets the zombie position randomly

var zombiey15 := Rand.Int (-50, -30) %|This sets the zombie position randomly

var zombiehealth12, zombiehealth13, zombiehealth14, zombiehealth15 := 1000 %|This sets the zombie health Wave 4

var giantx, gianty, giantx1, gianty1, zombiex12debug, zombiey12debug, zombiex13debug, zombiey13debug, zombiex14debug, zombiey14debug, zombiex15debug, zombiey15debug : real %|This is to prevent overflow in operations

var textparadebug, textparadebug2 := 0 %|This to be able to exit the text loop

var playercolour := 6 %|This is the player's ring colour

var playercenter := 18 %|This is the player's center colour

%------------------------------------Process/For-(Makes Code Efficient)-----------------------------------------------------------------------------------------------------%|

process reload %|This is the beginning of the reload process

reloaddebug (' ') := false %|This makes the (' ') false until the process is done

for loadanimation : 1 .. 20 %|This makes the load animation by making a growing box

bulletanimationx := bulletanimationx + 20 %|This adds 20 pixels to the x of the bullet box

ammo := loadanimation %|This makes the ammo equal the load animation

delay (100) %|Delays each bullet by 0.1 seconds

end for %|End For

reloaddebug (' ') := true %|Once the reload is over, then (' ') can be entered

end reload %|End the reload process

for i : 1 .. upper (reloaddebug) %|For 1 to the maximum reloaddebug value

reloaddebug (chr (i)) := true %|It is always true until reload is true

end for %|End For

proc bulletshoot %|Begin bullet shoot process

onscreen := 0 %|The time the bullets are on screen = 0

for i : 1 .. upper (bulletx) %|For every bullet that is created:

if bulletx (i) < maxx and bulletx (i) > 0 and bullety (i) < maxy and bullety (i) > 0 then %|If any bullet is within these requirements then:

Draw.FillOval (bulletx (i), bullety (i), bulletsize, bulletsize, black) %|Draw a brand new bullet

timeonscreen (i) += 1 %|The time on screen adds everyime there is a new bullet

onscreen += 1 %|on screen adds everyime there is a new bullet

end if %|This ends the if

if timeonscreen (i) >= 100 then %|This determines how long the bullets shouls stay on

bulletx (i) := maxx %|This causes the bullets to dissapear

bullety (i) := maxy %|This causes the bullets to dissapear

end if %|This ends the if

end for %|End For

Input.KeyDown (keys) %|This is so key info can be collected

if keys (' ') = true and firing = false and ammo > 0 and reloaddebug (' ') = true then %|If these requirements are met then:

bulletanimationx -= 20 %|Make the bullet go down by 1

ammo -= 1 %|Lose one bullet in ammo

firing := true %|Make firing become true

if shotdelay = 0 then %|If the shot delay is 0, then create a bullet info

new bulletx, upper (bulletx) + 1 %|This uses math to calculate the bullet travel

new bullety, upper (bullety) + 1 %|This uses math to calculate the bullet travel

bulletx (upper (bulletx)) := circlexposition %|(I had to google some stuff like how to keep

bullety (upper (bullety)) := circleyposition %|an object at a constant speed without change)

new xdestination, upper (xdestination) + 1 %|This uses math to calculate the bullet travel

new ydestination, upper (ydestination) + 1 %|This uses math to calculate the bullet travel

xdestination (upper (xdestination)) := round (circlexposition - (40 \* cosd (t))) %|This uses math to calculate the bullet travel

ydestination (upper (ydestination)) := round (circleyposition - (40 \* sind (t))) %|This uses math to calculate the bullet travel

new xsnapshot, upper (xsnapshot) + 1 %|This uses math to calculate the bullet travel

new ysnapshot, upper (ysnapshot) + 1 %|This uses math to calculate the bullet travel

xsnapshot (upper (xsnapshot)) := circlexposition %|This uses math to calculate the bullet travel

ysnapshot (upper (ysnapshot)) := circleyposition %|This uses math to calculate the bullet travel

new timeonscreen, upper (timeonscreen) + 1 %|This uses math to calculate the bullet travel

timeonscreen (upper (timeonscreen)) := 0 %|This uses math to calculate the bullet travel

end if %|This ends the if

else %|Else

end if %|This ends the if

for i : 1 .. upper (bulletx) %|For every bullet that is created:

bulletx (i) := bulletx (i) + round ((xdestination (i) - (xsnapshot (i))) \* 0.1) \* bulletspeed %|Calculate the slope and trijectory

bullety (i) := bullety (i) + round ((ydestination (i) - (ysnapshot (i))) \* 0.1) \* bulletspeed %|Calculate the slope and trijectory

end for %|End For

end bulletshoot %|End the bullet shoot process

proc mousecrosshairs %|This begins mouse crosshairs

Mouse.Where (x, y, b) %|Gathers info on mouse position

Draw.FillOval (x, y, 5, 5, red) %|Draws graphics for corsshairs

Draw.FillBox (x - 3, y + 8, x + 3, y + 23, 17) %|Draws graphics for corsshairs

Draw.FillBox (x - 3, y - 8, x + 3, y - 23, 17) %|Draws graphics for corsshairs

Draw.FillBox (x + 8, y - 3, x + 23, y + 3, 17) %|Draws graphics for corsshairs

Draw.FillBox (x - 8, y + 3, x - 23, y - 3, 17) %|Draws graphics for corsshairs

end mousecrosshairs %|This begins mouse crosshairs

proc shooterarm %|This begins the shooterarm process

if x = circlexposition and y > circleyposition then %|This uses math and trig to calulate with quadrant CAST

t := 270 %|tangent = 270

elsif x < circlexposition and y = circleyposition then %|This uses math and trig to calulate with quadrant CAST

t := 0 %|tangent = 0

elsif x = circlexposition and y < circleyposition then %|This uses math and trig to calulate with quadrant CAST

t := 90 %|tangent = 90

elsif x > circlexposition and y = circleyposition then %|This uses math and trig to calulate with quadrant CAST

t := 180 %|tangent = 180

elsif x = circlexposition and y = circleyposition then %|This uses math and trig to calulate with quadrant CAST

t := 360 %|tangent = 360

elsif x > circlexposition then %|This uses math and trig to calulate with quadrant CAST

t := arctand ((y - circleyposition) / (x - circlexposition)) + 180 %|Value given to t

else %|Else

t := arctand ((y - circleyposition) / (x - circlexposition)) %|Value given to t

end if %|This ends the if

Draw.ThickLine (circlexposition, circleyposition, circlexposition - round ((40 + (3 \* bulletspeed) / 2) \* cosd (t)), circleyposition - %|This draws the shooter arm using the information

round ((40 + round (3 \* bulletspeed) / 2) \* sind (t)), bulletsize + 25, black) %|This draws the shooter arm using the information

end shooterarm %|This begins the shooterarm process

proc quitgameanimation %|This is the quit animation process

for i : 1 .. 860 %|This is the growing box animation

Draw.FillBox (0, 0, i, i, 7) %|This draws the growing box

delay (3) %|Delay for smooth animation

end for %|End For

delay (100) %|Delays 0.1 second

Window.Close (winID) %|Closes Window

Window.Close (winID) %|Closes Window

end quitgameanimation %|End the quit animation process

proc zombiemove1 %|This is the zombie one movement process

if zombiehealth1 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex1, zombiey1, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex1 - 30, zombiey1 - 30, zombiex1 + 30, zombiey1 - 25, 21) %|Draws the black bar behind healthbar

Draw.FillBox (zombiex1 - 30, zombiey1 - 30, zombiex1 + 30 + zombiedamage1, zombiey1 - 25, zombiecolour1) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex1 then %|If the player pos is greater than zombie pos

zombiex1 += 3 %|Increase the x pos of the zombie

end if %|This ends the if

if circlexposition < zombiex1 then %|If the player pos is lower than zombie pos

zombiex1 -= 2 %|Lower the x pos of the zombie

end if %|This ends the if

if circleyposition > zombiey1 then %|If the player pos is greater than zombie pos

zombiey1 += 3 %|Increase the y pos of the zombie

end if %|This ends the if

if circleyposition < zombiey1 then %|If the player pos is lower than zombie pos

zombiey1 -= 2 %|Lower the y pos of the zombie

end if %|This ends the if

distancebetweencenter1 := sqrt ((zombiex1 - circlexposition) \*\* 2 + (zombiey1 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter1 <= 50) then %|If both radius >= distance between the two:

if zombiehealth1 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Lower the player's health bar

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth1 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour1 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth1 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour1 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth1 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour1 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove1 %|End zombie movement 1 process

proc bulletcollision1 %|Begin bullet collision 1 process

if zombiehealth1 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter2 := sqrt ((zombiex1 - bulletx (i)) \*\* 2 + (zombiey1 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter2 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth1 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage1 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision1 %|End bullet collision 1 process

proc zombiemove2 %|Begin zombie movement 2 process

if zombiehealth2 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex2, zombiey2, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex2 - 30, zombiey2 - 30, zombiex2 + 30, zombiey2 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex2 - 30, zombiey2 - 30, zombiex2 + 30 + zombiedamage2, zombiey2 - 25, zombiecolour2) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex2 then %|This is for zombie movement

zombiex2 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex2 then %|This is for zombie movement

zombiex2 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey2 then %|This is for zombie movement

zombiey2 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey2 then %|This is for zombie movement

zombiey2 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter3 := sqrt ((zombiex2 - circlexposition) \*\* 2 + (zombiey2 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter3 <= 50) then %|If both radius >= distance between the two:

if zombiehealth2 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Subtract player's health

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth2 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour2 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth2 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour2 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth2 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour2 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove2 %|End zombie movement 2 process

proc bulletcollision2 %|Begin bullet collision 2 process

if zombiehealth2 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter4 := sqrt ((zombiex2 - bulletx (i)) \*\* 2 + (zombiey2 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter4 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth2 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage2 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision2 %|End bullet collision 2 proces

proc zombiemove3 %|Begin zombie movement 3 process

if zombiehealth3 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex3, zombiey3, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex3 - 30, zombiey3 - 30, zombiex3 + 30, zombiey3 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex3 - 30, zombiey3 - 30, zombiex3 + 30 + zombiedamage3, zombiey3 - 25, zombiecolour3) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex3 then %|This is for zombie movement

zombiex3 += 3 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex3 then %|This is for zombie movement

zombiex3 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey3 then %|This is for zombie movement

zombiey3 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey3 then %|This is for zombie movement

zombiey3 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter5 := sqrt ((zombiex3 - circlexposition) \*\* 2 + (zombiey3 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter5 <= 50) then %|If both radius >= distance between the two:

if zombiehealth3 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Subtract player's health

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth3 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour3 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth3 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour3 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth3 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour3 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove3 %|End zombie movement 3 process

proc bulletcollision3 %|Begin bullet collision 3 process

if zombiehealth3 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter6 := sqrt ((zombiex3 - bulletx (i)) \*\* 2 + (zombiey3 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter6 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth3 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage3 -= bulletdamage %|Subtract zombie health by bullet damage

else %|else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision3 %|End bullet collision 3 process

proc zombiemove4 %|Begin zombie movement 4 process

if zombiehealth4 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex4, zombiey4, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex4 - 30, zombiey4 - 30, zombiex4 + 30, zombiey4 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex4 - 30, zombiey4 - 30, zombiex4 + 30 + zombiedamage4, zombiey4 - 25, zombiecolour4) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex4 then %|This is for zombie movement

zombiex4 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex4 then %|This is for zombie movement

zombiex4 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey4 then %|This is for zombie movement

zombiey4 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey4 then %|This is for zombie movement

zombiey4 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter7 := sqrt ((zombiex4 - circlexposition) \*\* 2 + (zombiey4 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter7 <= 50) then %|If both radius >= distance between the two:

if zombiehealth4 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Else

else %|End If

end if %|This ends the if

end if %|This ends the if

if zombiehealth4 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour4 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth4 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour4 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth4 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour4 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove4 %|End zombie movement 4 process

proc bulletcollision4 %|Begin bullet collision 4 process

if zombiehealth4 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter8 := sqrt ((zombiex4 - bulletx (i)) \*\* 2 + (zombiey4 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter8 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth4 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage4 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision4 %|End bullet collision 4 process

proc zombiemove5 %|Begin zombie movement 5 process

if zombiehealth5 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex5, zombiey5, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex5 - 30, zombiey5 - 30, zombiex5 + 30, zombiey5 - 25, 21) %|Draws black bar behind health bar

Draw.FillBox (zombiex5 - 30, zombiey5 - 30, zombiex5 + 30 + zombiedamage5, zombiey5 - 25, zombiecolour5) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex5 then %|This is for zombie movement

zombiex5 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex5 then %|This is for zombie movement

zombiex5 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey5 then %|This is for zombie movement

zombiey5 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey5 then %|This is for zombie movement

zombiey5 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter9 := sqrt ((zombiex5 - circlexposition) \*\* 2 + (zombiey5 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter9 <= 50) then %|If both radius >= distance between the two:

if zombiehealth5 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Subtract health of player

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth5 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour5 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth5 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour5 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth5 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour5 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove5 %|End zombie movement 5 process

proc bulletcollision5 %|Begin bullet collision 5 process

if zombiehealth5 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter10 := sqrt ((zombiex5 - bulletx (i)) \*\* 2 + (zombiey5 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter10 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth5 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage5 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision5 %|End bullet collision 5 process

proc zombiemove6 %|Begin zombie movement 6 process

if zombiehealth6 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex6, zombiey6, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex6 - 30, zombiey6 - 30, zombiex6 + 30, zombiey6 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex6 - 30, zombiey6 - 30, zombiex6 + 30 + zombiedamage6, zombiey6 - 25, zombiecolour6) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex6 then %|This is for zombie movement

zombiex6 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex6 then %|This is for zombie movement

zombiex6 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey6 then %|This is for zombie movement

zombiey6 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey6 then %|This is for zombie movement

zombiey6 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter11 := sqrt ((zombiex6 - circlexposition) \*\* 2 + (zombiey6 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter11 <= 50) then %|If both radius >= distance between the two:

if zombiehealth6 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Subtract player's healthbar

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth6 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour6 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth6 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour6 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth6 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour6 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove6 %|End zombie moveemnt 6 process

proc bulletcollision6 %|Begin bullet collision 6 process

if zombiehealth6 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

distancebetweencenter12 := sqrt ((zombiex6 - bulletx (i)) \*\* 2 + (zombiey6 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter12 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth6 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage6 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision6 %|End bullet collision 6 process

proc zombiemove7 %|Begin zombie movement 7 process

if zombiehealth7 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex7, zombiey7, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex7 - 30, zombiey7 - 30, zombiex7 + 30, zombiey7 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex7 - 30, zombiey7 - 30, zombiex7 + 30 + zombiedamage7, zombiey7 - 25, zombiecolour7) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex7 then %|This is for zombie movement

zombiex7 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex7 then %|This is for zombie movement

zombiex7 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey7 then %|This is for zombie movement

zombiey7 += 3 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey7 then %|This is for zombie movement

zombiey7 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter13 := sqrt ((zombiex7 - circlexposition) \*\* 2 + (zombiey7 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter13 <= 50) then %|If both radius >= distance between the two:

if zombiehealth7 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 2 %|Subtract player's health

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth7 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour7 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth7 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour7 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth7 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour7 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove7 %|End bullet movement 7 process

proc bulletcollision7 %|Begin bullet collision 7 process

if zombiehealth7 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created

distancebetweencenter14 := sqrt ((zombiex7 - bulletx (i)) \*\* 2 + (zombiey7 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter14 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth7 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage7 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision7 %|End zombie movement 7

proc zombiemove8 %|Begin zombie movement 8

if zombiehealth8 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex8, zombiey8, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex8 - 30, zombiey8 - 30, zombiex8 + 30, zombiey8 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex8 - 30, zombiey8 - 30, zombiex8 + 30 + zombiedamage8, zombiey8 - 25, zombiecolour8) %|This draws the zombie healthbar

else %|Else

end if %|This ends the if

if circlexposition > zombiex8 then %|This is for zombie movement

zombiex8 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex8 then %|This is for zombie movement

zombiex8 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey8 then %|This is for zombie movement

zombiey8 += 1 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey8 then %|This is for zombie movement

zombiey8 -= 2 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter15 := sqrt ((zombiex8 - circlexposition) \*\* 2 + (zombiey8 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter15 <= 50) then %|If both radius >= distance between the two:

if zombiehealth8 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 3 %|Subtract player's health bar

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth8 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour8 := 14 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth8 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour8 := 43 %|This changes the zombie healthbar colour

end if %|This ends the if

if zombiehealth8 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour8 := 40 %|This changes the zombie healthbar colour

end if %|This ends the if

end zombiemove8 %|End zombie movement 8

proc bulletcollision8 %|Begin bullet collision 8

if zombiehealth8 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet created:

distancebetweencenter16 := sqrt ((zombiex8 - bulletx (i)) \*\* 2 + (zombiey8 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter16 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth8 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage8 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|This ends the if

end for %|End For

end if %|This ends the if

end bulletcollision8 %|End bullet collision 8 process

proc zombiemove9 %|Begin zombie movement 9 process

if zombiehealth9 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex9, zombiey9, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex9 - 30, zombiey9 - 30, zombiex9 + 30, zombiey9 - 25, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex9 - 30, zombiey9 - 30, zombiex9 + 30 + zombiedamage9, zombiey9 - 25, zombiecolour9) %|This draws the zombie healthbar

else %|else

end if %|This ends the if

if circlexposition > zombiex9 then %|This is for zombie movement

zombiex9 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circlexposition < zombiex9 then %|This is for zombie movement

zombiex9 -= 3 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition > zombiey9 then %|This is for zombie movement

zombiey9 += 2 %|Changes zombie value relation to player

end if %|This ends the if

if circleyposition < zombiey9 then %|This is for zombie movement

zombiey9 -= 1 %|Changes zombie value relation to player

end if %|This ends the if

distancebetweencenter17 := sqrt ((zombiex9 - circlexposition) \*\* 2 + (zombiey9 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter17 <= 50) then %|If both radius >= distance between the two:

if zombiehealth9 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 2 %|Lower healthbar of player

else %|Else

end if %|This ends the if

end if %|This ends the if

if zombiehealth9 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour9 := 14 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth9 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour9 := 43 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth9 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour9 := 40 %|This changes the zombie healthbar colour

end if %|End If

end zombiemove9 %|End zombie 9 movement process

proc bulletcollision9 %|Bullet collision 9 process

if zombiehealth9 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created

distancebetweencenter18 := sqrt ((zombiex9 - bulletx (i)) \*\* 2 + (zombiey9 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter18 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth9 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage9 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision9 %|End bullet collision 9 process

proc zombiemove10 %|Zombie movement 10 process

if zombiehealth10 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex10, zombiey10, 20, 20, red) %|Draws zombie

Draw.FillBox (zombiex10 - 30, zombiey10 - 30, zombiex10 + 30, zombiey10 - 25, 21) %|Draw black bar behind healthbar

Draw.FillBox (zombiex10 - 30, zombiey10 - 30, zombiex10 + 30 + zombiedamage10, zombiey10 - 25, zombiecolour10) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex10 then %|This is for zombie movement

zombiex10 += 1 %|Changes zombie value relation to player

end if %|End If

if circlexposition < zombiex10 then %|This is for zombie movement

zombiex10 -= 2 %|Changes zombie value relation to player

end if %|End If

if circleyposition > zombiey10 then %|This is for zombie movement

zombiey10 += 3 %|Changes zombie value relation to player

end if %|End If

if circleyposition < zombiey10 then %|This is for zombie movement

zombiey10 -= 1 %|Changes zombie value relation to player

end if %|End If

distancebetweencenter19 := sqrt ((zombiex10 - circlexposition) \*\* 2 + (zombiey10 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter19 <= 50) then %|If both radius >= distance between the two:

if zombiehealth10 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 2 %|Lower healthbar of player

else %|Else

end if %|End If

end if %|End If

if zombiehealth10 <= 45 then %|If the zombie health is less than 45 then:

zombiecolour10 := 14 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth10 <= 25 then %|If the zombie health is less than 25 then:

zombiecolour10 := 43 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth10 <= 10 then %|If the zombie health is less than 10 then:

zombiecolour10 := 40 %|This changes the zombie healthbar colour

end if %|End If

end zombiemove10 %|End zombie 10 movement process

proc bulletcollision10 %|Begin zombie 10 collision process

if zombiehealth10 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet created:

distancebetweencenter20 := sqrt ((zombiex10 - bulletx (i)) \*\* 2 + (zombiey10 - bullety (i)) \*\* 2) %|Calculates collision

if (distancebetweencenter20 <= bulletsize + 20) then %|If both radius >= distance between the two:

zombiehealth10 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage10 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision10 %|End bullet collision zombie 10

proc zombiemove11 %|Begins zombie 11 movement process

if zombiehealth11 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex11, zombiey11, 120, 120, 12) %|Draws zombie

Draw.FillOval (zombiex11, zombiey11, 100, 100, 4) %|Draws zombie

Draw.FillBox (zombiex11 - 70, zombiey11 - 140, zombiex11 + 70, zombiey11 - 165, 21) %|Draws black bar behind healthbar

Draw.FillBox (zombiex11 - 70, zombiey11 - 140, zombiex11 + 70 + round (zombiedamage11 / 14.2971428), zombiey11 - 165, zombiecolour11) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex11 then %|This is for zombie movement

zombiex11 += 3 %|Changes zombie value relation to player

end if %|End If

if circlexposition < zombiex11 then %|This is for zombie movement

zombiex11 -= 2 %|Changes zombie value relation to player

end if %|End If

if circleyposition > zombiey11 then %|This is for zombie movement

zombiey11 += 3 %|Changes zombie value relation to player

end if %|End If

if circleyposition < zombiey11 then %|This is for zombie movement

zombiey11 -= 3 %|Changes zombie value relation to player

end if %|End If

distancebetweencenter21 := sqrt ((zombiex11 - circlexposition) \*\* 2 + (zombiey11 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter21 <= 150) then %|If both radius >= distance between the two:

if zombiehealth11 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 10 %|Player health bar loses 10 pixels

else %|Else

end if %|End If

end if %|End If

if zombiehealth11 <= 1400 then %|If the zombie health is less than 1400 then:

zombiecolour11 := 14 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth11 <= 700 then %|If the zombie health is less than 700 then:

zombiecolour11 := 43 %|This changes the zombie healthbar colour

end if %|End If

if zombiehealth11 <= 400 then %|If the zombie health is less than 400 then:

zombiecolour11 := 40 %|This changes the zombie healthbar colour

end if %|End If

end zombiemove11 %|End zombie movement process

proc bulletcollision11 %|Begin zombie 11 bullet collision

if zombiehealth11 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bullety) %|For every bullet that is created:

giantx := (bulletx (i) - zombiex11) %|Calculates collision

gianty := (bullety (i) - zombiey11) %|Calculates collision

giantx1 := giantx \*\* 2 %|Calculates collision

gianty1 := gianty \*\* 2 %|Calculates collision

distancebetweencenter22 := sqrt (giantx1 + gianty1) %|Calculates collision

if (distancebetweencenter22 <= 120 + bulletsize) then %|If both radius >= distance between the two:

zombiehealth11 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage11 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision11 %|End Process zombie 11 collision

proc zombiemove12 %|Process zombie 12

if zombiehealth12 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex12, zombiey12, 55, 55, 12) %|This draws zombie

Draw.FillOval (zombiex12, zombiey12, 40, 40, 4) %|This draws zombie

Draw.FillBox (zombiex12 - 70, zombiey12 - 110, zombiex12 + 70, zombiey12 - 95, 21) %|This draws the black bar behind healthbar

Draw.FillBox (zombiex12 - 70, zombiey12 - 110, zombiex12 + 70 + round (zombiedamage12 / 7.1485714), zombiey12 - 95, zombiecolour12) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex12 then %|This is for zombie movement

zombiex12 += 2 %|Changes zombie value relation to player

end if %|End If

if circlexposition < zombiex12 then %|This is for zombie movement

zombiex12 -= 2 %|Changes zombie value relation to player

end if %|End If

if circleyposition > zombiey12 then %|This is for zombie movement

zombiey12 += 1 %|Changes zombie value relation to player

end if %|End If

if circleyposition < zombiey12 then %|This is for zombie movement

zombiey12 -= 3 %|Changes zombie value relation to player

end if %|End If

distancebetweencenter23 := sqrt ((zombiex12 - circlexposition) \*\* 2 + (zombiey12 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter23 <= 85) then %|If both radius >= distance between the two:

if zombiehealth12 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 10 %|Player health bar loses 10 pixels

else %|Else

end if %|End If

end if %|End If

if zombiehealth12 <= 700 then %|If the zombie health is less than 700 then:

zombiecolour12 := 14 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth12 <= 400 then %|If the zombie health is less than 400 then:

zombiecolour12 := 43 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth12 <= 100 then %|If the zombie health is less than 100 then:

zombiecolour12 := 40 %|Change the zombie healthbar colour

end if %|End If

end zombiemove12 %|This ends zombie movement process for zombie 12

proc bulletcollision12 %|This begins the bullet collision for zombie 12

if zombiehealth12 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet created:

zombiex12debug := (zombiex12 - bulletx (i)) \*\* 2 %|Calculates Collision

zombiey12debug := (zombiey12 - bullety (i)) \*\* 2 %|Calculates Collision

distancebetweencenter24 := sqrt (zombiex12debug + zombiey12debug) %|Calculates Collision

if (distancebetweencenter24 <= 65 + bulletsize) then %|If both radius >= distance between the two:

zombiehealth12 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage12 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision12 %|This ends bullet collision for zombie 12

proc zombiemove13 %|This begins bullet movement for zombie 13

if zombiehealth13 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex13, zombiey13, 55, 55, 12) %|This draws the zombie

Draw.FillOval (zombiex13, zombiey13, 40, 40, 4) %|This draws the zombie

Draw.FillBox (zombiex13 - 70, zombiey13 - 110, zombiex13 + 70, zombiey13 - 95, 21) %|This draws the black bar behind healthbar

Draw.FillBox (zombiex13 - 70, zombiey13 - 110, zombiex13 + 70 + round (zombiedamage13 / 7.1485714), zombiey13 - 95, zombiecolour13) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex13 then %|This is for zombie movement

zombiex13 += 2 %|Changes zombie value relation to player

end if %|End If

if circlexposition < zombiex13 then %|This is for zombie movement

zombiex13 -= 2 %|Changes zombie value relation to player

end if %|End If

if circleyposition > zombiey13 then %|This is for zombie movement

zombiey13 += 2 %|Changes zombie value relation to player

end if %|End If

if circleyposition < zombiey13 then %|This is for zombie movement

zombiey13 -= 3 %|Changes zombie value relation to player

end if %|End If

distancebetweencenter25 := sqrt ((zombiex13 - circlexposition) \*\* 2 + (zombiey13 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter25 <= 85) then %|If both radius >= distance between the two:

if zombiehealth13 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 10 %|Player health bar loses 10 pixels

else %|Else

end if %|End If

end if %|End If

if zombiehealth13 <= 700 then %|If the zombie health is less than 700 then:

zombiecolour13 := 14 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth13 <= 400 then %|If the zombie health is less than 400 then:

zombiecolour13 := 43 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth13 <= 100 then %|If the zombie health is less than 100 then:

zombiecolour13 := 40 %|Change the zombie healthbar colour

end if %|End If

end zombiemove13 %|This ends the zombie movement 13

proc bulletcollision13 %|This begins the bullet collision for zombie 13

if zombiehealth13 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

zombiex13debug := (zombiex13 - bulletx (i)) \*\* 2 %|Calculates collision

zombiey13debug := (zombiey13 - bullety (i)) \*\* 2 %|Calculates collision

distancebetweencenter26 := sqrt (zombiex13debug + zombiey13debug) %|Calculates collision

if (distancebetweencenter26 <= 65 + bulletsize) then %|If both radius >= distance between the two:

zombiehealth13 -= bulletdamage %|Subtract zombie health by bullet damage

zombiedamage13 -= bulletdamage %|Subtract zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision13 %|This ends bullet collision for zombie 13

proc zombiemove14 %|The begins the zombie 14 movement process

if zombiehealth14 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex14, zombiey14, 55, 55, 12) %|Draws the zombie ring

Draw.FillOval (zombiex14, zombiey14, 40, 40, 4) %|Draws the zombie center

Draw.FillBox (zombiex14 - 70, zombiey14 - 110, zombiex14 + 70, zombiey14 - 95, 21) %|Draws the black bar behind zombie heatlh

Draw.FillBox (zombiex14 - 70, zombiey14 - 110, zombiex14 + 70 + round (zombiedamage14 / 7.1485714), zombiey14 - 95, zombiecolour14) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex14 then %|If player position is greater than zombie then:

zombiex14 += 2 %|Zombie x position increases by 2

end if %|End If

if circlexposition < zombiex14 then %|If player is less than zombie then:

zombiex14 -= 2 %|Zombie x position decreases by 2

end if %|End If

if circleyposition > zombiey14 then %|If player is greater than zombie then:

zombiey14 += 2 %|Zombie y position increases by 2

end if %|End If

if circleyposition < zombiey14 then %|If player is lower than zombie then:

zombiey14 -= 2 %|Zombie y position decreases by 2

end if %|End If

distancebetweencenter27 := sqrt ((zombiex14 - circlexposition) \*\* 2 + (zombiey14 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter27 <= 85) then %|If both radius >= distance between the two:

if zombiehealth14 > 0 then %|If the zomvie health is greater than 0 then:

healthbar -= 10 %|Player health bar loses 10 pixels

else %|Else

end if %|End If

end if %|End If

if zombiehealth14 <= 700 then %|If the zombie health is less than 700 then:

zombiecolour14 := 14 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth14 <= 400 then %|If the zombie health is less than 400 then:

zombiecolour14 := 43 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth14 <= 100 then %|If the zombie health is less than 100 then:

zombiecolour14 := 40 %|Change the zombie healthbar colour

end if %|End If

end zombiemove14 %|This ends movement process for zombie 14

proc bulletcollision14 %|This begins the bullet collision for zombie 14

if zombiehealth14 > 0 then %|If the zombie health is greater than 0 then:

for i : 1 .. upper (bulletx) %|For every bullet that is created:

zombiex14debug := (zombiex14 - bulletx (i)) \*\* 2 %|This calculates zombie collision with bullet

zombiey14debug := (zombiey14 - bullety (i)) \*\* 2 %|This calculates zombie collision with bullet

distancebetweencenter28 := sqrt (zombiex14debug + zombiey14debug) %|This calculates the distance between bullet and zombie

if (distancebetweencenter28 <= 65 + bulletsize) then %|If both radius >= distance between the two:

zombiehealth14 -= bulletdamage %|The zombie health subtracts by bullet damage

zombiedamage14 -= bulletdamage %|The zombie health subtracts by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision14 %|This ends the zombie bullet collision

proc zombiemove15 %|This begins the zombie movement process

if zombiehealth15 > 0 then %|If the zombie health is greater than 0 then:

Draw.FillOval (zombiex15, zombiey15, 55, 55, 12) %|Draws zombie ring

Draw.FillOval (zombiex15, zombiey15, 40, 40, 4) %|Draws zombie center

Draw.FillBox (zombiex15 - 70, zombiey15 - 110, zombiex15 + 70, zombiey15 - 95, 21) %|Draws original healthbar in black

Draw.FillBox (zombiex15 - 70, zombiey15 - 110, zombiex15 + 70 + round (zombiedamage15 / 7.1485714), zombiey15 - 95, zombiecolour15) %|This draws the zombie healthbar

else %|Else

end if %|End If

if circlexposition > zombiex15 then %|If the player pos is bigger than zombie then:

zombiex15 += 2 %|Increase the x pos of the zombie

end if %|End If

if circlexposition < zombiex15 then %|If the player pos is less than zombie then:

zombiex15 -= 2 %|Lower the x pos of the zombie

end if %|End If

if circleyposition > zombiey15 then %|If the player pos is bigger than zombie then:

zombiey15 += 1 %|Increase the y pos of the zombie

end if %|End If

if circleyposition < zombiey15 then %|If the player pos is less than zombie then:

zombiey15 -= 3 %|Lower the y pos of the zombie

end if %|End If

distancebetweencenter29 := sqrt ((zombiex15 - circlexposition) \*\* 2 + (zombiey15 - circleyposition) \*\* 2) %|Caluclated the distance between zombie and player

if (distancebetweencenter29 <= 85) then %|If both radius >= distance between the two:

if zombiehealth15 > 0 then %|If the zombie health is greater than 0 then:

healthbar -= 10 %|Player health bar loses 10 pixels

else %|Else

end if %|End If

end if %|End If

if zombiehealth15 <= 700 then %|If the zombie health is less than 700 then:

zombiecolour15 := 14 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth15 <= 400 then %|If the zombie health is less than 400 then:

zombiecolour15 := 43 %|Change the zombie healthbar colour

end if %|End If

if zombiehealth15 <= 100 then %|If the zombie health is less than 100 then:

zombiecolour15 := 40 %|Change the zombie healthbar colour

end if %|End If

end zombiemove15 %|End the zombie 15 process

proc bulletcollision15 %|This is the bullet collision for zombie 15

if zombiehealth15 > 0 then %|If the zombie health is greater than 0

for i : 1 .. upper (bulletx) %|For every bullet that is created:

zombiex15debug := (zombiex15 - bulletx (i)) \*\* 2 %|This calculates collision distance

zombiey15debug := (zombiey15 - bullety (i)) \*\* 2 %|This calculates collision distance

distancebetweencenter30 := sqrt (zombiex15debug + zombiey15debug) %|This is the formula for collision

if (distancebetweencenter30 <= 65 + bulletsize) then %|If both radius >= collision distance then:

zombiehealth15 -= bulletdamage %|subtract the zombie health by bullet damage

zombiedamage15 -= bulletdamage %|subtract the zombie health by bullet damage

else %|Else

end if %|End If

end for %|End For

end if %|End If

end bulletcollision15 %|The ends the bullet collision process

proc musicmenu %|Begins the music process

Music.PlayFileLoop ("Menu.MP3") %|Plays the selected song

end musicmenu %|Ends the music process

proc musicstory %|Begins the music process

Music.PlayFileLoop ("Story.MP3") %|Plays the selected song

end musicstory %|Ends the music process

proc musicgame %|Begins the music process

Music.PlayFileLoop ("In-Game.MP3") %|Plays the selected song

end musicgame %|Ends the music process

proc musicwin %|Begins the music process

Music.PlayFileLoop ("Winning.MP3") %|Plays the selected song

end musicwin %|Ends the music process

proc musiclose %|Begins the music process

Music.PlayFileLoop ("Death.MP3") %|Plays the selected song

end musiclose %|Ends the music process

proc mainmovementprocess %|This begins the main stuff process

mousecrosshairs %|This is calling the crosshair process

shooterarm %|This calls the shooter process

Draw.FillBox (0, 0, maxx, 120, 21) %|Control Panel Box

Draw.FillBox (15, 780, 35, 730, 255) %|Pause Button

Draw.FillBox (45, 780, 65, 730, 255) %|Pause Button

Draw.ThickLine (0, 120, maxx, 120, 10, 7) %|Border

Draw.ThickLine (0, maxy - 5, maxx, maxy - 5, 10, 7) %|Border

Draw.ThickLine (5, 120, 5, maxy, 10, 7) %|Border

Draw.ThickLine (maxx - 5, 120, maxx - 5, maxy, 10, 7) %|Border

delay (25) %|Delays the game so the game is more fluid

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, playercolour) %|Draws the player circle

Draw.FillOval (circlexposition, circleyposition, round (25 - (playermovement / 2) + 5) - 1, round (25 - (playermovement / 2) + 5) - 1, playercenter) %|Draws the player circle

Input.KeyDown (keys) %|This is a variable for movement

if keys ('a') = true then %|If 'a' is pressed then:

if circlexposition <= 30 then %|As long the xposition is less than 30 then:

else %|Else

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, 214) %|"Erases" the original circle

circlexposition := circlexposition - playermovement %|Changes the player's x pos

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, playercolour) %|Draws new player circle

Draw.FillOval (circlexposition, circleyposition, round (25 - (playermovement / 2) + 5) - 1, round (25 - (playermovement / 2) + 5) - 1, playercenter) %|Draws new player circle

end if %|End If

end if %|End If

if keys ('d') = true then %|If 'd' is pressed then:

if circlexposition >= 770 then %|As long the xposition is greater than 770 then:

else %|Else

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, 214) %|"Erases" the original circle

circlexposition := circlexposition + playermovement %|Changes the player's x pos

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, playercolour) %|Draws new player circle

Draw.FillOval (circlexposition, circleyposition, round (25 - (playermovement / 2) + 5) - 1, round (25 - (playermovement / 2) + 5) - 1, playercenter) %|Draws new player circle

end if %|End If

end if %|End If

if keys ('s') = true then %|If 's' is pressed then:

if circleyposition <= 150 then %|As long the yposition is less than 150 then:

else %|Else

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, 214) %|"Erases" the original circle

circleyposition := circleyposition - playermovement %|Changes the player's y pos

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, playercolour) %|Draws new player circle

Draw.FillOval (circlexposition, circleyposition, round (25 - (playermovement / 2) + 5) - 1, round (25 - (playermovement / 2) + 5) - 1, playercenter) %|Draws new player circle

end if %|End If

end if %|End If

if keys ('w') = true then %|If 'w' is pressed then:

if circleyposition >= 770 then %|As long the yposition is greater than 770 then:

else %|Else

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, 214) %|"Erases" the original circle

circleyposition := circleyposition + playermovement %|Changes the player's y pos

Draw.FillOval (circlexposition, circleyposition, round (30 - (playermovement / 2) + 5) - 1, round (30 - (playermovement / 2) + 5) - 1, playercolour) %|Draws new player circle

Draw.FillOval (circlexposition, circleyposition, round (25 - (playermovement / 2) + 5) - 1, round (25 - (playermovement / 2) + 5) - 1, playercenter) %|Draws new player circle

end if %|End If

end if %|End If

Draw.FillBox (20, 20, 240, 95, 23) %|This draws the control panel

Draw.FillBox (280, 60, 680, 100, 17) %|This draws the control panel

Draw.FillBox (280, 60, healthbar, 100, healthcolour) %|This draws the control panel

Draw.FillBox (280, 10, 680, 50, 17) %|This draws the control panel

Draw.FillBox (280, 10, bulletanimationx, 50, 43) %|This draws the control panel

Draw.FillBox (660, 10, 670, 50, 21) %|This draws the control panel

Draw.FillBox (640, 10, 650, 50, 21) %|This draws the control panel

Draw.FillBox (620, 10, 630, 50, 21) %|This draws the control panel

Draw.FillBox (600, 10, 610, 50, 21) %|This draws the control panel

Draw.FillBox (580, 10, 590, 50, 21) %|This draws the control panel

Draw.FillBox (560, 10, 570, 50, 21) %|This draws the control panel

Draw.FillBox (540, 10, 550, 50, 21) %|This draws the control panel

Draw.FillBox (520, 10, 530, 50, 21) %|This draws the control panel

Draw.FillBox (500, 10, 510, 50, 21) %|This draws the control panel

Draw.FillBox (480, 10, 490, 50, 21) %|This draws the control panel

Draw.FillBox (460, 10, 470, 50, 21) %|This draws the control panel

Draw.FillBox (440, 10, 450, 50, 21) %|This draws the control panel

Draw.FillBox (420, 10, 430, 50, 21) %|This draws the control panel

Draw.FillBox (400, 10, 410, 50, 21) %|This draws the control panel

Draw.FillBox (380, 10, 390, 50, 21) %|This draws the control panel

Draw.FillBox (360, 10, 370, 50, 21) %|This draws the control panel

Draw.FillBox (340, 10, 350, 50, 21) %|This draws the control panel

Draw.FillBox (320, 10, 330, 50, 21) %|This draws the control panel

Draw.FillBox (300, 10, 310, 50, 21) %|This draws the control panel

Draw.FillBox (280, 10, 290, 50, 21) %|This draws the control panel

Font.Draw (intstr (round (((healthbar - 280) / 4))), 700, 70, font6, 7) %|This draws the control panel

Font.Draw (intstr (ammo), 700, 15, font6, 7) %|This draws the control panel

Font.Draw ("Kills: " + intstr (killcounter), 45, 38, font3, 7) %|This draws the control panel

if (healthbar - 280) / 4 <= 75 then %|If the player's health is >75

healthcolour := 45 %|This changes the colour of the health

end if %|End If

if (healthbar - 280) / 4 <= 50 then %|If the player's health is >50

healthcolour := 41 %|This changes the colour of the health

end if %|End If

if (healthbar - 280) / 4 <= 25 then %|If the player's health is >25

healthcolour := 40 %|This changes the colour of the health

end if %|End If

if (healthbar - 280) / 4 <= 0 then %|If the player's health is >0

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

for i : 1 .. 860 %|This increases the box size

Draw.FillBox (0, 0, i, i, 7) %|This draws the growing box

delay (3) %|Slows the growing for smooth animation

end for %|End For

Music.PlayFileStop %|This stops the play of music

musiclose %|This plays the closing death music

Font.Draw ("You Died!", 15, 610, font7, 5) %|This prints text

Font.Draw ("All of humanity is in ruins!", 10, 510, font1, 55) %|This prints text

Font.Draw ("Sagar Patel", 210, 410, font1, 55) %|This prints text

Font.Draw ("April 28th, 2019", 140, 310, font1, 55) %|This prints text

Font.Draw ("Darren Moore", 168, 210, font1, 55) %|This prints text

Font.Draw ("ICS201", 280, 110, font1, 55) %|This prints text

Draw.FillBox (50, 10, 750, 70, 52) %|This draws a rounded rectangle

Draw.FillOval (50, 40, 30, 30, 52) %|This draws a rounded rectangle

Draw.FillOval (750, 40, 30, 30, 52) %|This draws a rounded rectangle

delay (5000) %|This waits 6 seconds

Window.Close (winID) %|This closes the screen

Window.Close (winID) %|This closes the game

end if %|End If

if keys ('r') = true and ammo = 0 then %|If 'r' is presses and ammo = 0:

fork reload %|call the reload procedure

end if %|End If

if keys (' ') = false then %|If the space is not presses then:

firing := false %|Firing is false

end if %|End If

if zombiehealth1 <= 0 and killtandf1 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf1 := false %|Make the counter stop after that

end if %|End If

if zombiehealth2 <= 0 and killtandf2 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf2 := false %|Make the counter stop after that

end if %|End If

if zombiehealth3 <= 0 and killtandf3 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf3 := false %|Make the counter stop after that

end if %|End If

if zombiehealth4 <= 0 and killtandf4 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf4 := false %|Make the counter stop after that

end if %|End If

if zombiehealth5 <= 0 and killtandf5 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf5 := false %|Make the counter stop after that

end if %|End If

if zombiehealth6 <= 0 and killtandf6 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf6 := false %|Make the counter stop after that

end if %|End If

if zombiehealth7 <= 0 and killtandf7 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf7 := false %|Make the counter stop after that

end if %|End If

if zombiehealth8 <= 0 and killtandf8 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf8 := false %|Make the counter stop after that

end if %|End If

if zombiehealth9 <= 0 and killtandf9 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf9 := false %|Make the counter stop after that

end if %|End If

if zombiehealth10 <= 0 and killtandf10 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf10 := false %|Make the counter stop after that

end if %|End If

if zombiehealth11 <= 0 and killtandf11 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf11 := false %|Make the counter stop after that

end if %|End If

if zombiehealth12 <= 0 and killtandf12 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf12 := false %|Make the counter stop after that

end if %|End If

if zombiehealth13 <= 0 and killtandf13 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf13 := false %|Make the counter stop after that

end if %|End If

if zombiehealth14 <= 0 and killtandf14 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf14 := false %|Make the counter stop after that

end if %|End If

if zombiehealth15 <= 0 and killtandf15 = true then %|If the zombie is dead then:

killcounter += 1 %|Add one to the kill counter

killtandf15 := false %|Make the counter stop after that

end if %|End If

end mainmovementprocess %|This ends the process

proc variablereset %|This is to reset the variables after a wave

zombiehealth1 := 60 %|Variable Reset

zombiecolour1 := 47 %|Variable Reset

zombiedamage1 := 0 %|Variable Reset

zombiex1 := Rand.Int (-300, -200) %|Variable Reset

zombiey1 := Rand.Int (800, 1000) %|Variable Reset

killtandf1 := true %|Variable Reset

zombiehealth2 := 60 %|Variable Reset

zombiecolour2 := 47 %|Variable Reset

zombiedamage2 := 0 %|Variable Reset

zombiex2 := Rand.Int (0, 800) %|Variable Reset

zombiey2 := Rand.Int (800, 1000) %|Variable Reset

killtandf2 := true %|Variable Reset

zombiehealth3 := 60 %|Variable Reset

zombiecolour3 := 47 %|Variable Reset

zombiedamage3 := 0 %|Variable Reset

zombiex3 := Rand.Int (800, 900) %|Variable Reset

zombiey3 := Rand.Int (0, 800) %|Variable Reset

killtandf3 := true %|Variable Reset

zombiehealth4 := 60 %|Variable Reset

zombiecolour4 := 47 %|Variable Reset

zombiedamage4 := 0 %|Variable Reset

zombiex4 := Rand.Int (0, 800) %|Variable Reset

zombiey4 := Rand.Int (800, 1000) %|Variable Reset

killtandf4 := true %|Variable Reset

zombiehealth5 := 60 %|Variable Reset

zombiecolour5 := 47 %|Variable Reset

zombiedamage5 := 0 %|Variable Reset

zombiex5 := Rand.Int (0, 800) %|Variable Reset

zombiey5 := Rand.Int (-30, -20) %|Variable Reset

killtandf5 := true %|Variable Reset

zombiehealth6 := 60 %|Variable Reset

zombiecolour6 := 47 %|Variable Reset

zombiedamage6 := 0 %|Variable Reset

zombiex6 := Rand.Int (0, 800) %|Variable Reset

zombiey6 := Rand.Int (-30, -20) %|Variable Reset

killtandf6 := true %|Variable Reset

zombiehealth7 := 60 %|Variable Reset

zombiecolour7 := 47 %|Variable Reset

zombiedamage7 := 0 %|Variable Reset

zombiex7 := Rand.Int (0, 800) %|Variable Reset

zombiey7 := Rand.Int (-30, -20) %|Variable Reset

killtandf7 := true %|Variable Reset

zombiehealth8 := 60 %|Variable Reset

zombiecolour8 := 47 %|Variable Reset

zombiedamage8 := 0 %|Variable Reset

zombiex8 := Rand.Int (0, 800) %|Variable Reset

zombiey8 := Rand.Int (-30, -20) %|Variable Reset

killtandf8 := true %|Variable Reset

zombiehealth9 := 60 %|Variable Reset

zombiecolour9 := 47 %|Variable Reset

zombiedamage9 := 0 %|Variable Reset

zombiex9 := Rand.Int (0, 800) %|Variable Reset

zombiey9 := Rand.Int (-30, -20) %|Variable Reset

killtandf9 := true %|Variable Reset

zombiehealth10 := 60 %|Variable Reset

zombiecolour10 := 47 %|Variable Reset

zombiedamage10 := 0 %|Variable Reset

zombiex10 := Rand.Int (0, 800) %|Variable Reset

zombiey10 := Rand.Int (-30, -20) %|Variable Reset

killtandf10 := true %|Variable Reset

ammo := 20 %|Variable Reset

bulletanimationx := 680 %|Variable Reset

healthbar := 680 %|Variable Reset

healthcolour := 48 %|Variable Reset

circlexposition := maxx div 2 %|Variable Reset

circleyposition := maxy div 2 %|Variable Reset

upgradedebug1 := false %|Variable Reset

upgradedebug2 := false %|Variable Reset

upgradedebug3 := false %|Variable Reset

upgradedebug4 := false %|Variable Reset

upgradedebug10 := false %|Variable Reset

upgradedebug20 := false %|Variable Reset

upgradedebug30 := false %|Variable Reset

upgradedebug40 := false %|Variable Reset

upgradedebug100 := false %|Variable Reset

upgradedebug200 := false %|Variable Reset

upgradedebug300 := false %|Variable Reset

upgradedebug400 := false %|Variable Reset

zombiex12 := Rand.Int (-50, -30) %|Variable Reset

zombiey12 := Rand.Int (820, 850) %|Variable Reset

zombiex13 := Rand.Int (-50, -30) %|Variable Reset

zombiey13 := Rand.Int (-50, -30) %|Variable Reset

zombiex14 := Rand.Int (830, 850) %|Variable Reset

zombiey14 := Rand.Int (820, 850) %|Variable Reset

zombiex15 := Rand.Int (830, 850) %|Variable Reset

zombiey15 := Rand.Int (-50, -30) %|Variable Reset

zombiehealth12 := 1000 %|Variable Reset

zombiehealth13 := 1000 %|Variable Reset

zombiehealth14 := 1000 %|Variable Reset

zombiehealth15 := 1000 %|Variable Reset

end variablereset %|Ends the process

proc completescreen %|This is the wave complete process

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (0, 0, maxx, maxy, 214) %|Colours the background

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a white circle

Font.Draw ("Wave", 230, 430, font9, 7) %|This prints text

Font.Draw ("Complete", 70, 280, font9, 7) %|This prints text

delay (1000) %|This delays for 1 second

end completescreen %|This ends the complete screen process

proc upgrade %|This is the upgrade text process

Draw.FillBox (0, 0, 800, 800, 30) %|This colours the background

mousecrosshairs %|This is calling the crosshair process

Font.Draw ("Upgrades", 10, 610, font7, 17) %|This is the title

Font.Draw ("Click a button to upgrade!", 230, 540, font2, 17) %|This is the instructions

Draw.FillBox (50, 50, 350, 240, 17) %|This draws the button

Font.Draw ("STRONGER", 70, 170, font3, 40) %|text

Font.Draw ("BULLETS", 70, 70, font3, 40) %|text

Draw.FillBox (450, 50, 750, 240, 17) %|This draws the button

Font.Draw ("FASTER", 470, 170, font3, 44) %|text

Font.Draw ("MOVEMENT", 470, 70, font3, 44) %|text

Draw.FillBox (50, 300, 350, 490, 17) %|This draws the button

Font.Draw ("BIGGER", 100, 425, font3, 55) %|text

Font.Draw ("BULLETS", 100, 325, font3, 55) %|text

Draw.FillBox (450, 300, 750, 490, 17) %|This draws the button

Font.Draw ("FASTER", 500, 425, font3, 48) %|text

Font.Draw ("BULLETS", 500, 325, font3, 48) %|text

end upgrade %|This ends the upgrade proccess

proc pausemenuprocedure %|This begins the pause menu process

mousewhere (x, y, b) %|Gathers information for the cursor

if b = 1 and x >= 10 and x <= 60 and y <= 790 and y >= 730 then %|If the pause button is pressed then:

pausedebug := true %|Variable is true

end if %|End If

end pausemenuprocedure %|Ends the pause button process

proc pausemenu %|This is the process for the pause button

mousewhere (x, y, b) %|Gathers information for the cursor

Draw.FillBox (0, 0, 800, 800, 17) %|Colours background

Font.Draw ("PAUSE", 30, 600, font5, white) %|Text

Draw.FillBox (50, 300, 250, 500, 246) %|Button/Logo

Draw.FillOval (50, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (50, 500, 10, 10, 246) %|Button/Logo

Draw.FillOval (250, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (250, 500, 10, 10, 246) %|Button/Logo

Draw.FillBox (40, 300, 260, 500, 246) %|Button/Logo

Draw.FillBox (50, 290, 250, 510, 246) %|Button/Logo

Draw.FillBox (300, 300, 500, 500, 246) %|Button/Logo

Draw.FillOval (300, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (500, 500, 10, 10, 246) %|Button/Logo

Draw.FillOval (500, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (300, 500, 10, 10, 246) %|Button/Logo

Draw.FillBox (300, 290, 500, 510, 246) %|Button/Logo

Draw.FillBox (290, 300, 510, 500, 246) %|Button/Logo

Draw.FillBox (550, 300, 750, 500, 246) %|Button/Logo

Draw.FillOval (550, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (550, 500, 10, 10, 246) %|Button/Logo

Draw.FillOval (750, 300, 10, 10, 246) %|Button/Logo

Draw.FillOval (750, 500, 10, 10, 246) %|Button/Logo

Draw.FillBox (550, 290, 750, 510, 246) %|Button/Logo

Draw.FillBox (540, 300, 760, 500, 246) %|Button/Logo

Draw.ThickLine (100 + 290, 100 + 230, 100 + 290, 150 + 230, 28, 8) %|Button/Logo

Draw.ThickLine (100 + 290, 100 + 230, 130 + 290, 125 + 230, 28, 8) %|Button/Logo

Draw.ThickLine (130 + 290, 125 + 230, 100 + 290, 150 + 230, 28, 8) %|Button/Logo

Draw.FillOval (650, 350, 30, 30, 8) %|Button/Logo

Draw.FillOval (150, 350, 30, 30, 8) %|Button/Logo

Draw.FillOval (150, 350, 20, 20, 246) %|Button/Logo

Draw.FillBox (140, 350, 160, 390, 246) %|Button/Logo

Draw.ThickLine (150, 360, 150, 390, 15, 8) %|Button/Logo

Font.Draw ("Quit", 110, 430, font3, 8) %|Text

Font.Draw ("Resume", 305, 430, font3, 8) %|Text

Font.Draw ("Rules", 590, 430, font3, 8) %|Text

Font.Draw ("Sagar Patel", 250, 210, font3, 8) %|Text

Font.Draw ("April 28th, 2019", 190, 150, font3, 8) %|Text

Font.Draw ("Darren Moore", 208, 90, font3, 8) %|Text

if x > 40 and x < 260 and y > 290 and y < 510 then %|If the cursor enters these places then:

if b = 0 then %|If the button is not clicked then:

Font.Draw ("PAUSE", 30, 600, font5, 246) %|Text

Draw.FillBox (50, 300, 250, 500, white) %|Button/Logo changes

Draw.FillOval (50, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (50, 500, 10, 10, white) %|Button/Logo changes

Draw.FillOval (250, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (250, 500, 10, 10, white) %|Button/Logo changes

Draw.FillBox (40, 300, 260, 500, white) %|Button/Logo changes

Draw.FillBox (50, 290, 250, 510, white) %|Button/Logo changes

Draw.FillOval (150, 350, 30, 30, 246) %|Button/Logo changes

Draw.FillOval (150, 350, 20, 20, white) %|Button/Logo changes

Draw.FillBox (140, 350, 160, 390, white) %|Button/Logo changes

Draw.ThickLine (150, 360, 150, 390, 15, 246) %|Button/Logo changes

Font.Draw ("Quit", 110, 430, font3, 246) %|Text

else %|Else

Draw.FillBox (50, 300, 250, 500, white) %|Button/Logo changes

Draw.FillOval (50, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (50, 500, 10, 10, white) %|Button/Logo changes

Draw.FillOval (250, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (250, 500, 10, 10, white) %|Button/Logo changes

Draw.FillBox (40, 300, 260, 500, white) %|Button/Logo changes

Draw.FillBox (50, 290, 250, 510, white) %|Button/Logo changes

Draw.FillOval (150, 350, 30, 30, 246) %|Button/Logo changes

Draw.FillOval (150, 350, 20, 20, white) %|Button/Logo changes

Draw.FillBox (140, 350, 160, 390, white) %|Button/Logo changes

Draw.ThickLine (150, 360, 150, 390, 15, 246) %|Button/Logo changes

Font.Draw ("Quit", 110, 430, font3, 246) %|Text

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

for i : 1 .. 860 %|For statement

Draw.FillBox (0, 0, i, i, 7) %|Draws a growing box

delay (3) %|Delay for smooth animation

end for %|End for

Window.Close (winID) %|Closes window

Window.Close (winID) %|Closes window

end if %|End If

end if %|End If

if x > 290 and x < 510 and y > 290 and y < 510 then %|IF the cursor is in this area then:

if b = 0 then %|If the button isn't clicked then:

Draw.FillBox (300, 300, 500, 500, white) %|Button/Logo changes

Draw.FillOval (300, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (500, 500, 10, 10, white) %|Button/Logo changes

Draw.FillOval (500, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (300, 500, 10, 10, white) %|Button/Logo changes

Draw.FillBox (300, 290, 500, 510, white) %|Button/Logo changes

Draw.FillBox (290, 300, 510, 500, white) %|Button/Logo changes

Font.Draw ("Resume", 305, 430, font3, 246) %|Text

Draw.ThickLine (100 + 290, 100 + 230, 100 + 290, 150 + 230, 28, 246) %|Button/Logo changes

Draw.ThickLine (100 + 290, 100 + 230, 130 + 290, 125 + 230, 28, 246) %|Button/Logo changes

Draw.ThickLine (130 + 290, 125 + 230, 100 + 290, 150 + 230, 28, 246) %|Button/Logo changes

else %|Else

Draw.FillBox (300, 300, 500, 500, white) %|Button/Logo changes

Draw.FillOval (300, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (500, 500, 10, 10, white) %|Button/Logo changes

Draw.FillOval (500, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (300, 500, 10, 10, white) %|Button/Logo changes

Draw.FillBox (300, 290, 500, 510, white) %|Button/Logo changes

Draw.FillBox (290, 300, 510, 500, white) %|Button/Logo changes

Font.Draw ("Resume", 305, 430, font3, 246) %|Text

Draw.ThickLine (100 + 290, 100 + 230, 100 + 290, 150 + 230, 28, 246) %|Button/Logo changes

Draw.ThickLine (100 + 290, 100 + 230, 130 + 290, 125 + 230, 28, 246) %|Button/Logo changes

Draw.ThickLine (130 + 290, 125 + 230, 100 + 290, 150 + 230, 28, 246) %|Button/Logo changes

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (0, 0, maxx, maxy, 214) %|Draws new background

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a white oval

Font.Draw ("RESUMING", 25, 400, font7, 7) %|Text

Font.Draw ("IN...", 310, 270, font7, 7) %|Text

delay (1000) %|Delay 1 second

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a white oval

Font.Draw ("3", 230, 200, font8, 7) %|Countdown text

delay (1000) %|Delay 1 second

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a white oval

Font.Draw ("2", 230, 200, font8, 7) %|Countdown text

delay (1000) %|Delay 1 second

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a white oval

Font.Draw ("1", 230, 200, font8, 7) %|Countdown text

delay (1000) %|Delay 1 second

View.Set ("offscreenonly") %|This resets the screen so it doesn't crash

pausedebug := false %|The pause button is then false

end if %|End If

end if %|End If

if x > 540 and x < 760 and y > 290 and y < 510 then %|If the cursor enters a certain area then:

if b = 0 then %|If the button is not clicked then:

Font.Draw ("PAUSE", 30, 600, font5, 246) %|Text

Draw.FillBox (550, 300, 750, 500, white) %|Button/Logo changes

Draw.FillOval (550, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (550, 500, 10, 10, white) %|Button/Logo changes

Draw.FillOval (750, 300, 10, 10, white) %|Button/Logo changes

Draw.FillOval (750, 500, 10, 10, white) %|Button/Logo changes

Draw.FillBox (550, 290, 750, 510, white) %|Button/Logo changes

Draw.FillBox (540, 300, 760, 500, white) %|Button/Logo changes

Draw.FillOval (650, 350, 30, 30, 246) %|Button/Logo changes

Font.Draw ("Rules", 590, 430, font3, 246) %|Text

else %|Else

loop %|Loop

View.Set ("offscreenonly") %|This sets the screen type

mousewhere (x, y, b) %|Gathers information for the cursor

Draw.FillBox (0, 0, maxx, maxy, 17) %|Button Box

Draw.FillBox (50, 70, 380, 200, 246) %|Button Box

Draw.FillBox (420, 70, 750, 200, 246) %|Button Box

Font.Draw ("RULES", 80, 550, font5, white) %|Text

Font.Draw ("1. Use 'wasd' to move the character!", 80, 450, font6, white) %|Text

Font.Draw ("2. Spacebar to shoot the zombies!", 80, 400, font6, white) %|Text

Font.Draw ("3. Try to survive all the waves!", 80, 350, font6, white) %|Text

Font.Draw ("4. You can upgrade your character!", 80, 300, font6, white) %|Text

Font.Draw ("5. Click 'r' to reload when empty!", 80, 250, font6, white) %|Text

Font.Draw ("QUIT", 140, 110, font1, white) %|Text

Font.Draw ("GO BACK", 445, 110, font1, white) %|Text

if b = 0 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|If the cursor hovers over the button then:

Draw.FillBox (420, 70, 750, 200, white) %|New button colour

Font.Draw ("GO BACK", 445, 110, font1, 246) %|Text

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|If the button is pressed

Draw.FillBox (420, 70, 750, 200, white) %|Draws new button

Font.Draw ("GO BACK", 445, 110, font1, 246) %|Text

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|If the cursor hovers over this button then:

Draw.FillBox (50, 70, 380, 200, white) %|Draws the new button

Font.Draw ("QUIT", 140, 110, font1, 246) %|Text

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|If the quit button is pressed then:

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (50, 70, 380, 200, white) %|This draws the new button

Font.Draw ("QUIT", 140, 110, font1, 246) %|Text

quitgameanimation %|This calls the quit animation

end if %|End If

View.Update %|Updates the screen

exit when b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 %|Exits the loop if the resum button is pressed

end loop %|This ends the loop

end if %|End If

end if %|End If

end pausemenu %|This ends the pause menu process

proc upgradestrength %|This begins the process

Draw.FillBox (50, 50, 350, 240, 40) %|Draws the button box

Font.Draw ("STRONGER", 70, 170, font3, 17) %|Text

Font.Draw ("BULLETS", 70, 70, font3, 17) %|Text

end upgradestrength %|End the music process

proc upgradespeed %|This begins the process

Draw.FillBox (450, 50, 750, 240, 44) %|Draws the button box

Font.Draw ("FASTER", 470, 170, font3, 17) %|Text

Font.Draw ("MOVEMENT", 470, 70, font3, 17) %|Text

end upgradespeed %|End the music process

proc upgradesize %|This begins the process

Draw.FillBox (50, 300, 350, 490, 55) %|Draws the button box

Font.Draw ("BIGGER", 100, 425, font3, 17) %|Text

Font.Draw ("BULLETS", 100, 325, font3, 17) %|Text

end upgradesize %|End the music process

proc upgradefast %|This begins the process

Draw.FillBox (450, 300, 750, 490, 48) %|Draws the button box

Font.Draw ("FASTER", 500, 425, font3, 17) %|Text

Font.Draw ("BULLETS", 500, 325, font3, 17) %|Text

end upgradefast %|End the music process

%------------------------------------Font-----------------------------------------------------------------------------------------------------------------------------------%|

font1 := Font.New ("Bauhaus 93:50") %|These are the fonts

font2 := Font.New ("Berlin Sans FB Demi:20") %|These are the fonts

font3 := Font.New ("Bauhaus 93:40") %|These are the fonts

font4 := Font.New ("Bauhaus 93:60") %|These are the fonts

font5 := Font.New ("Bauhaus 93:200") %|These are the fonts

font6 := Font.New ("Berlin Sans FB Demi:30") %|These are the fonts

font7 := Font.New ("Bauhaus 93:130") %|These are the fonts

font8 := Font.New ("Bauhaus 93:500") %|These are the fonts

font9 := Font.New ("Bauhaus 93:110") %|These are the fonts

%------------------------------------Start Menu-----------------------------------------------------------------------------------------------------------------------------%|

loop %|Loop

Music.PlayFileStop %|Stops the music

musicmenu %|Plays the menu music

View.Set ("offscreenonly") %|Sets the screen type

loop %|Loop

Draw.FillBox (0, 0, maxx, maxy, 214) %|Colours background

Font.Draw ("SURVIVAL ADVENTURE", 20, 690, font4, white) %|Text

Font.Draw ("Developed and Designed by Sagar Patel", 150, 10, font2, white) %|Text

Draw.FillBox (50, 500, 750, 600, 249) %|Draws button

Draw.FillBox (50, 350, 750, 450, 250) %|Draws button

Draw.FillBox (50, 200, 750, 300, 252) %|Draws button

Font.Draw ("START", 300, 525, font1, white) %|Text

Font.Draw ("INSTRUCTIONS", 180, 380, font1, white) %|Text

Font.Draw ("QUIT", 310, 230, font1, white) %|Text

mousecrosshairs %|This is calling the crosshair process

if b = 0 and x >= 50 and x <= 750 and y >= 500 and y <= 600 then %|If the button is hovered then:

Draw.FillBox (50, 500, 750, 600, white) %|Draws new button

Font.Draw ("START", 300, 525, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

end if %|End If

if b = 1 and x >= 50 and x <= 750 and y >= 500 and y <= 600 then %|If the button is pressed then:

Draw.FillBox (50, 500, 750, 600, white) %|Draws new button

Font.Draw ("START", 300, 525, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

exit %|exits the loop if it's true

end if %|End If

if b = 0 and x >= 50 and x <= 750 and y >= 350 and y <= 450 then %|If the button is hovered then:

Draw.FillBox (50, 350, 750, 450, white) %|Draws new button

Font.Draw ("INSTRUCTIONS", 180, 380, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

end if %|End If

if b = 1 and x >= 50 and x <= 750 and y >= 350 and y <= 450 then %|If the button is pressed then:

Draw.FillBox (50, 350, 750, 450, white) %|Draws new button

Font.Draw ("INSTRUCTIONS", 180, 380, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

loop %|Loop

Draw.FillBox (0, 0, maxx, maxy, 214) %|Colours the background

Draw.FillBox (50, 70, 380, 200, 250) %|Draws Button

Draw.FillBox (420, 70, 750, 200, 250) %|Draws Button

Font.Draw ("RULES", 80, 550, font5, white) %|Text

Font.Draw ("1. Use 'wasd' to move the character!", 80, 450, font6, white) %|Text

Font.Draw ("2. Spacebar to shoot the zombies!", 80, 400, font6, white) %|Text

Font.Draw ("3. Try to survive all the waves!", 80, 350, font6, white) %|Text

Font.Draw ("4. You can upgrade your character!", 80, 300, font6, white) %|Text

Font.Draw ("5. Click 'r' to reload when empty!", 80, 250, font6, white) %|Text

mousecrosshairs %|This is calling the crosshair process

Font.Draw ("QUIT", 140, 110, font1, white) %|Text

Font.Draw ("GO BACK", 445, 110, font1, white) %|Text

if b = 0 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|If the cursor hovers over the button then:

Draw.FillBox (420, 70, 750, 200, white) %|Draws the button

Font.Draw ("GO BACK", 445, 110, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|If the button is pressed then:

Draw.FillBox (420, 70, 750, 200, white) %|Draws the button

Font.Draw ("GO BACK", 445, 110, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|IF the cursor hovers over it then:

Draw.FillBox (50, 70, 380, 200, white) %|Draws the new button

Font.Draw ("QUIT", 140, 110, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|If the button is pressed then:

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (50, 70, 380, 200, white) %|Draws the button

Font.Draw ("QUIT", 140, 110, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

quitgameanimation %|This calls the growing box animation

end if %|End If

View.Update %|Updates the screen

exit when b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 %|Exit the loop if this is true

end loop %|This ends the loop

end if %|End If

if b = 0 and x >= 50 and x <= 750 and y >= 200 and y <= 300 then %|IF the cursor hovers the button then:

Draw.FillBox (50, 200, 750, 300, white) %|Draws the new button

Font.Draw ("QUIT", 310, 230, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

end if %|End If

if b = 1 and x >= 50 and x <= 750 and y >= 200 and y <= 300 then %|If the button is presses, then:

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (50, 200, 750, 300, white) %|Draws the new Button

Font.Draw ("QUIT", 310, 230, font1, 249) %|Text

mousecrosshairs %|This is calling the crosshair process

quitgameanimation %|This calls the growing box animation

end if %|End If

shooterarm %|This calls the shooter arm process

Draw.FillOval (maxx div 2, 70, 30, 30, 6) %|This draws The player for intro

Draw.FillOval (maxx div 2, 70, 25, 25, 18) %|This draws The player for intro

View.Update %|Updates the screen

end loop %|This ends the loop

%------------------------------------Story----------------------------------------------------------------------------------------------------------------------------------%|

Music.PlayFileStop %|Stops the music from playing

musicstory %|Plays the story music

Draw.FillBox (0, 0, maxx, maxy, 7) %|Colours the background

for i : 16 .. 31 %|Causes the text to fade black to white

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (790, 90, 500, 10, i) %|Fades the button in

Font.Draw ("You are one of the only survivors in the", 20, 750, font6, i) %|Text

Font.Draw ("year 2173. The outbreak of the Death", 20, 690, font6, i) %|Text

Font.Draw ("Virus has affected billions around the", 20, 620, font6, i) %|Text

Font.Draw ("world and they have become zombies that", 20, 550, font6, i) %|Text

Font.Draw ("try to eat anything. You must survive a", 20, 480, font6, i) %|Text

Font.Draw ("total of 4 waves in order to survive the", 20, 410, font6, i) %|Text

Font.Draw ("virus. You have developed the technology", 20, 340, font6, i) %|Text

Font.Draw ("to track how much health you have, how", 20, 270, font6, i) %|Text

Font.Draw ("much ammo is in your clip, and how many", 20, 200, font6, i) %|Text

Font.Draw ("zombies you have killed. You begin", 20, 130, font6, i) %|Text

Font.Draw ("with 100 health...", 20, 60, font6, i) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

delay (100) %|Delays by 0.1 seconds

end for %|End for

loop %|Loop

View.Set ("offscreenonly") %|Sets the screen for loop

Draw.FillBox (0, 0, maxx, maxy, 7) %|Colours the background

Draw.FillBox (790, 90, 500, 10, 31) %|Colours the button

mousewhere (x, y, b) %|Calculates the mouse position

Font.Draw ("You are one of the only survivors in the", 20, 750, font6, white) %|Text

Font.Draw ("year 2173. The outbreak of the Death", 20, 690, font6, white) %|Text

Font.Draw ("Virus has affected billions around the", 20, 620, font6, white) %|Text

Font.Draw ("world and they have become zombies that", 20, 550, font6, white) %|Text

Font.Draw ("try to eat anything. You must survive a", 20, 480, font6, white) %|Text

Font.Draw ("total of 4 waves in order to survive the", 20, 410, font6, white) %|Text

Font.Draw ("virus. You have developed the technology", 20, 340, font6, white) %|Text

Font.Draw ("to track how much health you have, how", 20, 270, font6, white) %|Text

Font.Draw ("much ammo is in your clip, and how many", 20, 200, font6, white) %|Text

Font.Draw ("zombies you have killed. You begin", 20, 130, font6, white) %|Text

Font.Draw ("with 100 health...", 20, 60, font6, white) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

if b = 1 and x > 500 and x < 790 and y > 10 and y < 90 then %|If the requirements meet it will run

textparadebug := 10000 %|The variable is set to 10000

else %|Else

textparadebug := textparadebug + 1 %|The variable increases as time continues

end if %|End If

View.Update %|Updates the screen

exit when textparadebug >= 10000 %|Exits the loop if the variable is greater than 10000

end loop %|End loop

for decreasing i : 31 .. 16 %|Fades everything out

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (790, 90, 500, 10, i) %|Resets the screen

Font.Draw ("You are one of the only survivors in the", 20, 750, font6, i) %|Text

Font.Draw ("year 2173. The outbreak of the Death", 20, 690, font6, i) %|Text

Font.Draw ("Virus has affected billions around the", 20, 620, font6, i) %|Text

Font.Draw ("world and they have become zombies that", 20, 550, font6, i) %|Text

Font.Draw ("try to eat anything. You must survive a", 20, 480, font6, i) %|Text

Font.Draw ("total of 4 waves in order to survive the", 20, 410, font6, i) %|Text

Font.Draw ("virus. You have developed the technology", 20, 340, font6, i) %|Text

Font.Draw ("to track how much health you have, how", 20, 270, font6, i) %|Text

Font.Draw ("much ammo is in your clip, and how many", 20, 200, font6, i) %|Text

Font.Draw ("zombies you have killed. You begin", 20, 130, font6, i) %|Text

Font.Draw ("with 100 health...", 20, 60, font6, i) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

delay (100) %|Delays by 0.1 seconds

end for %|End for

Draw.FillBox (0, 0, maxx, maxy, 7) %|Colours Background

for i : 16 .. 31 %|Fades text in

Font.Draw ("and 20 bullets in each clip (an infinite", 20, 750, font6, i) %|Text

Font.Draw ("amount of clips). You must survive each", 20, 690, font6, i) %|Text

Font.Draw ("wave, and as each wave goes on, the", 20, 620, font6, i) %|Text

Font.Draw ("zombies get faster and sometimes more", 20, 550, font6, i) %|Text

Font.Draw ("will spawn. However, you will be", 20, 480, font6, i) %|Text

Font.Draw ("able to get upgrades as you progress", 20, 410, font6, i) %|Text

Font.Draw ("and you may be able to upgrade your", 20, 340, font6, i) %|Text

Font.Draw ("health. Your health restores to 100", 20, 270, font6, i) %|Text

Font.Draw ("every time a wave is done. Your objective", 20, 200, font6, i) %|Text

Font.Draw ("is to survive the 3 waves of the zombies!", 20, 130, font6, i) %|Text

Draw.FillBox (790, 90, 500, 10, i) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

delay (100) %|Delays by 0.1 seconds

end for %|End for

loop %|Loop

View.Set ("offscreenonly") %|Sets Screen

Draw.FillBox (0, 0, maxx, maxy, 7) %|Colours background

mousewhere (x, y, b) %|Calculates mouse position

Draw.FillBox (790, 90, 500, 10, 31) %|Colours button

Font.Draw ("and 20 bullets in each clip (an infinite", 20, 750, font6, white) %|Text

Font.Draw ("amount of clips). You must survive each", 20, 690, font6, white) %|Text

Font.Draw ("wave, and as each wave goes on, the", 20, 620, font6, white) %|Text

Font.Draw ("zombies get faster and sometimes more", 20, 550, font6, white) %|Text

Font.Draw ("will spawn. However, you will be", 20, 480, font6, white) %|Text

Font.Draw ("able to get upgrades as you progress", 20, 410, font6, white) %|Text

Font.Draw ("and you may be able to upgrade your", 20, 340, font6, white) %|Text

Font.Draw ("health. Your health restores to 100", 20, 270, font6, white) %|Text

Font.Draw ("every time a wave is done. Your objective", 20, 200, font6, white) %|Text

Font.Draw ("is to survive the 3 waves of the zombies!", 20, 130, font6, white) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

if b = 1 and x > 500 and x < 790 and y > 10 and y < 90 then %|If the requirements meet it will run

textparadebug2 := 10000 %|It increases the text when button is presses

else %|Else

textparadebug2 := textparadebug2 + 1 %|Increases the time counter

end if %|End If

View.Update %|Updates the screen

exit when textparadebug2 >= 10000 %|Exits after 10 seconds

end loop %|This ends the loop

for decreasing i : 31 .. 16 %|Decreases i, for colour

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (790, 90, 500, 10, i) %|Draws button

Font.Draw ("and 20 bullets in each clip (an infinite", 20, 750, font6, i) %|Text

Font.Draw ("amount of clips). You must survive each", 20, 690, font6, i) %|Text

Font.Draw ("wave, and as each wave goes on, the", 20, 620, font6, i) %|Text

Font.Draw ("zombies get faster and sometimes more", 20, 550, font6, i) %|Text

Font.Draw ("will spawn. However, you will be", 20, 480, font6, i) %|Text

Font.Draw ("able to get upgrades as you progress", 20, 410, font6, i) %|Text

Font.Draw ("and you may be able to upgrade your", 20, 340, font6, i) %|Text

Font.Draw ("health. Your health restores to 100", 20, 270, font6, i) %|Text

Font.Draw ("every time a wave is done. Your objective", 20, 200, font6, i) %|Text

Font.Draw ("is to survive the 3 waves of the zombies!", 20, 130, font6, i) %|Text

Font.Draw ("Continue", 560, 35, font6, 7) %|Text

delay (100) %|Delay 0.1 second

end for %|End For

Draw.FillBox (0, 0, maxx, maxy, 7) %|Colours background

for i : 1 .. 860 %|Increases box size

Draw.FillBox (0, 0, i, i, 214) %|Draws Growing Box

delay (3) %|Makes smooth animation for box

end for %|End For

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|This draws a circle behind text

Font.Draw ("STARTING", 45, 400, font7, 7) %|Text

Font.Draw ("IN...", 310, 270, font7, 7) %|Text

delay (1000) %|Delay 1 second

for decreasing i : 3 .. 1 %|This causes the number to decrease

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|This adds a cricle behind text

Font.Draw (intstr (i), 230, 200, font8, 7) %|Draws the 3,2,1 timer

delay (1000) %|Delay for 1 second

end for %|End For

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|This draws a while circle behind text

Font.Draw ("Wave 1", 130, 330, font7, 7) %|This draws text

delay (800) %|This delays by 1 second

Music.PlayFileStop %|This stops the current music

musicgame %|This plays the new music file

View.Set ("offscreenonly") %|This sets the screen type

%------------------------------------Game: Wave 1---------------------------------------------------------------------------------------------------------------------------%|

variablereset %|Resets the variable

musicgame %|PLays the music for the game

loop %|Loop

pausemenuprocedure %|This determines if the pause is presses

if pausedebug = true then %|If the pause button is clicked then:

pausemenu %|This calls the pause menu process

View.Update %|Updates the screen

else %|Else

Draw.FillBox (0, 0, maxx, maxy, 215) %|Colours the background

bulletshoot %|This calls the bullet process

bulletcollision1 %|This is the collision process for zombies

zombiemove1 %|This calls the zombie process

bulletcollision2 %|This is the collision process for zombies

zombiemove2 %|This calls the zombie process

bulletcollision3 %|This is the collision process for zombies

zombiemove3 %|This calls the zombie process

bulletcollision4 %|This is the collision process for zombies

zombiemove4 %|This calls the zombie process

bulletcollision5 %|This is the collision process for zombies

zombiemove5 %|This calls the zombie process

mainmovementprocess %|This calls the movement/main things process

View.Update %|Updates the screen

end if %|End If

exit when zombiehealth1 <= 0 and zombiehealth2 <= 0 and zombiehealth3 <= 0 and zombiehealth4 <= 0 and zombiehealth5 <= 0 %|This exits the loop when all zombies die

View.Update %|Updates the screen

end loop %|This ends the loop

completescreen %|Calls the "Wave Complete" Text

%------------------------------------Upgrades-------------------------------------------------------------------------------------------------------------------------------%|

loop %|Loop

View.Set ("offscreenonly") %|This sets the screen type

upgrade %|This prints the title and instructions

if b = 0 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|If the cursor is over the bullet strength

upgradestrength %|This prints the button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|If the cursor is over the button

upgradespeed %|This prints the button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|If the cursor is over the button

upgradesize %|This prints the button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|If the cursor is over the button

upgradefast %|This prints the button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|If the cursor is over the button

upgradestrength %|This prints the button and text

playercenter := 16 %|This changes the player center colour

bulletdamage := bulletdamage + 1 %|This increases bullet damage

upgradedebug100 := true %|This helps exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|If the cursor presses the button

upgradespeed %|This prints the button and text

playermovement := playermovement + 3 %|This increases the player movement

upgradedebug200 := true %|This helps exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|If the cursor presses the button

upgradesize %|This prints the button and text

bulletsize := bulletsize + 10 %|This increases the bullet size

upgradedebug300 := true %|This helps exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|If the cursor presses the button

upgradefast %|This prints the button and text

bulletspeed := bulletspeed + 2 %|This increases the bullet speed

upgradedebug400 := true %|This helps exit the loop

playercolour := 68 %|This changes the player's ring colour

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

mousecrosshairs %|This is calling the crosshair process

exit when upgradedebug100 = true or upgradedebug200 = true or upgradedebug300 = true or upgradedebug400 = true %|This exits when a button is presses

View.Update %|Updates the screen

end loop %|This ends the loop

%------------------------------------Reset----------------------------------------------------------------------------------------------------------------------------------%|

variablereset %|Resets the variable

%------------------------------------Game: Wave 2---------------------------------------------------------------------------------------------------------------------------%|

Draw.FillOval (maxx div 2, maxy div 2, 400, 400, white) %|Draws a Cricle Behind Text

Font.Draw ("Wave 1", 130, 330, font7, 7) %|Prints "Wave 1"

delay (1000) %|Delays by 1 second

loop %|Loop

View.Set ("offscreenonly") %|This sets the screen type

pausemenuprocedure %|This determines if the button is pressed

if pausedebug = true then %|IF the button is pressed, then:

pausemenu %|This calls the pause menu process

else %|Else:

Draw.FillBox (0, 0, maxx, maxy, 216) %|This colours the background

bulletshoot %|This calls the shooting bullets process

bulletcollision1 %|This calls the collision for zombie and bullets

zombiemove1 %|This calls the zombie assigned to this process

bulletcollision2 %|This calls the collision for zombie and bullets

zombiemove2 %|This calls the zombie assigned to this process

bulletcollision3 %|This calls the collision for zombie and bullets

zombiemove3 %|This calls the zombie assigned to this process

bulletcollision4 %|This calls the collision for zombie and bullets

zombiemove4 %|This calls the zombie assigned to this process

bulletcollision5 %|This calls the collision for zombie and bullets

zombiemove5 %|This calls the zombie assigned to this process

bulletcollision6 %|This calls the collision for zombie and bullets

zombiemove6 %|This calls the zombie assigned to this process

bulletcollision7 %|This calls the collision for zombie and bullets

zombiemove7 %|This calls the zombie assigned to this process

bulletcollision8 %|This calls the collision for zombie and bullets

zombiemove8 %|This calls the zombie assigned to this process

bulletcollision9 %|This calls the collision for zombie and bullets

zombiemove9 %|This calls the zombie assigned to this process

bulletcollision10 %|This calls the collision for zombie and bullets

zombiemove10 %|This calls the zombie assigned to this process

mainmovementprocess %|This calls the movement/main things process

View.Update %|Updates the screen

end if %|End If

View.Update %|Updates the screen

exit when zombiehealth1 <= 0 and zombiehealth2 <= 0 and zombiehealth3 <= 0 and zombiehealth4 <= 0 and zombiehealth5 <= 0 and %|Exits when all the zombies health are zero

zombiehealth6 <= 0 and zombiehealth7 <= 0 and zombiehealth8 <= 0 and zombiehealth9 <= 0 and zombiehealth10 <= 0 %|Exits when all the zombies health are zero

end loop %|This ends the loop

completescreen %|Calls the "Wave Complete" Text

%------------------------------------Upgrades-------------------------------------------------------------------------------------------------------------------------------%|

loop %|Loop

View.Set ("offscreenonly") %|This sets the screen type

upgrade %|This places the upgrade title and quick instructions

if b = 0 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|If the cursor is on the button:

upgradestrength %|This places the new button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|If the cursor is on the button:

upgradespeed %|This places the new button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|If the cursor is on the button:

upgradesize %|This places the new button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 0 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|If the cursor is on the button:

upgradefast %|This places the new button and text

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|If the button is pressed then:

upgradestrength %|This places the new button and text

playercenter := 248 %|Visual Upgrade: Changes player's center

bulletdamage := bulletdamage + 1 %|Increases bullet damage

upgradedebug1 := true %|This helps to exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|If the button is pressed then:

upgradespeed %|This places the new button and text

playermovement := playermovement + 3 %|Increases player movement

upgradedebug2 := true %|This helps to exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|If the button is pressed then:

upgradesize %|This places the new button and text

bulletsize := bulletsize + 10 %|This increases the bullet size

upgradedebug3 := true %|This helps to exit the loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End If

if b = 1 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|If the speed button is pressed then:

upgradefast %|This places the new button and text

bulletspeed := bulletspeed + 2 %|Increases the bullet speed

upgradedebug4 := true %|This helps to exit the loop

playercolour := 43 %|Visual Upgrade: Player Colour changes

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|Ends If

mousecrosshairs %|This is calling the crosshair process

exit when upgradedebug1 = true or upgradedebug2 = true or upgradedebug3 = true or upgradedebug4 = true %|It exits the loop when and upgrade is pressed

View.Update %|Updates the screen

end loop %|This ends the loop

variablereset %|Resets the variable

%----------------------------------------Game: Wave 3-----------------------------------------------------------------------------------------------------------------------%|Starts loop

loop %|Loop

pausemenuprocedure %|This calls the pause menu process

if pausedebug = true then %|If the button is pressed, pause is true

pausemenu %|This puts the pause menu if the pause is true

else %|Else

Draw.FillBox (0, 0, maxx, maxy, 214) %|Colours the background

bulletshoot %|This also calls bullet process for the player

bulletcollision11 %|This calls the collision for zombie and bullets

zombiemove11 %|This calls the zombie assigned to this process

mainmovementprocess %|This calls the movement/main things process

View.Update %|Updates the screen

end if %|End if

exit when zombiehealth11 <= 0 %|Exits when zombie dies

View.Update %|Updates the screen

end loop %|Ends loop

completescreen %|Calls the "Wave Complete" Text

%------------------------------------Upgrades-------------------------------------------------------------------------------------------------------------------------------%|

loop %|Loops the upgrade

View.Set ("offscreenonly") %|This sets the screen type

upgrade %|This calls the basic things (title, buttons, etc)

if b = 0 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|This if only runs if this is true

upgradestrength %|This calls the text to print this on the button

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 0 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|This if only runs if this is true

upgradespeed %|This calls the text to print this on the button

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 0 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|This if only runs if this is true

upgradesize %|This calls the text to print this on the button

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 0 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|This if only runs if this is true

upgradefast %|This calls the text to print this on the button

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 1 and x >= 50 and x <= 350 and y >= 50 and y <= 240 then %|This if only runs if this is true

upgradestrength %|This calls the text to print this on the button

bulletdamage := bulletdamage + 1 %|Increased bullet damage by 1

playercenter := 7 %|Changes the player center to 7

upgradedebug10 := true %|Changes the upgrade to false to exit loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 1 and x >= 450 and x <= 750 and y >= 50 and y <= 240 then %|This if only runs if this is true

upgradespeed %|This calls the text to print this on the button

playermovement := playermovement + 3 %|Increased player movement by 3

upgradedebug20 := true %|Changes the upgrade to false to exit loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 1 and x >= 50 and x <= 350 and y >= 300 and y <= 490 then %|This if only runs if this is true

upgradesize %|This calls the text to print this on the button

bulletsize := bulletsize + 10 %|Increased bullet size by 10

upgradedebug30 := true %|Changes the upgrade to false to exit loop

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

if b = 1 and x >= 450 and x <= 750 and y >= 300 and y <= 490 then %|This if only runs if this is true

upgradefast %|This calls the text to print this on the button

bulletspeed := bulletspeed + 2 %|Increased bullet speed by three

upgradedebug40 := true %|Changes the upgrade to false to exit loop

playercolour := 44 %|This changes the player's colour

mousecrosshairs %|This is calling the crosshair process

View.Update %|Updates the screen

end if %|End if

mousecrosshairs %|This is calling the crosshair process

exit when upgradedebug10 = true or upgradedebug20 = true or upgradedebug30 = true or upgradedebug40 = true %|This exits upgrade when a button is pressed

View.Update %|Updates the screen

end loop %|Ends the loop

%----------------------------------------Reset------------------------------------------------------------------------------------------------------------------------------%|

variablereset %|Resets the variable

%----------------------------------------Game: Wave 4-----------------------------------------------------------------------------------------------------------------------%|

loop %|This loops the wave until it can exit or die

View.Set ("offscreenonly") %|Sets the screen to prevent glitching

pausemenuprocedure %|This determines if pausedebug is f or t

if pausedebug = true then %|If the pause button is pressed

pausemenu %|This calls the pause menu

else %|Else

Draw.FillBox (0, 0, maxx, maxy, 215) %|This colours the background

bulletshoot %|This calls the bullet process

bulletcollision12 %|This calls the collision for zombie and bullets

zombiemove12 %|This calls the zombie assigned to this process

bulletcollision13 %|This calls the collision for zombie and bullets

zombiemove13 %|This calls the zombie assigned to this process

bulletcollision14 %|This calls the collision for zombie and bullets

zombiemove14 %|This calls the zombie assigned to this process

bulletcollision15 %|This calls the collision for zombie and bullets

zombiemove15 %|This calls the zombie assigned to this process

mainmovementprocess %|This calls the movement/main things process

View.Update %|Updates the screen

end if %|End if

exit when zombiehealth12 <= 0 and zombiehealth13 <= 0 and zombiehealth14 <= 0 and zombiehealth15 <= 0 %|Exits the loop when the zombies die

View.Update %|Updates the screen

end loop %|Ends the loop

completescreen %|Calls the "Wave Complete" Text

%-----------------------------------------Congrats!-------------------------------------------------------------------------------------------------------------------------%|

variablereset %|Resets all variables

circleyposition := 270 %|Puts player position in special location

View.Set ("offscreenonly") %|Sets the screen to prevent glitching

Music.PlayFileStop %|Stops previous music

musicwin %|Plays music for winning

loop %|Loops the congrats page

Draw.FillBox (0, 0, maxx, maxy, 25) %|Background colour

for i : 1 .. 30 %|Does confetti 30 times a second

randint (confettix, 0, 800) %|Random confetti position

randint (confettiy, 0, 800) %|Random confetti position

Draw.FillOval (confettix, confettiy, 2, 2, 49) %|This puts confetti

end for %|End for statement

Font.Draw ("Congrats!", 20, 625, font7, white) %|Text

Font.Draw ("You survived the virus", 65, 525, font1, white) %|Text

Font.Draw ("and saved all of humanity!", 10, 425, font1, white) %|Text

Font.Draw ("Kills: " + intstr (killcounter), 115, 325, font1, white) %|This prints the final kills

Font.Draw ("Sagar", 495, 325, font1, white) %|This prints my name

mousecrosshairs %|This calls the crosshairs process

shooterarm %|This calls the arm of the player

Draw.FillOval (maxx div 2, 270, 30, 30, 42) %|This is the player

Draw.FillOval (maxx div 2, 270, 25, 25, 18) %|This is the player

Draw.FillBox (50, 70, 380, 200, 42) %|This is the button box

Draw.FillBox (420, 70, 750, 200, 42) %|This is the button box

Font.Draw ("QUIT", 140, 110, font1, white) %|This is the button text

Font.Draw ("RETRY", 485, 110, font1, white) %|This is the button text

if b = 0 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|These are the requirements the run this

Draw.FillBox (420, 70, 750, 200, white) %|This is the button box

Font.Draw ("RETRY", 485, 110, font1, 249) %|This is the button text

mousecrosshairs %|This is calling the crosshair process

View.Update %|This updates the screen without glitching

end if %|This ends the end if

if b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 then %|These are the requirements the run this

Draw.FillBox (420, 70, 750, 200, white) %|This is the button box

Font.Draw ("RETRY", 485, 110, font1, 249) %|This is the button text

mousecrosshairs %|This is calling the crosshair process

exit when b = 1 and x >= 420 and x <= 750 and y >= 70 and y <= 200 %|exits the congrats loop

View.Update %|This updates the screen without glitching

end if %|This ends the end if

if b = 0 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|These are the requirements the run this

Draw.FillBox (50, 70, 380, 200, white) %|This is the button box

Font.Draw ("QUIT", 140, 110, font1, 249) %|This is the button text

mousecrosshairs %|This is calling the crosshair process

View.Update %|This updates the screen without glitching

end if %|This ends the end if

if b = 1 and x >= 50 and x <= 380 and y >= 70 and y <= 200 then %|These are the requirements the run this

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

Draw.FillBox (50, 70, 380, 200, white) %|This is the button box

Font.Draw ("QUIT", 140, 110, font1, 249) %|This is the button text

mousecrosshairs %|This is calling the crosshair process

quitgameanimation %|This makes the growinf box animation

end if %|This ends the end if

View.Update %|This updates the screen without glitching

end loop %|ends the congrats sloop

View.Set ("nooffscreenonly") %|This resets the screen so it doesn't crash

for i : 1 .. 860 %|This is animation to retry game

Draw.FillBox (0, 0, i, i, 214) %|This is animation to retry game

delay (3) %|This is animation to retry game

end for %|End for statement

circlexposition := maxx div 2 %|Resets the player position

circleyposition := 70 %|Resets the player position

bulletsize := 10 %|Resets the upgrades

bulletspeed := 3 %|Resets the upgrades

playermovement := 5 %|Resets the upgrades

bulletdamage := 1 %|Resets the upgrades

end loop %|This ends the loop of the entire game

%------------------------------------End Program----------------------------------------------------------------------------------------------------------------------------%|