

SilverCoders

DIGITAL LITERACY IMPROVEMENT THROUGH EFFECTIVE
LEARNING EXPERIENCES FOR ADULTS



CHALLENGE #27 ASTEROIDS

CODING TRAINING PROGRAMME FOR +55 ADULTS



SILVER CODERS

ERASMUS+ No. 2020-1-SE01-KA227-ADU-092582



Co-funded by
the European Union

This document reflects only the author's view and the National Agency and the European Commission are not responsible for any use that may be made of the information it contains

STRUCTURE OF THE CHALLENGE

DESCRIPTION

This game we are going to create now is similar to the well known Asteroids game.

GENERAL GOAL

In this challenge we are going to develop an Asteroids type game while learning some instructions that allows us to repeat several times the same instruction.

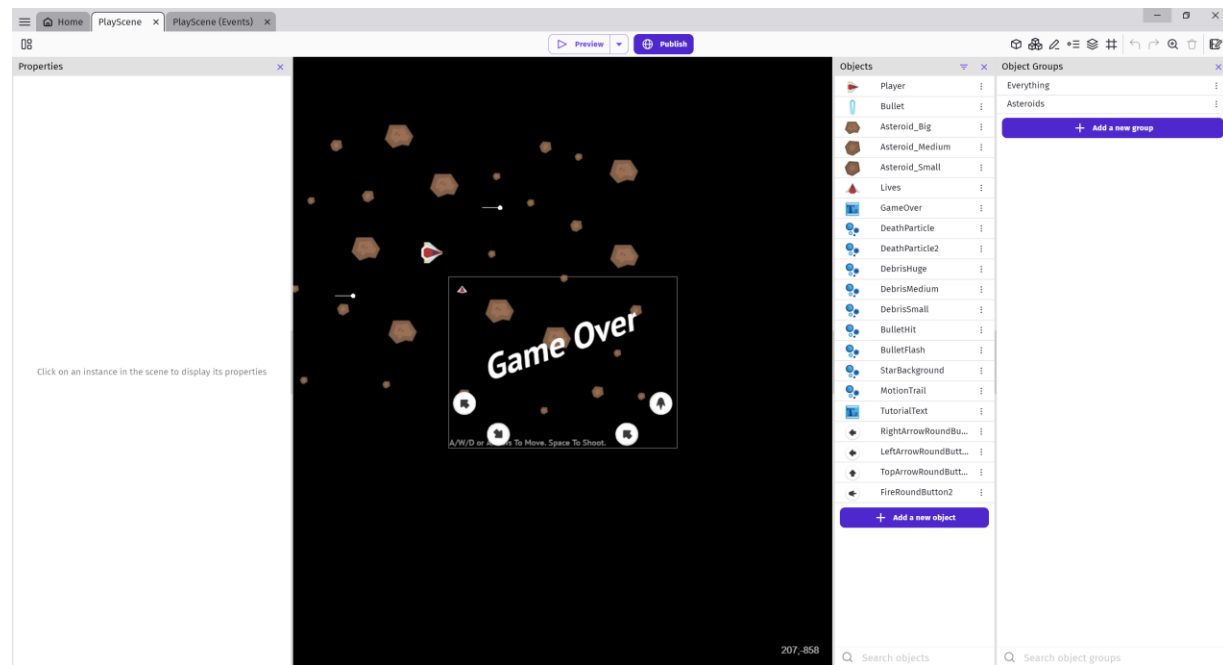
LEARNING OBJECTIVES

In the end of this challenge, you will be able ...:

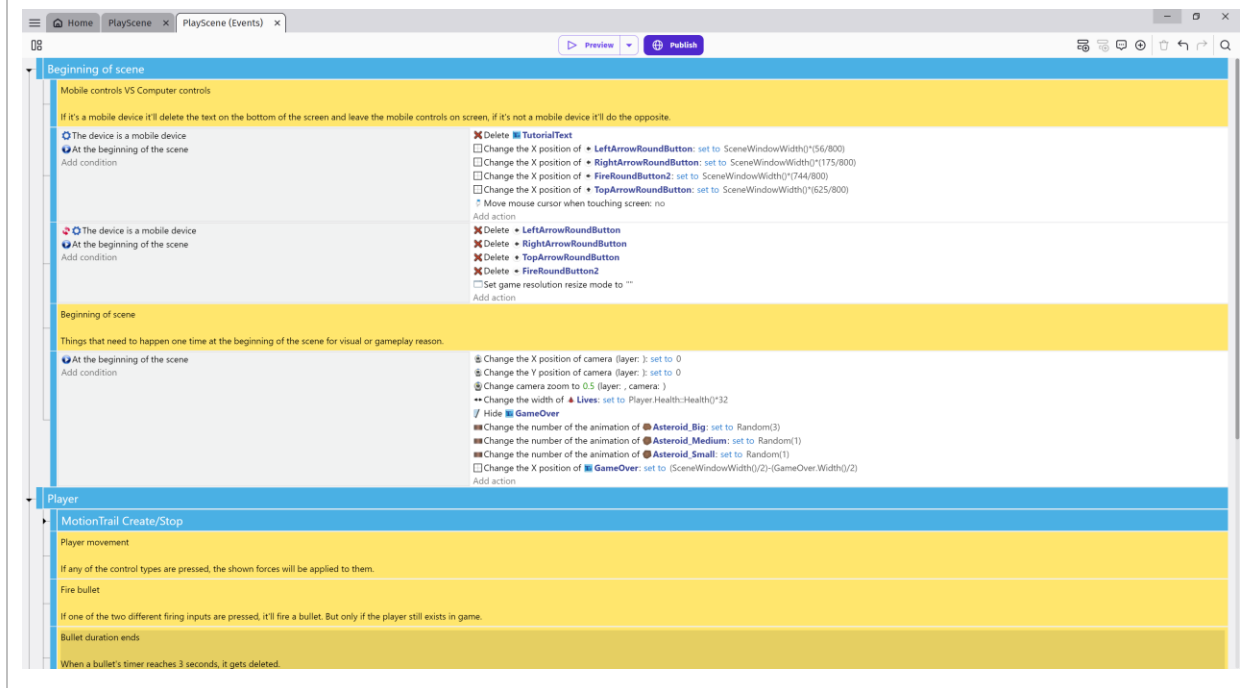
- To have experience with a visual programming suite and be able to code standard small piece of software with it.
- Know what statements and command lines are and what they mean for a compiler.
- To be able to write instructions using correct syntax and with minimal errors.
- Know what operators are, what they do and which symbols stand for which operators.
- To be able to understand the assignment of values to variables and how to change them.
- To know all the basic arithmetic operations and how to use them.
- Recognize and know how to use all the data structures related to numbers.
- To know the structures linked to the use of text, such as strings and characters.
- To be able to use If statements correctly to execute code according to a certain defined fixed condition.
- To be able to use loop control instructions.

INSTRUCTIONS

This is your initial setup. In this case we just provided the basic objects that you'll need for the game. As usual start by checking them carefully.



We also have the code that starts the game and we have the structure for the rest of the code.



Our objective is to create a game where we destroy asteroids when they are in the screen.

Let's start by allowing the player to control the ship, using keys or the mobile device.

MotionTrail Create/Stop	
Player movement	
If any of the control types are pressed, the shown forces will be applied to them.	
If one of these conditions is true: [A] w key is pressed [A] Up key is pressed [M] The cursor/touch is on + TopArrowRoundButton Add a sub-condition Add condition	Apply to Player a force of angle Player.Angle() and length 4.5 Add action
If one of these conditions is true: [A] a key is pressed [A] Left key is pressed [M] The cursor/touch is on + LeftArrowRoundButton Add a sub-condition Add condition	Apply to Player a torque of -0.5 Add action
If one of these conditions is true: [A] d key is pressed [A] Right key is pressed [M] The cursor/touch is on + RightArrowRoundButton Add a sub-condition Add condition	Apply to Player a torque of 0.5 Add action

We now define everything related to the firing and the bullets.

Fire bullet	
If one of the two different firing inputs are pressed, it'll fire a bullet. But only if the player still exists in game.	
If one of these conditions is true: [A] Space key is released [M] If all of these conditions are true: [M] The cursor/touch is on + FireRoundButton2 [M] A touch has ended Add a sub-condition Add a sub-condition [M] The number of Player objects > 0 [I] Trigger once Add condition	Play the sound LaserFire.wav , vol: 40, loop: no Create object Bullet at position Player.PointX("BulletSpawn");Player.PointY("BulletSpawn") (layer:) Create object BulletFlash at position Player.PointX("BulletFlash");Player.PointY("BulletFlash") (layer:) Rotate Bullet towards Player.Angle()+90 at speed 0 deg/second Rotate BulletFlash towards Player.Angle()+90 at speed 0 deg/second Add to Bullet a permanent force, angle: Player.Angle() degrees and length: 350 pixels Change the z-order of Bullet : set to Player.ZOrder()-2 Change the z-order of BulletFlash : set to Player.ZOrder()-1 Add action
Bullet duration timer	
When a bullet is fired, it starts a timer with that bullet so it can be deleted later.	
Add condition	Start (or reset) the timer "End" of Bullet Add action
Bullet duration ends	
When a bullet's timer reaches 3 seconds, it gets deleted. *This is important because otherwise your game could end up needlessly tracking hundreds/thousands of instances of the bullets by the end of a game.	
[M] The timer "End" of Bullet > 3 seconds Add condition	Delete Bullet Add action

We also address what happens if we shoot a bullet that hits nothing (remember we had this question a few challenges ago?).

Now we deal with getting hit and loosing.

Getting hurt	
If the player object collides with anything, other than the bullet because it doesn't have the physics behavior, then it gets hurt.	
Player is colliding with Everything Add condition	Play the sound Bump.wav, vol.: 60, loop: no Shake camera on "1" layer for 1 seconds. Use an amplitude of 2px on X axis and 2px on Y axis, angle rotation amplitude 1 degrees, and zoom amplitude 2 percent. Wait 0.1 seconds between shakes. Keep shaking until stopped: no Make Lives blink for 1.5 seconds Damage Player , removing 1 from its health Change the width of Lives set to $\text{Player.Health} \times 32$ Add action
Dying	
If the player's health reached 0 or below, it's considered dead and will apply the following effects.	
Player is dead Add condition	Play the sound Death.wav, vol.: 50, loop: no Create object DeathParticle at position $\text{Player.X}; \text{Player.Y}$ (layer:) Create object DeathParticle2 at position $\text{Player.X}; \text{Player.Y}$ (layer:) Rotate DeathParticle towards Player.Angle at speed 0 deg/second Delete Player Show GameOver Add action

Now we deal with the asteroids. In this game when we hit a big asteroid it breaks into medium asteroids and medium asteroids will brake into small asteroids. When we hit small asteroids they are destroyed.

Destroying each asteroid size	
For "each instance of each type of asteroid" that collides with a bullet, the following actions will happen.	
Repeat for each instance of Asteroid_Big: Bullet is in collision with Asteroid_Big Add condition	Play the sound Explosion.wav, vol.: 60, loop: no Create object Asteroid_Medium at position $\text{Asteroid_Big.X}; \text{Asteroid_Big.Y}$ (layer:) Create object Asteroid_Medium at position $\text{Asteroid_Big.X}; \text{Asteroid_Big.Y}$ (layer:) Create object DebrisHuge at position $\text{Asteroid_Big.X}; \text{Asteroid_Big.Y}$ (layer:) Create object BulletHit at position $\text{Bullet.PointX}(\text{"BulletHit"}); \text{Bullet.PointY}(\text{"BulletHit"})$ (layer:) Rotate Asteroid_Medium towards $\text{RandomFloatInRange}(0, 360)$ at speed 0 deg/second Apply to Asteroid_Medium a force of angle $\text{Asteroid_Medium.Angle}()$ and length 3 Delete Asteroid_Big Delete Bullet Add action
Repeat for each instance of Asteroid_Medium: Bullet is in collision with Asteroid_Medium Add condition	Play the sound Explosion.wav, vol.: 55, loop: no Create object Asteroid_Small at position $\text{Asteroid_Medium.X}; \text{Asteroid_Medium.Y}$ (layer:) Create object Asteroid_Small at position $\text{Asteroid_Medium.X}; \text{Asteroid_Medium.Y}$ (layer:) Create object DebrisMedium at position $\text{Asteroid_Medium.X}; \text{Asteroid_Medium.Y}$ (layer:) Create object BulletHit at position $\text{Bullet.PointX}(\text{"BulletHit"}); \text{Bullet.PointY}(\text{"BulletHit"})$ (layer:) Rotate Asteroid_Small towards $\text{RandomFloatInRange}(0, 360)$ at speed 0 deg/second Apply to Asteroid_Small a force of angle $\text{Asteroid_Small.Angle}()$ and length 3 Delete Asteroid_Medium Delete Bullet Add action
Repeat for each instance of Asteroid_Small: Bullet is in collision with Asteroid_Small Add condition	Play the sound Explosion.wav, vol.: 50, loop: no Create object DebrisSmall at position $\text{Asteroid_Small.X}; \text{Asteroid_Small.Y}$ (layer:) Create object BulletHit at position $\text{Bullet.PointX}(\text{"BulletHit"}); \text{Bullet.PointY}(\text{"BulletHit"})$ (layer:) Delete Asteroid_Small Delete Bullet Add action

Finally, we do something that is also typical from asteroids games: when something goes out of the screen it appears on the other side (X and Y wrap).

Screenwrap

X Wrap

If something goes outside of +/-840 on the X axis, its position will be switched to be the opposite.
 *It moved to be slightly closer to the center to avoid having it repeat the event by still being out of bounds when it's flipped.

Repeat for each instance of Everything:

Add condition	Add action
If one of these conditions is true: <input type="checkbox"/> The X position of the center of Everything < -840 <input type="checkbox"/> The X position of the center of Everything > 840 Add a sub-condition Add condition	<input type="checkbox"/> Change the X position of Everything : set to (Everything.X)*-0.95 Add action

Y Wrap

If something goes outside of +/-630 on the Y axis, its position will be switched to be the opposite.
 *It moved to be slightly closer to the center to avoid having it repeat the event by still being out of bounds when it's flipped.

Repeat for each instance of Everything:

Add condition	Add action
If one of these conditions is true: <input type="checkbox"/> The Y position of the center of Everything < -630 <input type="checkbox"/> The Y position of the center of Everything > 630 Add a sub-condition Add condition	<input type="checkbox"/> Change the Y position of Everything : set to (Everything.Y)*-0.95 Add action

Here we used the instruction Repeat that allows us to execute several times one instruction while a certain condition is met.

RESOURCES

Challenge 27 (Basic)