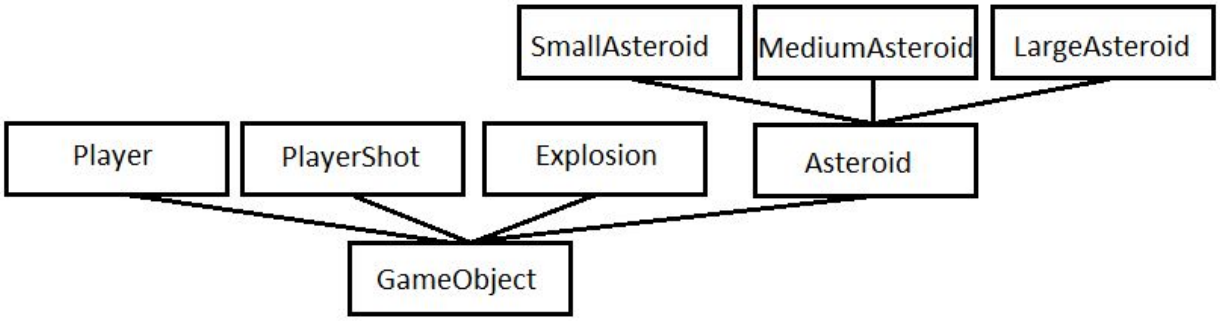
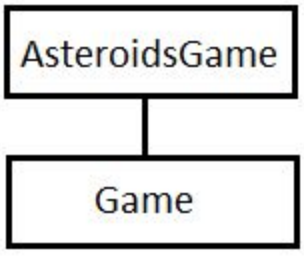
C# Asteroids

In this exercise you will complete a clone of the classic game, Asteroids. This implementation of the game relies heavily of inheritance and polymorphism. To get the game working, you will have to override several functions and provide new implementations.

The diagram shown below outlines the class hierarchy of the objects in the game.



Additionally, the game itself is derived from a class called Game, which implements some of the functionality common to any game. The Game classes primary job is to maintain a list of GameObjects representing the things in the game.



The game is driven by a Timer object on the main form. When the timer "ticks", the Form calls the game's Update() function. The game then loops through its list of objects and calls each object's Update() function. After calling the game's Update() function. The Form triggers a Paint event by calling Invalidate() on its PictureBox control. The paint event handler then

calls the game's Paint() function and passes it the Graphics object to draw on. The game then calls the Draw() function on each of the objects in its list.

Getting started

Download the Asteroids project from blackboard and unzip it.

Here is what you need to do to complete the game:

Implement the NumPlayerShots function

Loop over the list of objects and count up the number of objects which are PlayerShots (use the *is* keyword). Return this number.

Implement the NumAsteroids function

Loop over the list of objects and count up the number of objects which are Asteroids (use the *is* keyword). Return this number.

Draw the remaining lives

In the AsteroidGame class, override the Draw() function. Call the Draw() function in the base class, then call the DrawLives() function which is already implemented in AsteroidGame.

Add a constructor the LargeAsteroid class

Add a constructor that takes two integer parameters, x and y. Have the constructor call the constructor in the base class. The base class constructor takes three parameters x,y and the *image* variable which is part of the class. Pass x, y, and smallAsteroidImg to the base class constructor.

In the body of the constructor…

set Speed to SMALL_ASTEROID_SPEED.

Initialize the hitSound with snd_explosion1 (example below)

Example:

```
hitSound = new SoundPlayer(Properties.Resources.snd_explosion1);
```

Add a constructor the MediumAsteroid class

Add a constructor that takes two integer parameters, x and y. Have the constructor call the constructor in the base class. The base class constructor takes three parameters x,y and the *image* variable which is part of the class. Pass x, y, and mediumAsteroidImg to the base class constructor.

In the body of the constructor…

set Speed to MEDIUM_ASTEROID_SPEED.

Initialize the hitSound with a snd_explosion2

Add a constructor the LargeAsteroid class

Add a constructor that takes two integer parameters, x and y. Have the constructor call the constructor in the base class. The base class constructor takes three parameters x,y and the *image* variable which is part of the class. Pass x, y, and largeAsteroidImg to the base class constructor.

In the body of the constructor…

set Speed to LARGE_ASTEROID_SPEED.

Initialize the hitSound with a snd_explosion3

Override the Hit() function in the SmallAsteroid class

In the SmallAsteroid class, override the Hit() function. In the body, set Destroy to true.

Call Hit() in the base class

Override the Hit() function in the MediumAsteroid class

In the MediumAsteroid class, override the Hit() function.

In the body, set Destroy to true.

Create two new SmallAsteroid objects at the MediumAsteroid's coordinate and add them to the *objects* list.

Call Hit() in the base class

Override the Hit() function in the LargeAsteroid class

In the MediumAsteroid class, override the Hit() function. In the body, set Destroy to true.
Create two new MediumAsteroid objects at the LargeAsteroid's coordinate and add them to the *objects* list.

Call Hit() in the base class

Implement the function to check for player collisions with an asteroid

Implement the CheckForCrash() function in the AsteroidsGame class. This function needs to carry out the following logic:

```
For each object:
    If the object is an asteroid:
        If the distance to the player is less than the Asteroid's radius + 16:
            Call the DestroyPlayer() function
```

Break out of the loop

Note: The GameObject class has a Distance() function you can use to determine how far the player is from an Asteroid.

Implement the Update function in the PlayerShot class

The player's shots aren't supposed to live forever. The PlayerShot class has a tick counter in it that keeps track of how long it has been alive. We need to add new functionality while keeping the old. This means overriding Update(), adding new code, and then calling the base class version.

In the PlayerShot class, override the Update() function.
Call Update() in the base class (this makes the shot move).
Increment *ticks.* If *ticks* is greater than MAX_TICKS, set Destroy to true.

Implement the function to check for shot hits

This function loops over the list of objects in the game. If an object is a PlayerShot, the function compares its position to every other object. If that other object happens to be an Asteroid, check the distance to the shot. If the distance is less than the Asteroid's radius, the Asteroids Hit() function is called, and the Destroy property on the shot is set to *true*.

Make the Explosion draw properly

Override the Draw function in the Explosion object. In the new function, call the *sheet* object's *Draw()* function. This delegates drawing to the sprite sheet rather than the base class.

Stop the Explosion from looping forever

When the explosion is done playing it the frames in its sprite sheet, we want destroy it (because it's done). Here's how to do that:

Override the Update() function in the Explosion object.
In it, check the *Done* property on the *sheet* object. If it is *true,* set Destroy to *true.* The Game will now destroy the explosion object for you.

You should have a playable Asteroids game. Enjoy your hard work.