

Creature Kingdom

In this assignment you are to create a creature that moves around a kingdom (Canvas) based on the Creature class provided in the attached project. Your creature is to be implemented as a class that inherits from the Creature class defined in Creature.cs.

Your creature's class name is to be your last name, your first name, and the word "Creature". It is to be pascal-cased. Example: SmithMollyCreature

Any assets (images) used by your creature are to be placed in a folder using your last name and first name (pascal-cased). This folder will be placed in a folder named 'assets' in the same directory as the executable. The bitmap loader that is part of the Creature class can be used to load the image using the relative file path provided from the assets folder.

This means when you load the assets you need to include the name of the folder as part of the file path. Example:

- folder named: SmithMolly
- asset relative file path: SmithMolly\right.png

NOTE: remember to use @ in front of strings that you want interpreted literally. This is necessary if the string contains a \

Example: `String imagePath = @"SmithMolly\left.png";`

Creature Class Documentation

If you need additional information refer to Creature.cs

Constructor

```
public Creature(Canvas kingdom, Dispatcher dispatcher, Int32
waitTime = 100)
```

When your creature is created it will be supplied with a Canvas and a Dispatcher and optionally a waitTime. If a waitTime is not provided it will default to 100 millisecs. The waitTime is the time to wait in the threaded update loop.

Class member variables that can be accessed in the subclass:

```
protected Canvas kingdom;
protected Dispatcher dispatcher;
protected double x, y;
```

Properties

```
public Int32 WaitTime
```

- can be set or get.

```
public Boolean Paused
```

- can be set or get.
- when Paused is true the update loop should not update the creature.
- when Paused is false the update loop should update the creature.

Creature Kingdom

```
public double X
```

- can be set or get.
- the x (left) position of the creature.

```
public double Y
```

- can be set or get.
- the y (top) position of the creature.

Methods

```
public virtual void Place(double x = 100.0, double y = 200.0)
```

- calling Place on the creature is to cause it to appear and begin animating.
- the x and y indicate where the creature is to be placed on the canvas and defaults to (100,200) if not supplied.
- you will need to write your own Place method to override this one provided in the Creature class.

```
protected virtual BitmapImage LoadBitmap(String  
assetsRelativePath, double decodeWidth)
```

- can be used to load a bitmap image from a file placed in the assets directory in the application's directory.
- assetsRelativePath is the path of the image file relative to the assets folder.
- as per note above, you are to create a directory based on your name in the assets directory to hold your assets.
- for transparency to work the image needs to be a png.
- decodeWidth specifies how wide the image should be decoded as. The height will be set to keep the image from being distorted.

```
public virtual void Shutdown()
```

- this is called when the creature needs to go away. It is the place that any threads used by the creature are to abort.
- you need to write your own Shutdown method to override this one provided in the Creature class.

Additional Information

Your creature class must use a thread to update the position of the creature. The WaitTime property is to be used in the update loop in the thread to sleep between updates.

The kingdom (Canvas) is resized when the window size is changed. Your creature is to respond to the size of the kingdom as it animates. The current/rendered bounds of the kingdom must be used in the boundary checking. Remember that the Width property on a Canvas will not work because it is set to NaN when the Canvas is set to stretch to fill the container that holds it.

The name of the zip file that you upload to Blackboard for this assignment is to be your last name and first name camel cased. Example: SmithMolly.zip

Creature Kingdom

The following is a description of the contents of the zip file you are to submit:

- the folder based on your name containing your media (image) assets.
- the class file for your creature.
- a zip file of your entire project.

So, when I download and unzip your submission I should see:

- folder based on your name containing your media assets that I can drop into the assets folder in the app's directory.
- a .cs file that contains your creature class file that I can add to the project with everyone else's class files.
- a zip file of your entire project.

If I unzip your project I should be able to load it, run it and see the creature doing its thing.