GDD 1200 Programming Assignment 5
Mining Teddies

ASSIGNMENT DESCRIPTION

In this assignment, you'll be placing land mines to blow up teddy bears. Teddy bears will spawn at random times between 1 and 3 seconds, at random locations with random velocities. When a teddy bear collides with a land mine, the land mine blows up and both the land mine and the teddy bear disappear.

STARTING THE ASSIGNMENT

To start your work, someone in your guild should download the ProgrammingAssignment5Materials.zip file from the Programming Assignment 5 content area on Blackboard and extract the contents somewhere. The zip file contains a dll, documentation, and sprites for the assignment.

Next, create a MonoGame Windows Project (or MonoGame Mac Application (MonoMac) project) called ProgrammingAssignment5 in your shared Dropbox folder. Copy the dll from the zip file into the same folder as your Game1.cs file. Add the dll as a reference for the project and add a using statement for the `TeddyMineExplosion` namespace.

You also need to use the Pipeline tool to build content for all the sprites I provided to you in the zip file and add that content to the project.

REQUIREMENTS

Your program must do the following:

- Let the player place a mine by left clicking in the game window. When they do, add the new mine to a list of mines in the game. The player should be able to place as many mines as they want
- Spawn a new teddy bear at a random time (1 to 3 seconds) after the previous spawn. Add the newly spawned teddy bear to a list of teddy bears in the game
- When an active teddy bear collides with an active land mine, add a new explosion (located at the center of the mine) to a list of explosions in the game. Note that explosions start playing their animation as soon as they're created. Make the teddy bear and the land mine inactive
- On each update remove all the inactive teddy bears, mines, and explosions from their respective lists

Your solution to this problem must:

- Meet the problem specification (e.g., do what it’s supposed to)
- Comply with the GDD Coding Standards

HELPFUL HINTS

Write 5 or fewer lines of code, save, compile, test, repeat! Don't try to write a whole block of code at once – implement one small chunk at a time.

Add a `using TeddyMineExplosion;` statement at the top of the `Game1` class to get access to the classes in the dll.

Add land mine placement functionality first, then random teddy bear spawning, then collision detection and resolution, then all the processing to remove inactive entities from their lists.

Make sure you update all the teddy bears and explosions in their respective lists.

Make sure you draw all the teddy bears, mines, and explosions in their respective lists.

MY IMPLEMENTATION STEPS

I implemented the required functionality in the order shown below. You do NOT have to implement your solution in this order, and I encourage you to try the assignment on your own first, but some of you might find a little extra guidance helpful as you work on your solution. Note that all the code I added was in the Game1 class; I didn't make any changes to any of the other files I gave you.

1. Set the resolution and made the mouse visible in the Game1 constructor
2. Declared fields for the mine sprite and the list of mines in the Game1 class. Loaded the mine sprite in the Game1 LoadContent method.
3. Declared fields to support left click processing in the Game1 class. Added code to the Game1 Update method to add a mine to the list of mines when a left click is finished
4. Added code to the Game1 Draw method to tell each of the mines in the list of mines to draw itself. At this point, you should be able to place mines by left clicking in the game window
5. Declared fields in the Game1 class to support spawning teddy bears at random times. These fields include the fields for spawn timer support, a field to let us generate random numbers, the teddy bear sprite, and the list of teddy bears
6. Added code to the Game1 LoadContent method to load the teddy bear sprite and set the first spawn delay between 1 and 3 seconds
7. Added code to the Game1 Update method to update the spawn timer and (when the timer is finished) reset the spawn timer, set a new random spawn delay between 1 and 3 seconds, spawn a new teddy bear (with a random velocity as described above) and add it to the list of teddy bears

8. Added code to the Game1 Draw method to tell each of the teddy bears in the list of teddy bears to draw itself. At this point, you should be able to see teddy bears spawning randomly at 1 to 3 second intervals (though the teddy bears don't move)
9. Added code to the Game1 Update method to tell each of the teddy bears in the list of teddy bears to update itself. At this point, the spawned teddy bears should move after they're spawned
10. Declared fields in the Game1 class for the explosion sprite and the list of explosions
11. Added code to the Game1 LoadContent method to load the explosion sprite
12. Added code to the Game1 Update method to detect teddy bear collisions with mines and, if a collision is detected, make the teddy bear inactive, make the mine inactive, and add a new explosion to the list of explosions. At this point, when a teddy bear collides with a mine, both the teddy bear and the mine should disappear
13. Added code to the Game1 Draw method to tell each of the explosions in the list of explosions to draw itself.At this point, on a collision between a teddy bear and a mine you should see the first frame of the explosion animation
14. Added code to the Game1 Update method to tell each of the explosions in the list of explosions to update itself. At this point, you should see the complete explosion animation when a teddy bear collides with a mine
15. Added code to the Game1 Update method to remove inactive teddy bears, mines, and explosions from their respective lists

TURNING IN YOUR ASSIGNMENT

This programming assignment is worth 3% of your overall course grade. **Only provide one submission per guild**.

You're required to turn in ALL of the following by the beginning of the scheduled class time on the due date:

Electronic Copy

1. Zip up your entire assignment folder into a file named <yourguildname>.zip. Log into Blackboard and submit the file into the appropriate assignment.

IMPORTANT NOTE: If your zip file doesn’t contain all the required files or is zipped up using WinZip, 7Zip, or any other program different from the default Windows compression utility, you’ll receive an **AUTOMATIC 0** on this assignment. Since this assignment is worth a good chunk of your overall course grade, I strongly suggest you

use the Windows compression utility and check your zip file to make sure it’s complete before submitting it.

LATE TURN-INS

- Turn-ins are due at the beginning of the scheduled class time on the specified due date
- No late turn-ins will be accepted