

GDD 1200 Programming Assignment 4 Ted the Collector

ASSIGNMENT DESCRIPTION

Your job for this assignment is to develop a MonoGame game that meets the requirements listed below.

STARTING THE ASSIGNMENT

To start your work, someone in your guild should download the ProgrammingAssignment4Materials.zip file from the Programming Assignment 4 content area on Blackboard and extract the contents somewhere. The zip file contains code files and sprites for the assignment.

Next, create a MonoGame Windows Project (or MonoGame Mac Application (MonoMac) project) called ProgrammingAssignment4 in your shared Dropbox folder. DON'T call the project something else, it needs to be called ProgrammingAssignment4 for the next steps to work properly. Copy all the code from the Windows or Mac code folder into the appropriate place (you'll have to confirm replacing the template Game1.cs file the IDE generated for you) and add all the files you just copied to the project (except Game1.cs, which is already in the project). These are all the classes you need as well as stubs for all the methods you need. I even threw in some extra code snippets where I thought they might be helpful.

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.

The Game1.cs, TeddyBear.cs, and Pickup.cs files from the zip file includes comments in the code indicating where you should add your own code or modify the provided code; all those comments start with the word STUDENTS. Changing code that I already provided to you, except where indicated, is both unnecessary and unwise.

REQUIREMENTS

Your program must do the following:

- Start with a TeddyBear object, centered in the window, not moving
- On every right mouse click, add a Pickup object where the mouse was clicked
- When the player left clicks the TeddyBear object, the teddy starts collecting the pickups in the order in which they were placed
- The TeddyBear collects a Pickup by colliding with it, but this only works for the Pickup the TeddyBear has currently "targeted for collection"
- Once the last Pickup object has been collected, the TeddyBear object stops moving

- If the player adds more Pickups while the TeddyBear is moving, the TeddyBear picks them up as well
- If the player adds more Pickups while the TeddyBear is stopped, the player has to left click on the TeddyBear again to start it collecting again

Your solution to this problem must:

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

You are NOT allowed to use any constructs we haven't learned in class yet (we just started on loops, so I've provided the required loops for you). Using those would certainly be easier (if you already know them) but you won't learn what I want you to learn from this assignment.

HELPFUL HINTS

When you spawn a new Pickup you should add it to the end of the list of pickups.

When the TeddyBear collects a Pickup, you should remove the Pickup from the list of pickups and either start moving the TeddyBear toward the next Pickup or stop the TeddyBear.

If you use the intuitive “move the teddy bear draw rectangle every update using velocity and elapsed game time” approach in your solution, you'll sometimes end up with accumulated rounding error that makes the teddy bear miss the pickup it's trying to pick up. Instead, you should update the location field in the TeddyBear class using velocity and elapsed game time, then center the drawRectangle field on the location X and Y values. Although your draw rectangle could end up off by a single pixel in both X and Y on a particular update, it will never be worse than that because the location field doesn't suffer from the accumulated rounding (because X and Y in the location field are floats).

Write a few lines of code, save, compile, test, repeat! Don't try to write massive chunks of code at once – implement one small chunk at a time.

IMPORTANT NOTE

The code I provided to you doesn't compile; getting the code to compile by getting the current mouse state and storing it in a `mouse` variable in the Game1 Update method should probably be your first step!

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 <yourguildnumber>.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