TeenCoder™: Game Programming
Second Edition Errata Sheet

Updated May 9th, 2011

This document lists the known typographical or other corrections to the *TeenCoder™: Game Programming* Second Edition course.

- In early printings, on page 128 in the **fireBeeShot**() discussion, the following paragraph may incorrectly reference the "beeShots" list instead of the "bees" list:

  "If you have not exceeded the maximum allowed shots, generate a random number between 0 and the total number of bees in the **`beeShots`** linked list. Use **`beeShots`**.**Count** as the ending number in order to ensure you a random number starting from 0 and going up to the last valid sprite index in the list! "

  You should use the total number of bees in the **bees** linked list, and use **bees**.**Count** as the ending number. The activity solution provided in the Solution Guide already contains the correct code.

- In early printings, on pages 145 and 146 in the Continuous and Short Animation discussion, we neglected to discuss the **Sprite**.**ContinuousAnimation** flag, which defaults to **true** in the **Sprite** class:

```
// public flag indicating whether or not animation frames should be continuously looped
public bool ContinuousAnimation = true;
```

  In cases where you want to perform an animation "short", such as when you are exploding the smoke sprayer in the Swarm game, make sure you set your sprite's **ContinousAnimation** flag = **false** when creating the sprite, otherwise the animation short will begin animating immediately. For instance, in the Swarm's **initializeSmokeSprayer**() method after creating the **smokeSprayer** sprite and setting the texture, add the following line:

```
// we will only animate this texture when it gets hit
smokeSprayer.ContinuousAnimation = false;
```

- In early printings, in Chapter 9 Lesson 2 when discussing the loading of sound effects from resources into a **SoundEffects** object, examples were given as follows:

```
mySoundEffect = Content.Load<SoundEffect>("laser");
```

Note that if you put your resource objects into folders as suggested in the text (e.g. “Audio”), then you will need to include that folder name in your Load() method call in order to find the resource. See for example the Swarm activity solution:

```
stingerSound = Content.Load<SoundEffect>("Audio\\stingerShot");
```

- In early printings, on page 235 the **DoSimpleAI**() method instructions ask you to check “**`Starship2`**`.velocity.`**`Y`**” in the third paragraph. You should actually be checking “**`StarShip2.GetVelocity().Y`**” as shown in the Activity Solution.