Exercises

1. The AcademyPopcorn class contains an IndestructibleBlock class. Use it to create side and ceiling walls to the game. You can ONLY edit the AcademyPopcornMain.cs file.
2. The Engine class has a hardcoded sleep time (search for "System.Threading.Sleep(500)". Make the sleep time a field in the Engine and implement a constructor, which takes it as an additional parameter.
3. Search for a "TODO" in the Engine class, regarding the AddRacket method. Solve the problem mentioned there. There should always be only one Racket. Note: comment in TODO not completely correct
4. Inherit the Engine class. Create a method ShootPlayerRacket. Leave it empty for now.

Exercises (2)

5. Implement a TrailObject class. It should inherit the GameObject class and should have a constructor which takes an additional "lifetime" integer. The TrailObject should disappear after a "lifetime" amount of turns. You must NOT edit any existing .cs file. Then test the TrailObject by adding an instance of it in the engine through the AcademyPopcornMain.cs file.
6. Implement a MeteoriteBall. It should inherit the Ball class and should leave a trail of TrailObject objects. Each trail objects should last for 3 "turns". Other than that, the Meteorite ball should behave the same way as the normal ball. You must NOT edit any existing .cs file.
7. Test the MeteoriteBall by replacing the normal ball in the AcademyPopcornMain.cs file.

Exercises (3)

8. **Implement an UnstoppableBall and an UnpassableBlock. The UnstopableBall only bounces off UnpassableBlocks and will destroy any other block it passes through. The UnpassableBlock should be indestructible.**
   Hint: Take a look at the RespondToCollision method, the GetCollisionGroupString method and the CollisionData class.
9. **Test the UnpassableBlock and the UnstoppableBall by adding them to the engine in AcademyPopcornMain.cs file**
10. **Implement an ExplodingBlock. It should destroy all blocks around it when it is destroyed. You must NOT edit any existing .cs file.**
    Hint: what does an explosion "produce"?

Exercises (4)

11. Implement a Gift class. It should be a moving object, which always falls down. The gift shouldn't collide with any ball, but should collide (and be destroyed) with the racket. You must NOT edit any existing .cs file.
12. Implement a GiftBlock class. It should be a block, which "drops" a Gift object when it is destroyed. You must NOT edit any existing .cs file. Test the Gift and GiftBlock classes by adding them through the AcademyPopcornMain.cs file.

Exercises (5)

13. **Implement a shoot ability for the player racket. The ability should only be activated when a Gift object falls on the racket. The shot objects should be a new class (e.g. Bullet) and should destroy normal Block objects (and be destroyed on collision with any block).**
    Use the engine and ShootPlayerRacket method you implemented in task 4, but don't add items in any of the engine lists through the ShootPlayerRacket method.
    Also don't edit the Racket.cs file.
    Hint: you should have a ShootingRacket class and override its ProduceObjects method.

Exercises (6)

14. * Bonus task (optional): Download JustBelot game source from https://github.com/NikolayIT/JustBelot (code commits are made often so be sure to always work on the latest game source). Write your own C# library called `JustBelot.AI.YourBotName` and write a class in it that implements `JustBelot.Common.IPlayer` interface. Implement your own AI belot player that will fight with other AI players. The winner will be awarded. Please send your players to academy@telerik.com and add them in the homework archive when you upload it. You are allowed to work in teams.
    This task is not obligatory. Discussions: here.