CS 175 | Scripting Languages
Programming Assignment 6

This assignment is focusing on the collision manager. It will be a very simple collision manager that we will expand on in future assignments. The estimated time for completion is about six hours.

The program given to you is based on the same small game that you made in assignment 4. You will be given all the needed assets and engine classes (.as files) except of one that you will have to implement.

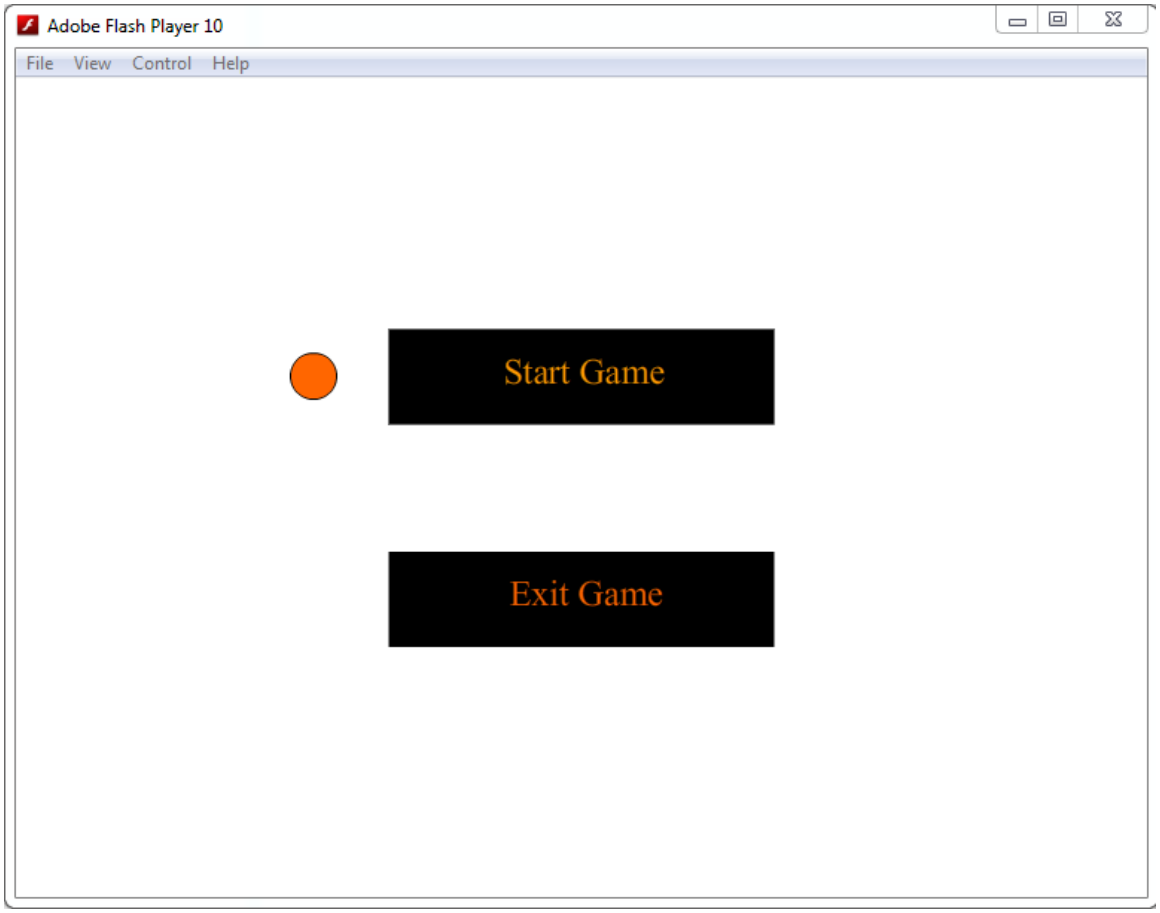
- **CollisionManager.as**

In addition, you will have to add a new game play level.

Details

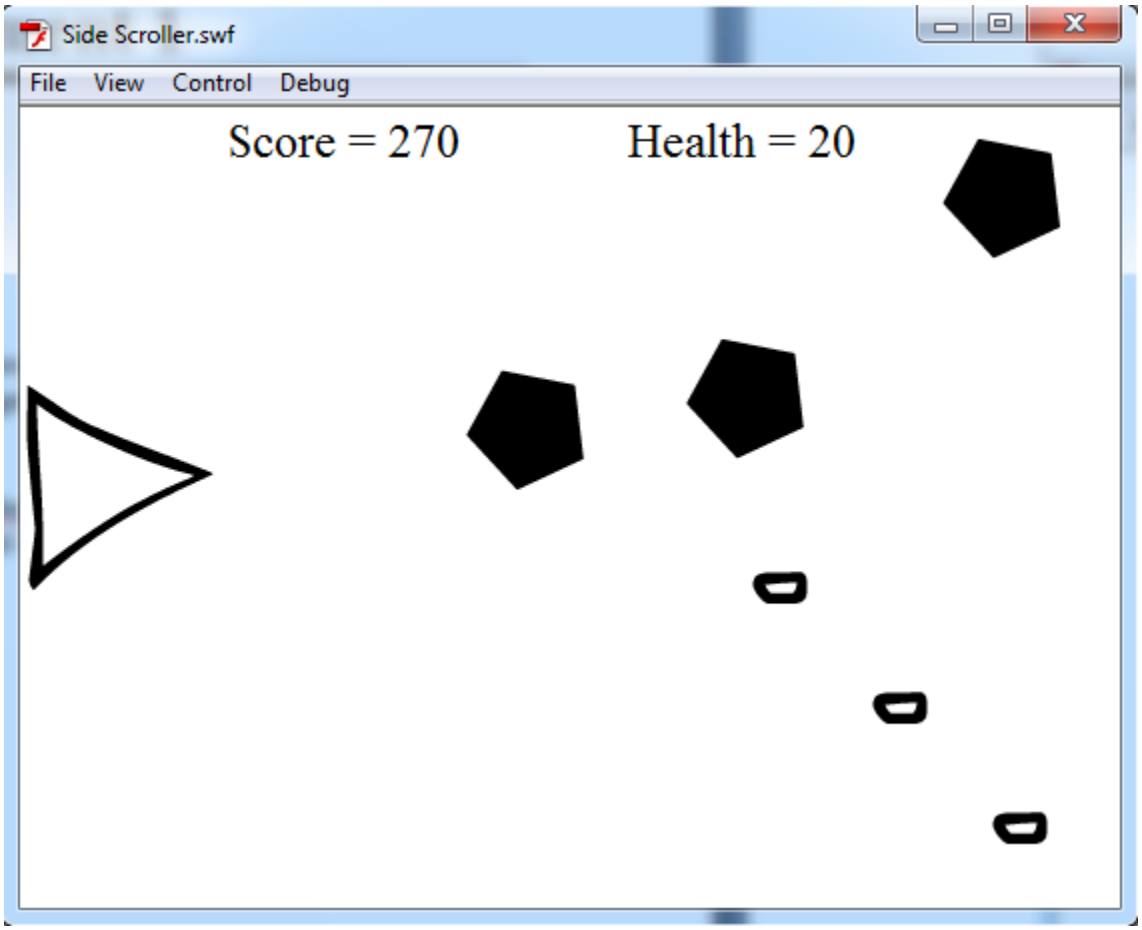
Game Logic Details:

Main Menu



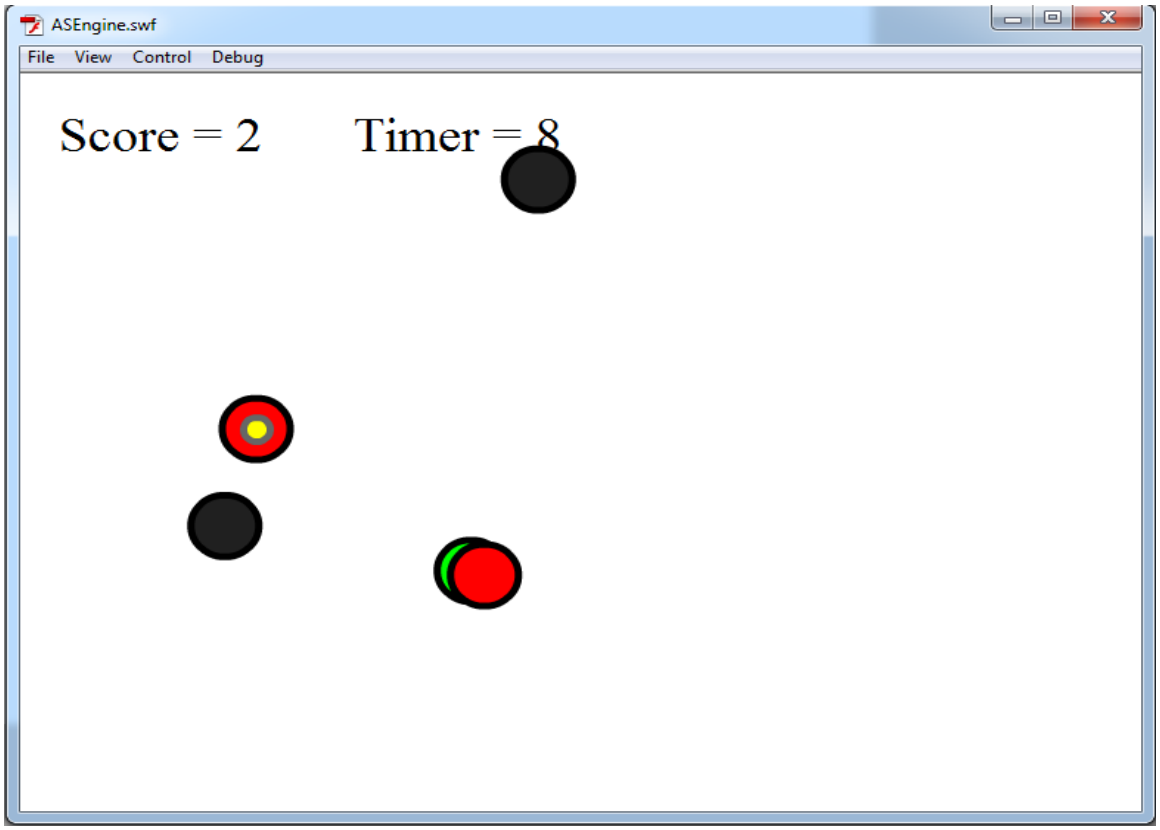
- ***In the MainMenu, two MovieClips have to be shown (StartGame and ExitGame)***
- ***The player uses the "UP" and "DOWN" keys in order to change the selection between the two***
- ***Pressing the "Space" bar will switch us to a different level (of course depending on the user's selection)***

Level 1



- ***Level 1 is a simple shooter***
  - ***The level starts with a simple countdown (no enemies are generated until the countdown is over).***
  - ***The ship goes up and down when user presses the "UP" or "DOWN" keys.***
  - ***The ship shoots bullets when the user presses the "SPACE" key.***
  - ***Enemies (black stars) get generated on the right and move towards the ship***
  - ***Enemies and Bullets are destroyed if***
    - ***they collide, which increases the score***
    - ***they are outside the screen (since we don't have use for them anymore)***
  - ***The ship's health is decreased if hit by an enemy.***
  - ***The game ends if the ship's health reaches 0 by going back to the main menu***
- ***Pressing the "M" key anytime during the game will take us back to the main menu***
- ***Pressing the "R" key anytime during the game will restart the level***
- ***Pressing the "N" key anytime during the level will take you to "Level 2"***

Level 2



- ***In level 2:***
  - ***The player is represented by a circle with a yellow dot in the middle.***
    - ***Moves Up, Down, Left and Right (arrow keys)***
    - ***Changes colors (Red, Black or Green)***
    - ***Collects circles with the same color. Score incremented by one when collecting a circle.***
    - ***Dies if collides with any circle with different color. Plays the death animation before being removed.***
    - ***Doesn't collide with circles when playing the death animation.***
  - ***Every 10 seconds the player randomly changes its color***
  - ***Collectable circles:***
    - ***generated every 25 frames***
    - ***have a random speed between 5 and 10***
    - ***Are generated on the perimeter of a circle positioned in the center of the viewport and with radius 400. Its initial direction is always going towards the center.***
    - ***Is destroyed when outside the viewport***
  - ***If player dies everything is removed from the stage except the score.***
- ***Pressing the "R" key anytime during the game will restart the level***
- ***Pressing the "M" key anytime during the game will take us back to the main menu***

PS: ***Just in case I missed to list things, check the given swf file for the full details or to play the game***

Code Details:

In assignment 4, collision was handled in one or two functions inside the object manager. Here, we will create a simple collision manager that will be used in the object manager.

Even though we will be going over all the classes during the lecture, it will be up to you to revisit them when doing the assignment in order to implement the required functions found in the CollisionManager.as :

- Initialize
- CheckDynamicCollision
- CollidedObjectsReaction
- Destroy

Check how every function is called in order to know what parameters each function needs.

Note:

- **In the Engine classes, you are only allowed to add code inside "CollisionManager.as" (basically implementing the 4 functions)**
- **As far as Level2's gameplay, the only restriction is that all the files have to be inside a "Level2" folder placed in the "GamePlay" folder.**

Comments

In this and future assignments, you are required to include:

- **A file header comment in every piece of source file. The format is shown in the "Comments.as" file given to you in the beginning of the semester and should be present at the very top of all your code.**

- **Function header for each function you create. The format is shown in the "Comments.as" file given to you in the beginning of the semester and should be present at the top of every function.**

- **Inline commenting for your code.**

Note:

- **Only comment the files you alter or create.**

What to submit

You must submit the "ASEngine v3.0" folder in a single .zip file named correctly (go to the class page on moodle and you will find the assignment submit link). **Do not change the hierarchy of the files inside it. Do not submit any other files than the ones listed.**

If you've forgotten how to submit files, the details about how to submit are posted in the syllabus. Failure to follow the instructions will result in a poor score on the assignment (and possibly a zero).

Special note:

The due date/time posted is the positively latest you are allowed to submit your code. Since the assignments can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. Moral: **Don't wait until the last day to do your homework.**