

Code Quality Reviews:

Sprint 2:

Author: Carson Lewis

Date: September 21

Name of File: idleMegaman.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 10

Comments: Quality is pretty good, but the different drawing sprites are hardcoded using an interval variable.

Change: I would utilize the update method to change the sprite so that he is animated instead of doing it all in the draw method with the interval variable.

Sprint 2:

Author: Matthew Weikel

Date: September 24

Name Of File: EnemySpriteFactory

Author of File: Tyler Burkett

Number of Minutes to complete review: 7

Comments: The code is easy to read and follows the format that was provided in class. The method names are descriptive and easy to understand.

Change: I would just recommend to add in more enemies and remove the boss so that the boss has its own sprite factory.

Sprint 3:

Author: Carson Lewis

Date: October 21

Name of File: KeyboardController.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 15

Comments: Quality is Decent. Everything seems to work but there is a lot of extra if statements that could be better placed and condensed.

Change: Make everything more organized as it is all over the place right now. Also make the if statements easier to read.

Sprint 3:

Author: Matthew Weikel

Date: October 21

Name of File: Pellet.cs

Author of File: Carson Lewis

Number of Minutes to Complete Review: 10

Comments: The code is readable, and most are short and have one purpose, just some names are misleading like isfacing left

Change: There will most likely need to be a variable holding the damage that a pellet inflicts for collisions. The variable name for isfacingleft should be ismovingleft. Double check if Megaman is needed for the init function.

Sprint 4:

Author: Carson Lewis

Date: November 11

Name of File: KeyboardController.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 10

Comments: It is looking better but the update method is way too long and messy.

Change: Extract each megaman action to a different method. For example, have a “TryJump” method and a “TryShoot” method to make it better.

Sprint 4:

Author: Matthew Weikel

Date: November 5th

Name of File: jumpingFlea.cs

Number of Minutes to Complete Review: 10

Comments: There seems to be a lot of unused global variables and magic numbers for frame values to animate the sprites. Other than that the code is easy to read and the variable names and method names are specific.

Change: The flea does not have any logic to track Megaman and only jumps one direction, so I would add more logic to test if megaman is behind or in front of the flea so taht it is able to follow him, also add logic for flipping directions.

Sprint 4:

Author Matthew Weikel

Date: November 11th

Name of File: healthBar.cs

Author of File: Carson Lewis

Number of Minutes to Complete Review: 8

Comments: The code is easy to read and has minimal variables, there is only the required methods. The only thing I would say is to make the variables private, as well as the getmultiplier logic has to many else if statements.

Change: The biggest change I would recommend is to edit the logic for the getmultiplier so that it does not have as many else if statements you could try and find a way to do it with math.

Sprint 5:

Author: Carson Lewis

Date: November 11

Name of File: Camera.cs

Author of File: Tyler Burkett

Number of Minutes to Complete Review: 10

Comments: This class itself is good, but it does not have the ability to pan across the screen, it just follows megaman and a lot of things are hardcoded, also not sure if zoom even works.

Change: I would add a separate camera controller class that can handle things like panning and making it so it has more power and more features than just following megaman directly.

Code Readability Reviews:

Sprint 5:

Author: Matthew Weikel

Date November 26th

Number of Minutes to Complete Review: 15

Name of File: SniperJoeProjectile.cs

Author of File: Shiva Nallapati

Comments: The code is readable and straightforward, some of the methods are unused. A lot of the public variables are not used at all and never set. This seems to be because the class is implementing `IenemySprite`, but it adds a new method for `isoffscreen`.

Changes: I would recommend creating a new interface for them so that you can remove a majority of the un needed variables and methods and so that the code will be cleaner and easier to read and set the projectiles to be more seperated from the enemies.

Sprint 5:

Author: Matthew Weikel

Date December 3d

Number of Minutes to Complete Review: 15

Author of File: Carson Lewis

Name of file: KeyboardMethods.cs

Comments: This file is much more clear than the keyboard controller. It is better after being separated into different methods for each case. But it still has the megaman flags inside of the code which is not good practice.

Changes: I would change it so that it does not use the megaman flags inside of the keyboard controller. I would have it call a method inside of the megaman class that uses the flags to execute the megaman controls and change the megaman values.

Sprint 2:

Author: Carson Lewis

Date: October 21

Name of File: GenericFlea.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 15

Comments: Looks pretty good and organized well. The only thing I would maybe change is the changeSprite method. I have no idea what the Boolean forward means or does, so I would change this variable name to a better one that describes the use better.

Sprint 3:

Author: Carson Lewis

Date: October 21

Name of File: CollisionDetector.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 15

Comments: Looks very good and organized. I like the use of enums. The logic also looks good and when I tested the megaman collision with blocks it seemed to work fine.

Sprint 3:

Author: Matthew Weikel

Date: October 21

Name of File: Floor.cs

Author of File: Tyler Burkett

Number of Minutes to Complete Review: 8

Comments: The code is readable, and most are short, but block width and height should be settable variables in case we need to have variable length blocks.

Sprint 4:

Author: Carson Lewis

Date: October 21

Name of File: CollisionHandler.cs

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 10

Comments: This class is very good. I think the readability of all the collision classes are perfect and do not need to be changed.

Sprint 5:

Author: Carson Lewis

Date: October 21

Name of File: SniperJoe.cs and SniperJoeProjectile.cs

Author of File: Shiva

Number of Minutes to Complete Review: 15

Comments: These classes are okay. I think they could be condensed and easier to follow. I also think that SniperJoeProjectile should not implement the IEnemySprite interface. It should implement the IPelletSprite interface. This way would make a lot more sense and I think it would help with issues down the road.

Author: Shiva Nallapati

Date: September 30th

Sprint 2

Name of file: Jumping flea

Author of File: Tyler Burkett

Number of Minutes to Complete Review: 15

Comments: In terms of quality, it is there. I would say that it should align more with the other enemies that were made in terms of actual code to ensure that all the enemies work in a similar pattern.

Change: I would work on aligning the constructor and general movement to be more aligned with the other enemies. Of course, the jumping flea has its own capabilities, but I would focus on changing it to be organized like the other enemies for easier readability. Also, should create a enemysprite factory for this

Author: Shiva Nallapati

Date: October 21

Sprint 3

Name of file: Megaman

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 20

Comments: This has the quality that we are looking for within Megaman and it's looking good so far. We just need to be able to add the other implementations in terms of easier jumping, shooting, climbing, and etc to be easier mechanics.

Change: For this class, it was organized well. The only real thing that could be improved is the formatting of the code itself and adding comments to make it easier to understand.

Author: Shiva Nallapati

Date: November 12

Sprint 4

Name of file: Megaman collisions

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 15

Comments: The quality within this file is amazing and it works perfectly with all of the other parts in terms of collisions.

Change: The only thing I would change is the future implementation of collisions for enemies as well, it seems to just be smaller classes but that's about it..

Author: Shiva Nallapati

Date: December 4th

Sprint 5

Name of file: Bombman

Author of File: Matthew Weikel

Number of Minutes to Complete Review: 10

Comments: The quality of this file is nice, though small as a class. This is a necessary file to help Bombman get animated and for that part of this to work.

Change: The animation likely needs to be better edited to where the bombman class actually gets the right values from the state machine that was created. This would allow the sprite to be completely animated.

Date: December 4th

Sprint 5

Name of file: SoundController

Author of File: Carson Lewis

Number of Minutes to Complete Review: 8

Comments: This file is necessary for the actual sound within the game and it was written well. It's a small file and it seems to be working all cases so no real improvements.

Change: In terms of change, I would maybe change the class name for PascalCase to sort of work with most of the other classes. Maybe renaming _content to contentManager within this file just as something to understand easier. But the changes I would make are just naming conventions for better code quality.