Sprint2_CodeReview

Yi Chen

Date: 3/2/2021

CollisionDetector

Author: JiashuZhang

Comments: The main part of this class is the class for initail and a class for DetectCollison.

In this class, I check the variable first, I find that the Game1 game just use in the initial part, so in my opnion this value "Game" can be deleted. The other part of intial is easy to read.

In the DetectCollision, it create a rectangle of the mario, and check all the sprite for position checking. And it have three commands dealing with enemy, item and block. The length of this class is not too long and also I think the main part of the code is just three foreach part.

Change: In the later I think we need have more sepcifi reaction of differnet kind of collide. I think we can have a big collisionDetect class, and in that class we can call many small method for doing different kind of job. And also we may need more detail checking for different object. For example, different item may can have different function. And for these functio I believe it can be solved in subclasss(MarioEnemyCollide, MarioItemCollide, etc)

Mintues: 25mins

LeveLoader and SpriteManager

Author: Yi Chen

Comments: The work for these two class is two read the information in a XML and record information to SpriteList. The informatiion in these two class is by a Enum. And SpriteManager get the information and direct have update and draw in SpriteManger.

Change: In the later, we may need to add Mario into, and this may cause trouble. And how can we control all the enemy in the spriteList and delete them as they die, this is problem still need to fix.

mintues: 10mins

Sprint1_CodeReview

Author: Yi Chen

Date: 2/16/2021

Game1.cs

Author: JiashuZhang, YiChen

Quality:On the whole, I think my code is not bad. My main concern now is that when our code becomes more complicated, code will become very chaotic in the main class.

Way to solve: I don't know much about Sprite factory since this part job I do not focus on this time. But I think we get a better way through group discussion in the future. Such as command design pattern we learn from class and use at Controller this time.

Minutes: Ten

SpriteFactory.cs

Author: JiashuZhang

Quality: Overall, I feel that this file is written very well. There is the core step of creating sprite, "creatSprite". It uploads all the Content and the corresponding paths of multiple pictures. In my opinion, this file seems to be relatively concise. For example, in the public enum sprites, Jiashu has made more detailed plans including Mario, Item,etc in different forms.

Improve: For this file, I think it is very basic but important. The first I know may can be better is that this file is directly connect to the main class(Game1.cs), so in this way we need to write game1.cs, we need several update, initial new variable, etc. I think we may can find a way to solve this problem in later project. Maybe we can build up a class, it have record the item we need to update, and in the main we can just have one update them we will update all the sprites. And I think as we going future, we may learn some new pattern which help us to we do it better.

Minutes: twenty.

fireMarioRun.cs

Author: JiashuZhang

Quality: In this review, I hope to focus on Sprite, because this time I mainly focus on Controller, Command and the corresponding reciever, while I just watched another large sprite. After reading the SpriteFactory and its running on main, I decided to start observing some more detailed code. First of all, this file is very similar to the animated one I wrote last time, but it has made a great improvement. They used gameTime on update to control the update speed of sprite, so that sprite would not be blurred because of too fast refresh.

Improve: I have watch several other sprite class after this, I find that every sprite need to animated have all the similar content, so I think we may can create a animated class to help them. (I am not sure at this method is better or worse.)

Minutes: ten