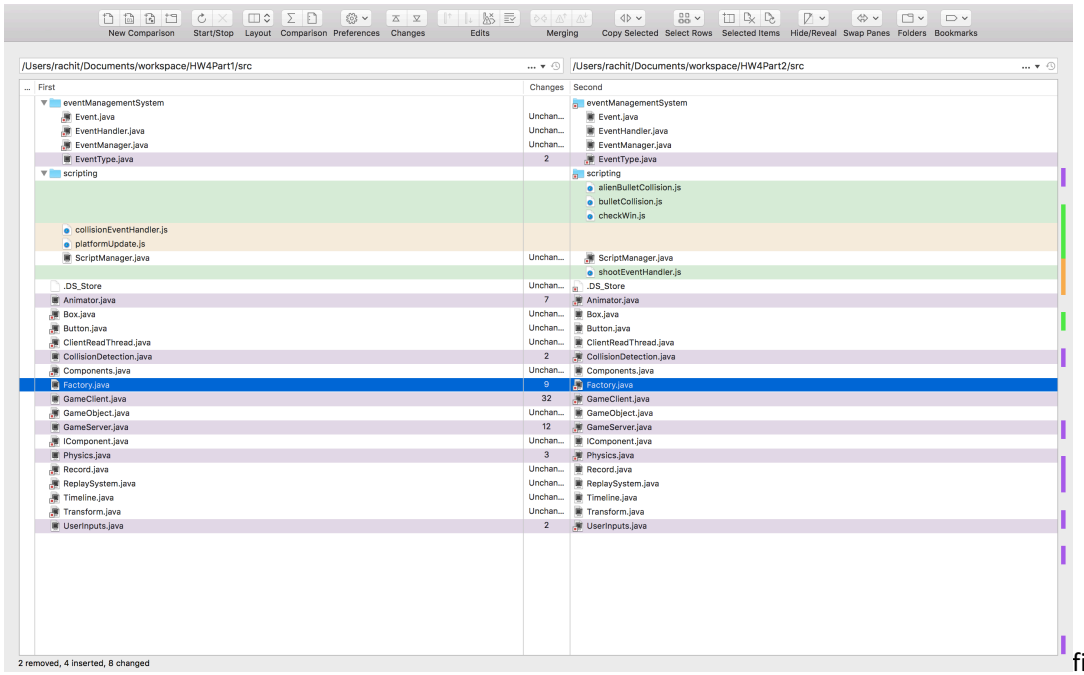
Assignment 4 – Part 4 – Reflection

CSC591 – Game Engine Foundations
Rachit Shrivastava

Once the 2D platformer game engine was up and running, there were few tweaks made at my end in the functionality to accommodate the requirements for two new games – Space Invaders (Part2) and Pong (Part3).

Attached are screenshots for the diff between all three game engines code mentioned above. I'm using Araxis Merge as the diff tool to compare two projects. I've also attached the number of line differences between two projects at a time. I will also be mentioning the changes made per file.



fig(i)a

Summary for all files

Between Folders 1 and 2
1466 lines in 15 pairs of unchanged files
2662 lines in 8 pairs of changed files
4 inserted files
2 removed files

	Between Folders 1 and 2			
Description	Lines (Folder 1)	Lines (Folder 2)	Lines (Total)	Text Blocks
Unchanged	1868	1868	3736	91
Changed	70	157	227	37
Inserted	0	158	158	32
Removed	7	0	7	6

fig (i)b: 1 is 2d platformer, 2 is space invaders

Fig (i)a, depicts the difference between **2d platformer** and **space invaders** codes, it can be observed that it involved a lot of code change.

GameClient.java involved a lot of change count (32), which mostly involved the following-

- New render functions for 2 types of bullets, one form player other from aliens
- Script calls have been added for some eventHandling, mostly collision component
- Removal of gravity factor which was being used form Physics component in 2d platformer, in both the games it was of no use.
- Since this is a single player game, platform(alien) and player co-ordinates/events are not transmitted over to other clients via server.

EventType.java includes two new events for shooting, ALIEN_SHOOT_EVENT and PLAYER_SHOOT_EVENT

GameServer.java – init data for platforms have been changed and the regular updates for their position now also calls the animator y axis change.

Physics.java – component has been devoid of using the gravity factor, it has been reset to zero and the default fall has been eliminated

Animator.java – the animation component for platforms lagged movement in y axis, adding this functionality was useful discovery for the game engine

CollisionDetection.java – the collision component was completely removed, since it wasn't designed to be a generic one, and for the games mentioned in homework, collision is mostly handled using scripts rather than through the component.

Factory.java – this designs the gameObjects to be rendered on screen, since it involved two new gameObjects, there two initialization functions were created for alienBullet and playerBullet.

UserInput.java – the component has a new trigger call for player shoot event, on spacebar input

Codes for GameObjects, Timeline, Events, Transform, Box component haven't been touched at all. According to the data obtained from diff, fig (i)b suggests that 12-15% of the code is changed or added, (227+158-7)/3736.

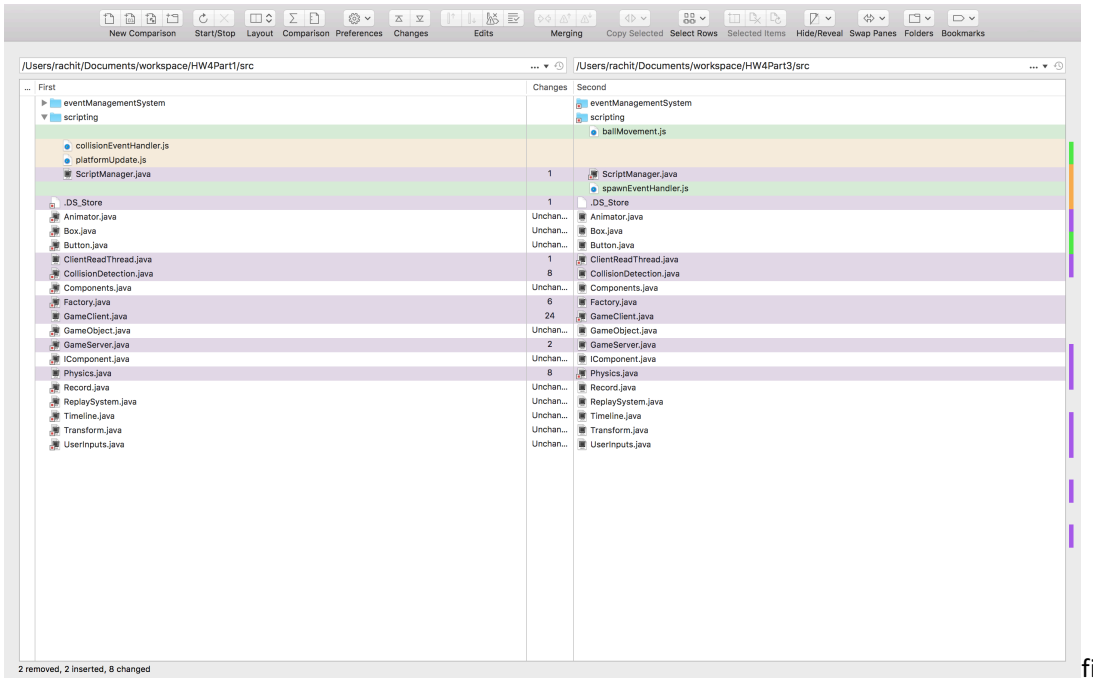


fig (ii)a

Summary for all files

Between Folders 1 and 2
1694 lines in 15 pairs of unchanged files
2269 lines in 8 pairs of changed files
2 inserted files
2 removed files

Description	Between Folders 1 and 2			
	Lines (Folder 1)	Lines (Folder 2)	Lines (Total)	Text Blocks
Unchanged	1889	1889	3778	72
Changed	51	56	107	30
Inserted	0	73	73	19
Removed	5	0	5	6

fig (ii)b: 1 is 2D platformer, 2 is Pong

Fig (ii)a, depicts the difference between **2d platformer** and **Pong** codes, it can be observed that it doesn't involve much changes as compared to Space Invaders.

GameServer.java – init data for platforms have been changed and the regular updates for their position now also calls the animator y axis change.

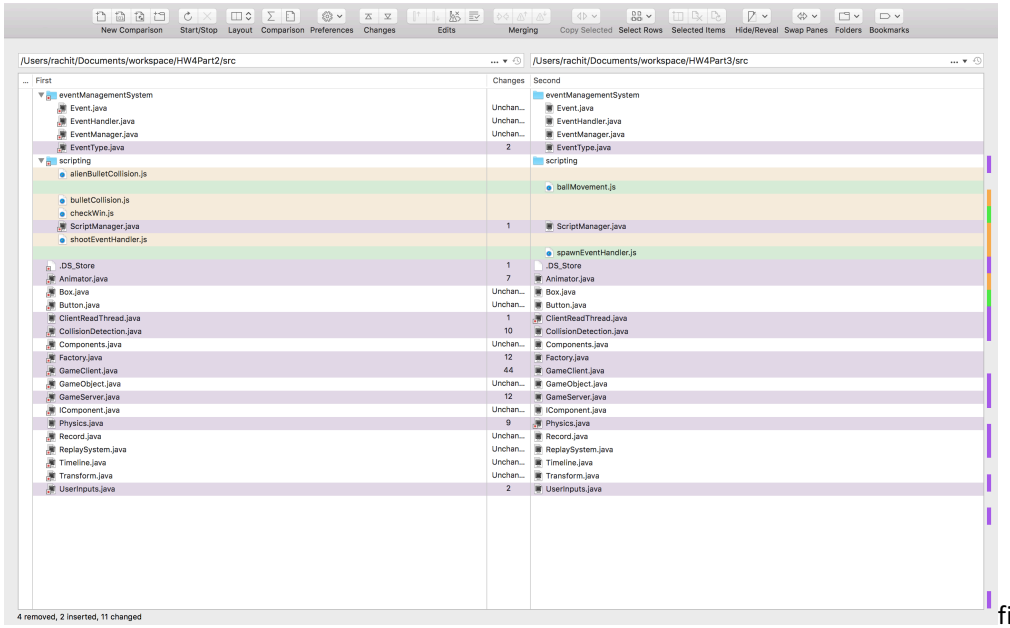
Physics.java – component has been devoid of using the gravity factor, it has been reset to zero and the default fall has been eliminated

Animator.java – the animation component for platforms lagged movement in y axis, adding this functionality was useful discovery for the game engine

CollisionDetection.java – the collision component was completely removed, since it wasn't designed to be a generic one, and for the games mentioned in homework. It had to be re-engineered.

GameClient.java – it mostly has codes to include scripts and bind the java objects to javascript.

Pong majorly uses the same code as 2D platformer, basic changes involving, removal of gravity factor, init data values, here the player is changed into shape of a platform and platform is changed into a square ball, functionality remains completely same, with platform moving in x-y directions both. Thus < 10% of the code is modified, seen from fig(ii)b.



fig(iii)a

Summary for all files

Between Folders 1 and 2
1292 lines in 12 pairs of unchanged files
2909 lines in 11 pairs of changed files
2 inserted files
4 removed files

Description	Between Folders 1 and 2			
	Lines (Folder 1)	Lines (Folder 2)	Lines (Total)	Text Blocks
Unchanged	1874	1874	3748	121
Changed	195	111	306	55
Inserted	0	33	33	19
Removed	114	0	114	33

fig(iii)b: 1 is Space Invaders, 2 is Pong

since Pong is more or less same as 2D platform, the data obtained by comparing Space Invaders and Pong projects is same as comparing Space Invaders and 2D platformer.

For both the games, my Collision, Physics components didn't work as planned. Collision had to be completely taken care off by scripting and for Physics some parameters had to be reset. Init data will always vary game to game, but there should have been a config file like structure to

set the default values while starting the server. I would have redesigned my CollisionDetection subsystem completely keeping all the scenarios in mind to be more generic, instead of game specific.
The best part was that with one Factory file I could create any gameObject using any combination of components available in my list, which made creating bullet objects easily. This creation was quick and my most of the time spent for this assignment was over redefining the Collision Detection functionality for both the games. Also the Transform and Box components were adequately designed to tackle all requirements for any square/rectangle figure.