# Side Scrolling Shmup Design Document

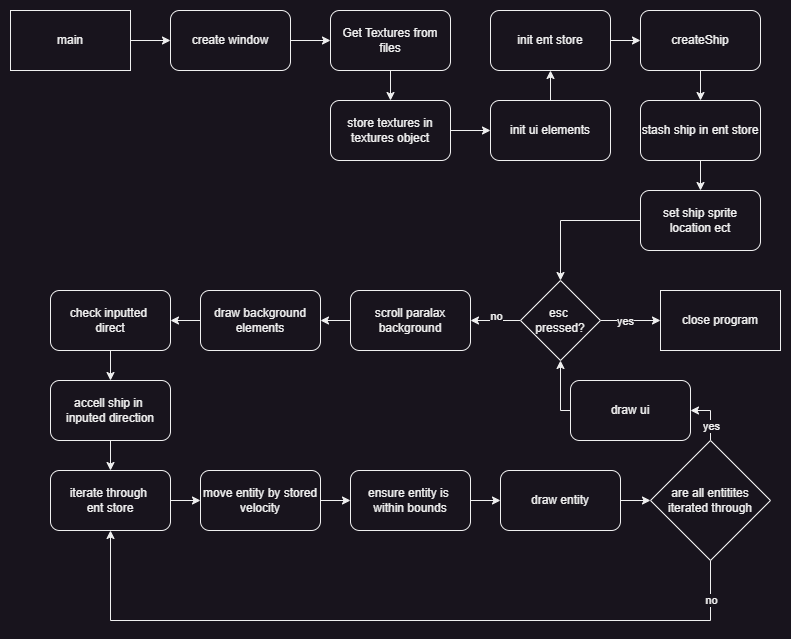
### Understanding the template

Due already having implemented a full entity system i will be using the my homework 3 solution as a jumping off point, rather then solution 11.

#### Procedure Dependency Diagram



#### Flowchart



## Upgrading the template

Alot of the ground work for this project is already present thanks to my work on homework 3, a robust entity system and entity store to easily manipulate a large number of entities.

In order to meet to specification i plan to extend the entity class to include collision as well as logic for collision cases. I plan to make a pool of both asteroids and bullets, in order to handle the large number of entities in an efficient manner, managing them with the entity store. This ship must be able to shoot, with the life of the bullets and asteroids handled correctly, respawning and activating/deactivating pooled entities correctly. The game must also have a fail state if a ship collides with an asteroid.

### Collision detection and anti collision

When testing the ideas to avoid overlapping asteroids, i was experiencing difficult to debug issues regarding collision detection based on predicting asteroid location, due to ensuring that the asteroids are no half off screen, i realised that by simply storing the lifetime of the entity (as i was planning on doing anyway) i could use that to move newly spawned asteroids when they, rather then relying on the prediction, as i was going to have to implement this solution for the bullet system later anyways. As such we do not need to run this prediction in the planned program.

E.g.

|  |
| --- |
| onAsteroidSpawn: if ((lifetime < SOMECONST) AND (ent.isColliding)) then {respawn} |

So collision function will look something like this

|  |
| --- |
| Foreach (entity : i):  if ((i.active) AND (i != currentEnt) AND (i.isColliding){  if ((i.type == asteroid) AND (i.lifetime < i.LIFETIMERESPAWNLIMIT)){  i.respawn()  }  if ((i.type == bullet) AND (currentEnt.type == asteroid)){  currentEnt.respawn()  i.deactivate()  }  if ((i.type == asteroid) AND (currentEnt.type == ship))  entStore.deactivateAllEnts()  deathMessage.activate()  }  }  } |

### Pooling entities

In order to make the game more preformant and the entity pointers more manageable all entities are going to be created at the start of the game, then simply manipulated in order to give the illusion that they are being created and destroyed. For the asteroids this is simple enough as they just need to be moved from one side of the screen to the other when they leave vision of the player. However for bullets we will require activating and deactivating them as required from the pool, to avoid them colliding with asteroids when deactive. All entities in the project are stored within the entStore object.

### Fail State

Due to how simple the gamestate is when this ship collides with a i plan on deactivating all entities and showing a simple text element over the screen.

### Storing Textures

In order to allow for a easy, performant and extendable texture management system i would like to store each texture only once (for performance after the game is loaded), be able to easily find these textures in memory in a human readable manner, and have a single simple place in the code to add or remove these textures from the project. In order to do this we will store all textures in one array, using an enum to index this in a human readable way. In order to populate this array we make a array of paths for indiviual textures, and use iterators for large batches of textures like asteroids, this array of paths is then passed to the texture loader.

|  |
| --- |
| populateTexturePathArray {  texturePaths[0] = "data/ship.png"  texturePaths[1] = "data/bullet.png"  for (i : NUMOFASTEROIDTEXTURES) {  string path = "data/asteroidSprs/" + i + “.png”  texturePaths[i+ PREASTEROIDTEXTUREAMOUNT] = path  }  } |

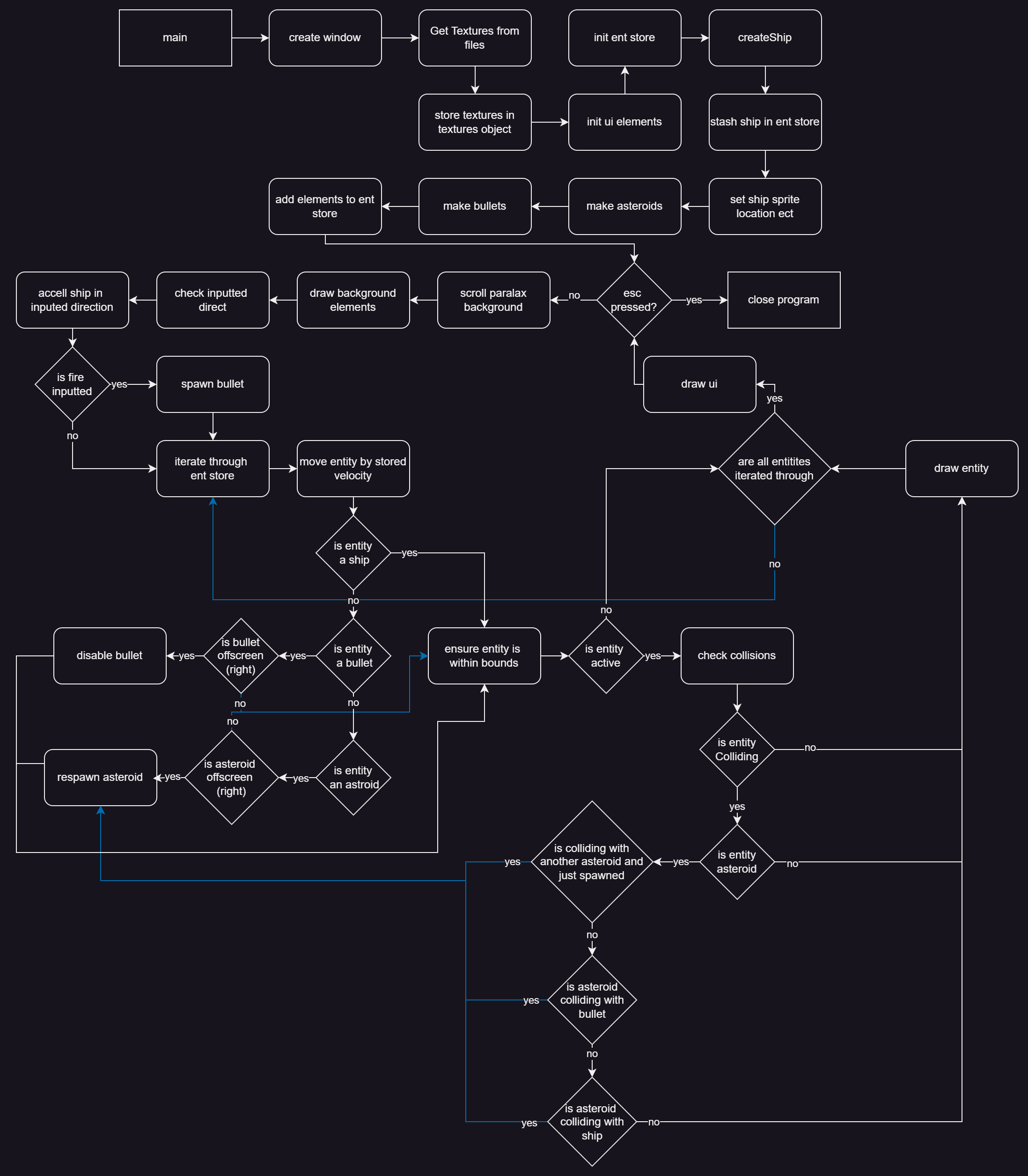
|  |
| --- |
| populateTextureArray {  counter = 0  foreach (i : texturePaths) {  myTexture = loadTextureFromFile(i)  if (failed to load texture){  throwError(“failed to load texture”)  }  textureArray[counter] = myTexture  c += 1  }  } |

### Animation

In order to store animation data we will define a custom struct, that can be iterated though and store the pointers to the frames in the animation, in order to make it framerate independent each frame duration is stored.

|  |
| --- |
| current frame = currentFrameLifetime%frameDuration |

### Upgrade plan flowchart



## Upgrade plan procedure dependency diagrams

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Postmortem

## Unplanned additions

#### Invulnerability frames

During playtesting sometimes i would encounter dying on spawn, in order to counteract this the player gets 1500 frames of invulnerability when the game starts.

#### Shooting cooldown

Due to the lack of an input buffer, and to save you fingers, the player is allowed to hold down the fire button to continue fireing. However this lead to a stream of bullets that both broke the pool system, and just felt ridiculous. In order to solve this a cooldown to the ability to shoot was added.

### The bad

#### Its not fun

The game is simply not fun, due to the lack of a real objective or score, as well as the ease of the game, there is little engagement to the player.

#### Missing juice

Due to the lack of sfx and vfx there is little satisfaction that comes with the player actually interacting with the game and its systems

#### Inconsistent timing

The game inconsistently uses delta time between frames and framerate for its timings, framerate with a consistent set framerate should be used, or delta time to allow the game to function independently of framerate should be used, due to the current unlimited framerate systems like the “air resistance” of the ship currently have different magnitudes of effect on different systems and at different times on the same system.

#### Missing content

Due to unplanned events i ended up having about 2 weeks less time then expected to work on coursework at the end of the semester, coupled with some minor overscoping in the backend and some strange and difficult to debug errors, this has left a few things i would like to add. Most obviously the animation system that was planned in the doc was cut due to time. I had also planned to make the a much more advanced ui system, however it was the least important of all the elements so it was cut for time. The game is also desperately in need of an input buffer, as well as some squash and stretch on the ship to make the movement fell more powerful.

# Git Log

|  |  |
| --- | --- |
| commit 6692c923720b355bcd67cfe2a7bd5ded35f489dd  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Wed Dec 18 23:17:22 2024 +0000  spellings fixed  commit fee4da648a26e757b1d7c4f3c9e31bdba36e13fe  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Wed Dec 18 23:08:14 2024 +0000  postmortemDone  commit b2df308539e1d44208ac79c25744a442e92b7b46  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Wed Dec 18 21:12:29 2024 +0000  tone of hw4 writeup done  commit 232dae4fc5b1848ad26705a2fb8b441809c74eff  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Dec 17 06:12:52 2024 +0000  refactor complete  commit d621c119c64d0a8eb942dd2ddfba889a6d2326c3  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Dec 17 05:23:19 2024 +0000  reformat into header files properly  commit 19a03bb98e3409ace6dedcb2e057ee24d65a83f2  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 16 21:09:41 2024 +0000  Update main.cpp  commit 86dcaa0b0f6d7bf7360db59f491901f236ab8219  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 16 17:39:27 2024 +0000  Update entStore.h  commit a6e82cebe899edfc6d0fe64bd9e05d1dba6e0944  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 16 17:39:17 2024 +0000  all functonality in fr this time  commit 9e10dbc3a46107879cefda3e03f7d079e2f2cc14  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 16 12:19:34 2024 +0000  just need rework+death screen  commit f25e9f983a37c5249f8e395e7d87acdace2c8504  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 16 11:00:18 2024 +0000  namespace no work  commit 3045c80845221a91db6ff873aca6082808ca341a  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sun Dec 15 05:14:11 2024 +0000  refactor - whao the nhs is sooo slow  commit f92c305c6466204a185e7f11d721764bf9d836fd  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sun Dec 15 04:17:01 2024 +0000  functionality all in  commit a673310d55bffe69a57bb12a33dbbeff36eea514  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sat Dec 14 22:32:42 2024 +0000  bullets done  commit a16676081f78f95d8fc9321133a8ad9e448abbdf  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sat Dec 14 22:09:48 2024 +0000  bulletsSortaWork  commit 2f1a918228821d2178b3c7c44649aea84a3d030b  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sat Dec 14 17:50:57 2024 +0000  fixTheHeisenBug  commit aca5af7cc25c7ffb2e0b438ebc2ef0ee6ca8e82b  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Sat Dec 14 15:39:59 2024 +0000  pre entVect rework  commit 4ae6eb0374d6de21a245ac569d4eedf6f14a1e34  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Dec 13 14:15:46 2024 +0000  pooling didnt fix  commit 3fe1cd2638bc9ca19a3cdfc4d30f082c3921f017  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Dec 13 12:11:19 2024 +0000  bulletNotWorky  commit ff55df0e91ad5f3ec827b1eced13ed4a55f45b54  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Thu Dec 12 18:37:36 2024 +0000  fixxxxed | commit 405b84aa861f53d3815f2d90f51b74013f5ff804  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 9 21:03:12 2024 +0000  whyshipMoveNotWorkyNow  commit a74f501799db4808c1b011e5fa611ae014b1381c  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 9 19:06:08 2024 +0000  insane definition bug - wtf do u mean 6 it doesnt take 6 vars  commit db9f8131f967f2da777eab779bd5d1ee833b256f  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 9 18:55:26 2024 +0000  collionsOnInitWorking  commit a03c93b8b50466b24662fb23b4b71b52524d39b6  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Mon Dec 9 17:47:54 2024 +0000  collision scoffold  commit 45173d17fbdd4951b70c16a2c15c9d1b1f81071d  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Dec 3 20:15:56 2024 +0000  manyFiles  commit a73bffce092977f5e97b52a01b7cb376e057f740  Author: Field <c4024637@my.shu.ac.uk>  Date: Mon Dec 2 11:38:59 2024 +0000  worked on a buncha non asteroid stuff  commit 3bdec56be42f9c6d941537deccac4861e2702e5c  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Nov 29 14:16:47 2024 +0000  asteroidsNotMovin  commit 5c529a5b8d6eabb7702b634193c78b9d8f082c51  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Nov 29 12:58:46 2024 +0000  i hate these asteroids  commit e384667a092c921387f3d229c061e54d0d25c0db  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Nov 29 02:01:36 2024 +0000  asteroid collision - whyyyyyyyyyyyyyyyyyyyyyy  commit 00a3c7e14613675ba80b10cbd61baea06cb84e71  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Wed Nov 27 19:11:02 2024 +0000  more design doc stuff  commit 6ce15c87504ece106724b5d83bce27c9be874d33  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Nov 27 09:35:20 2024 +0000  mostly done pre plan  commit 1add5116acb5be02ff813a78937b17a454203e0e  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Nov 24 09:12:03 2024 +0000  collision detection plan  commit 23eace2320219202f81ae4c788750988a44c1dd7  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Nov 24 08:59:42 2024 +0000  planned flowchart done  commit 7eb8e705cd4800570a0f28f30c14c2e0b5b191f1  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Tue Nov 24 08:10:36 2024 +0000  startedHw4Writeup  commit a0850c633a9da27ea120bf2976d727e03cce887f  Author: dr1zzleGDShu <c4024637@my.shu.ac.uk>  Date: Fri Nov 24 00:30:47 2024 +0000  hw4 |

## Testing

|  |  |  |
| --- | --- | --- |
| Test | Expected outcome | Actual outcome |
| Player spawns on asteroid | Game over | As expected |
| Bullet hits asteroid | Asteroid “dies” and respawns off screen | As expected |
| Asteroid respawns | Player cant see respawn, doesnt overlap with other asteroids | As expected |
| Player fires as many bullets as possible | No changes | As expected |
| Player collides with wall | Player gets stuck in bounds | As expected |
| Player fires while colliding with all 4 walls | No changes | As expected |
| Bullets leaves screen | Bullet deactives | As expected |
| Asteroid leaves screen | Asteroid “dies” and respawns off screen | As expected |
| Player collides with bullet | No Difference | As expected |
| Game played on faster machine | No difference | Spawn Invulnerability is shorter, player slows down faster and moves slightly slower |
| Long run of 10+ mins | No difference or crash | As expected |

## Time spent estimations

## 

|  |  |  |
| --- | --- | --- |
| Task | Estimated time (approx hours) | Actual Time (approx hours) |
| Plan and diagram improvement | 2 | 5 |
| Create asteroids | 1 | 1 |
| Create collision system | 1 | 1 |
| Debug asteroids spawning on each other | 1 | 5 |
| Bullets | 1 | 4 |
| Create ui | 0.5 | 1 |
| Comment codebase | 1 | 2 |
| Polish | 2 | 4 |
| Post mortem | 2 | 1 |
| Fin docs | 2 | 2 |
| Total | 13.5 | 26 |