Radical Conversion Documentation

Concept & Inspiration:

When choosing the project to convert, I decided to go with the space invaders style example because I could see a lot of changes I could make to have it resemble the original space invaders more.
This meant that my inspiration from the start was the original space invaders.

As we can see here, the alien ships can shoot back at the player, and the players ship has multiple lives.

This is what I attempted to impliment into my own project.

However, knowing that copying a previous game is not what is being asked, those two aspects were the only two that I took and implimented into my own game.

Since the ship and aliens were simply blocks, I started to work on textures and had to look for some ideas of what to draw.
For the players ship, I decided to take inspiration from the tank in Metal Slug.

The reason for taking inspiration from this artwork was because of how exadurated the idea of the Meatl Slug tank was.

The tracks were oversized, along with the body, and the whole style of it was cartoonish and created it's own atmosphere.
This was exactly what I wanted for my player as well, so decided to go with this.

Ofcourse, the ship I created had to be shooting upwards instead of side to side, meaning that necessary changes were made.

Since I had decided to go with a silly / cartoonish style for the game, I started to look for a similar style alien ship when I came across this image:

To start with, I relied heavily on this image to design my alien ships.

The reason for this is because the idea of having a face on the ship instead of aliens in the ship was the furthest from serious that I could get; perfect for my game style.

After this design, I decided to create my own versions of this style ship for future levels.

Development Process:

To begin with, I decided to start changing the sprites for the ship and aliens, using the inspiration pieces mentioned before, because that was the simplest part of the process that I knew what I needed to do for it.
I used a website called piskelapp.com which is a pixel art creation website. With the tools on here I was able to get the correct style for my game.
The sprites that I made can be found here: https://www.piskelapp.com/user/6063982606548992#
After creating the sprites, all I had to do was download them into the directory asset file with the same name as the previous sprite in order to replace the existing one with my new creation.

Once the first set of sprites were drawn up, I was able to use them as reference for the later levels because I could change certain aspects of them, such as colour and shape (to an extent).

The next step after creating the initial sprites was to have the aliens shoot back at the player. The reason for this was to make the game more challenging, and to make it more like the original space invaders game.
I used this phaser example (https://phaser.io/examples/v2/games/invaders) to find out how they would track the players constantly changing position and act acordingly when the plater was hit with a projectile.

Whilst looking at the 'enemyFires' finction, I saw that this example also had reference code for multiple lives, which was another aspect that I wanted to implement into my game to make it more similar to the original.
This was good for me because the code was made by the same person, so seeing how the enemy bullets function and the player lives function interacted was a good guide on what I had to do with my existing code.

On top of these two main functions, I had to look into the respawning of the player ship once it was hit the first time. This proved to be a big problem for a while.
Once I figured out how to make the player respawn, it would default to the starting position. This doesn't seem too bad on the face of it, but after playtesting I found that if the ship was hit near the default spawn then the same bullet could collide with the ship twice in it's life.
To stop this from happening, I made a random generation for the ship every time it was hit with a bullet.

Now that I had all of the sprites for this level made and implemented, the aliens were shooting back at the player, and the multiple life function was working, I needed a way of making a more dramatical change to the code.

This is when I had the idea of making multiple levels. The reason behind this was to give the highscore function more reason of being there.
As a result of the first level being quite simple, I decided to have more levels with increasing difficulty to make it more of a competition between players, like it would have been in the arcade years ago.

In order to have multiple levels, I had to make more states in which the new information for the level was held, and at the end of one level, I had to code that the next levels state needed to be called. This would then look at the state that was called and then run that state until told otherwise.

What this meant was that a lot of the information needed to be copied and placed into the next state. Once the code was copied into the next state, I had to figure out what code was not needed a second time, to reduce the overall amount code that needed to be read.
Next, changing the respective fields to make each level harder was simple enough. The second level had faster moving enemies and the third level had even faster enemies that fired more bullets at the player. This could continue further to a point where it would be nearly impossible to complete, but I decided to go with three levels because of the time limitations and my knowledge of the phaser code.
Along with changing the numerical values, the sprites also needed to be updated, which is what I was mentioning previously. I used the first sprites as reference for the level two and three sprites.

Once levels two and three were completed with their respective difficulty changes, I found that the flow of gameplay was far too rushed in the sense that you were thrown into the first level and were not able to stop at all.
This is when I decided to add in a start screen to the game, and pop up 'congratulations' screens in between each level to give the player a break.

Again, this was a simple matter of adding in more states, which I had gathered more knowledge of by the addition of levels.
I also used the 'game over' state as a baseline for these pop ups and menu screens, so are all images and can be closed by clicking. The screens were made in GIMP 2.0 using an arcade style font to recreate the pixel 8-bit style of game that I was going for.

Problems & Insights I had:

The first major issue that I ran into was getting the players ship to respawn after being hit by an alien. This took me roughly two days to figure out and was solved by a simple addition of a line of code in the function that controlled the enemy bullets and the players ship.
The line of code is as follows:

```
this.ship.reset(this.game.rnd.integerInRange(0, this.game.width * 1), game.world.height * 0.92);
```

This line is placed into an 'if' statement saying that this is what will happen if there are lives left for the player. This also shows the fix to the next problem I ran into.

This next problem was that the players ship could potentially lose multiple lives off of one enemy bullet, so I decided to add in peramiters for a random generator for the ship to spawn on.
The "rnd.integerInRange" part of the code is what controls this, and was resolved after a few hours of searching for a way to impliment random generators into a setting like the one I had.

The final major issue that I ran into was how to go about adding more levels to the game.
As a result of google searches being too vague about the situation, I asked my games design teacher and they pointed me in the direction of using multiple states for each level. There was also another method that was mentioned when I asked for help. However, I decided to go with multiple states instead because the other way would have me defining all of the speeds and ammuntiion of the aliens to begin with as a set value. This was not what I wanted as I needed the difficulty to increate (things like the speed of the aliens and amount of bullets that were being fired).
Multiple states was perfect for what I had planned.

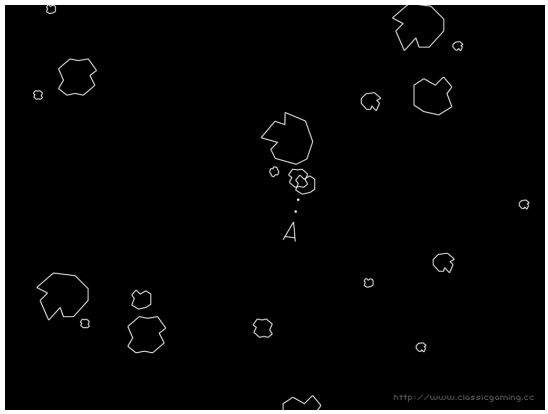
Other than these main problems, there were a few minor problems, such as debugging, that I ran into that all got resorted after a matter of minutes by following the command console on the atom live server in Chrome.

Future Upgrades:

The clear option is to add in more levels with increasing difficulty to make it a true challenge for the player to beat every level.
In addition to this, I had the idea of making the levels more difficult in other ways such as flipping the screen upside down, or maybe inverting the controls of the player. These kinds of changes would be unexpected by the player and enguage them more to achieve the goal of completing the level.

As for aesthetic changes that I would make to the game, I would like to add in a background that is more interesting to look at than a grey screen. This will add to the story of the game and may make it more enjoyable to some players.
The background theme would be that of a ruined / abandoned city, or military base of some sort.

On the other hand, improvements for the game as a whole, I might try to change it to be more like asteriods.
What I mean by this is that the ship is in the middle and asteroids are closing on your location so you have to shoot them before they hit you, taking one of your lives.



A simpler change would be to have a never ending wave of aliens coming at the player that increasingly gets faster and they gradually shoot faster and more often.
To go about this I would have to utilize the game time feature to set a minimum and maximum speed and apply the speeds and capacity amounts to respond acordingly.

Repository Link:

https://github.com/ZyaxitronED/Phaser-Conversion