Our Future is in your hands.



05-2017

João Costa

IPCA

**Resume**

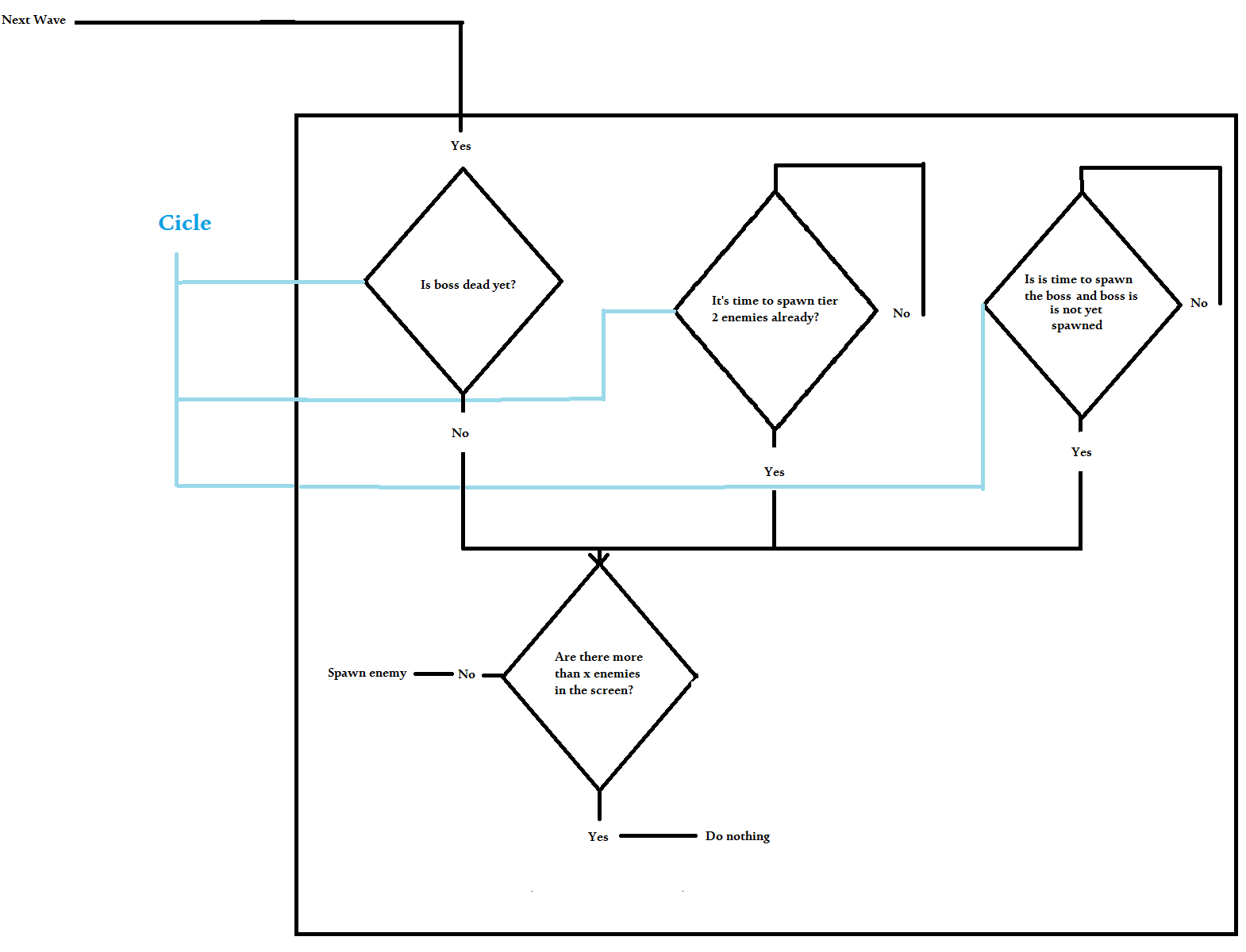
Gliese is a single player 2D *Shoot-‘em-up* arcade game developed with c# and monogame (XNA based).

The game exposes the conflict between Terraquian humans and Mars colony for the conquest of a new planet to call home. The planets being disputed are two of the Gliese group of planets. By Gliese group of planets I am actually referring to planets orbiting a star named Gliese, there are several stars called Gliese by the humans, yet the one in the game goes by the name 581, and the planets being fought for 581c and 581g.

In the game the player is the Captain of a ship and his job is to defeat the most number of enemies possible, accordingly to their tier and category you will gain points, points which will unlock new ships.

My inspiration to make this game came especially from space impact game from the old Nokia phones, but also Galaga, Chicken invaders and some other Android titles that I played.

**Game Blocks Diagram**

****

**Game Systems**

Score

The score in the game is an attempt to tell how well the player performed. The score increases every time an enemy is destroyed. The amount of points gained is determined by the collision damage of the enemy, those two systems were combined in purpose to make bosses really worth to kill (because they are the really hard thing about the game) and also to make sure the player ship was always destroyed when colliding with a boss.

Achieving a certain amount of points will unlock new ships.

Breaking the game loop

The game can end in three ways: the player quits, the player dies and the player finishes the game.

Bonuses

Apart from the original stats a player can pick, there are also power ups that will appear randomly in the screen. These will increase the player HP by a flat amount, projectile damage, shooting speed and ship damage. On the other side being hit by an enemy projectile will reduce these stats, both increase and decrease are capped.

Some stats aren’t increased/decreased on purpose, while at first it may seem pick endurance stat (that reduces the damage taken by a %) is the best for endurance, especially because power ups are flat amount and not HP based the reality is endurance only works against collisions with enemies but not enemy projectiles.

**Controls**

The controls are very simple, as the ship has natural speed degradation as way to go backwards there is no key associated with going backwards, however to move the ship forwards, up and down the player can use the keys D,W and S respectively.

To shoot the player simply needs to press the space bar, the amount of projectiles shot is determined by a timer ( or shooting speed, however you want to call it), timer this changed by the ways I mention above in game systems section.

While in the upgrades menu the player can change between ships and increase/decrease the points spent on an ability using the arrow keys and the enter key.

**Implementation**

Under the hood of the game the thing more frequent to find is random trickery integers, integers made to avoid the creation a whole new class or a more complex structure. Lists are also very common as the spawning and unpawning of objects is constant, handling it without lists would be near to impossible and most important of all, support classes that will sit in *game1* class.

The 3 Major classes are: *game1*, *LVL* and *Ship*. Some functions that should be implemented in one class were implemented in the other and vice versa. These are definitively the most important ones, they coordinate all other classes.

*LVL* class coordinates the enemy waves spawn, some collisions and power ups.

*Ship* class is responsible for ship and ship bullets drawing, stats handling

*Game1* class coordinates all others, handle some collisions, timers and all music is controlled by this class, along with button trigger delay, among other functions.

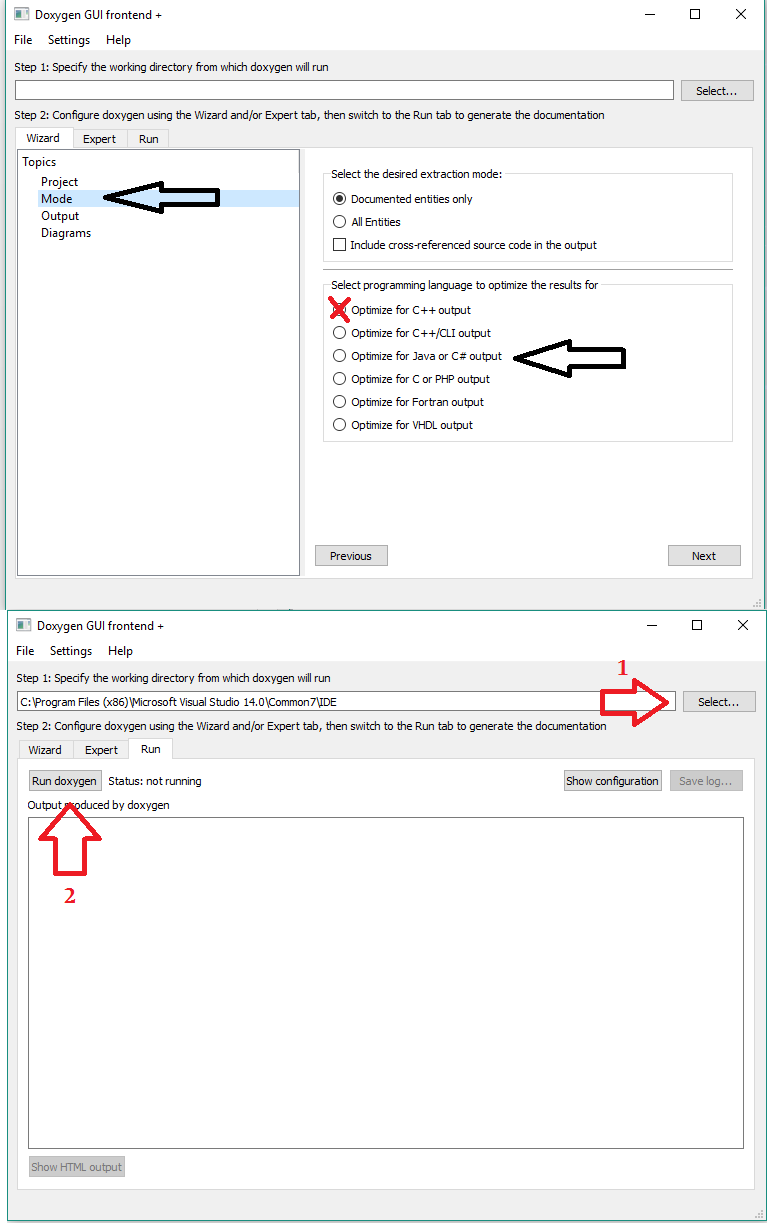
A big help was the tutorials by *gamerdad81* in YouTube:

<https://www.youtube.com/channel/UCCHFS0odXntBz9-5SvB5Ikg>

Major Difficulties: Some random unexplainable errors; the one that took me more time to figure out was one image without update to content.

Typos errors, such as the lack of “;” or “}” that instead generating an error would run the code differently than expected.

Within the main folder of this project is the documentation of the source code of Gliese made by Dyoxigen. In order to obtain such documentation I downloaded the installer from <http://www.stack.nl/~dimitri/doxygen/download.html> and applied basic settings to my project.



**Art**

All images came from:

<https://pixabay.com/> A high resolution images gallery with most images under Creative Commons 0, public domain license.

<http://millionthvector.blogspot.pt/> *millionthvector* is an excellent artist who puts out for free a lot of spaceship images.

<https://opengameart.org/> probably the largest site with images for use in games.

<https://pt.cooltext.com/> A site with lots of free text generators.

Sounds came from <https://www.freesound.org/>

And music from <http://freemusicarchive.org/>

**Development**

The Base idea evolved slightly, I wanted to do a Mars VS Earth space shooter thing and two Gliese planets where a good reason and also a good name.

Along the development the idea remained the same but lack of time started cutting it, for example where I had imagined the player picking the faction to fight for, the player is put on Mars side whether he/she wants or not.

The planned schedule was not totally accomplished, not just I started doing the game latter than supposed, in part because I didn’t know how to start, but also because I lost my files. So instead 6 weeks for code development and 2 weeks for art and assets I had only three weeks.

Apart from the code I had no difficulties. Images and sounds where handy and though scaling was a problem at first, doing some undertricks with screen width and height made everything work nice.

**Future Development**

A lot of ideas where left behind in the making of the game, within all, these are the ones I remark as the ones I would more eagerly implement:

* Additional Levels;
* Different power up and stat system;
* Narration of the history and small cuts along the progress of the game with a little corner pop up of a character and also spoken dialog;

**Copyrights**

**Font: *Halo***

Made by Raymond Larabie

<http://www.1001fonts.com/neuropol-x-free-font.html>

**Images: Enemies**

**Enemy 1 wave 1, enemy 2 wave 1, boss wave 1 and enemy1 wave 2:**

Made by wubitog and Skorpio http://opengameart.org/users/skorpio

**Enemy 2 Wave 2 and Boss Wave 2:**

Made by Dravenx

https://opengameart.org/users/dravenx

**Enemy 1 wave 3, enemy 2 wave 3, boss wave 3 and megaboss:**

Made by millionthVector

<http://millionthvector.blogspot.pt/>

**Images: *Player Ship***

Scoundrel, Chaser and Goliath ships that can be used by the player were made by millionvector

<http://millionthvector.blogspot.pt/>

**Images: *Power up***

Made by TRBRY

https://opengameart.org/content/power-up-icons

**Images: Explosion**

Made by: Master 484

https://opengameart.org/content/explosion-set-2-m484-games

**Music: *Falling Sky***

Made by Ketsa

<http://freemusicarchive.org/music/Ketsa/May_Starlight_Find_You/falling_sky>

**Music*: Space Blockbuster***

Made by Soulbringer

<http://freemusicarchive.org/music/Soulbringer/Soulbringer/Soulbringer_-_Space_Blockbuster>

**Sounds:**

**Space Ship3 shoot sound:**

Made by debsound

<https://www.freesound.org/people/debsound/sounds/339169/>

**Space Ship 3 Startup:**

Made by tyops

<https://www.freesound.org/people/tyops/sounds/250843/>

**Wrong button sound:**

Made by SgtPepperArc360

<https://www.freesound.org/people/SgtPepperArc360/sounds/341732/>

Disclaimer:

Some images came from pixabay, pixabay has no requirements of referencing the poster of the image in any way.

Most sounds are also under the license 3.0 O (public domain) so no reference is necessary

**Conclusion**

This project was a big leap for me, not just in game designing, but also in C#. Before doing this project I was not aware of some things such as constructors, or what the main game loop was. It was definitively enriching and I learned a lot of gamming programing and development.

