Technical Design Document

Updated Technical Design Document

Team List and updated responsibilities

Alliance of the Auld is the name given to our team that was composed of 2 French students, one Scottish and a Greek student within the Development Team. We all have different backgrounds and that makes us more interesting since we have 4 different points of view.

- The team Leader is James Keohane, he is a French student that studied 2 years of web development. In the AoA team (Alliance of the Auld) he acts like the Project Manager and the Lead Designer during game implementation he will also act as a game dev.
- The Lead Dev is Ryan Kennedy, he is a Scottish student that did 2 years of Game Development at the College of the West of Scotland and then moved onto his 2nd year at UWS. In the AoA team he is our coding referee on the Unreal Engine because of his knowledge and experience. Ryan will be mainly working on the game implementation and multiplayer side of the game.
- The Level Designer is Vincent Berthet, he is a French student that studied 2 years of electrical engineering and industrial computing in AoA team he will design and implement the GUI of the game.
- Neoptolemos “Thomas” Papadiofantous has re-joined our team two weeks after the development started. He will work has a game developer on Unreal Engine with Ryan. Thomas did 2 years of Computer Game Development in UWS. During implementation, Thomas will assist Ryan in his work.

Unfortunately, in the middle of the development Ryan and Neoptolemos left the group.

Overview of Project

The concept idea came from a mixed of Stellaris and a modded map from Warcraft 3 called missile war.

The team wanted to make a game that focuses on planets, shooting missiles, and controlling resources. Stellaris is a 4X game that focus more on tactical choices, strategies and politics than big battles whereas missile war is only building stuff to make your shield stronger or organize your weaponry to shoot more missiles. So the Project Auld was born.

The player is the Grand Leader of a nation that live not that peacefully in a solar system where there is another player that have his nation settled in a neighbouring planet. The two players will have to make their nations expand and evolve through various development branch to pressure their opponent into submission or annihilation.

This game is mainly resource management and tactical choices. The main planet is protected by a shield that prevents the opponent weaponry from harming the population. This population will have to produce energy and tech points in order to make weapons powerful enough to overcome the opponent shielding.

The resources are scattered through the entire map on moons and asteroids and can be salvage by workers, space stations or settlements. These resources will aid the player into developing its technology faster, expanding further and building more powerful weapons.

An army can also be built to attack everything that stands in the outer space. Their main purpose is to take control the opponent settlements, destroy and scavenge the workers and space stations.

Religion is another way to turn the tide of battle by praying that something strange will happen in space …

Final Implementation

The final implementation of the game before first release includes the core mechanics: Main Planet, Missiles, Shield, Multiplayer and features: Power of Faith and Technology Tree.

The nations will be implemented in a second cycle of development. Only a meta-nation without background will be implemented. Implementing the nations is also a big piece of work on its own because it modifies every aspect of the game and its gameplay but it is not required to implement it first as the game could be played without nations.

Population will be the last feature to be implemented to the game but it still has to be decided if it could be a fun implementation or not.

See in Game Design Document: Project Auld for any design details about the game.

Updated Risk Analysis

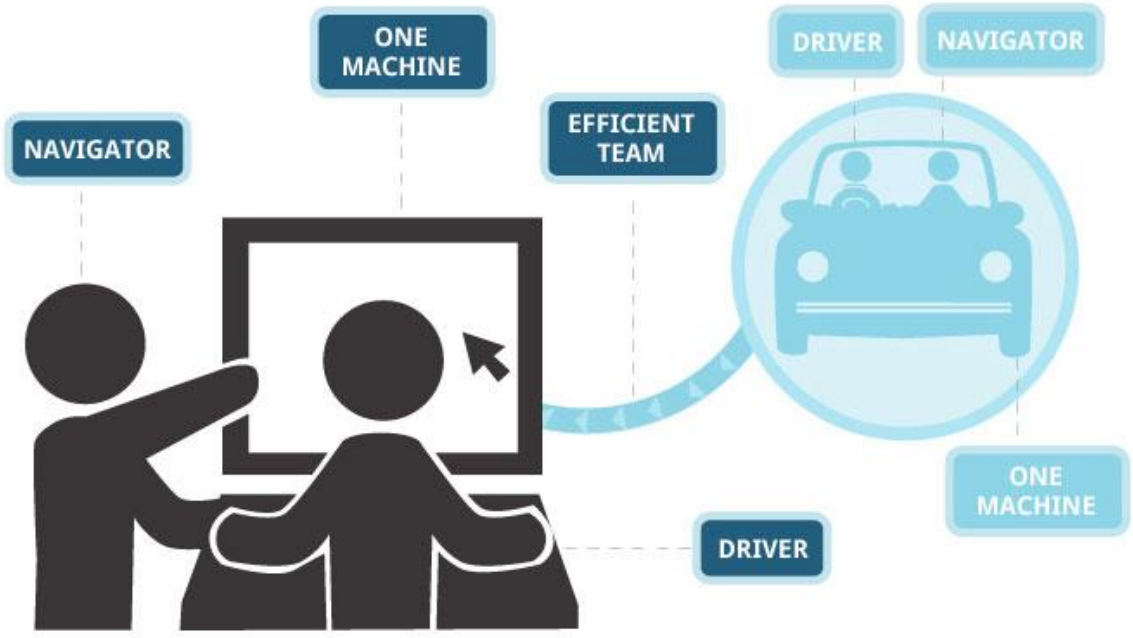
Risk	Score	Recognition	Resolution
Team Member absence	EP = 2 EI = 4	Sometimes people get ill so when illness arrives the work cannot be done properly. Working is also time consuming.	Task that haven't been should be given to someone that can manage them and wait until the team member is healed in order to give him another task.
Equipment Failure	EP = 2 EI = 5	The personal computer can crash or something can go wrong with the software.	A task that can be accomplished with a UWS computer will be given.
Incomplete Task	EP = 4 EI = 3	A task can be unfinished, not well done or too hard. It will make the development process longer.	If a task is incomplete 2 team members work on it to finish it correctly.
Unreal Engine learning	EP = 3 EI = 5	In order to make the proof of concept, team members should learn some basics from the unreal engine in order to code.	Books from the library are good ways to learn Unreal Engine 4.
Source Control fail	EP = 2 EI = 4	GitKraken is a bit hard to understand for the first time using it. So sometimes the code isn't correctly shared between team members.	Help the team member struggling with GitKraken by sharing screen on skype or assisting him by whatever ways possible.

Team/Project working

Pair Programing was the key at finishing our project even though the game is not playable.

At the beginning of the second trimester we all different task to work on that were on Trello but then Neoptolemos and Ryan stopped working so we changed our methods and we were two developers working on one computer: Pair Programing.

We knew that we could not finish in time so we worked in pair to speed up the development and the learning of Unreal Engine 4.



The benefits are real:

- Coding efficacy
- Less learning to do
- Less bugs
- Faster reflexion

Game Implementations

Player Controls	
Main Planet	
Hit Points	
Shields	
Missiles	
Projectile	
Shield Destroyer	
Crippler	
Gatherers	
Worker	
Moon Station	
Settlement	
Resources	
Material	
Energy	
Tech Points	
Faith Points	
Multiplayer Features	
User Interface	
Console	
Resource Indicators	
Missile buttons	
Gatherer Buttons	
Main Planet life and shield bar	
Game Time	

See the game features in our youtube video:
https://www.youtube.com/watch?v=DXS8MZFRZzk

Quality Assurance Plan

The project had attained his worst scenario with Ryan disappearing and the graphic card of my computer that died in January. So, to limit the waste of time I've bought a new graphic card as soon as possible but unfortunately I couldn't replace Ryan.

Maybe merging with another group that also lost their team members could be another good idea into creating a game and learning more from their experience.

In our methodology, we planned to work onto an agile method but the method is pointless if there is no coordinator and since we were only 2 team members we switched to Pair Programming so we could boost the development and minimise some risks like job not done.

Testing

The debug testing was mostly done during development time since the game couldn't be playable.

We had two different kinds of testing: visuals and game features.

Whenever a piece of code has been written it is immediately tested to see if it works by using the Print Screen node to witness the success or failure of the piece of code. If it fails, we check what was going wrong. If it works, we continue implementing game mechanics or improving the code until it was satisfactory.

The second kind of testing during development was the visual aspects of the game elements or the UI.

For the game elements, every game feedback had to be visually representative of what they do in the game so, for example, the missiles had to leave a trail behind them and explode on impact to show the player the harm of the projectile that he had launched. So, testing their visual behaviour was primal into our game.

For the UI, the challenge was that the user interface had to fit a screen in high resolution perfectly and resize onto smaller screens. So, when it was ready for testing on 1920* 1080 resolution we tested several times in that same resolution for every flaws like a UI element that was too big or a button that didn't have his place until every element.

Individual Critical Appraisal

James Keohane-B00313851

I felt the end of the project like a waste of time because I knew that the game couldn't even be playable without any research onto how to make a multiplayer game, even a simple one between two computers.

But it was interesting to work in pair with Vincent because he learned how to make a user interface in Unreal Engine 4 and he taught me how to work with his user interface without wasting much time so I found it efficient and that was a good point from the project.

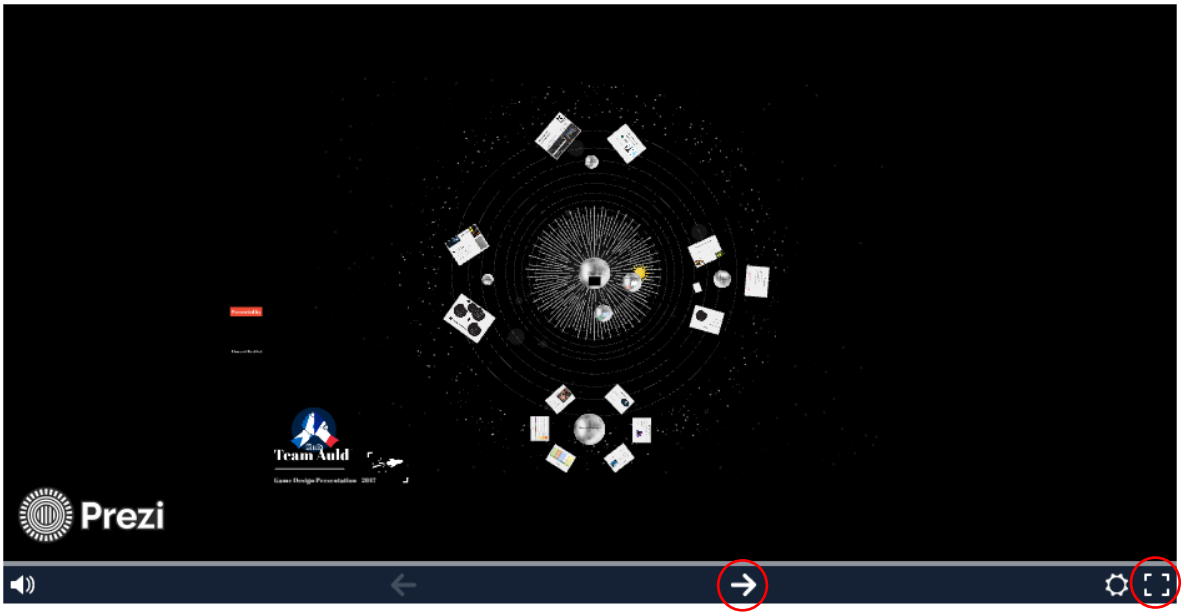
Now that I know how to make the basic control of an RTS I am quite glad of this experience. I am looking forward to code a multiplayer lobby and try to make people connect onto an RTS style with the same controls that I have coded in this project. Could be fun.

In the overall, the project was interesting and I learned a lot from this experience though I lost passion from making this game because of our team mates. So, in the future I'll rather work alone or find a team already composed that works well like a company. Though there is not much to say.

Presentation

The presentation is available online on Prezi.

Click on the link: https://prezi.com/9cvgvtua6kz6/project-auld/



2 : Enjoy the slides

1 : Get full screen