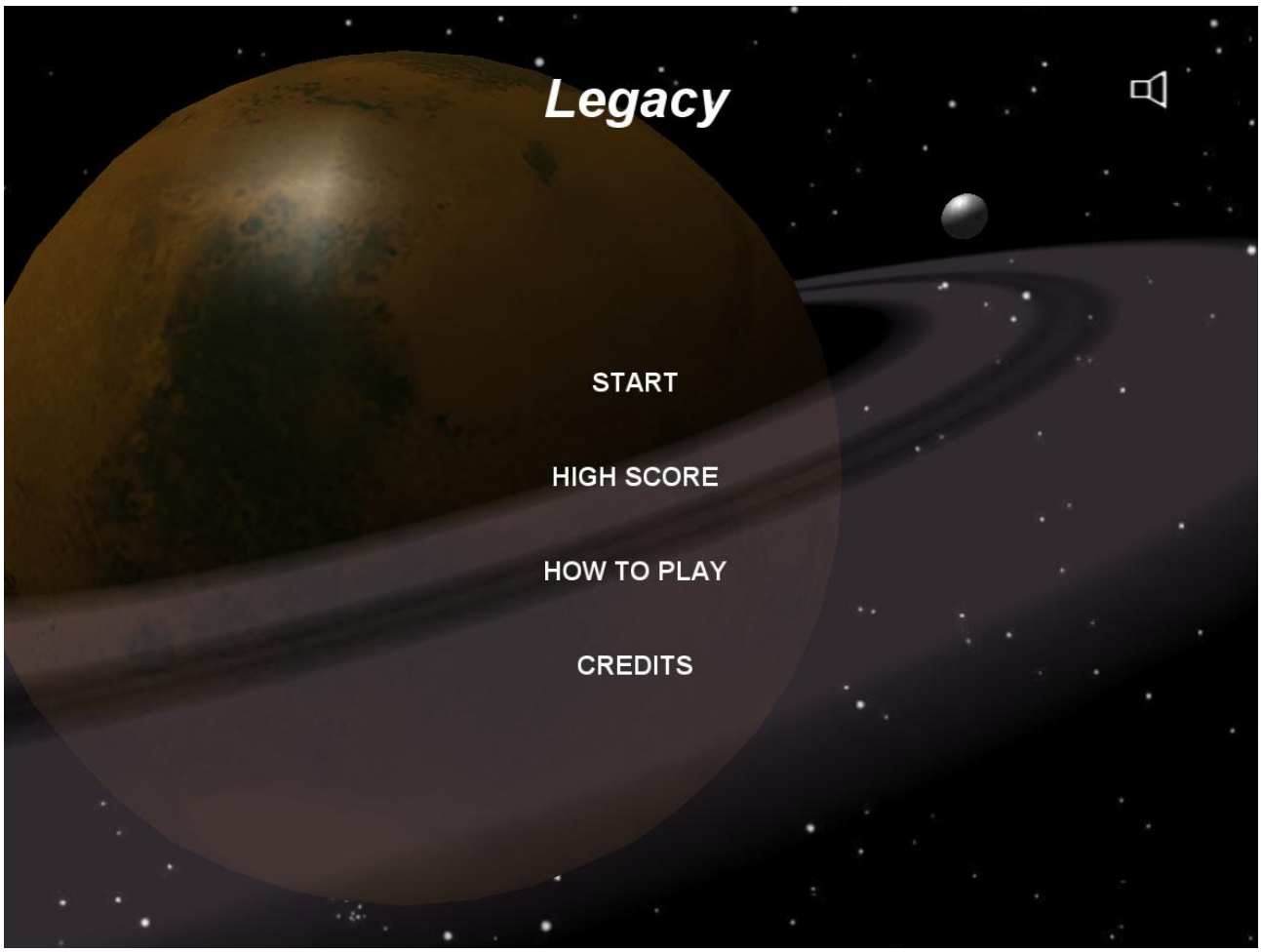
02825 Introduction to Computer Game Prototyping – Legacy

Fall semester 2010



By

Matti Välimäki (s101500)

Jens-Kristian Nielsen (s102601)

Contents

Introduction

By Matti Välimäki

When we started thinking about the course project we had two ideas: a noir-esque platformer and a planetary bombardment game. We ended up choosing the one that was doable with our limited 3d modeling and animation abilities ergo the planet bombardment game, which only required primitive 3d shapes to implement. The idea came originally from Mass Effect 2 in which there is an orbital resource scanning and gathering mini-game. We wanted to do something similar, but it had to be able to stand on its own and not just as a mini-game of something else. We also looked at EVE Online for inspiration as it had recently been upgraded to include planetary interaction, which had some similarities with what we were trying to achieve. The concept we ended up with was almost the reverse if the classic arcade game Missile Command, where the player had to defend against incoming missiles.

The game itself consists of three levels with varying objectives and difficulties but also armaments. You can rotate freely around the planet while trying to evade planetary defenses and destroying the objectives on the planet. We have created several different kinds of enemies to bombard on the surface. Some of these enemies will try to harm you either directly or indirectly and some will be merely targets for wanton destruction. There are three different weapons that vary in damage, blast size and rate of fire. In addition, you have a smaller mouse-controlled defense grid used to ward off planetary defenses.

The report goes over the development of the game from game design to implementation. It includes a detailed description of the rules of the game, as well as an analysis of the playability of game and reflection on tester feedback.

The game can be played at http://tothemathmos.com/legacy .

Detailed Description of the game

By Jens-Kristian Nielsen

This chapter gives a detailed description of the game rules and how Legacy was designed.

Game Design and Rules

In Legacy, the player controls a spaceship in orbit around a planet. The goal is to destroy certain targets on the surface of the planet, and the type of enemy that needs to be destroyed varies from level to level. The player must avoid losing his/her *shield level*, which is reduced when the player is hit by *missiles* or *lasers*. The shield can also be damaged by the players own weapons; each weapon increases the *reactor level*, which when it reaches 100 % starts damaging the shield.

The player must complete the level objectives within two minutes.

Controls

The player may move freely around the planet, with a fixed distance from the surface and with a fixed speed. Movement is controlled with the “WASD-keys”.

The player also controls two weapons: The offensive primary weapon has its crosshair at the center of the screen, and is controlled by moving the spaceship. It is fired by pressing the spacebar. The defensive secondary weapon is controlled using the mouse cursor and is fired using the left mouse button.

Weapons

The primary weapon is used to damage and destroy enemies on the surface of the planet. It has three different settings, *beam*, *shell gun* and *asteroid magnet*, which each have their pros and cons.

The *beam* has a very high firing rate and does very little collateral damage, which makes it good for precision attacks, but bad for clusters of enemies. It only costs very little of the reactor level per shot, but the high firing rate means that it is quite expensive over time.

The *shell gun* has a much slower rate of fire, but covers a larger area and generally better for small clusters of tougher targets. It costs significantly more per individual shot than the beam, but because of the low firing rate, it is not very expensive over time.

Finally, the *asteroid magnet* places a marker on the surface of the planet which draws in an asteroid, which a few seconds later crashes into the planet. This does a large amount of damage over a large area, it however also costs a lot energy, which means that it can almost only be used when the players reactor level is very low.

The secondary weapon is used to shoot down incoming missiles and satellites which shoot lasers at the player. This also uses the reactor, so one has to extra careful if using the primary and secondary weapon at the same time, as the reactor level will rise very quickly.

Enemy types

Enemies in Legacy are indicated by a range of symbols. Each type of enemy has its own unique properties. The different enemies also have different amounts of durability.

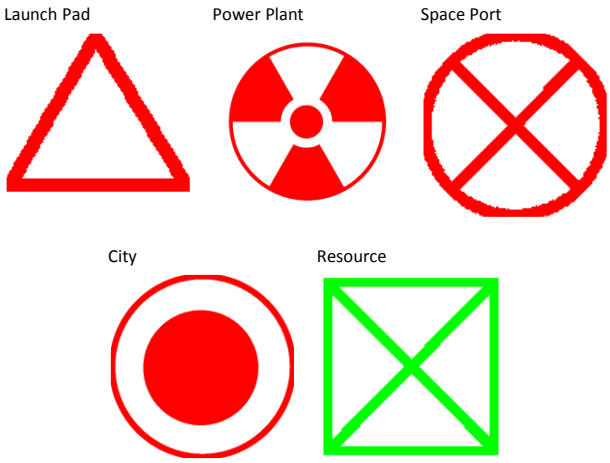
Launch pads shoot *missiles* when the space ship passes over them, and work as the primary obstacle in the game. When the player comes within a certain range of a *launch pad*, it launches an accelerating homing missile at the player. Once a missile is launched, the player has two method of dealing with: Either shoot it down with the secondary weapon, or evade it at the last minute. Fleeing from a missile is not a possibility, as it will eventually catch up because it accelerates. *Launch pads* have low durability.

Power plants are surrounded by 3 *launch pads* and explode when they are destroyed. The explosion causes the surrounding *launch pads* to be destroyed. This mechanic forces the player to choose whether to risk multiple missile launches and destroy the surrounding *launch pads* all at once, or slowly pick off the individual *launch pads*. *Power plants* have medium durability.

Space ports periodically spawn orbiting *satellites*, which shoot *lasers* at the space ship when they are within a certain range. Unlike *missiles lasers* hit every time. The player may choose to avoid *satellites* by staying out of their range, or shoot them down with the secondary weapon. *Space ports* have 2 surrounding enemies and medium durability.

Cities have no special functions, but they have very high durability.

Resources also have no special functions, but unlike the other target, the player should avoid destroying them. *Resources* are surrounded by two *launch pads* and have low durability.



The figure above shows the different symbols for indicating the types of enemies that can be encountered in Legacy.

Levels and Objectives

Legacy features three different levels, each with different objectives. If the level objectives are completed, the player can proceed to the next level, if the player dies, runs out of time, or fails to complete the objective, the game is over and he/she has to start over.

Level 1 is set in orbit around an alien planet with a Saturn-like ring. The settler's on this planet are using valuable resources in power plants, hence the objective is to destroy all the power plants and preserve the resources. This level is relatively made to be relatively easy and features only launch pads, power plants and resources. Only the beam is available in level 1.

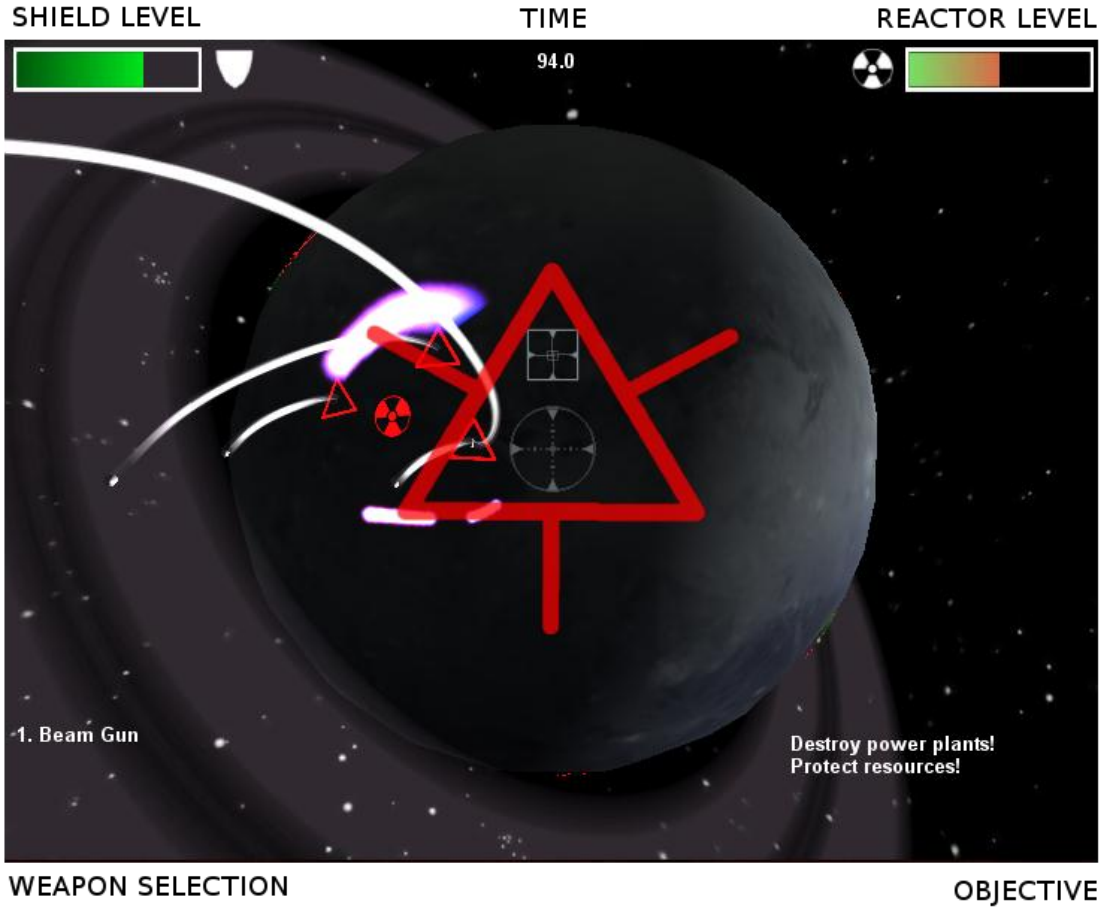
Level 2 is set on a colonized Mars and is significantly more difficult than level one, since satellites are introduced. The objective in this level is to disable the planets air capabilities by destroying all the space ports that are scattered over the planet's surface. The shell gun is introduced in this level.

Level 3 is set on the home planet of the colonists, Earth, and is the final and most difficult level. In this level the objective is to destroy cities. In this level the final weapon, the meteor magnet, is introduced.

Scores

Destroying red enemies is awarded with points depending on enemy types but destroying the green ones e.g. resources give negative points. The total points and amounts of enemies destroyed, is presented when the level is completed. If all three levels are completed, then the player may submit his/her score to the high score list, which shows the top 15 scores.

Heads-up Display (HUD)



The figure above shows the HUD in Legacy.

The heads-up display provides the player with information needed in gameplay. The HUD shows current shield level (life), current reactor level, which weapon is currently selected, level objectives, a clock indicating how much time is left and a pulsating warning symbol in the middle of the screen indicating incoming missiles.

Unimplemented Features

The following is features that were not implemented, because of the limited amount of development time.

Motivational Descriptions and Explanations

The game lacks a short description at the beginning of each level describing the objectives and the motivation for them as well as explaining game mechanics different to the level before, which would possibly make the game easier to learn.

Feedback when shooting

The game also lacks feedback when shooting, because there are no weapon animations, such as beams coming from the ship when firing the beam weapon and projectiles when shooting the shell weapon. The feel of the game could also be improved by including particle effects when enemies are hit, both when they just take damage and when are actually destroyed.

Sound effects

Even though there is no sound in space, the feel of the game would greatly improve from having sound effects. Sound effects provide to the experience with atmosphere, and possibly subtle indications such as warnings.

Landing parties

One possibility for a scenario was that the planet would have to be enslaved. This meant getting rid of defenses but avoiding the cities. One possibility would have been to set friendly landing parties to conquer cities. Enemy landing parties could have also been used to portray a threat to some ancient ruin or resource which had to be dealt with.

Advanced ship damage distribution

The damage could be divided to specific parts of the ship and not just to shields in general. This would allow deleterious effects such as slowing down or losing control of the secondary gun. The effects would probably have to be temporary as they would easily make the game impossible should they be permanent.

Time-based scoring

Time could be taken into consideration when calculating the score. As each level has a set amount of time, it would be possible to award faster reaching of objectives by adding or multiplying the score in some way using the time left.

Craters and other signs of devastation

Signs of destruction on the planet, caused by the player firing on it, would improve the player experience as it would clearly show where he/she has been and that firing upon the planet has an actual effect.

Anticipated Technical Challenges

When we initially set out to make the game, we anticipated that there would be certain technical challenges. Some of the issues we anticipated were player movement behavior, generating levels, movement behavior of missiles, ray-tracing and handling objectives.

The last two of those issues proved to be simple, because there was built-in functionality for ray-tracing and once we had set up the framework for the game, objective management was easy, because all the appropriate information was there at that point. The other anticipated issues did however prove to be true and they are described in more detail throughout the report.

Implementation

By Matti Välimäki

This chapter describes the implementation of Legacy and how it was created using the Unity 3d engine.

General Design and Reasons

We decided to use simple shapes because that would eliminate the need for external models as the Unity engine already has the necessary models. The planets are spheres as are the moons, the enemies and rings on planets are planes and the ship is a cube. The missiles are capsule shapes and the satellites are spheres too. Their colliders were found to be too small to be reasonably easy to hit so they are actually larger than the object itself. This was done on the ship collider as well. The colliders on the enemies are spherical so as to make sure we're able to hit them.

The explosions are created by using special objects in Unity called particle systems. These particle systems were then given spherical colliders so we could easily manage the area the explosion does damage in. When enemy colliders meet with explosion colliders the enemies in question are dealt damage and if the damage lowers the enemies' health below zero they are destroyed and a kill of the specific enemy type is noted.

All rotations (namely of the ship and satellites) are done via an empty game object created as the parent of the object in the middle of the planet. The ship rotation around the planet was originally limited by excluding the poles to maintain a sense of direction e.g. north is up and South is down. However, due to tester feedback, we implemented a rotation system that allows free movement around the planet.

Scripts

AsteroidMagnetScript.cs is attached to an asteroid magnet prefab that is spawned instead of an explosion. It animates the target tag on the planet, spawns an asteroid, drags it to the planet and creates the explosion when the planet and asteroid collide.

BeamOScript.cs has the functionality for the beam that shoots from the satellite to the ship. It creates the actual beam, targets it toward the player and tells the ship to sustain damage.

ButtonScript.cs has the functionality for changing scenes when buttons are pressed. This script is attached to the GUIText objects used as buttons. Also changes the color of the object when the cursor hovers over it.

ChangeHelpScript.cs switches visibility of two GUIText objects describing the game help. This script is attached to the appropriate buttons.

CityScript.cs functionality for the City type enemy object including health, damage and death functions.

ControlManager.cs has the functionality for rotating ship around planet. Attached to the empty game object located at the center of the planet that is the parent of the ship.

EnterHighScore.cs has the functionality for getting a player name for a high score.

HighScoreScript.cs has the functionality for saving and getting high scores from the server to be presented in the HighScoreScene.

HUD.cs has most of the C# functionality for the Heads-up Display including reactor and shield levels and appropriate icons as well as weapon crosshairs.

HUDGunScript.cs has the C# functionality for the parts of the HUD that shows which weapons are available and in use.

JGUIDangerScript.js is JavaScript that creates a flashing icon on the HUD when missiles exist to warn the player of their impending danger. Unfortunately the pulsating effect can only be achieved in JavaScript so we were unable to do the entire project in C#.

JGUIReactorDangerScript.js is JavaScript that crates a flashing icon on the HUD when reactor level is too high. This is achieved in the same way as in the JGUIDangerScript.

JGUIReloadingScript.js is JavaScript that creates a flashing icon on the HUD when main weapon is loading. The script utilizes the same JavaScript functionality as before in the JGUIDangerScript.

JGUIWeaponSwitchingScript.js is JavaScript that creates a flashing icon on the HUD when switching the main weapon. This should never be used at the same time as JGUIReloadingScript as they utilize the same position on the HUD. It also uses the same functionality as the ones above.

LaunchPadScript.cs has the functionality for the Launchpad type enemy object including health, death and missile spawning.

LevelManager.cs manages the level set-up, such as initiating and distributing enemy clusters and. It also checks game over and victory conditions and handles logic regarding shield level, reactor level and the timer.

LifetimeScript.cs kills the object after a specified amount of time.

MainGunTarget.cs ray traces from the center of this object to the center of the main camera and creates the appropriate explosion in the hit location and raises reactor level if possible. This script is attached only to the planet as it is the only thing that is shot with the main gun.

MissileControl.cs controls missile movement including speed and rotation as well as destruction.

MuteButtonScript.cs has the functionality for mute button in start screen including changing the color if the cursor hovers over it.

MuteScript.cs checks if mute is on and sets audio objects on mute if necessary. This is attached to audio objects on each scene.

MyUnitySingleton.cs is used in the menu backgrounds. Checks if there is an object of this type and if there is, destroys all but the oldest. This way we can keep using the same one and also suffer no breaks in music.

PowerPlantScript.cs has the functionality for the Power plant type enemy object including health and explosion in case of death.

ResourceScript.cs has the functionality for the Resource type enemy object including health and death.

SatelliteOrbitScript.cs controls the satellite orbit/rotation around the planet. This is attached to every satellite and makes them orbit the planet in an organized manner.

SatelliteScript.cs has the functionality for the satellites: firing, health etc.

ScoreScript.cs calculates scores according to enemy objects killed and shows them in the score screen.

SecondaryGunTarget.cs ray traces from the center of this object to the position of the cursor on viewport, creates an explosion if object is hit and raises reactor level. This script is attached to missiles and satellites which are the only things that the secondary gun can shoot.

SetCreditsScript.cs is a setter for the GUIText in CreditScene. Setters are necessary because GUITexts that have several lines of text are difficult to set in the Inspector view in Unity.

SetHelpTextScript.cs is the setter for the GUITexts in HelpScene. It sets both HelpText1 and HelpText2 so that ChangeHelpTextScript can change between them.

SetObjectiveTextScript.cs is the setter for the objective GUIText in different levels. It uses the current level to find which objective text should be shown.

SetTimerScript.cs has the functionality for the timer including counting down and ending the game if time runs out.

ShipScript.cs has the damage and game over functionality for the ship.

SpacePortScript.cs has the functionality for the Space port type enemy object including health, death and spawning of the satellites.

Implementation Issues

We had difficulties controlling the flight paths of the missiles as regular rotations seemed to create weird effects like the missile flying off into a seemingly random direction. We believe this is because when the rotation reaches 360 degrees it actually reverts back to zero. This might cause some scripts to function in unexpected ways. We also tried using the physics system by giving the missiles rigidbodies and pushing them in the appropriate direction but that did not give the behavior that we were looking for. Finally we ended up using the Vector3.RotateTowards function to adjust the heading of the missiles towards the ship.

Prototypes

During the development process we tested different prototypes to evaluate possible game mechanics.

Top-down Camera

We tried using the scene view as a secondary camera where you could be able to see around the ship as well and not just from inside it. The purpose of this was to be able to react to missiles coming from outside the view. It was deemed not really helpful as you could not see at what depth the missiles were coming at the ship.

Missile Behavior - Slow to turn, fast to go straight

To avoid a possible exploit in the game (namely continuously strafing to one side to avoid missiles) we tried making the missiles go faster when they're going straight and slow down when they turn. This ended up not solving the problem because the strafing is actually rotation around the planet, following the spherical shape and as such meant that the missiles would be turning constantly.

Ray Particle Effect – For blinding effect

We tried using a particle effect for the beams of the satellites as it had a nice blinding effect. Should this have been used, the damage received from the beam would have been minimal and the main point of the satellites would have been to blind the player. We could not however make the particle effect look good enough so we opted for a different solution.

Analysis

By Jens-Kristian Nielsen

In this chapter we discuss the playability of the game as well as review the feedback given by the tester and how it affected the game design.

Playability and Strategy

Legacy challenges the player with fast paced action and multitasking. The player constantly needs to be aware of incoming missiles, shield level, reactor level and satellites and also to make sure that he/she finishes the level objectives within the time limit. This all adds up to create an atmosphere in which the player is continuously challenged.

We found that it was important to incorporate strategy in Legacy and we tried to execute that by constantly forcing the player to make choices during gameplay. We anticipated that by incorporating decision as a core part of the gameplay; the player can develop strategies to deal with different reoccurring problems. Some of the choices that we tried to make available were:

- Choice of weapon: Beam, shell gun or asteroid magnet.
- Dealing with missiles: Shoot or evade.
- Dealing with satellites: Shoot or flee.
- Destroying power plant clusters: Shoot power plant and destroy surrounding launch pads or pick off launch pads one by one and then safely destroy power plant.
- Dealing with space ports: Destroy early in the game and prevent spawning of further satellites, or deal with space ports and satellites as you go along.

Feedback

At the end of the development period we put the game online in order to get feedback from our friends via email.

Summary of initial round of feedback

- Invert the movement y axis.
- The missiles are too difficult to hit.
- The player should not spawn over enemies.
- It is possible to avoid missiles by constantly moving to one side and picking off targets on the way.
- The weapons lack feedback, because there is no sound, projectiles, rays or animation.
- The game should maybe be playable without a how to play description.
- Level 2 is too difficult.
- It is difficult reacting to missiles coming from the sides, because the player does not know what direction they are coming from or their distance to the player space ship.
- It takes too long to charge weapons.
- Movement around the polar axis feels restrictive.

Analysis of the initial round of feedback

After reviewing the feedback obtained from emails and talking to people, we decided to prioritize the issues that multiple people had noticed first. This meant, first of all, changes to the way movement worked.

Originally, the W and S keys would move the ship longitudinally (north and south) and the A and D keys would move the ship latitudinally (east and west). The idea behind this was that this type of movement would preserve the cameras orientation relative to the planet, so it would be easier to navigate. However, this caused a calculation error when the player moved the ship near the poles, which meant we had to restrict movement to the poles. It also meant that the player would move slower the closer they got to the poles. This was something that the testers responded to, which caused us to change this so that the W and S would instead simply move up and down, and the A and D key would move left and right.

Multiple testers also reported that they would be immediately shot down if the randomly placed enemy clusters would be right under the player spawning position at the start of the game. This was naturally fixed as well.

There also seemed to be a general problem with people discovering an exploit, in which they would simply constantly move to one side, and never get hit by the missiles, because they were slightly slower than the space ship. We could not just make the missiles faster, because then they would be more difficult to shoot down. Instead we made the missiles accelerate, which entirely removed the possibility of fleeing from them, however preserving the possibility of evading by doing a last minute maneuver.

The problem with the missiles being difficult to hit was dealt with by increasing the size of the collider of the missile prefab.

Regarding the lack of feedback on weapons, we found the best solution would be to implement sound and different types of animations and particle effects, however due to lack of development time, this was mostly not implemented. We did however make an animation in which the asteroid would crash into the planet when using the asteroid magnet weapon; originally, the weapon just instantiated an explosion on the surface of the planet.

The number of enemies in level 2 was greatly decreased to make it easier.

Summary of the second round of feedback

- Lack of information regarding cause of death.
- Lack of appropriate symbols in objective text description.
- There no point in the “protect resources” objective.
- It is still difficult reacting to missiles coming from the sides, because the player does not know what direction they are coming from or their distance to the player space ship.
- Satellites are sometimes out of sight, but they can still attack.
- Particle effects sometimes block vision.
- Learning curve is too steep.

We didn’t have any development time after this round of feedback, but we can still analyze it.

We believe that a possible solution to the steep learning curve would be to have a bit of explanatory text at the beginning of each level, to explain the concepts as they come along, as mentioned in the unimplemented features section. The objectives could also be simplified so that there would only be one type of target that would be needed to be destroyed in order to pass the level, instead of varying it every

level. A line of additional text when dying, describing the cause and how to avoid it in the future would perhaps also help.

The concept of the resources was to increase this value of the beam weapon, by having a greater need of precision. However, currently the player can simply avoid them completely. This could be solved by adding an additional objective in which the player has to clear the enemies surrounding the resources.

Multiple testers seemed to have problems with missiles coming from the sides, where the player could not see them. This could perhaps be solved by using a third person perspective for the space ship instead of the current first person one. This would allow the player to see the missiles and perform a proper evasion. This would also give a wider viewing angle, revealing satellites near the edge of the screen.

Conclusion

By Matti Välimäki and Jens-Kristian Nielsen

Initially the concept of the game was very simple. The player would be able to bombard a planet from orbit. As time passed we discovered the need of a more complicated and challenging mechanic, so we decided to introduce tactics or strategy to the game. For the player to be able to develop strategy, he/she needs to be able to make decisions, hence multiple types of new weapon, enemies, formations and mechanics were introduced that each had their strengths and weaknesses.

Were we to start this project now based on what we have learned, we would use a different enemy retaliation method than the missiles. Missiles have proved to be very difficult to balance, because of the complexity of their movement pattern. We have also never found a good solution for handling missiles coming from the sides of the space ship, where the player cannot see them. Another thing that we likely would do would be to implement sound throughout the process, instead of waiting to the end; that way we could use sound in the gameplay as a substitute to some of the graphical indicators that are currently used.

In hindsight the originality of the concept was a double-edged sword, because we had to develop the game mechanics from scratch opposed to an already existing genre, where a lot of the basics are already established. Choosing a concept of an existing genre would have allowed us to focus more on technical aspects, but likely also less on game design.

If we were to continue the development of this project we would most likely first try to implement the unimplemented features: the motivational backstory, craters, sound effects, etc. We would then work with our testers to try and balance the game as well as possible as well as making it truly fun and engaging. After all this is done the possibility of new levels is definite.

Resources and Inspirations

By Matti Välimäki

Because we did not make a game of an already existing genre or mechanic, we did not have references as such, but rather inspirations.

The concept was derived from the resource gathering mini-game from the block buster Mass Effect 2 in which the player scans planets with a cursor to detect veins of resources and then shoots probes to obtain them.

Some inspiration was also spawned by the Eve Online Tyrannis update which allowed players to build and war on planets in a previously space bound game.

Helping us along the way was the Unity community as we searched and used the Unity Answers forum to get ahead.

Links

Mass Effect 2, http://masseffect.bioware.com/ (for the mining mini-game see http://sarcasticgamer.com/wp/wp-content/uploads/2010/01/mining-pic-mass-effect.jpg)

Eve Online - Tyrannis, http://www.eveonline.com/tyrannis/

UnityAnswers, http://answers.unity3d.com/

Appendix: Feedback Emails

First round of feedback

From: Frederik Højlund

Date: 21. nov. 2010 23.56

Hej hej

Byt om på W og S. Det er også lidt mærkeligt at man bevæger sig hurtigt rundt om ækvator men langsomt ved polerne.

Er det mig der er dårlig eller er missilerne bare rigtig svære at ramme?. Måske der ikke skulle komme så mange i starten og så flere senere. Det er vel oldschool arkade stilen.

Efter et par spil finder man ud af at det nemmeste bare er at hele tiden køre til siden hvorved man undgår alle missiler og så bare køre op og nede og skyde ting med våben 2.

Det lykkedes mig at score 2000point i første level. Ved ikke om det er godt.

Jeg synes ideen er fed! Man føler sig sikker når man kan "nuke the site from orbit".

-Frederik

From: Michael Josefsen

Date: 26. nov. 2010 21.41

Hey, Michael fra DADIU produktionen her!

Prøvede lige det spil du har linket til på facebook.
Looking good so far!

Her er en smule input:

Det første der skete var at jeg trykkede how to play, og det crashede
(dvs holdt op med at reagere på museklik, tilsyneladende)

ellers ingen deciderede bugs da jeg gennemførte alle 4-5 baner

jeg kunne se at den sagde "reloading" efter hver gang jeg trykkede på space, men man
kunne "rapid-fire" uden begrænsninger ved at tappe løs på space med to fingre,
så en fail-proof strategi jeg hurtigt fandt var at glide til siden med turretet indtil jeg var over et mål, så hamre løs på space med to fingre, og flyve videre. på den måde kunne man smadre hvert mål før missilerne nåede at

komme ret langt op.

ellers ser det godt ud so far. planeten og baggrunden plus musikken virker fedt sammen,
og konceptet er simpelt, men godt.

måske nogle forskellige typer af modstand, så svaret mod alt ikke bare er at bevæge sig til siden konstant (det virkede også mod lasers)

var også en smule forvirret første gang jeg prøvede at skyde. ikke kun fordi der ikke er lyd, men også fordi der ikke er noget projektil eller en slags mundingsild. troede lige først at den pink laserkugle på jordoverfladen var en slags powerup eller noget :P
ellers oplevede jeg ikke nogen problemer. kan godt se det ikke er færdigt endnu.

det var alt jeg havde. Keep up the good work :D

From: Christopher Aaby

Date: 27. nov. 2010 00.20

Overall - jeg kunne ikke regne ud hvad jeg skulle uden at læse "how to play", så det er ikke det mest intuitive... når man først ved at man skal skyde ting og man har lært kontrollerne, er det dog rimelig simpelt... green = good, red = bad, det er til at forstå.

Jeg synes det fedeste i spillet er perspektiv-følelsen når missilerne eller strålerne kommer imod en, det virker rigtig godt.. desværre er der også nogle problemer med missilerne, se herunder.

Jeg savner allermest noget mere pow-wow på spillerens egne våben, og mere følelse af at jeg kan stille noget op mod fjenden. Jeg spillede spillet 10-ish gange, og hver gang nåede jeg til level 2 for at blive skudt ned af de beams der... det føles bare lidt som om der ikke er noget at stille op. Jeg er klar over at de kan skydes ned, det er bare for svært, eller de skader for meget, eller et eller andet. Og ens egne våben... ideen med sådan en ion-cannon agtig ting er rigtig fed, men der er ingen lyd på, og ikke ret meget WOW over den. Det ville hjælpe meget.

Detaljeret rapport herunder.

Hilsen Christopher

UI

Der mangler noget feedback når man klikker på knapper. Bare en mouse-over og en mouse-down state, så er den godt på vej.

Click for more knappen på første skærm af "how to play" virker kun på "more" området

Teksten spilder ind over "back" knappen på samme skærm

Teksten er centreret til venstre på 2. how to play skærm... check skærmen for at se hvad der er galt.

Der mangler noget feedback når man er game over eller har klaret en bane. Hver gang er jeg lidt nervøs for at spillet er gået i stå eller at mine input ikke virker.

Tallene på score skærmen står ikke over hinanden.

Det ville være godt med noget feedback på om man rent faktisk rammer målene eller ej. De kunne f.eks. skifte farve jo mere skade de tager, eller der kunne bare være en anden farve "beam" når man rammer noget / små eksplosioner / noget. Det ville bestemt også føje til awesome-følelsen når man skyder.

Der er helt klart også brug for noget bedre feedback når man skyder med musekanonen.

Highscore skærmen er fucked.

Gameplay

Det er frustrerende at man ikke kan se missilerne når de kommer mod en fra siden. Enten burde de ikke kunne ramme en fra siden, eller også burde man have en advarsel om at de kommer... eller også burde man kunne skyde dem på en eller anden måde. Måske kan man kaste flak eller noget.Jeg forstår ikke hvordan jeg skal undgå at blive ramt af de grønne beams i level 2... kan man det? Hvis man kan er det ikke åbenlyst hvordan.

Det tager laaaang tid for at skifte våben... det lægger ikke ligefrem op til at man skal skifte frem og tilbage. Hvorfor tager det så lang tid? Er det ikke sjovere hvis man skal skifte våben ofte, og tilpasse ens valg af våben til fjenderne netop nu? (I stedet for f.eks. bare at skifte våben een gang per bane, bum)

En lille smule auto-aim ville være en god ting, så sigtet f.eks. snapper til et mål når man stopper med at bevæge det og målet er tilstrækkelig tæt på. Det kan godt være lidt svært at placere det liiige rigtigt... minor issue.

Det ville være et godt lille gameplaymæssigt greb hvis de forskellige typer mål (missil launchers, reactors, ...) tog ekstra skade af visse våben, så der var mening i at skifte våben, i stedet for at et af våbnene "bare er bedst" eller at de kun findes for at give spilleren et valg af præference.

Generelt

Hvorfor kan man ikke rulle over toppen af planeterne? Det virker mærkeligt, og der er ikke nogen visuel indikation af det. Man burde kunne rulle rundt på planeterne frit, uanset planetens orientering.

Det ville være rart at have noget radar / GTA-pil agtig navigationshjælp til hvor ens mål er henne, incoming missiler (der ikke kan ses når de kommer skråt fra), etc.

Der kunne godt være lyd på nogle flere ting. Ens eget våben f.eks... sådan en beam cannon ting kunne føles rigtig fed at skyde af, hvis bare der var noget mere lir, f.eks. en god BZZZZZ lyd. Incoming missiler kunne også give nogle panik-bip der både tjener til at indikere hvor tæt man er på at blive ramt, og give noget mere panik-stemning.

Lyd på ens power-meter oppe til højre ville også være godt... man kigger ikke derop så naturligt, så lyd-feedback på den ville være godt.

Det kunne også være rimelig sweet hvis ens eget våben satte nogle mærker på jorden, det ville føles en del mere mægtigt. Og så kan man tegne smileyfaces på planeter!

Man kan bruge sine våben og sådan noget mens der står "game over" eller "victory"... det burde man ikke. Virker lidt mærkeligt.

Second round of feedback

From: Jussi Perttola

Date. 1. dec. 2010 12.39

Hi.
Let's just get to the point. The game is just horrible! :D

1) The game is really really hard and there are many reasons for this.

- In the first stage I died a dozen of times before I realized it's because I had shot at the wrong kind of symbols (resources). I recommend fixing this by adding the appropriate symbols next to the mission objectives so you wouldn't have to memorize the whole how-to-play part. Also, if you destroy resources, the game over screen should point out the reason of game over. I always thought it was a random missile I didn't see. You should also consider removing the objective "protect resources" as there is no way to actually protect them, as no one is actually attacking them. Seems pointless. Or maybe change it to "don't destroy resources"?

- I often got to a situation where I got hit by something I didn't see. Happened more often in the second stage where the satellites shoot me as soon as they get line of sight. Because of the wide screen you can more easily see enemies coming from the sides than from above or below. Yes, I've yet to make it past the second stage.

- You really shouldn't have to start the game in a position where you get shot at even before you move!

- Maybe a sound indicating when you are hit?

- When you blow up a station or shoot down missiles there is a nice explosion effect. However this prevents the player from seeing other missiles fired from the same location! Makes the game really really hard. Maybe decrease the rate of fire on missile platforms?

- The shell gun sucks. The beam gun can destroy bases much faster. Maybe increase its power?

2) Other issues I can think of

- The game over screen. I get it all the time. First thing is that the game over text is just ugly. Please don't

just increase the font size as it gets blurry. The same goes for the "victory" text (which I didn't see that often though). Another problem with game overing is the delay. You should be able to click the screen to start over again. Now you have to feel the shame of losing for like five seconds!

- How to play now seems to work ok. However the text in the bottom right corner says "click for more". When I click the screen nothing happens. So it means I have to click the "click for more" text which it didn't actually say. You should be able to click forward anywhere. Otherwise the caption should be "Click here for more" or just "more...". Also the how to play layout is terrible. The last line is on the same level as the buttons which is confusing. Also the symbols of the stations should be positioned near its description. Is there a reason for the "back to menu" button in how to play? Shouldn't it be simpler if it would just go back to the many after seeing both pages?

- What happens when I go over the reactor level?

- Do you really need "how to play" page? The main menu could simply show the keys used and give info like "destroy *symbol*, "protect *symbol*", "beware *symbol*". Simple is beautiful. You don't have to tell the player that the game ends after life reaches zero! I think that all the symbols should have just one meaning. What I mean is that the player should always try to destroy the same kind of symbols in every stage. For example, every mission is to destroy cities. However you obviously benefit from destroying launch pads and spaceports and if the former are located around a power plant, you also benefit from destroying that. Simple?

- You should change the symbols to 3d-graphics or more graphical symbols. The only one I really recognize is the power plant!

- It would be cool if the power plant exploded in a mushroom cloud. I see this compulsory.

3) Things I like

- You made it possible to go round the planet in every direction. That was a must!

- The idea of two weapons is nice.

That's about it. I hope my extra-critical designed approach helps! :)

--
Jussi Perttola

From: Stine Rosenbeck
Date: 2. dec. 2010 17.45

From a casual gamer's perspective I find the learning curve too steep. Tried it out three times and then gave up. Had trouble figuring out how far the missiles would chase me too. When I died, I didn't know why, which was rather demotivating; was is the missiles or the radiation thingy? Also, I would prefer the fire

button to be on the mouse instead of the spacebar. Maybe the defensive thingy could be the right mousebutton? The music kicks ass, though.