**Astro Crisis!**

**An OpenGL project for Comp315-Advanced programming**



**Project Proposal Documentation**

**Authors:**

Kreason Naidoo, Muhammad Bassa, Shaherin Dehaloo, Liam Barnard

2015 Comp315 Advanced Programming: project proposal

**1 - Project theme:**

A long way away in a far ago time, the Home Planet was a happy and nice place. But then, one day, deep space scans reveal a large amount of asteroids heading straight for Home Planet! The asteroids have been sent by the evil being known as Story Plot... How evil! The Home Planet sends their champion, the mighty and good-looking "Player 1", to go and defeat the evil asteroid army, thus saving Home Planet from certain annihilation! How brave!

Astro crisis is a space-based defense shooter. The player takes control of a spaceship and defends their objective, the "Home Planet" from being damaged, by shooting down "asteroids" that are on a collision course.

The Player moves in the XY plane with “Home Planet” behind them. The Z value is kept constant as the game is about defending “Home Planet”. The mouse is used for pointing the ship but is limited such that the player is required to move around to hit all the targets.

The asteroids approach the base in waves, starting off as a few smaller ones, and eventually become greater and greater in size and number. The player can shoot apart asteroids to prevent them from damaging their home base. A new wave begins after a certain amount of time, or once all asteroids have been destroyed

The asteroids come in one of three sizes: small, medium, and large. Large asteroids are the most dangerous and require a few shots to break them apart. Large asteroids break into two medium asteroids. Medium asteroids are less resilient than larger ones, and upon their "death", they break into two small asteroids. The small asteroids are destroyed in a single shot.

The player wins if they make it past the twentieth wave, upon which a victory message will play and ask them if they would like to exit or play again.

Should the base be destroyed, or the player themselves be destroyed by colliding with too many objects, they lose the game. A defeat message will play, and prompt them to either play again or exit the application.

**2 - The Player**

The actions listed here are the most critical in order to get the standard functionality of the game up and running. Assume all movement is relative to viewing direction and orientation.

|  |  |  |
| --- | --- | --- |
| **2.1) LIST OF BASIC ACTIONS** | | |
| **Key** | **Effect** | **Description** |
| **a** | Strafe left | Translates player left as long as the button is held |
| **d** | Strafe right | Translates player right as long as the button is held |
| **w** | Strafe up | Translates player up as long as the button is held |
| **s** | Strafe down | Translates player down as long as the button is held |
| **esc** | Pause menu | Pauses the game and brings up a menu |

|  |  |  |
| --- | --- | --- |
| **2.2) LIST OF ADVANCED ACTIONS** | | |
| **Key** | **Effect** | **Description** |
| **LMB** | Shoot | Creates a small projectile at the front of the player object that moves forward at great speed. If it enters the same area as another object, it is destroyed and a set amount of damage dealt to the object with which it collided |

Mouse movement would affect the player’s viewing orientation (LookAt), much like the viewing mode in any first-person game. However the mouse curser (the sight) will not be able to access the entire screen but rather only a limited space around the ship, forcing the player to move the ship in order to hit all the targets.

Additionally, if the player comes into contact with another asteroid or the base, they will be "bounced" back a short distance, I.e.: no two objects must occupy the same location in space. The direction of the bounce will be a logical product of the two colliding objects' vectors at the time of impact. Damage will be dealt to the player and the asteroid will take damage / the base will take damage.

If an asteroid hits the base it will break and the base will take damage.

**3 - NPC's and/or Dynamic Aspects of the World**

3.1) Basic (single NPC) State and Behaviour

HOME PLANET (PRIMARY OBJECTIVE, FRIENDLY NPC)

The "Home Planet" is a large sphere, skinned to look like a planet. Its’ defense serves as the primary objective of the game, and is thus unique. It remains stationary in terms of position, but rotates slowly on its’ axis in order to simulate a basic planetary rotation. Initially, the NPC has a "health" value, set so some value representing the number of "hits" it can take. Should an object collide with the Home Planet, this value will be decrease. If the value reaches 0, the game is lost.

3.2) Extensions to NPCs

ASTEROIDS (TRIPLE 'STATE', HOSTILE NPCs)

The asteroids will be procedurally generated, randomly, in various size categories. The damage an asteroid can do is dependent on its size at the moment of impact. They spawn at the edges of the world, moving on a general collision course with the Home Planet NPC while rotating on its’ axis. Their speed of approach remains constant, although each may differ in speed, from one another, to a degree – within specific parameters. Each instance of an asteroid will have variable properties such as: *x-position, y-position, z-position, angular velocity (including direction) and velocity.* All of which will be generated by randomization. Their fixed properties will be fields such as size *category (s/m/l),* and *“health” (number of hits required to transition it to a lower size category)* will be affected by gameplay.

**4 - World Environment**

4.1) Basic World Environment

The basic world environment is not complex – a single light source represents the solar systems’ primary star – the equivalent of our sun. In addition, the "Home Planet" NPC, positioned at the centre, would serve as a good point of reference to ensure the movement keys are working properly. Player 1’s movement would be restricted to the planet behind him and the asteroids in front of him, allowing gamers to not lose focus of the planet they must defend.

Asteroids will spawn from the far z plane and move towards the camera / player 1, with their final destination being the “Home Planet”

4.2) Additional lighting, textures

The “world” will have a backdrop, achieved by a star texture being painted onto the inside of a large sphere or cube (the outer universe) that would serve as the worlds’ boundary. The light source would need to be inside the sphere / cube but not so to distract from gameplay.

**5 - Possible Extensions**

Increased Graphics:

* Flash effects (muzzle flare) around players weapon firing.
* A jet effect on the player’s ship’s thrusters
* Asteroid trails that disappear over time
* Dynamic texture changes to the NPCs
* Asteroid explosions on impact

Multiplayer

* To replace asteroid defense with an active player 2, allowing a “capture the flag” style gameplay dynamic.

Additional Weapons

* Secondary weapons and power-ups for the player, swappable weapons as well as limited ammo for the additional weapons.

Asteroid Movement

* Simulate (the inverse square law of) gravity when asteroids are within range of the planet, visible gravitational field using translucent shapes.

Healing

* Introduce healing of ship / planet health based on “life packs”, game progression reward or time.

Sound effects

* Adding of sound effects for asteroid impact, ship movement and weapon firing (“pew pew”)

Additional Player Movement

* See below key movements:

|  |  |  |
| --- | --- | --- |
| **<-** | Roll left | Rotates player view and orientation counter-clockwise |
| **->** | Roll right | Rotates player view and orientation clockwise |
| **RMB** | Shoot | Fire with Secondary weapon |
| **c** | Switch | Cycle secondary weapon |