**Mega G(alaga)**

**TODO**

* Go through GDD to finalize changes.

**TODO [DONE]**

* Alienmanager removal of aliens .destroy() bug
* Firebombrate does nothing for the bomb; the rate is only good for when the key is held down (gives the time in CE engine); I most likely need to use real time for the firebombrate. Check CE for real time; like deltaTime or getTimer
* Need to tie in the \_ce.input.addKeyAction for the options menu at least once so it can open
* Alien bullets can be destroyed by hero bomb.
* Continue with refining options manager keys.
* For key customization; possibly make the static constants to just public var that can be changed and accessed easily through hero from game. Probably can keep static but const needs to go since I cant modify it.
* Create UI bg in flash cs 6 or photoshop.
* Search for embed swf UI for games. Flash etc.
* Add points to each alien.
* Hard alien fire projectile (triple)
* Now I have options key pausing the game and a separate pause key; bad; what can I do to fix this.
  + Fix: one turns off the other, next time don’t have this.
* Volume slider; master volume
* Sounds
* Check Galaga .setUpStarling(true), see if the enemies are counted as being drawn by Starling.
  + According to the draw call debug; they are not being drawn to the GPU.
* Find out why objects are not being drawn to GPU or maybe I’m mistaken on how the Starling draw call display works. Best way is to create a mock test; testing CitrusSprite with Starling has shown draw calls appear. Test CitrusEngingStarling with Starling Sprites separately.
  + Mock test shows that all Starling Sprites are being drawn to the GPU.
  + From a StarlingState using this.addChild(new Enemy()); it is being drawn to the GPU.
  + Do a test with just starling and .showStats enabled.
  + Why in Galaga project are Starling Sprites not being shown in stats as drawn triangles? Outside this project anything drawn using Starling Sprites is drawn as triangle.
  + Holy moly; thought of this completely wrong. The reason there are increased draw calls is because I am using a new texture for each object. If I reuse the texture then the draw calls will be reduced to a single call for that texture. See tips.docx.

**TODO [SCRAPPED]**

* Research workable sizes for the games width and height compared to the enemy and player size. If the player is 32 x 32 and game size is 640 x 480 is that a good ratio? Maybe 16 by 16.
* Calculate the proper speed that the enemies should move down the screen. Should it be a set integer value? Or it doesn’t matter?
* Try keypoll starling first and others if applicable.
* See old file about keyboard settings to change keycode to string type of invalid keys (game was splatterhouse)
* Add score class and etc.
  + Added to HUD.

**Programming**

AS3 – Flixel/FlashPunk/Starling

* Advantages
  + AS3
  + Starling is fast (stage3d for 2d) and well documented
  + All 3 engines are robust and tested
* Disadvantages
  + Not cross platform (?) according to Starling it is cross platform

Haxe – HaxeFlixel/HaxePunk/Starling

* Advantages
  + IDE
  + Cross Platform
  + Haxe similar to AS3
* Disadvantages
  + Have to learn Haxe (disadvantage because it will take time)

Javascript – GameEngine

* Advantages
  + More javascript
* Disadvantages
  + No IDE
  + Maybe not cross platform (directly)

Unity – C#

* Advantages
  + Programming IDE
  + C#
* Disadvantages
  + 2D isn’t native; must use a secondary package library for support
  + Cross platform costs a lot of money

**Decision**

I will be starting off with AS3 programming language and Starling to get familiar with Starling API again. I will use the Flash Builder IDE.

**Gameplay**

A simple game based on Galaga. The player is tiny ship that goes up against enemy ships. Every 20 seconds the game gets harder. More and more enemy ships appear. Harder more intelligent ships appear. Movement is not snapped like in the old Galaga, for example, we will not be moving in grid-like format. Movement will be fluid. There will be a score system. Score will be saved locally.

Although Galaga featured a patterned enemy spawning gameplay this will not be the case here. It will be tuned more to a top-down shooter where enemies come in and then leave.

**Controls**

The player can move freely in all direction ~~but is limited to the bottom half of the screen~~. (bad idea and unnecessary).

The ~~mouse button~~ space bar (ctrl for now, my spacebar is screwed) is used to fire the gun.

The movement controls are arrows keys. Escape to pause.

Next GDD

* Introduction
  + Short game intro; describe gameplay in a few sentences and what the point of the game is. What needs to be accomplished and why someone will play this game.
* Game Mechanics
  + Controls and what they do in detail
  + Breakdown each element here and describe mechanics
  + Player
    - Movement
  + Enemies
  + Enemies AI
* NPC’s (may not be needed if its under mechanics)
* Story
* Gameflow / Game progression
* Maps
* Menus

**Things need fixing.**

**Updating**

Each alien to ~~update itself and~~ have its ~~own pool~~. Aliens shouldn’t update their own movement because there may be a lot of aliens on screen and there will be too many updates running. Cant have its own pool because they are all separate classes; cant keep track of multiple pools like that, AlienManager seems to be the way to go.

Thought about this. Could have a for loop iterate through each Alien and call the .update method. Need to update the ones in the pool, the ones already active can be left alone. .update wont work since the Alien has no idea what their pool is and even if they did they would need there exact location in the pool, now seems like too much work and resources to use.

**Spawn Rates etc**

I may have to make enemy spawning not random --- that leads to unpredictable gameplay for the next iteration of the game. Need to think up a wave system.

**Entities**

* **Screens**
  + Width
    - 640
  + Height
    - 480
  + Start Screen
    - Press start and game begins.
  + HUD
    - There are HUD parts visible from the start screen such as the volume changer.
    - Display quick volume change.
      * Visible from start screen.
      * A slider is preferable.
    - Display lives (3)
      * Visible only during game play and pause.
  + Pause screen
    - Displays the word “PAUSE”
  + Controls
    - Allow user to change keys.
    - Defaults:
    - WASD
    - W – forward
    - S - back
    - A – left
    - D – right
    - H – Fire gun
    - J – Bomb
      * Fire only 1 bomb at a time.
      * Bomb should do a full animation before being allowed to shoot another.
      * No travel time. Bomb spawns around the player.
    - Esc – pause
  + Background
    - Parallax background of space, stars and debris.
    - Slight increase of scroll speed as game gets harder. May need a maximum cap on speed just in case players are really good and get really far.
* **Player**
  + 32 by 32
  + Ship
  + Minus one life when dead.
  + Comes back to life and is invincible for a short period of time (1.5 – 2 seconds).
* **Enemy**
  + There are a wide variety of enemies and as the game progresses the enemies difficulty goes up and new enemies appear.
  + Enemies travel down the screen slowly and then as the game progresses they get faster. Enemies spawn with a certain speed so they don’t get faster themselves but are spawned with their downward speed.
  + Each enemy will take only 1 hit to destroy; the challenge is the number of enemies spawning and the AI of the enemy.
  + Basic
    - 32 by 32
    - Brainless, moves downward at a set rate.
    - Has no attack
    - Base score is 10 points.
  + Medium
    - 32 by 32
    - Moves downward
    - Has a single attack; 1 projectile.
    - Projectile will fire to players location of when it was fired (not homing)
    - Base score is 25 points.
  + Hard
    - 64 by 64
    - Moves downward.
    - Can side step.
    - ~~Can fire as many times as necessary. Random fire rate.~~
    - Fire 3 projectiles towards the hero. Triangular pattern. Fires once.
    - Base score is 60 points.
  + Enemies will stop firing after crossing a certain boundary to prevent problems.
  + Other enemies can range from 32 by 32 to 64 by 64
* **Explosion**
  + Scales to the size of the enemy.
  + The explosion may handle its own animation and sound?
  + Has an animation.
* **Sounds**
  + Player
    - ~~Hit by enemy bullet~~
    - ~~Player shoots regular gun~~
    - ~~Player shoots bomb~~
  + Enemy
    - Each enemy will have a unique explosion sound when they are hit by a regular bullet.
    - Bomb explosion hits will all be the same.
    - When appears on screen (?)
  + Projectiles
    - ~~Regular bullet hits an enemy~~
    - Bomb hits an enemy
  + UI
    - ~~Press escape (pause)~~
    - ~~Press escape (unpause)~~

**Images**

* ~~Player ship~~
* ~~Enemy ships~~
  + ~~1~~
  + ~~2~~
  + ~~3~~
* ~~Asteroids~~
* ~~Start image~~
* ~~Gameover image~~
* Hud lives icon
* ~~Paused image~~
* ~~Background~~
* ~~Player Projectile~~
* ~~Player bomb~~
* Explosion bomb
* ~~Explosion player~~
* ~~Explosion enemy~~
  + Re-use player explosion
* ~~Enemy projectile~~
  + ~~1~~
  + ~~2~~
  + ~~3~~

**Interesting Things**

This is a compilation of noteworthy things I discovered and may want to remember.

* Filter Forge has a spaceships renderer. I can render mini spaceships then trace over them with pixels.
* SpriteCraft is a minecraft pixel maker but it can be used for anything really.
* Cgmusic software can automatically make music.

**Advanced Ideas**

These ideas are above the scope of the basic gameplay I want to complete before moving onto more complicated ideas that may take some time or be scrapped altogether.

* Start screen touched up.
* Game over screen touched up.
* Good transitions between screens.
* Multiplier for scores.
* **Maps**
  + Moving through destructible debris.
  + Indestructible debris.
  + Enclosed areas as if you were inside a building or large abandoned spaceship (gradius levels)
* **Additional enemies**
  + Meteors
  + Each enemy type has a specific attack pattern and movement.
  + Could introduce one enemy called “bionic android” that has intelligent movement (unpredictable). (black car from ridge racer skill)
* An innate shield ability that recharges over-time or can be recharged.
  + This is a buffer to the player’s health and gets taken down first.
* Special power-ups, multiplier bonuses etc flow down the screen for the player to pick up. These should be temporary.
* Secret areas that are accessed while playing the game. For example in star fox there were secret levels if you flew into a specific ring or flew through rings in a certain order or shoot a meteor with a face on it and destroyed it and slammed into it.
* The background will transition to another background as the game gets harder.
  + Problems: Since the game is “infinite” the background will have to stop transitioning. Or better yet the background has no bearing on the difficulty of the game. The background image will transition as a queue that the game is getting harder?
* Transitioned backgrounds.
* The player is able to choose their starting ship.
* **Stats System**
  + The player comes with default stats can later be upgraded or changed in various ways.
* **Upgrades**
  + The ability to upgrade your ship.
  + Upgrades can be similar to the old SNES side-shooter plane game that had those “options” around the plane that are like AI. (Gradius)
  + Upgrades can be visual upgrades. The exchanging of parts of the ship. Upgrading a gun to a missile shooter.
  + Upgrades can be similar to the game Armored Core. The user can build their ship from various parts.
    - Body
    - Weapons
    - Boosters
    - Wings
    - Fuel
  + There can be various different parts to mix and choose that boost overall stats.
* **In-Game Store**
  + An in-game store to buy power-ups.
    - Move speed
    - Lives
    - Projectile speed
    - Projectile damage
    - Shield packs
* **Animations**
  + Enemies
    - Particle trails
  + Player
    - Fire trails
  + Background
    - Animated
    - Flickering stars/lights
    - Scrolling stars/asteroids
* **Weapon System** 
  + Similar to Gradius that only allowed choices of weapons, more can unlock as time goes on. Weapon switching is real time but there is a cool down every time you switch weapons then the player is unable to switch.
  + Weapon Charging
    - Hold down the fire button and your weapon will charge for a stronger attack.
  + Weapon switching
    - Switching weapons in-game (on the fly). Cooldown between switching weapons.
  + Weapon garage
    - You go to a special garage to equip your ship with weapons, armors, power-ups etc.
    - You can place the weapons anywhere on your ship. The weapons will fire from the place you put them.
  + Some weapons have automatic targeting and will rotate and fire at targets.
  + Cooldown of hero weapons get shorter the more you kill.
* **Collect**
  + When you kill enemies they will leave behind items to collect that can do various things.
    - Increase score
    - Power-ups
    - Think of medals from those 1940’s plane games
  + Random (or not random but pre-set) power-ups and coins scattered about to collect.
* **Advanced Level Select**
  + When you start the game you go to a level select. The level select is like a galaxy screen in Star Fox. Each level will be different. The basic levels are like Galaga.
  + Some levels will be free roam type levels.
  + There will be secret levels.
  + You can choose multiple paths.
  + This will have to break the design of an endless level game.
* **Game Modes**
  + Endless
    - Features top scores for endless mode which is separate from other game modes.
  + Story mode
  + Perma-Death (possible branch off game idea here)
    - You have 1 life.
    - You go through different levels. Killing enemies, waves, collecting stuff, possibly even doing quests.
    - Each play through progresses your character some. You can see all your stats.
    - Can choose a character type, kind of like an rpg.
* **Achievements**
  + Implement if applicable; this is going to be based on the website its uploaded to. Each site has their own API for achievements.
  + Achievement types
    - Endless mode
    - Story mode
    - Enemies
      * How many killed by type
      * How many killed by projectile type
    - Player
      * Time played
      * Deaths
      * Pixels moved
      * Shots fired per projectile type
* **Particle Effects**
  + Explosions when enemies killed
  + Explosions when player dies
  + Fire trails
  + Bullet trails
  + Movement trails, fade, blur
  + Effects when picking up power-ups, coins

Sources

* Font
  + http://www.dafont.com/04b-30.font