

8 (crazy) weeks plan

Week #1: Easy stuff first

* ~~Feature: Turrets~~
  + ~~Create the new turrets as described in the game design document;~~
  + ~~Integrate the new turrets in the game;~~
  + ~~Playtest the new turrets;~~
  + ~~Produce the standard assets for the new turrets;~~
* ~~Feature: Power-ups~~
  + ~~Create the new power-ups as described in the game design document;~~
  + ~~Integrate the new power-ups in the game;~~
  + ~~Playtest the new power-ups;~~
  + ~~Produce the standard assets for the new power-ups;~~
* ~~Feature: Enemies~~
  + ~~Create the new enemies as described in the game design document;~~
  + ~~Integrate the new enemies in the game;~~
  + ~~Playtest the new enemies;~~
  + ~~Produce the assets for the new enemies;~~
  + ~~Produce the standard assets for the new enemies;~~
* Feature: Editor
  + Finish the editor so I can build levels for menus, worlds and of course, game levels;
  + Produce the first World with the Editor to test the feature
* Feature: Tweaking
  + Build a tweaking platform so I can easily make modifications to the gameplay at one place and (optionally) see the result in real-time.

Week #2: More planning

* Feature: Story
  + Produce a brief background story about the *Commodore*;
  + Produce a brief background story about the turrets;
  + Produce a brief background story about the aliens;
  + Produce a brief background story about the war between the humans and the aliens.
* Feature: Levels
  + Create a level design document that describes each level in terms of turrets, power-ups, enemies, waves, resources availability and other data like which world/environment, which elements are introduced for the first time and associated tutorials/tips, etc.
* Feature: World
  + Create the layouts (8) for the worlds;
  + Define which levels in a world must be completed to unlock a new World.
* Feature: Cutscenes
  + Produce the storyboards (9) for the cutscenes (intro, outro and worlds transitions) based on the main story.
* Feature: Level Endings
  + Produce storyboards (8) for the different level ending animations.
* Feature: Balance and tweaking
  + Revise the document describing the weapons and enemies properties to create a perfect balance.

Week #3: Plumbing & Textual assets

* Feature: Worlds
  + ~~Generalize the actual worlds system to support different worlds;~~
  + Implements the layouts for the worlds with the editor;
  + Plug the transitions from a World scene to a Cutscene scene.
* Feature: Cutscenes
  + ~~Create a new cutscene scene which support skipping back (go to the previous world) and forth (go to the next world);~~
  + Create the transition between a World and a cutscene and vice-versa;
  + Serialize the number of times a cutscene has been seen so a player don’t have to see the cutscene every time he goes in a world;
  + Test the feature.
* Feature: Level endings
  + Generalize the actual level ending animations system to support different endings;
  + Add a new ending tag in the level descriptor;
  + Test an alternative ending.
* Feature: Levels
  + Add a new assets package tag in the level descriptor;
  + Test an alternative assets package.
* Feature: The *Commodore* quotes
  + Produce the quotes of the *Commodore* based on the background stories.
* Feature: Turret quotes
  + Produce the quotes of the turrets based on the background stories;
  + Implement the quotes of the turrets;
  + Tweak the turrets quotes system to display the quotes in an optimal way.

Week #5: Levels

* Feature: Levels
  + Create the levels layouts (50) with the level editor;
  + Create the waves layouts for each level;
  + Finalize the level with the data from the level design document;
  + Plug the levels in the Worlds.
* Feature: Tutorial
  + Implement the tutorials elements described in the levels document in the actual tutorial system (i.e.: level descriptor).
* Feature: The *Commodore* quotes
  + Implement the quotes of the *Commodore* in the actual tutorial system (i.e.: level descriptor).

Week #6: Assets

* Feature: Levels
  + Produce environments assets as described in the design document, which include: backgrounds, enemies, turrets, things to protect and turrets placeholders, animations, sound effects, particle effects and music;
  + Package the environments assets in the package descriptor file;
  + Assign the environment assets to the levels.
* Feature: Cutscenes
  + Create the assets necessary for the cutscenes;
  + Implement the storyboards for the cutscenes (i.e. create the animations).
* Feature: Level endings
* Create the assets necessary for the level ending animations;
* Implement the level ending animations.

Week #7: Tweaks and playtest

* Feature: Balance
  + Implement the data described in the balance document;
  + Playtest the data described in the balance document.
* Feature: Levels
  + Playtest and tweak the levels;
* Feature: Enemies
  + Playtest and tweak the enemies;
* Feature: Power-ups
  + Playtest and tweak the power-ups;
* Feature: Turrets
  + Playtest and tweak the turrets;

Week #8: Polishing

* Change the help menu to include the new turrets, power-ups and enemies.
* Fix the mention bugs in the game design document;
* ~~Remove the product key protection in the Windows release;~~
* Pass the XBLIG marketplace evil check list;
* Create a tiny shortcut system to go directly to an unlocked world.

if (jobs.peek() == null) {

* Feature: Assets styles
  + Create the “realistic” assets style;
  + Create the “cartoonish” assets style;
  + Create the “black & white” assets style;
  + Create the “vectorial” assets style;
  + Create the “retro” assets style;
  + Create the “isometric” assets style;
  + Add each assets styles to the package descriptor;
  + Apply each assets style to some levels.
* Feature: Highscores
  + Capture the logged on user on the session as the name of the player;
  + Add a “submit highscore” button when a new highscore is made;
  + Implement the logic necessary to handle highscore submission;
  + Create the highscores webpages in PHP (with the database);
  + Create a global highscores scene where the player can see the highscores for each level;
  + Implement the logic necessary to handle global highscores fetching.
* Feature: Multiplayer
  + ~~Add supports for multiple opened menus and inputs in the GUI;~~
  + ~~Add player log in / log out in main menu;~~
  + Design specific multiplayer levels;
  + Change the help menu to include instructions for multiplayer;
  + Add supports for multiple mouse inputs (optional);
  + Test Xbox 360 integration.