Design Document for:

“fortris”

aka Blastris

Written by Min Chang

Version # 1.00

Design History

Concept and design by Min Chang with input from Joseph Huckaby.

Version 1.00

1. This is the first version of Fortis.
2. This version of Fortris is a single player only experience.
3. This version of Fortris is meant to run on iOS devices (iPhone, iPad) only.

Game Overview

1. Fortris was conceived as a unique update to the classic game of Tetris.
2. Fortris intentionally uses the familiar shapes and controls of Tetris in order to make the game instantly familiar to players.
3. Fortris is intended to only run on iOS devices. Screen dimensions and control schemes are optimized for play on iOS only.

Common Questions

What is the game?

Fortris is a unique hybrid of shape matching games (like Tetris) and tower defense games (like Plants vs Zombies).

Why create this game?

We want to create a game that is easy to pick up and play, but also invites players to get more "involved".

Where does the game take place?
Fortis takes place in a simple grid based environment.

What do I control?
In single player mode, players control the placement of blocks used to create their fortresses. This version of the game does not currently include a multi-player mode.

What is the main purpose of playing the game?
The goal is to build and maintain a fortress or as long as possible.

What's different about this game to other similar games?

There are many Tetris-like games that involve shape matching, or building currently on iOS. There are also some very successful tower defense games as well.

Fortris attempts to combine the two types of games to appeal to the "pick up and play" style casual gamer, but rewards repeat gamers with unique upgrades to the game play itself (rather than advancing a story or becoming involved with a non-game related incentive structure.

Feature Set

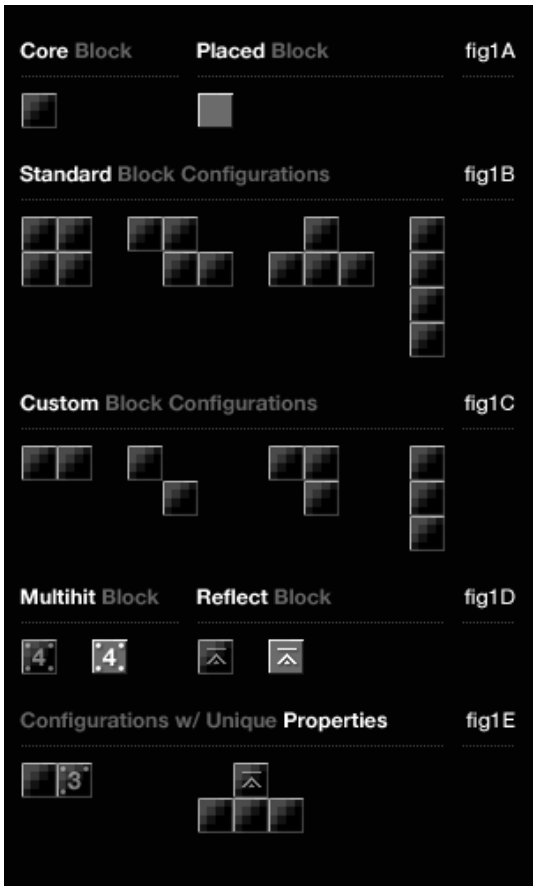
General Features

Easy to understand gameplay and game world
Optimized for iPhone/iPad screens

Gameplay

Blocks

1. blocks appear sequentially (one at a time) from a "well" in the center of the screen
    1.1. these blocks are composed in varying configurations of the same core element (a small square, see **figure F1A**)
    1.2. these blocks will initially appear in the same configurations known as "tetrominos" as Tetris. Again, this is the reinforce the familiarity in terms of gameplay. (**figure F1B**)
    1.3. blocks will continue to appear as long as player has space to "place" them in their fortress.
        1.3.1. if they have no area in which to place the block, the well simply "holds" until an appropriate space is cleared.



2. blocks are manipulated similarly to Tetris
    2.1. shapes can be moved in all four cardinal directions (north, south, east, west) one space at a time.
    2.2. shapes can be rotated in 90 degree increments
    2.3. shapes can be "dropped" into position (where they remain until destroyed)

3. Game controls mimic Tetris intentionally. This makes it easy for anyone who has played Tetris to understand the mechanics of how the blocks fit together and how to manipulate them (move, rotate) on screen.

4. As the player advances, these blocks will appear in unique configurations (see **figure F1C**):

4.1. the blocks will, in the later levels, intermittently also take on certain "properties" which affects gameplay:
  4.1.1. absorb multiple hits
    4.1.1.1. the number of impacts this block can "take" will be displayed on the block as a number. This number will update to always show how many impacts it can absorb remains.
    4.1.1.2. deflect projectiles
      4.1.1.2.1. this block will deflect the projectile back in the direction it came
      4.1.1.2.2. this deflected projectile will also destroy any one projectile in it's path.
      4.1.1.2.3. this block reverts to being a "normal" block after the deflection

Walls

1. the game begins with a well and "inner walls" in the center of the screen (represented by a white square with lines, see **figure F2A**).

2. also on screen at the start of the game is an "outer wall" composed of prepositioned blocks, 16 blocks to a side (see **figure F2C1**).
  2.1. the player places blocks in the "area" (see **figure F2B**) between the well and the outer wall to build up their fortress.
    2.1.1. this "area" is represented by a flat black area, where the grid background is not displayed.
    2.1.2. projectiles that impact a block destroys it (according to the block rules cited above)

3. as the game progresses, the outer wall increases in size (perimeter - see **figure F2C2**), by

4. the game ends (or the player loses a "life") when a projectile impacts a side of the "well" or one of the "inner walls".
  4.1. most of the player's time will be spent looking within the confines of their fortress (given the blocks can only be placed within that area). To mitigate possible player frustration, the fortress itself will provide a visual clue to activity going on at the time:
    4.1.1. when a projectile is close in proximity to the fortress, the side the projectile is on will turn red.
    4.1.2. "active" blocks will have a texture and a glow to indicate that is the piece in play.
    4.1.3. "placed" blocks will be a flat grey color to indicate they can not be moved.

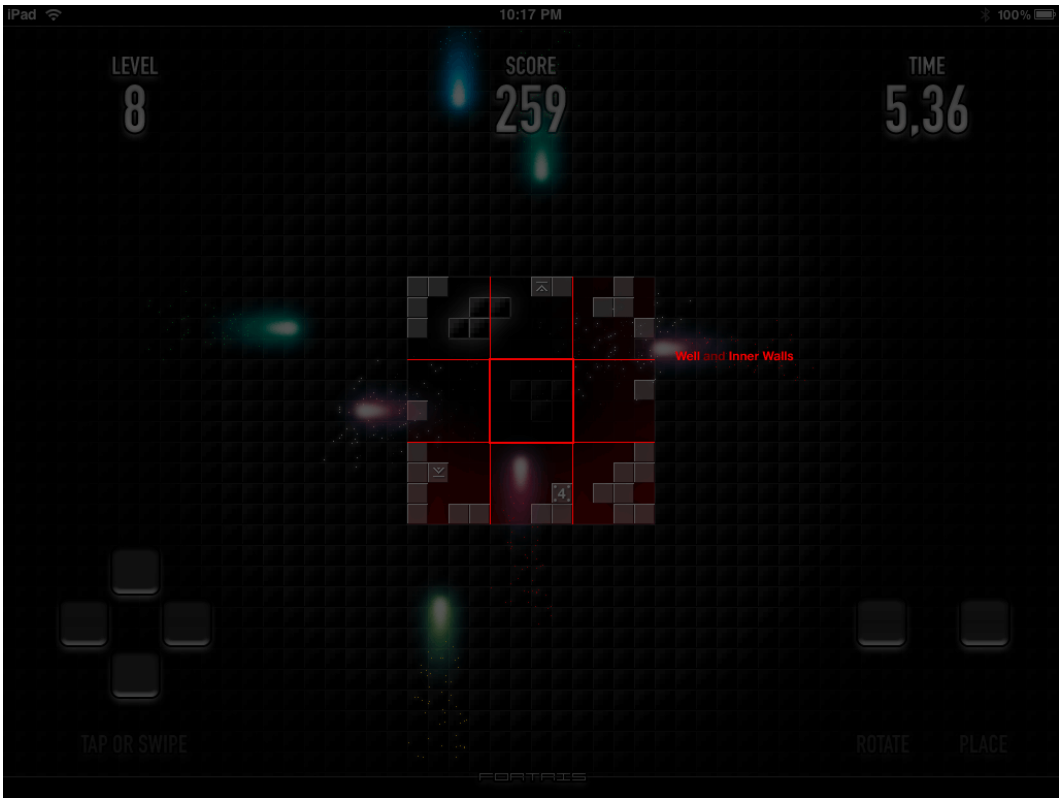
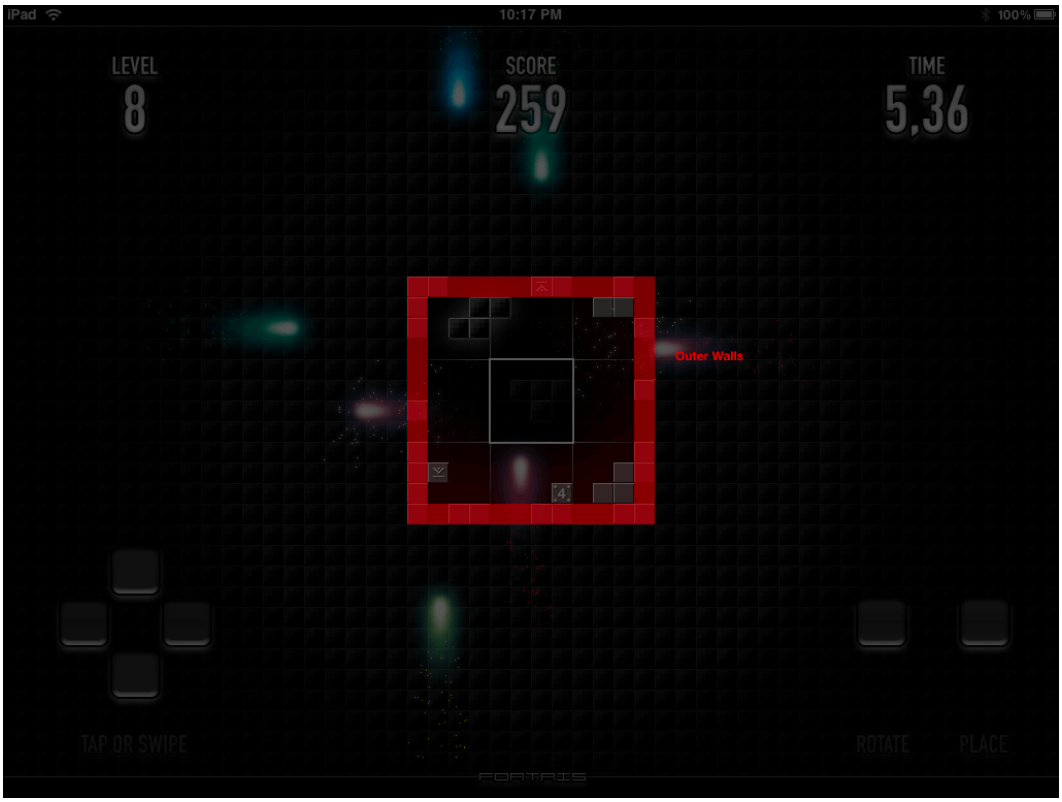


figure F2A (above), figure F2C1 (below)



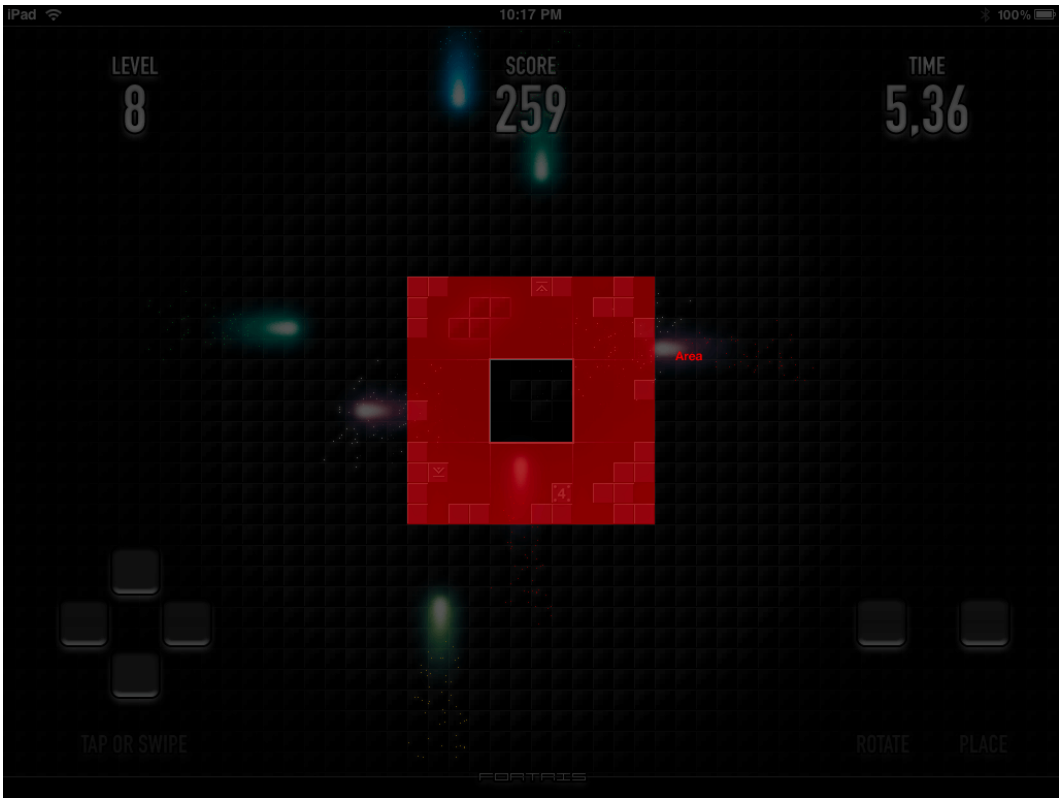
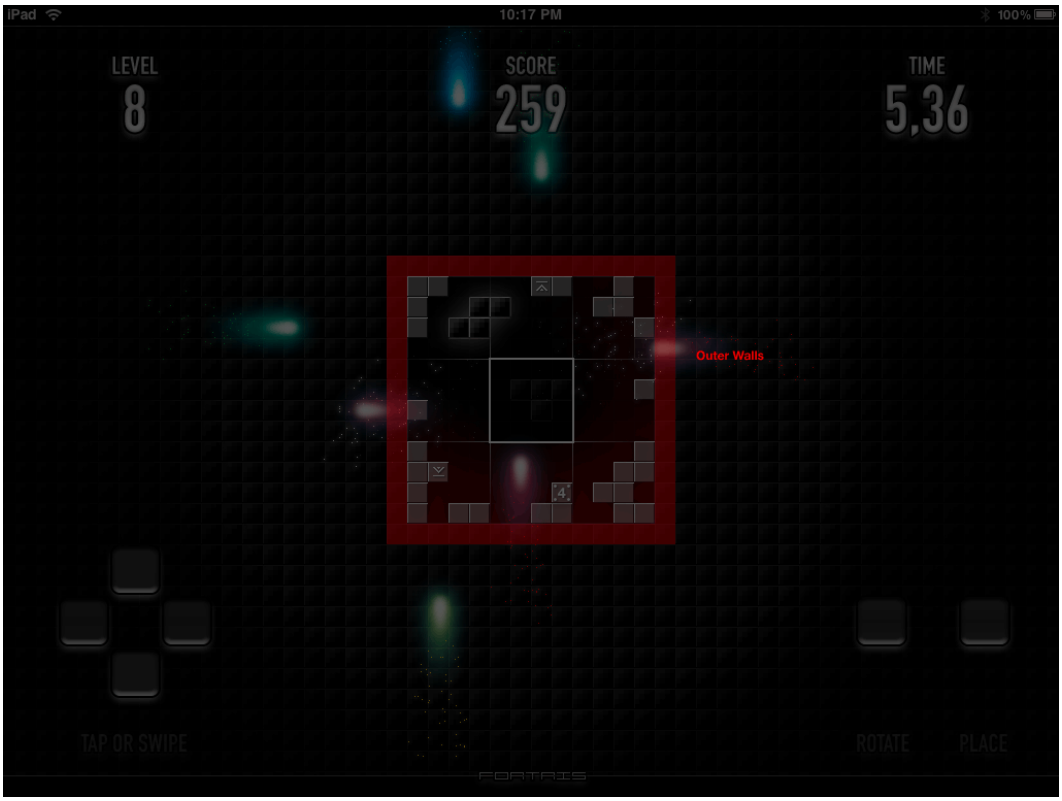
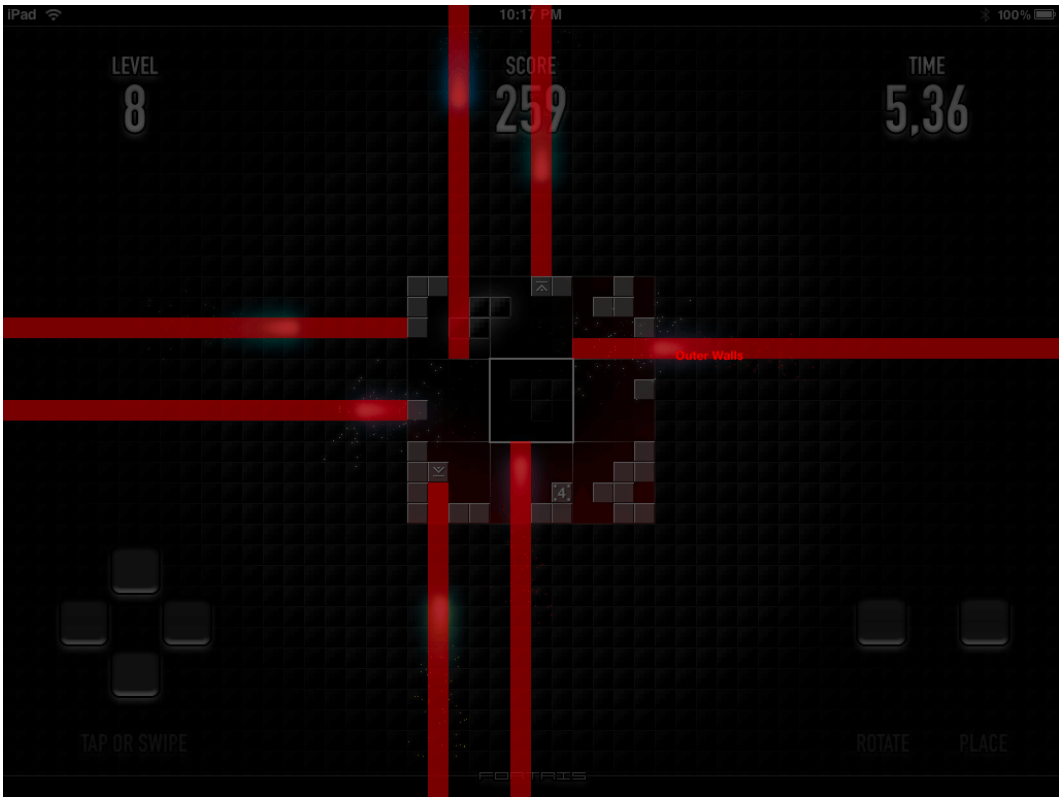


figure F2B (above), figure F2C2 (below)



Projectiles

1. 1. opposing the player are projectiles that travel in a straight line, from one of the four sides of the screen (see **figure F2D**).



2. projectiles move at a constant rate toward the center of the screen, although they will increase in frequency as the game progresses.

3. to add some color to an otherwise dark playing field, the projectiles will change color based on their proximity to the fortress.
   3.1. projectiles start off in the color blue (as they appear on the edge of the screen)
   3.2. as they get closer to the fortress, the projectiles will shift color from blue to green then yellow and finally red.
   3.3. the projectiles leave a trail of particles in their wake, again, to add a hint of color to an otherwise dark play field. the particles also reinforce the "motion" of the game. destructed blocks, inner walls and the well also leave white particles.

4. although there is a fog-of-war applied, the iOS device screens aren't square. projectiles that come in from the shorter sides (if held in portrait mode, they would be to the left and right of the fortress / if held in landscape mode, they would be above and below the fortress) appear less frequently than the projectiles on the longer sides.

Environment

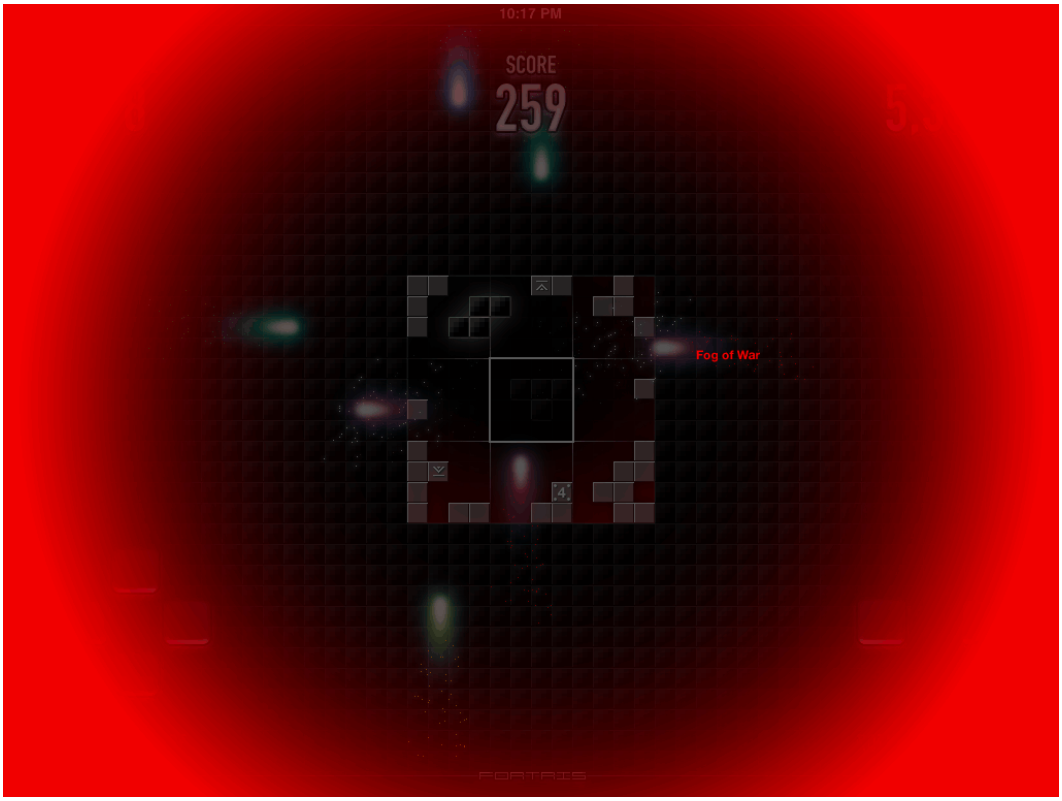
The game world is based on a fixed grid of blocks. All movement and placement of blocks, movement of projectiles aligns to this grid. The grid and a "fog-of-war" are the only two elements of the game environment. The game environment should be kept simple and non-distracting allowing the player to focus on maintaining their fortress.

The Grid

1. The base grid itself is simple in that it's a uniform grid, each cell of the grid is 1:1 in it's aspect ratio. The grid dictates the size of nearly all the game assets:
    1.1. A single block matches a single grid cell in dimensions
    1.2. A block can not misalign with the grid when "placed". The game engine should know to "snap" the block to align with the grid.

Fog of War

1. Covering the grid is a fog-of-war (**figure F2E**), which serves two purposes:
    1.1. It normalizes the aspect ratio of the screen against the play field. The disparity between the long and short sides of the device screen isn't as big a factor.
    1.2. It allows for some space to render UI elements without interfering with gameplay (High Score, Timer, Level, etc)



User Interface

Overview

This section current being written and designed.

Musical Scores and Sound Effects

Overview

Music isn't a major consideration in this game, although sound effects should be used to provide basic feedback of gameplay. The only real requirement for audio at this point is to have some controls in the UI to allow muting and or volume setting.

Sound Design

Again, given the stereo output of the iOS devices as well as the ability of the user to change orientation at will, sound design beyond basic game feedback isn't a strict priority.

Single-Player / Multiplayer Game Types

Overview

This version of Fortris is intended to be a single player experience only. A multiplayer experience is in the early phases of concept and design, but is irrelevant to this release.

Story

While a story can be applied, to encourage a deeper level of player investment, the story would not influence gameplay in any way for this release.

Game Duration / Game Progression

A typical Fortris match should run for "a few minutes". Fortris' difficulty ramp will be tuned throughout the course of game play testing.

To ease new players into the game and prevent early-game frustration for repeat players, we will use an exponential difficulty curve.

For example (*the following is NOT a hard set formula and is only provided as an illustrative point*):

1. Level 1 will start slowly, giving new players ample time to familiarize themselves with the core game play mechanics (positioning and placing of blocks).
    1. The standard set of Tetris blocks are introduced.
    2. Projectiles are delivered infrequently. Progresses until 15 blocks are destroyed.
2. Level 2 increases rate of projectile delivery to 1.4 times more frequently.
3. Level 3 introduces the non-standard configurations of blocks.
    1. Projectiles increase rate of delivery to 1.6 times more frequently.
    2. Progresses until an additional 15 blocks are destroyed.
4. Level 4 increases rate of projectile delivery to 1.8 times more frequently.
5. Level 5 introduces the multi-hit block.
    1. There is an increase to the size of the fortress.
    2. Projectiles increase rate of delivery to 2.0 times more frequently.
    3. Progresses until an additional 10 blocks are destroyed.
6. Level 6 introduces the reflection block.
    1. Projectiles remain at the same rate of delivery.
    2. Progresses until an additional 10 blocks are destroyed.

Further levels vary between increasing rate of projectile delivery and increasing the size of the fortress. Number of blocks to be broken before advancing levels will be capped.

A wildcard may need to be introduced to maintain a good balance and "feel" to the gameplay. For example, we may need to monitor rate of block placement against the rate of projectiles delivered.

Victory Conditions

Fortris is meant to be a continuous play game. We may need to change the levelup structure in order to address play balance and feel.

An alternative option is to introduce formal "levels", with specific goals or themes (survive 50 projectiles, stay alive for 2 minutes, score a chain using the reflection block, etc)

Saving and Loading

While a true "save" function is irrelevant to Fortris, suspend functionality should be incorporated if possible.

The player should be able to "exit" the game via a tap on the iPhone/iPad home button (circular physical button). The player should also then be able to launch Fortris to return directly to their game in progress, but paused.

Miscellaneous Notes

Design

Design of screens outside of core gameplay continues. I am currently working on a title screen, and a credits screen, as well as finalizing the UI for game play.

“Document Figures Appendix”

This appendix lists the figures referred to in the document that illustrates concepts. All full size images are included in the folder labelled “Document Figures”.

- F1.png "blocks and block configurations"
- F2a.png "well and inner walls"
- F2b.png "placement area"
- F2c1.png "outer walls"
- F2c2.png "outer walls expanded"
- F2d.png "projectile paths"
- F2e.png "fog of war"

“Objects Appendix”

“User Interface Appendix”