# VOOGA Functional Spec

By Benjamin Wolf

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Overview</td>
<td>2</td>
</tr>
<tr>
<td>Goals</td>
<td>2</td>
</tr>
<tr>
<td>User Stories:</td>
<td>2</td>
</tr>
<tr>
<td>Gamer Goals:</td>
<td>3</td>
</tr>
<tr>
<td>Developer Goals:</td>
<td>3</td>
</tr>
<tr>
<td>Additional Goals:</td>
<td>3</td>
</tr>
<tr>
<td>Non-goals:</td>
<td>3</td>
</tr>
<tr>
<td>Design – Developers</td>
<td>4</td>
</tr>
<tr>
<td>Overview:</td>
<td>4</td>
</tr>
<tr>
<td>Design – Gamers</td>
<td>4</td>
</tr>
<tr>
<td>Overview:</td>
<td>4</td>
</tr>
<tr>
<td>The Splash Screen:</td>
<td>4</td>
</tr>
<tr>
<td>Splash Screen Overview:</td>
<td>4</td>
</tr>
<tr>
<td>The Game Information Screen:</td>
<td>6</td>
</tr>
<tr>
<td>The Game:</td>
<td>7</td>
</tr>
<tr>
<td>Overview:</td>
<td>7</td>
</tr>
<tr>
<td>Starting the game:</td>
<td>7</td>
</tr>
<tr>
<td>Playing the game:</td>
<td>7</td>
</tr>
<tr>
<td>Losing a Game:</td>
<td>8</td>
</tr>
<tr>
<td>Completing a Level:</td>
<td>8</td>
</tr>
<tr>
<td>Design – Breakout</td>
<td>9</td>
</tr>
<tr>
<td>Requirements:</td>
<td>9</td>
</tr>
<tr>
<td>Breakout Use Cases:</td>
<td>9</td>
</tr>
</tbody>
</table>
ABSTRACT

Video Object-Oriented Gaming Architecture (VOOGA) is a gaming framework that allows easy development of arcade style video games. In addition to being a gaming framework, it is a playable video arcade (What use is a framework if there are no games to play on it?).

OVERVIEW

VOOGA is both a gaming framework and an arcade. The idea is that VOOGA will have a reusable core that is responsible for playing the game. Developers will not need to touch this core in order to build new VOOGA games. This allows developers to develop new games for VOOGA in a timely fashion. VOOGA will also provide developers with tools that help with rapid game development.

VOOGA will be a playable arcade when it launches. There will be many games for the user to choose from in the first version of VOOGA (including breakout). Keeping this in mind, the user will not need to be technical in order to play games on VOOGA. They will be able to open the application and choose which game they want to play. Adding new games (that have already been developed) to the application will be a seamless task for more advanced users (however these advanced users are not necessarily developers).

GOALS

USER STORIES:

Bob is a simple gamer. He loves playing simple arcade games on his computer. His friend tells him about VOOGA, an arcade that he can play lots of cool games on. He downloads VOOGA and opens it up. Bob sees the variety of games that he is able to play. By clicking on a game, he gets to a screen that gives him the instructions on how to play as well as a way to start the game.

Mary is an advanced computer user. She is very comfortable installing program and following instructions that require her to download new files. She installs VOOGA on her computer (because she needs a distraction from her homework). The VOOGA site says that VOOGA can play a cool version of Asteroids, however it is not included with the VOOGA package. She is able to download some files, follow a simple set of instructions, and add Asteroids as one of the games she is able to choose on the VOOGA splash screen.

Professor Duvall loves writing arcade games but he never has enough time to write one in its entirety. Likely, his students have written VOOGA, a gaming architecture that allows for quick
development for games. He can write a very flashy clone of Geometry Wars (a very popular arcade game available on xbox live) in a few hours. He also doesn’t need to read his student’s code as the documentation for VOOGA explains exactly what he needs to do to write a new game.

GAMER GOALS:

The gamer should be able to:

- select which game they want to play at startup
- switch between games after running the program
- access a record of all high scores for each game
- learn the rules of each game (if they don’t know how to play)
- pause their game and resume it after it has been paused
- know their current score and remaining lives while their playing the game

DEVELOPER GOALS:

The developer should be able to:

- Write a new game in 2 hours or less
- Write a new game without touching the core code
- Write a new game without having to understand how the core code works
- Add new levels to existing games and new games (preferable without physically writing code)
- Modify existing levels in existing games (again without writing code)
- Change the theme for their game (explosions, sounds, images, etc)
- Develop a multi-player game
- Develop games with different sets of rules
  - Game over when time expires
  - Game over when gamer loses all of their lives

ADDITIONAL GOALS:

- VOOGA will ship with 3 games
  - One of these games will be Breakout
  - See Appendix A for a suggestion of games to implement

NON-GOALS:

- Support 3D graphics
• Ensure games are compatible with other VOOGA engines

**DESIGN – DEVELOPERS**

**OVERVIEW:**

Developers will rely heavily on the documentation in order to figure out how to improve existing games and write new games. This section should detail the design of the framework and explain the steps a developer needs to take in order to modify an existing game or write a new game.

This section is left intentionally incomplete as it relies too heavily on the technical design. Students should complete this section of the functional spec.

**DESIGN – GAMERS**

**OVERVIEW:**

The gaming experience should be fairly simple for the gamer. This section explains the gaming experience from beginning to end. A gamer will have the opportunity to select a game, start the game, and be notified when they win (or lose). The basic program flow looks like this:

![Diagram of game flow](image)

One can argue that the game information screen is an unnecessary extra step, however it does provide a way for each game to feel like a separate game. Having an information screen gives developers a place to put game options and show off the high scores accomplished in that game.

**THE SPLASH SCREEN:**

**SPLASH SCREEN OVERVIEW:**

The splash screen is a place for the user to select which game they want to play. As VOOGA is a framework, there should not be a limit on how many games are displayed on the splash screen. The following is a mockup of a sample splash screen. It shows the basic features that are necessary for a successful splash screen. It is by no means a complete mockup and should be used as a starting point. Additional features can be added to this screen and are encouraged.
Some things to note in this mockup:

- An image for each game with the name of the game appearing below the image
- Placing a mouse over an image highlights it and shows a description of the game in the right column
- Scrollable left column

Some additional ideas:

- Sort games into categories and create category browser
- Search games or game descriptions

NOTE: Please ignore features that require a user to log into VOOGA. It would be cool to have each user customize their own splash screen, however, supporting user accounts in VOOGA is no longer a requirement. This means ignore any lines that say Bob as well as the customize
splash screen menu item. Also, this mockup still needs more detail. A background image might make it look cooler. Does the user get a pop-up asking them if they are sure they want to exit before they exit?

THE GAME INFORMATION SCREEN:

If VOOGA were just one of the games that it supported, the game information screen would replace the welcome screen. From this screen, you will be able to start a new game, change the game options, and load a saved game (if supported by that game. Again, the following is a rough mockup meant to show the minimum amount of features expected on this screen. It should be redesigned to look cooler and additional functionality can be added if necessary.

NOTE: The options screen should be designed in this spec and a reference to the page it is on should be found here.

Ways to improve this mockup:

- Use a list of options instead of buttons
- Decide on whether the description at the bottom of the screen works for you, maybe you want to add an option (or button) called “about”
- Add file menus relevant to the game (if you have a good reason to).
- Add a way to view high scores (or display high scores on this screen)

THE GAME:

OVERVIEW:

Playing the game is the most complicated part of VOOGA. Although each game is different, it is easy to find enough similarities to make the gaming experience for all games. For example, almost all games have a notion of score, lives remaining (or time remaining), and level. This section of the spec will walk you through starting the game, playing the game, losing the game, and completing a level.

STARTING THE GAME:

Games developed for VOOGA should be very friendly to the user. The games should not start until the users are ready to start them. Clicking on “New Game” from the game information screen does not necessarily mean that the user wants to start playing. There should be either a timer (3 . . . 2 . . . 1 . . . START) or a prompt for user input (“Click the left mouse button to start”) when a new game is loaded.

PLAYING THE GAME:

Each game has a different set of rules. There are however things common to all games, such as score, lives, and levels. The following screen shot is taken from Ballistik. Notice the information found on the screen about score, lives, and levels. It’s all there and easy to see. VOOGA will be able to handle displaying this information for any game as well as have the ability to provide more information. Some additional information that developers might want to make available to the gamers are: high score, time remaining, items remaining (for games that involve limited bullets, bombs, or other items).
NOTE: This picture is taken from a 3D version of breakout. Implementing 3D graphics in VOOGA is not required and is strongly discouraged.

**LOSING A GAME:**

Each game may have a different losing condition. Regardless, VOOGA will make it obvious to the gamer that they have lost. On this game lost screen, the gamer will be able to see all of the high scores for the game. If they have achieved a high score, it should be highlighted on the list of high scores. The name associated with the high score should be the user name of the user currently logged in. VOOGA will then allow the user to start a new game, return to the game information screen, or return to the main menu.

NOTE: on the topic of high scores - since there are user accounts, you will want to have a list of personal high scores and global high scores.

**COMPLETING A LEVEL:**

After a gamer finishes a level in a VOOGA game, there will be a screen that summarizes how they did. Each level will be separated by this screen and the next level will not start until the user is ready (see discussion on count-downs versus prompting for an action in the game start section). Most games are won when all the levels have been completed. In this case, there should be a game won screen that is similar to the game lost screen.
**DESIGN – BREAKOUT**

**REQUIREMENTS:**

- Breakout must be a fully functioning game with at least 3 levels.
- Breakout must have bricks that take more than one hit to be destroyed
- Some bricks should drop power-ups that can be collected by a paddle
  - Some example power-ups
    - Firing bullets (it would be nice if there were a limit to how many you could fire)
    - Bigger Paddle
    - Smaller Paddle
    - Sticky Paddle (ball sticks to the paddle until the user releases it)
    - Ball on the screen turns into 3 balls
    - Ball goes through bricks instead of bouncing
- Game should be robust
  - Ability to change brick color / image without too much trouble
  - Ability to write new levels quickly
- The game must be cool (in order of least cool to most cool)
  - Sound effects
  - Moving bricks
  - Untraditional breakout (see breakout clones)
  - Making bricks explode when they are hit

**BREAKOUT USE CASES:**

1. A gamer selects “New Game” from the Breakout Information Screen. The game loads up and waits for the gamer to press spacebar (or click a mouse button) before starting.
2. The gamer presses spacebar to start the game. The gamer presses the right arrow key to move the paddle to the right. The gamer presses the left arrow key to move the paddle to the left.
3. A ball hits a brick causing it to disappear, and bounces off the brick. After the brick disappears, the game’s score is updated.
4. A brick contains a power-up that drops down towards the drain (bottom of the screen) when the brick is hit. The user moves his paddle below the power-up and there is a collision between the paddle and the power-up. This power-up causes the paddle to grow longer.
5. A brick contains a power-up that drops down towards the drain (bottom of the screen) when the brick is hit. The gamer ignores it and it goes to the bottom of the screen and disappears.

6. When the ball is moving towards the drain, if it misses the paddle it goes through the drain and results in a loss of life for the player.

7. A gamer loses his last life and loses the game. The gamer does not have a high score, so only the game lost screen is shown.

8. A ball hits a brick that only dies after 3 hits. The ball bounces off of the brick.

9. The paddle collides with a power-up. This power-up causes two new balls spawn from the location of the current ball.

10. Two balls collide and bounce off of each other.

11. A power-up collides with a ball. The ball goes through the power-up having no effect on it.

12. The paddle collides with the right wall and can no longer move to the right.

13. The last brick is hit, the level is finished, and the level completed screen is shown. The user presses spacebar to continue to the next level and the next level is loaded.

14. A power-up is attained that makes a ball go through all the bricks, destroying them but not bouncing off them.

15. A gamer pauses the game or focuses on a different screen.

16. A gamer breaks a previous high score while playing breakout. The gamer's high score is added to the high score list when they win (or lose).

17. A gamer completes the last level, winning the game. They see a screen that proclaims that they have won.