



Request For Proposal

for

Software Product Development

Background

MyGameMill.com is a web platform which allows game designers (board, card, and other tabletop games) to design and playtest their games all online.

Games are created by uploading assets (board images, dice images, card images, rules, etc...) to create game components. Players will be able to choose from a library of pre-existing components, as well. Components are then assigned mechanics like random dice rolls or card shuffling. Once the components are set a beginning game state is created which is saved for future playtests.

Games can be shared publicly or privately for playtesting. During playtest players will be able to chat with one another via IM, Video, or Voice. Players will also be able to leave written feedback on the games they play.

Company Profile

Please provide a company profile including at least the following:

- Name of Company
- Address
- Contact Person
- Telephone
- 3 References (minimum)
- Examples of similar projects (Javascript user interactions, HTML5 graphics, node.js backend)

Terms of Engagement

Please respond by September 27, 2013 if you are interested in bidding on this project. Proposals will need to be provided by October 8, 2013 to be considered. This project is expected to be funded via Kickstarter campaign, and cost will be a major consideration in project award. The initial cost estimate for the "Basic" project as listed below is \$17,000 US.

Mandatory Items

The My Game Mill codebase is stored in a git repository hosted at GitHub. Pull access will be granted to the contractor, and code deliveries will be done at each milestone via GitHub pull request.

Source code delivered must be human-readable, unobfuscated, and fully commented. Deliveries must pass code review and deficiencies corrected prior to acceptance. Code copyright will be assigned to the customer.

Customer Furnished Products

Note that terms in **bold** are defined in the glossary at the end of this document.

At the beginning of the contractor's period of performance, an initial codebase will be provided by the customer which provides the following capabilities:

1. **User** authentication mechanism using Facebook authentication
2. **User** chat capability
3. Ability to create private game playing areas based around a user-defined set of game components.
4. **Designer** ability to upload images for use in **Game Boxes** and **Components**.
5. **Designer** ability to define a **Board**.
6. Ability of **players** to join a **table**, view the **board**, and to navigate and zoom to examine the **board**.
7. **Designer** ability of the define "token" **components**.
8. **Player** ability to move the **components** about the **table** and **board**, and to have those motions reflected on the displays of other **players**. **Players** will observe movements made by remote **players** to be animated.
9. Each **player** is assigned a "seat" around the **table**, which affects the angle at which they see the **table**, **board**, and **components**.
10. Other **players** can see where each **player** is "seated" around the **table**.
11. The **table** is not automatically deleted when all **players** have left it, allowing asynchronous play.
12. The **designer** may delete a **table** at any time.

Project Requirements

New functionality for MyGameMill.com will be added to the existing MyGameMill codebase. Extensions must conform to current practices, including:

1. The majority of the server-side code is written in node.js, using the sails.js framework.
2. End users interact with MyGameMill using a browser, via HTTPS and Websockets.
3. Flash will not be used.
4. User interface should be implemented using CSS, HTML, and Javascript only.
5. Server-side persistent datastore is provided using MongoDB.
6. Primary user authentication is provided by Facebook.
7. Server-side templating is done using Jade markup.
8. Data is sent to the client using Websockets.
9. Third-party modules and libraries may be included, however they must be used under an appropriate license (MIT, Apache). GPL-licensed code may not be used.
10. Third-party modules must be available either in the npm database, copied into the existing codebase in the "third-party-products" database, or installable via aptitude on ubuntu 12.

11. All build steps must be automatically completed on an Ubuntu 12 server in response to "npm install --production".

Specific Project Requirements (Basic)

Note that terms in **bold** are defined in the glossary at the end of this document.

Please provide a proposal including cost and estimated time to extend the Customer Furnished Products described above to add the following:

1. Facebook Integration
 - a. **Users** should have the ability to message other **users** when they are offline (use facebook messaging).
 - b. **Users** should be provided with a scheduling capability (integrated with facebook events).
 - c. **Users** should have the ability to request permission to message another user, and ability to block other users from messaging them.
2. Capabilities involving game design (the **workshop**)
 - a. **Designers** should have the ability to define "card" **components**.
 - i. Two sided **components** with a defined physical size.
 - ii. Independent images can be defined for each side of a card.
 - iii. The creation date and time of each **component** will be stored in the MongoDB database when the **component** is created.
 - iv. The last edited date and time of each **component** will be stored in the MongoDB database when the **component** is edited.
 - b. **Designers** should have the ability to define decks
 - i. Multiple cards can be grouped together into decks, which are contained in a **game box**.
 - ii. The same card can occur more than once in the same deck.
 - iii. It should be possible both to create a new instance of a card in a deck (where editing one copy edits all instances of that card), OR to create a duplicate of a card (where the cards start out identical, but editing one card does not affect the other).
 - iv. Designers should be able to toggle if the number of items remaining in a deck should be displayed next to the deck's image. The default for this setting is to have the tag displayed.
 - v. Designers should be able to define an empty deck to place on the **table** (e.g. a discard pile).
 - c. **Designers** should have the ability to define dice.
 - i. Any number of sides can be specified.
 - ii. An independent image can be defined for each side.
 - iii. Instead of an image, users can define numerical values for the die sides.
 - d. **Designers** should have the ability to define game rules
 - i. **Designers** can upload game rules in plain text or PDF format to associate with a given game.
 - e. **Designers** should be able to "clone" game **components** and **game boxes**
 - i. **Designers** should be able to make a copy of an existing **component** and edit it, without changing the original component.

- ii. **Designers** should be able to make a copy of an existing **game box** with a new name, and edit it, without changing the original **game box**.
 - f. **Designers** should have the ability to define which **components** will be made available upon game launch.
 - i. **Designers** should be able to define the initial number of each **component**.
 - ii. **Designers** should be able to define the initial location of each **component**.
 - iii. **Designers** should be able to define which **components** are present in **players' hands** at game launch.
- 3. Capabilities involving game play
 - a. **Players** should be able to manipulate **components**.
 - i. Players should be able to move any **component** around the **table**.
 - ii. Players should also be able to rotate any **component** on the **table**.
 - iii. Shortcuts should be available to quickly rotate **components** 90, 180, or 270 degrees.
 - iv. For any **component** that has multiple associated images (cards, dice), **players** should be able to select the displayed image.
 - v. For any **component** that has multiple associated images (cards, dice), **players** should be able to view all the associated images.
 - vi. For dice, **players** should be able to "roll the die", resulting in a random image and rotation being displayed.
 - b. **Players** should have a **hand**.
 - i. **Players** should be able to move **components** from the **table** to their **hand** and back.
 - ii. **Players** should be able to manipulate **components** in their **hand** as they can on the **board**.
 - iii. **Player** actions in their **hand** will not disclosed to other **players** at a **table**.
 - iv. **Players** should be able to "pass" an item from their **hand** to another **player's hand**.
 - c. **Players** should have the ability to create an arrow between any two points on the **table** (drawn on top of all **components**). This arrow should be displayed to all **players** and can be removed by any **player**.
 - d. **Players** should have the ability to add textual notes to any point on the **table** or on **components**. These notes should be displayed to all **players** at the **table** and can be removed by any **player** at the **table**.
 - e. **Players** at a **table** should have the ability to place a numerical "score counter" on the **table**.
 - i. The "score counter" should be easily incrementable or decrementable, either with a single click to add or subtract a single point, or with a button that allows the user to specify the amount to add or subtract in a dialog box.
 - ii. The "score counter" should be settable to a given value (requested via dialog box from the **player**).
 - f. If **components** obscure each other, **players** must be able to adjust the display order (place on top of, place under)
 - g. **Players** should be able to combine cards into "stacks"
 - i. When a card is placed on another card on the **table** or in the player's **hand**, the **player** should have an option to combine the two cards into a stack either by placing the new card on top of the card, or underneath it.

- ii. When a card is placed on top of a stack, the **player** should have the option to place it into the stack on top, on the the bottom, or at a random location in the stack.
- iii. When a stack is placed on top of a stack, the **player** should have the option to add the new stack on top of the existing one, on the bottom of the existing one, or by shuffling into the stack (where the cards are intermixed randomly to create a new stack).
- iv. The number of cards in a stack should be displayed next to the stack.
- v. A card stack should be displayed in the system as one image from the top card.
- vi. **Players** should be able to move card stacks around on the **table**, or in their **hands**.
- vii. **Players** should be able to rotate a stack arbitrarily, with or with shortcuts to rotate it in increments of 90 degrees.
- viii. **Players** should be able to draw a single card from the top or bottom of a stack, directly into the **hand**.
- ix. **Players** should be able to draw an arbitrary number of cards off the top or bottom of the stack into their **hand**, where they should become a new stack.

Specific Project Requirements (First Extension)

Note that terms in **bold** are defined in the glossary at the end of this document.

Please provide a proposal including cost and time to extend the proposed basic system described above to add the following:

1. **Players** should be able to manipulate a die after it has been “rolled”.
 - a. **Players** should be able to add subtract a specified amount from the displayed value (for numerical-value dice).
 - b. **Players** should be able to “turn a die upside down” - change the displayed value or image to the one which would be on a physical die.
 - c. **Designers** can edit the game rules in an online editor once they have been uploaded.
2. **Designers** will have access to tools for collecting **playtest** data
 - a. All **components** can be given names in the **workshop**. Unique names will be assigned by the system by default at creation.
 - b. All moves in a **playtest** will be logged and made available to game designers.
 - c. A **designer** will be able to define surveys to send to **players** after a **playtest**. Integration with a free, third party service is acceptable.
 - d. Survey results for **playtests** using a given **game box** are visible in the **workshop** in the section where that **game box** can be edited.
3. **Designers** should be able to print out all **components** in a **game box** for physical testing. Physical size settings on **components** should be respected in printouts.
4. **Designers** should be able to design the full set of **components** in a **game box** by uploading a Comma-Separated Value (CSV) file.
 - a. Image files should be uploadable in a zip file and be referenceable in the CSV file by their original filenames.

5. **Designers** should be able to define a “Draw Bag”.
 - a. Draw bags can contain any combination of any other **components**.
 - b. The draw bag has a physical location and size on the **table** or in a **player’s hand**, but no orientation.
 - c. Designers should be able to toggle if the number of items remaining in a draw bag should be displayed next to the bag’s image. The default for this setting is to have the tag displayed.
 - d. **Players** can only remove random items from a draw bag.
 - e. Any **component** can be moved from the **table** or a **player’s hand** into the draw bag.
6. **Players** should have the ability to show a **component** on the **table** or in their **hand** to another player without moving it from its location.
 - a. The **player** viewing the **component** should be able to view all the associated images and textual notes for that **component** (view sides of a die, sides of a card, etc.)
7. **Designers** should be able to define arbitrary **component** properties as key/value pairs
 - a. **Designers** should be able to see a list of all the **components** in a stack, draw bag, or **game box** in a tabular format, sorted by any combination of the properties.
 - b. Game designers should be able to see the number of **components** in a stack, draw bag, or **game box**, in a tabular format with all the possible values for a single given property.
8. All My Game Mill functions should function properly on iPad and Android Tablets.
9. **Designers** should be able to name card stacks and draw bags on the **table**.
 - a. **Players** should be able to draw **components** from a named stack or draw bag with a single-click, even if that stack/draw bag is not visible on screen.
 - b. **Players** should be able to move a **component** to a named stack/draw bag with a menu action attached to that **component**, even if that stack/draw bag is not visible on-screen.
10. **Players** should be able to select multiple **components** on the **table** or in their **hand**.
 - a. Selection should be possible either by holding a modifier key and clicking those **components**, or via “lasso” selection.
 - b. **Components** selected as a group should be moved or rotated as a group.

Specific Project Requirements (Second Extension)

Note that terms in **bold** are defined in the glossary at the end of this document.

Please provide a proposal including cost and time to extend the proposed “first extension” system described above to add the following:

1. **Players** should be able to make their own copy of a game box (if permitted by the **game box’s** original **designer**).
 - a. The original **designer** of the **game box** must be able to set distribution controls: No duplication allowed, duplication allowed once, duplication and redistribution allowed.
2. **Players** should be able to undo moves at a **table**.
3. **Players** and **designers** should be able to edit the **game box** rules during a **playtest**, from the **table**.
 - a. The **designer** should be able to enable or disable editing by other **players**.
 - b. The **designer** should be able to view the rules revision history, view previous revisions, and reset the rules to a previous revision.
4. **Tables** should provide a component “linking” capability.

- a. Any **components** can be linked.
 - b. Linked **components** can be moved and rotated as a group, retaining their relative position and orientation.
 - c. **Components** can be individually added to and removed from groups.
- 5. **Players** should have the ability to “expand” a stack of cards on the **table** or in their **hand**.
 - a. When a stack is “expanded”, all cards in the stack are visible at once.
 - b. cards can be added to an expanded stack at any location (top, bottom, middle of the stack).
 - c. Cards can be removed from an expanded stack at any location in the stack.
 - d. Expanded stacks can be collapsed back into normal stacks at any time.
- 6. **Designers** should be able to add a timer to a **game box** as a **component**.
 - a. timers can count up from 0 or down from a defined minute/second value.
 - b. Timers have single-click options to start, stop, and reset.
 - c. Timers can be tied together so that stopping one timer starts the other.
 - d. Timers should be synchronized so that they display the same value to all players at any given time, accounting for network latency.
 - e. Timers have no physical location on the **table**, but are instead “docked” to a **player’s** browser window.
 - f. Timers have names which can be defined by the **designer** in the **workshop**. The system will assign a default empty name.
 - g. Timer names are displayed in the **table** view.

Glossary

The following terms are used throughout prior sections of this document, with the following meanings:

Board	An image which is placed on the Table which provides context and guidance for Players .
Component	Any item that can be defined in a Game Box and manipulated by Players at a Table . Including, but not limited to: • Dice • Cards • Tokens
Designer	A user who creates a Game Box in the workshop , and is considered the “owner” of that Game Box . Only the designer who owns a game box can create a table for that game box from the Lobby .
Game Box	Top-level container that contains Components .
Hand	A private area which is unique for each player and table . It can contain any component and its contents are not displayed to any other player at the table .

Lobby	Along with the Workshop , one of two major sections of the My Game Mill website. The lobby is where any User can go in order to communicate with other Users who are not currently at a Table .
Player	A user who has been granted access to a particular table by a designer .
Playtest	A session where a Designer invites players to a table to play a game in a given game box , for the purpose of testing game rules and graphics.
Table	A semi-private page which can be accessed players , linked to a particular game box and primarily displaying the Board from that game box .
User	Any person who connects to My Game Mill, and logs in using their Facebook account.
Workshop	Along with the Lobby , one of two major sections of the My Game Mill website. The workshop provides tools for Designers to define and edit Game Boxes , and the components contained in them.