

Using Assembla.com Tools in Botball

Jeremy Rand

Norman High School

jeremy@asofok.org

Using Assembla.com Tools in Botball

1. Introduction

The challenge of Botball is multifold. The most commonly cited challenges involve application of engineering, computer science, and the scientific method. But another challenge that is less publicized, and yet often harder for many teams, is the ability for the team to organize, plan, and collaborate effectively on a project whose scope is so wide that no single member of the team is likely to know everything that the team is doing at one time. For maximum efficiency, the engineers and the programmers should be aware of what issues each other are having, so that a programmer can offer a software solution to a bothersome engineering issue, or vice versa. Programmers should know what other programmers are doing, so that work doesn't get duplicated, and so that if one programmer encounters an issue, other programmers are aware of it and may be able to avoid it or offer their own solution (and the same applies to engineers talking to engineers). Documenters should be able to get up-to-date information from the engineers and programmers without having to harass them for the past two weeks' updates when there's less than 24 hours to go before the documentation deadline. All team members should be aware of what the other members are expecting of them, and the team should be aware of when tasks are expected to be completed (and any relevant bottlenecks that are hindering progress). And keeping track of all this information should be transparent and easy enough that it doesn't become a bottleneck in itself.

Teams struggle on all of these points. I saw my old Botball team fall apart in 2007 due to poor communication and teamwork, and even otherwise successful teams encounter trouble in these areas. This paper will describe an excellent solution to these problems: Assembla.com.

Assembla.com is a project management tool which facilitates efficient collaboration by a team. They are aiming the site at geographically distributed teams, but the site is also extremely useful for any team which needs better communication and collaboration, such as Botball. They provide a large number of web-based tools which make these tasks relatively simple and routine rather than the mess that most Botball teams deal with. These tools include a wiki, a ticketing system (task list), a milestone system, Subversion (a server for sharing source code with the team), a message board, daily/weekly reports, and e-mail alerts and RSS feeds. Even if the whole team hasn't worked with some of these concepts, the tools are quick to learn.

2. Setting Up

Setting up should be a matter of minutes of work. Visit Assembla.com [1], and click "Choose a plan and sign up." Don't panic when it shows prices; you won't have to pay to use Assembla with student projects (e.g. Botball). Scroll down to "Free Public Plan" and click "Create a free

public space.” You will be shown a list of configurations; Botball teams will probably want “Subversion Hosting with Integrated Tickets.” You then will be asked to register on Assembla. Choose your username and password, enter your e-mail address, and register.

Once your workspace is set up, there’s one last thing you have to do. The Free Public Plan makes your workspace visible to the public. As a Botball team, you probably don’t want other teams seeing what you’re doing. Luckily, Assembla offers free Private/Professional plans for student projects such as Botball teams. Instructions for applying for a student plan are available on the Assembla website [2]. Once you’ve followed the instructions, you should probably e-mail Assembla customer service [3] to notify them that you’re waiting for a student plan to be approved (they don’t often check the student project queue unless they’re asked). While you’re waiting for them to approve your application, you can still use your Assembla workspace to learn how the setup works; just keep in mind that other teams will be able to see what you post until Assembla approves your application. Please note that while Assembla is nice enough to sponsor student projects, they pay for those spaces out of their own pocket, and they need revenue from paid spaces to continue to offer free student spaces. So if your team has a bit of money, or if you’d like to use Assembla for your non-student projects, please do purchase a space.

3. Admin Tab

From your new space, you can click the Admin tab to change settings. You can add non-default Tools (tabs for your space) here. If this paper mentions a tab that you don’t have, simply add the Tool in the Admin tab. You can also set security preferences. If you’re a Botball team, you’ll probably want to set Non-Member Access to None, so that other teams aren’t able to see your space. If you’re using Assembla for a non-Botball open-source project, you might set Non-Member Access to View. You can also set different permissions for different Tools, although this has limited utility in Botball. Other customizations such as the Appearance of your space (e.g. a banner image) and making backups of your data are also available from the Admin tab.

4. Team Tab

The Team tab allows you to add new team members, either by e-mail address or by Assembla username. You can also see contact information for your teammates (if they have elected to share it with you), and get reports on their recent activity in the space.

5. Wiki

Documentation can be hard to collaborate on. Often, teams will just have a programmer or engineer write the technical documentation, and e-mail it to the documenters, who post it. This is not a particularly optimal solution, because the documentation is not readily available to the rest of the team. Assembla’s solution is to use a wiki to manage documentation. Just like Wikipedia allows everyone to contribute and improve the content, Assembla’s wiki allows any team member to edit the documentation at any time. When the documentation deadline comes, the documenters simply copy-and-paste the documentation from the wiki into the Botball Home Base. Not only does this help your documentation scores, but it also helps less experienced

programmers and engineers, who can look at documentation for the code and designs by the lead programmers and engineers and learn from them.

6. Ticketing

Team members often try to mentally remember what their responsibilities are, and what tasks they have delegated to other members. This is a recipe for disaster, because issues will invariably rise in which members argue about who was responsible for what task that didn't get finished. And by the time that argument happens, it may be too late to get the task finished. Assembla offers a ticketing system to solve this problem. Every task is represented as a "ticket." Tickets contain a brief summary of the task, who requested the task, who is assigned to complete it, what component of the project is involved (e.g. "Programming: Seeding Create Bot"), the relative priority of the task, whether the task is unstarted, started, awaiting testing, or finished and working, and an estimate of the difficulty of the task (in terms of required work hours). Tickets also host a discussion thread, so for a ticket about fixing a major programming bug, two programmers can post their findings in that ticket, allowing all of the information on the task to be located in a centralized place. Creating a ticket is easy, and can even be done through e-mail. Unfortunately, the process of creating a ticket does not support AJAX [4] yet, so creation of tickets is not completely optimal in terms of efficiency. We're hoping Assembla will remedy this soon. Tickets support searches and filters, and various metrics can be viewed which graphically summarize how much work is being done.

7. Milestones

Many teams like to impose deadlines on specific tasks so that the team doesn't get behind. Assembla makes this easy with the Milestones tool. You can define a set of milestones which progress from the start of the season, through your regional tournament. Each milestone can have a due date and any number of tickets associated with it. For example, you might make a milestone for every two-week period between the workshop and regional, or you might make a milestone for each meeting (with tickets for each goal for that meeting). Or you might have parallel milestones; one timeline for each robot. Assembla will accommodate whatever your style of benchmarking yourself is. Assembla does recommend that you assign tickets that you don't plan to work on in the immediate future to the Backlog milestone, and moving a ticket from Backlog to another milestone once you need to work on it soon. Assembla can show a calendar view of your milestones, and can even integrate with iCal or Google Calendar (so if you set your meetings as milestones, they will show up in your calendar automatically – a handy alternative to manually notifying team members of meeting dates).

8. Burndown

The Tickets and Milestones features support a graphical view called Burndown. Once you have defined the tickets, expected required time, and due date for a milestone, you can view a graph showing how the total expected remaining work time in order to complete the milestone has changed over time. If the graph has a positive slope (more work being created than being completed), it usually is an indication that things are not going to plan. If the graph has a

negative slope (more work being completed than being created), it indicates that you're getting things done effectively. The workloads of the individual team members are also shown.

9. Scrum Reports

Often, you'll want to know what someone else on the team has been working on, and whether they need anything from the rest of the team to continue working effectively. Scrum Reports fill this function. Periodically, team members fill out a simple form, asking three questions: "What did I do?" "What will I do next?" and "What obstacles do I need help with from the rest of the team?" Assembla suggests that Scrum Reports be filled out anywhere from daily to weekly; we have found that for Botball, it is best to submit a Scrum Report after every meeting. Filling out a scrum report takes about two minutes, and it drastically increases effectiveness of communication. It also gives the documenters readily-available information about what each team member is doing. When a team member states in a Scrum report that he or she help with obstacles, you may wish to have a standardized way of dealing with that. For example, some teams may use a "Quality Circle" in which at the beginning of each physical meeting, the team discusses what team members need from each other. Or an electronic method (such as the message board and chat features discussed later in this paper) may be better suited to your team.

10. Message Board

For free-form comments between Botball team members, some teams use e-mail. E-mail works for this (somewhat), but it's not designed for the task. For example, members will have to enter the e-mail address of every person on the team in each e-mail. There is no centralized archive of this communication, and when a new member joins the team, he or she doesn't have access to the previous e-mails. And wouldn't it be nice if the comments appeared with your other team stuff (in the Assembla space) rather than in Outlook or Thunderbird, mixed in with all your other e-mail? Assembla's solution is a simple message board, in which team members can discuss anything team-related without the problems of e-mail which were just listed. Most people are familiar with the concept of a message board, so further elaboration here is probably unnecessary, except for this nice feature: if you have some team members who are addicted to their e-mail client (e.g. Outlook), and don't want to give it up, they can still use the message board. Received messages can be e-mailed to the user automatically, and users can post to the message board by sending e-mail to a specific e-mail address set up by Assembla.

11. Persistent Chat

The message board is a great replacement for e-mail, but what about IM? IM shares all of the problems with e-mail that were discussed above, but is even worse for a team than e-mail is, for the chief reason that if a couple members of the team have an IM conversation, no one else on the team knows what was said. Posting logs manually is something that you just plain will not be able to get every team member to do. That said, instant electronic communication is a highly useful concept. So what to do? Luckily, Assembla has addressed this problem (somewhat) with their Persistent Chat feature. It works like a web-based chat room, complete with audio notifications and a list of currently chatting members. But unlike most chat rooms, when you

leave the room, the chat history is saved in a convenient, date-based, searchable format. If everyone leaves the room, and comes back the next day, the conversation will still be there, ready to be resumed. If your team is ever in a situation where everyone needs to meet but some people can't be physically present, Persistent Chat is exactly what you need.

12. Subversion

Subversion is probably the most useful Assembla tool for programmers. Subversion is like a file server (usually used for source code), except it automatically keeps track of every revision made to the code. This is extremely useful, since if you mess up the code, you can get back to an earlier version. To use Assembla's Subversion server, we recommend TortoiseSVN [5]. TortoiseSVN is a complete GUI for Subversion which integrates into Windows Explorer. Once you've installed TortoiseSVN, visit the Source/SVN tab in your Assembla space. It will give you an SVN URL. Next, you're going to Checkout the Subversion repository. You will only need to do this once for your space. Make a directory on your computer (it doesn't matter where), and right-click in Windows Explorer from within that directory. Click "SVN Checkout." Enter the URL (however, you should change the "http" to "https", as this increases both compatibility and security), leave all the other fields at the default, and click OK. It will probably ask you for your Assembla username and password; enter them. Once complete, the contents of your Subversion repository will be in the directory in which you right-clicked. To add files or folders initially, simply copy them into that directory, right-click the files or folders, click "SVN→Add," then right-click them again and click "SVN Commit." When you change the files later, and you want to share the new versions with your team, right-click the files (or any Subversion folder containing them) and click "Commit". To receive updated files from your team, right-click the files (or any Subversion folder containing them) and click "Update." If you and someone else on your team have both updated the same file, your work won't be deleted. Just Update the file, and then click Commit. Usually, TortoiseSVN will merge in the changes without issue. In the event that you and someone else changed the same lines of the code, TortoiseSVN will show you a side-by-side comparison, with the opportunity to manually merge the changes that conflicted (any changes that don't conflict will still automatically merge).

13. Code Browser

If you haven't checked out the entire Subversion repository (e.g. if you're on a shared computer), you can still look at the code, and even download files from a web interface. Click the Source/SVN tab, and choose Browse Source. You can choose files to look at (with syntax highlighting for many languages), see older revisions of files (with the differences between revisions highlighted, as well as summaries by the programmer of what was changed), and download files. If you're ever stranded without access to the Subversion code that you checked out, but you need to quickly look at the code, the Code Browser is an excellent tool.

14. E-mail and RSS Alerts

Most people will not be persuaded to check a particular website every day, no matter how hard you push them. So how can you guarantee that they will find out about things on your Assembla

space if you don't know that they're checking the website on their own? Assembla offers e-mail and RSS alerts to solve this issue. By default, team members will be notified by e-mail of all events in the space. If you don't like receiving large quantities of e-mail, you can also set the alerts to send hourly or daily digests from the Stream tab (click Change Alert Settings) or from the link at the bottom of the alert e-mails. For the tech-savvy members, RSS feeds [6] are also available. Alerts can be disabled by members, so if you want to be certain that members receive important notices, make sure that they know to not disable alerts. It is also possible to integrate alerts with Twitter; we have not used this functionality, so we can't comment on it. If your team members frequently use Twitter, you may wish to try this feature out.

15. Dashboard

The Dashboard tab shows a summary of activity in the other tabs. This is an excellent way to know "at a glance" what's going on. It lists recent scrum reports, ticket changes, Subversion code commits, wiki changes, messages, and all tickets currently assigned to you. You may wish to set the Dashboard tab to be the default tool that team members see when they enter your space (you can do this from the Admin tab's Appearance settings).

16. Time Tracking

Assembla has a Time tool, which can be used to log time spent on a project. We haven't found much utility for this tool in Botball, but it could make life a lot easier if you're using Assembla for a project that you're paid for.

17. Conclusion

Assembla has made collaboration and communication much easier for our Botball team, and we think it can help yours too. If you'd like to contact me, I frequent the Botball Community chat room and forums [7]. Please don't e-mail me with questions unless you've tried but haven't had luck contacting me on the Botball Community, or if it's an emergency. Special thanks to Jeff Carl and Andy Singleton of Assembla for reviewing this paper for accuracy and offering a couple suggestions.

References

- [1] Assembla. <http://www.assembla.com/> , 2009.
- [2] Assembla. Free Spaces for Student Projects. <http://blog.assembla.com/assemblablog/tabid/12618/bid/7056/Free-spaces-for-student-projects-how-to-get-them.aspx> , 2009.
- [3] Assembla. Customer Service. service@assembla.com
- [4] Wikipedia. AJAX (Programming). [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) , 2009.
- [5] TortoiseSVN. <http://tortoisesvn.tigris.org/> , 2009.
- [6] Wikipedia. RSS. <http://en.wikipedia.org/wiki/RSS> , 2009.
- [7] Botball Youth Advisory Council. Botball Community. <http://community.botball.org/> , 2009.