

Sprint 2 Plan
Twitch Giveaway Tracker
Sprint Completion Date: 11/17/2019
Revision: 1.1, Date: 2019/11/5

Goal: To be able to allow users to join multiple channels, analyze them and notify users if giveaways are detected.

Kelin **Weichi** **Coby** **Brendan** **Scott**

User Stories

User Story 1: As a user, I want to be able to join multiple channels at the same time while applying some filters, so I can focus on getting the giveaways I want.

Task 1: Get a list of all channels then join (2 hour)

- Be able to choose a game or category, then get a list of all the channels in that directory and have the program join them.
 - (Get list of Channels based on a set criteria/category using twitch API, put them into an array and feed that array into the programs joining function)

Task 10: Sub-windows to filter/search channels to join, include stuff in

(https://twitch-tools.rootonline.de/channel_previews.php?stream_type=live) (4 hour)

- Create an interface that allows you to filter the channels you want the bot to monitor by title, game and number of viewers.
 - (Would link selectable options in the interface to the code we wrote in the other tasks that gathers titles, streams from a certain category etc so you could choose which streams you wanted to monitor at any given time.)

Total for user story: 6 hours

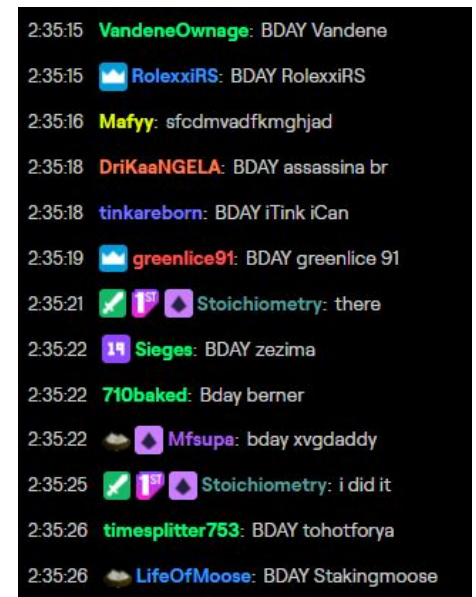
User Story 2: As a user, I want the software to detect giveaways instead of me finding it through every channel.

Task 2: Analyze by bots (2 hour)

- Look at the streams viewer list to see if certain bots that are known for hosting giveaways are in the channels viewer list (StreamElements, Nightbot, etc)
 - (Use the API to get a list of the current viewers, put the list of viewers into an array and check that array against a list of known bots to see if one of the bots is in the list of current viewers)

Task 3: Analyze by chat patterns (8 hour) - (Probably multiple people / sprints)

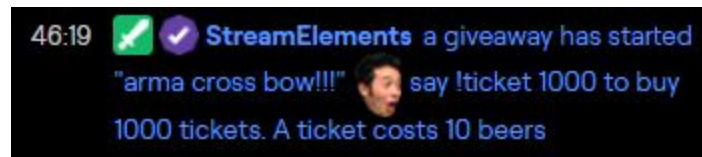
- Look for patterns in chat, such as people typing a keyword like !enter or !join.
- Look for multiple similar messages in a row from people which might indicate that message is a keyword for a giveaway.
 - (Have to parse the chat and notify if a keyword (!enter, !join, etc) is typed in the stream chat. May have to check if it's typed more than once in a small time frame to avoid false positives (i.e. someone randomly typing one of those things without a giveaway actually happening).)
 - (Can parse chat and notify if a word is said several times in a short period, which would indicate a bunch of people typing a certain keyword to join a giveaway)




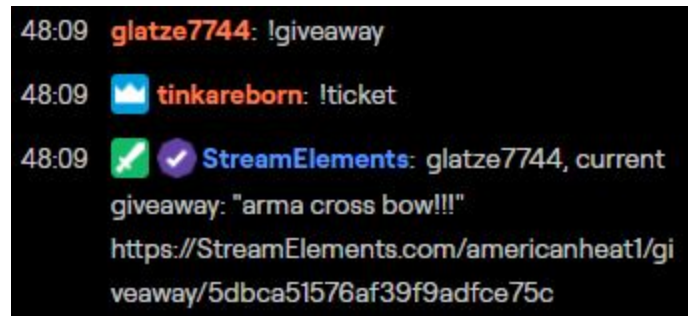
- Look at the titles for keywords specified by the user, such as giving or giveaway or raffle.
 - (Use the Twitch API to get a list of the titles of the streams in whatever category you choose, then find the ones from that list that contain the keyword you're looking for.)



- Look at the chat messages to see if you can tell what's being given away
 - (Parse the chat messages to see if there are any announcements from a bot used for giveaways that shows the prize, then if it exists add that to the notification that we send the user)

(Ex. Two different ways bots announce what is being given away in a channel, in this case) an "arma cross bow")



46:19  **StreamElements** a giveaway has started "arma cross bow!!!" say !ticket 1000 to buy 1000 tickets. A ticket costs 10 beers



48:09 **glatze7744**: !giveaway
48:09  **tinkareborn**: !ticket
48:09  **StreamElements**: glatze7744, current giveaway: "arma cross bow!!!"
<https://StreamElements.com/americanheat1/giveaway/5dbca51576af39f9adfce75c>

user has won

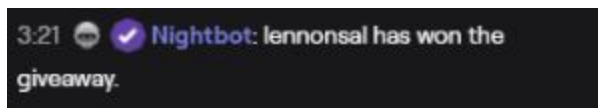
the chat to messages


Task 6: Detect if a giveaway (2 hour)

- Look at check for from bots that indicate who won the giveaway, and if the user has their username included in that message, send them an alert..
 - (Parse the messages in twitch chat for a message from a bot that says a giveaway has been won by someone, then if the username of the user of the program is in that message, send them a notification.)

Task 7: detect if a giveaway has ended (2 hour)

- Look for the message that says a user has won a giveaway from a bot, if you find that message mark the giveaway as completed.
 - (Parse the messages in twitch chat for a message from a bot that says a giveaway has been won by someone, if you find that message, send a notification so the user knows the giveaway is over)



3:21  **Nightbot**: lennonsal has won the giveaway.



0:38 **StreamElements** @cjhcarter won the giveaway with a 4.34% chance to win 

(Some examples of how bots will display the winners of a giveaway)

Total for user story: 20 hours

User Story 3: As a user, I want to have a graphical notification for when a giveaway is happening.

Task 8: Show a pop-up for when a raffle is detected, sound included (4 hours)

- Send the user a notification when we detect a giveaway is occurring in that channel, include the name of the channel and possibly the keyword that we detected.
 - (Not sure how to implement this as I don't know much about the UI system we're using or how to make pop-up windows, I know that it's possible with chatty however so we can potentially learn from their implementation)

Task 9: Generalize the pop-up so it can be used for other purposes (2 hours)

- Make the popup that appears a general popup so we can call it in other functions that we need to make a notification for.

Total for user story: 6 hours

- (If we're able to call the function to display a pop-up we could use it to indicate if a giveaway has ended as well)

User Story 4: As a user, I want to log in with a simple click, so I can avoid fetching oauth tokens and configuring the set up manually.

Task 11: Automate the login process (2 hours)

- Let the user login to an account and have the program automatically find and enter the OAuth key on its own
 - (Take the info needed to get an Auth key and put it into the Oauth generator, may require some additional research but can copy how chatty does it)

Total for user story: 2 hours

Team roles:

Brendan Reilly-Langer - project owner, developer

Wei-Chieh Hung - scrum master, developer

Scott Liu - developer

Coby Forrester - developer

Kelin Erickson - developer

Initial Task Assignment -

Brendan Reilly-Langer - task 1, 2, 4

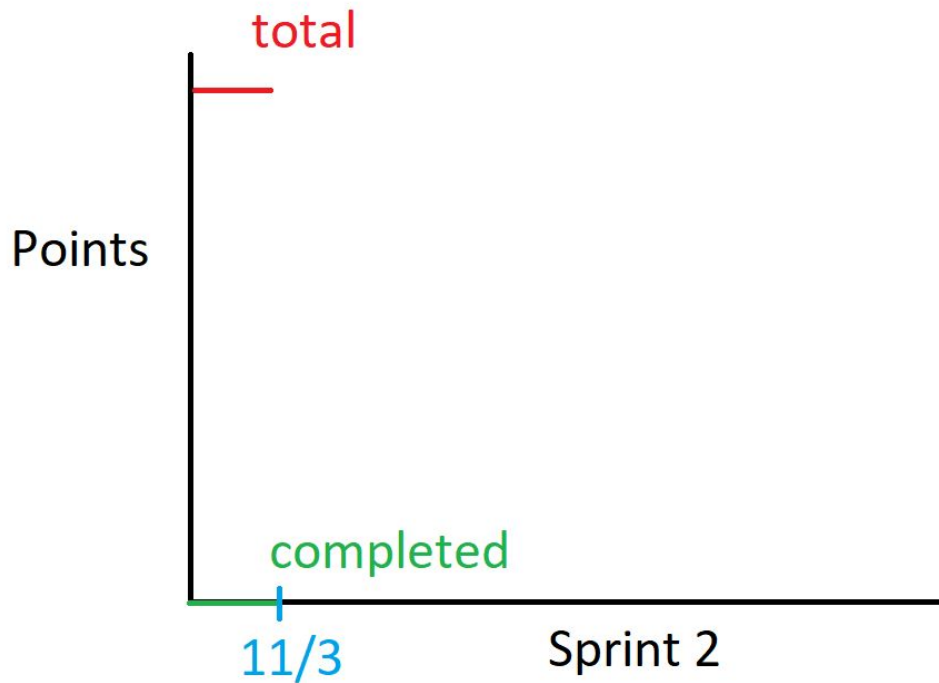
Wei-Chieh Hung - task 11, 8, 9

Scott Liu - task 10, 5

Coby Forrester - task 3

Kelin Erickson - task 6, 7

Initial Burnup Chart - Room 134A



Initial Scrum Board - Room 134A

User Story	Not Started	In Progress	Completed
User story 1	1, 10		
User story 2	2, 3, 4, 5, 6, 7		
User story 3	8, 9		
User story 4	11		

Scrum Times:

Monday, Wednesday, Friday

4:00 pm ~ 4:15 pm

Backlog:

Discord bot

Task 11: Set up a discord bot (10 hours)

- Connect the program to a discord bot that posts links to giveaways when it finds them in a channel, with the keyword it thinks it found, then deletes the giveaways from that channel when they are finished.
 - (Would involve learning the Discord API to connect to a channel and post a link. We would then link it to our program so that when it detects a

giveaway it posts it in the channel similar to how it notifies the user. It would also then delete the link from the channel when our script detects a giveaway has ended.)