

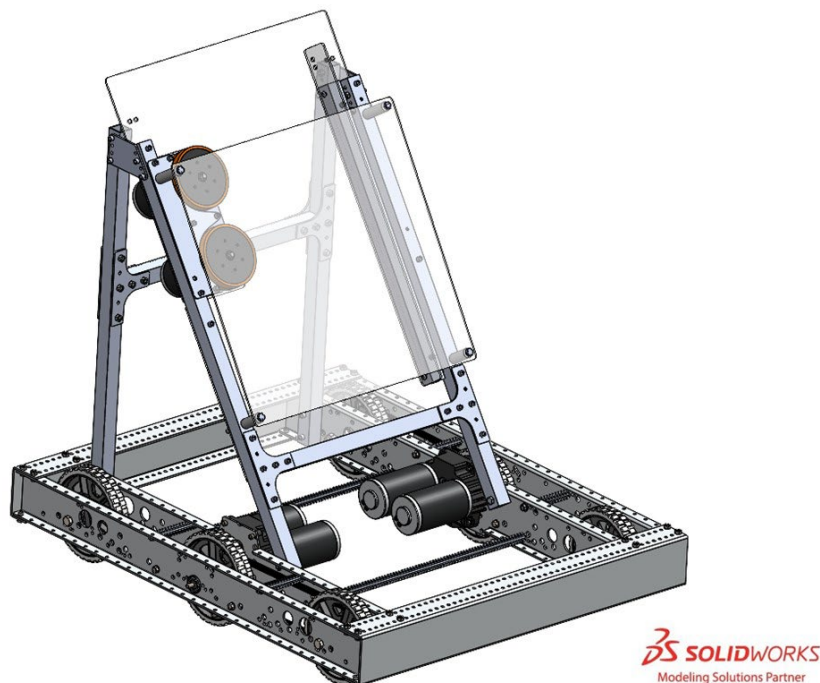
Compétition de robotique **FIRST[®]** 2024 Guide d’itération/amélioration pour le KitBot

Table des matières

1	Introduction	3
2	Quand commencer à penser à des changements	4
3	Amélioration – Ajout de nouvelles fonctionnalités.....	5
3.1	Analyse du jeu	5
3.2	Identification des améliorations	7
3.3	Prochaines étapes	7
4	Itération – Amélioration des capacités existantes.....	8
4.1	Tests utiles	8
4.2	Faire un remue-méninges sur les améliorations possibles.....	9
4.3	Prochaines étapes	14
5	Choisir – Transformer les idées en un plan.....	15
5.1	Établissement des priorités.....	15
5.2	Analyse – Estimation des ressources et de la portée	15
5.3	Analyse – Définissez votre « budget ».....	16
5.4	Faire des plans	17
5.5	Prochaines étapes	18
6	Annexe A – Exemple d’analyse de jeu de base, 2023.....	20
6.1	Exemple - Liste des tâches du robot	20
6.2	Exemple - Comment bien se classer	21
6.3	Exemple - Comment gagner des matchs	22
6.4	Exemple - Améliorations.....	24

1 Introduction

Figure 1: KitBot 2024



Le KitBot CRESCENDO^{MS} présenté par Haas est capable d’effectuer les actions suivantes. Certaines actions nécessiteront que l’équipe ajoute explicitement du code pour rendre cela possible (par exemple, le code AUTO) :

- Faire le tour du terrain (ailleurs que sous la scène) à l’aide d’une transmission différentielle (également communément appelée « tank ») conçue pour une vitesse maximale réalisable d’environ 15 pieds par seconde (~ 4,5 m / s).
- Précharger une note pour une utilisation en mode AUTO
- Marquer des points de DÉPART
- Marquer des notes dans le haut-parleur
- Recueillir des notes à partir de la source
- Jouer la défense

Il s’agit de capacités assez basiques en ce qui concerne toutes les tâches possibles dans le jeu. De plus, le KitBot a été conçu comme une solution très simple, ce qui signifie qu’il peut y avoir des opportunités d’itérer et d’améliorer les capacités de base. Dans cet esprit, les équipes peuvent choisir d’ajouter des composants supplémentaires pour permettre au robot de ramasser des pièces de jeu sur le sol, de monter sur la scène, et plus encore ! Ce document décrit un processus possible que vous pourriez suivre pour faire un remue-méninges et décider des améliorations possibles.

2 Quand commencer à penser à des changements

Avant ou après la construction de votre KitBot, vous voudrez peut-être revoir le jeu et déterminer s’il y a des capacités supplémentaires que vous souhaitez ajouter au robot de votre équipe. Le KitBot est conçu avec de futurs ajouts à l’esprit, et nous encourageons les équipes à le construire et à jouer avec lui un peu avant de le modifier ou de faire des ajouts, mais il y a des avantages potentiels à envisager des ajouts avant la construction. Ce sont quelques compromis qui peuvent vous aider à décider s’il faut faire ce premier exercice avant ou après avoir construit le KitBot.

Avant la construction du KitBot

- Avantages
 - Permet aux membres du groupe de design de travailler immédiatement au lieu d’avoir à attendre que la construction soit terminée
 - Il est plus facile d’apporter des modifications au design des pièces de base avant qu’elles ne soient assemblées
- Inconvénients
 - Possibilité de concevoir plus que vous ne pouvez construire avec succès
 - Possibilité d’apporter des modifications à des parties du design de base que vous ne comprenez pas entièrement

Après la construction du KitBot

- Avantages
 - Certitude que vous avez un robot qui peut jouer certains aspects du jeu
 - Meilleure position pour être en mesure de prioriser l’itération (améliorer les capacités existantes) vs l’amélioration (ajout de nouvelles fonctionnalités)
 - Meilleure compréhension de la fonction du design de base et de ce qui peut et ne peut pas changer
 - Les tests et les pratiques peuvent continuer simultanément avec le travail sur les améliorations
 - Les améliorations apportées au développement des programmes peuvent être testées immédiatement
- Inconvénients
 - Rien à concevoir tant que la construction n’est pas terminée
 - Changer les pièces de base peut nécessiter de démonter des choses
 - Vous devrez peut-être utiliser des écrous et des boulons au lieu de rivets pour démonter les choses plus facilement

3 Amélioration – Ajout de nouvelles fonctionnalités

Cet exercice peut s’effectuer avant ou après la construction du KitBot (voir la [section 2](#) pour les avantages / inconvénients de chaque option). Si vous terminez cet exercice après la construction du KitBot, nous vous recommandons de terminer tous les remue-méninges liés aux améliorations et à l’itération (Sections [3](#) et [4](#)), puis de les combiner lors de l’exercice de sélection ([Section 5](#)) pour prioriser les changements que votre équipe poursuivra.

Bien que cela ne soit pas strictement nécessaire, avoir des objectifs d’équipe est très utile pour l’exercice de sélection. Pour plus d’informations sur la définition des objectifs de l’équipe, consultez la vidéo « Goal Setting » (voa) sur la page Web [Team Management Resources](#) dans la section « Team Organization ».

Les exercices et les techniques présentés ici comprennent une version abrégée d’une analyse de jeu complète et d’un processus de stratégie pour le jeu avec le robot. Pour une ressource plus complète sur ce processus, consultez la section Strategic Design section of the « Effective FIRST® Strategies for Design & Competition presentation » de la page Web [Technical Resources](#) sous la section « Other Technical Resources->Scouting/Strategy section ».

3.1 Analyse du jeu

3.1.1 Analyse du jeu - Que pourrait faire d’autre notre robot ?

La première étape dans la réflexion sur des améliorations potentielles est d’analyser le jeu et de trouver une liste de choses possibles que le robot pourrait faire. Quelques questions figurent ci-dessous en guise d’exemples. Pour un processus plus complet, consultez le document « Kickoff Worksheet » sur la page Web [Technical Resources](#) sous la section « Other Technical Resources->Kickoff ». C’est différent d’une simple liste de tâches liées au jeu (bien que ce soit un bon point de départ!). Ne commencez pas encore à penser à la façon dont le robot effectuera ces tâches. Vous trouverez ci-dessous une liste de questions qui peuvent vous aider à dresser cette liste :

1. Quelles sont toutes les façons de marquer des points dans le jeu ?
2. L’un ou l’autre de ces éléments peut-il être décomposé davantage (p. ex., la marquer à différentes distances peut être une capacité en soit) ?
3. Comment le robot acquiert-il des pièces de jeu ? Souvent, il existe plusieurs façons d’acquérir les éléments de jeu et qui nécessitent des capacités uniques.
4. Y a-t-il d’autres tâches qui pourraient aider l’alliance (p. ex., des pièces de jeu sur le terrain à manipuler, des façons d’aider les partenaires, etc.) ?
5. Existe-t-il des moyens de ralentir le score des adversaires qui nécessitent des capacités de robot uniques (c’est-à-dire pas seulement la défense à l’aide du groupe motopropulseur du robot) ?

Vous cherchez un exemple ? Consultez l'[Annexe A – Exemple d’analyse de jeu de base, 2023](#) pour une liste de tâches possibles en lien avec le jeu FRC 2023.

3.1.2 Analyse du jeu - Comment nous classons-nous ?

Si les objectifs de votre équipe impliquent d’essayer de se classer le plus haut possible (par exemple, essayer d’être capitaine d’alliance, ou essayer de maximiser les points de district pour se qualifier pour un championnat de district, etc.), il est important de comprendre les critères selon lesquels les équipes sont classées. Si les objectifs de votre équipe n’impliquent pas un classement élevé (par exemple, être choisie dans une alliance, concevoir un mécanisme cool, gagner un prix, etc.), vous pouvez ignorer cette section. Les critères de classement se trouvent généralement dans la section Tournoi du [manuel](#) et sont généralement basés sur les points de classement (RP), gagnés pour les victoires ou les égalités et pour réussir des objectifs de jeu spécifiques. En supposant un jeu typique en Compétition de robotique *FIRST*® avec un classement suivant ce style, posez-vous les questions suivantes :

1. Outre les matchs gagnés, comment les alliances peuvent-elles gagner des points de classement ?
2. Pour chacun de ces points de classement RP, quelles capacités du robot sont nécessaires pour :
 - a. Maximiser notre capacité en tant qu’alliance à gagner le point de classement ?
 - b. Obtenir le point de classement en jouant avec 2 autres robots comme le nôtre ?
 - c. Faire la contribution minimale pour que notre alliance obtienne le point de classement ?
3. Quels sont les 1^{er} et 2^e bris d’égalité après les points de classement (il est généralement peu probable d’avoir une égalité au-delà de cela et très peu probable qu’une telle égalité décide de plus d’un ou deux rangs) ? Ceux-ci ne devraient pas exercer une influence majeure dans la sélection des capacités, mais peuvent faire pencher la balance dans le cas d’une décision difficile entre deux capacités lors de la priorisation.

3.1.3 Analyser le jeu - Gagner des matchs ?

Si les objectifs de votre équipe impliquent de gagner des matchs, vous devrez passer du temps à analyser le jeu et à réfléchir à ce à qui fait une alliance gagnante. Selon vos objectifs exacts, certaines de ces questions peuvent être plus ou moins pertinentes.

1. Répertoirez toutes les tâches pour enregistrer un score dans le jeu avec les valeurs de points.
 - a. Combien de fois une alliance peut-elle accomplir chacune de ces tâches ? La réponse est peut-être complexe, certains jeux peuvent avoir des tâches qui s’excluent mutuellement.
2. Pour chaque tâche, estimez à quelle vitesse vous pensez que la tâche peut être accomplie.
 - a. Les réponses varieront fortement en fonction du robot. Vous voudrez probablement choisir une « classe » particulière de robot et répondre de manière cohérente avec ce niveau. Pour enregistrer un score impliquant une pièce de jeu, n’oubliez pas d’inclure le temps pour aller chercher cette pièce de jeu et de revenir à l’emplacement pour marquer.

3. Pour certaines tâches, vous pouvez également envisager un « taux de réussite ». C’est le pourcentage d’occasions que vous pensez que le robot terminera la tâche dans le temps que vous avez estimé.
 - a. Pour une tâche de tir, cela peut paraître évident : à quelle fréquence la pièce de jeu ira dans le but. Pour d’autres tâches, cela peut être moins évident; mais essayez de considérer à quelle fréquence vous devrez peut-être vous réaligner et réessayez, par exemple. Exprimez la réponse sous forme décimale ou de fraction (p. ex. taux de réussite de 80 % = 0,8)
4. Utilisez les évaluations des trois étapes précédentes pour calculer les points /seconde pour chaque tâche contribuant au score (succès * points / seconde) et trier du plus efficace au moins efficace.

En fonction du temps et de la liste d’efficacité que vous avez dressée, évaluez ce qui suit (ignorez tout ce qui n’est pas approprié aux objectifs de votre équipe) :

1. À quoi ressemble un match pour une alliance gagnante moyenne en qualifications ?
2. À quoi ressemble un match pour une alliance qui gagnerait environ 80% des matchs de qualification ?
3. À quoi ressemble un match pour une alliance qui peut gagner un match éliminatoire ?
4. À quoi ressemble un match pour une alliance qui gagne le tournoi ?

À l’aide des objectifs de vos équipes et d’une évaluation de vos capacités, estimez votre rôle dans chacune de ces alliances ci-dessus:

1. Combien de points essayez-vous de contribuer pour l’alliance ?
2. Y a-t-il des tâches spécifiques dans l’alliance que vous pensez devoir contribuer ?
3. Y a-t-il des tâches spécifiques que vous souhaitez laisser aux partenaires de l’alliance ?

3.2 Identification des améliorations

Comparez maintenant les capacités du KitBot à la liste des tâches et des totaux de points que vous venez de générer en fonction des objectifs de votre équipe. Quelles capacités devez-vous ajouter afin de répondre aux points et aux capacités identifiés ? Laquelle d’entre-elles est la plus importante?

3.3 Prochaines étapes

Si vous envisagez des améliorations avant de construire votre KitBot, passez à la [Section 5](#) pour quelques idées sur la façon de transformer ces idées en un plan. Si vous avez déjà construit votre KitBot, passez à la [Section 4](#) pour plus d’informations sur la façon de tester efficacement votre robot et d’identifier d’autres améliorations potentielles à inclure dans votre processus de planification.

4 Itération – Amélioration des capacités existantes

L’itération est un élément central du processus de conception technique et la conception d’un robot *FIRST* n’est pas différente. Une fois que vous avez construit votre KitBot, vous voudrez utiliser tout le temps précédant, ou entre vos événements, pour faire deux choses : pratiquer et itérer. Pour plus d’informations sur la sélection et la pratique des pilotes, consultez les documents « Guide to Selecting Drivers » et « Improving Driver Performance » sur la page Web [Technical Resources](#) dans la section « Other Technical Resources ».

Un processus simple d’itération est décrit ci-dessous et comprend les étapes suivantes :

- Tests
- Idéation
- Planification
- Exécuter

4.1 Tests utiles

La première étape d’un processus d’itération consiste à identifier les domaines à améliorer. Pour votre KitBot, nous vous recommandons de le faire via des tests dédiés. La majorité de ces tests impliqueront l’utilisation de votre robot comme vous vous y attendez pendant un match, ce qui servira également de pratique aux pilotes. La seule exception est le test d’échec. À condition que vous ayez suffisamment de temps (nous vous recommandons au moins une semaine) avant votre événement, vous pouvez intentionnellement tester votre robot dans des situations compromettantes telles que se lancer vers des objets de manière agressive, acquérir trop de pièces de jeu, appuyer sur des boutons dans un ordre incorrect, etc. pour identifier ce qui va flancher. Pendant que vous testez le robot, notez les réponses aux questions suivantes :

1. Qu’est-ce qui s’est cassé ? Comment cela s’est-il cassé ?
2. Que fait le robot de manière incohérente (par exemple, ne pas acquérir de pièces de jeu, laisser tomber des pièces de jeu, manquer des occasions de marquer, etc.) ?
3. Quelles tâches les pilotes prennent beaucoup/trop de temps à accomplir (par exemple, acquérir des pièces de jeu, traverser le terrain, s’aligner pour marquer, libérer la pièce de jeu, etc.) ?

Si vous avez plus d’une pièce de jeu disponible, vous voudrez peut-être faire une comparaison minutieuse des performances du robot avec chaque pièce de jeu pour voir s’il y a une variation. Ensuite, prenez une pièce de jeu ou deux et changez-les similairement à ce que vous vous attendez pendant le jeu (sur ou sous-gonflé, déformé, éraflé ou déchiré, etc.). Répétez le test du robot et documentez tout changement de comportement.

4.2 Faire un remue-méninges sur les améliorations possibles

Maintenant que vous avez identifié les domaines d’amélioration potentiels, la prochaine étape consiste à générer des idées pour améliorer ces faiblesses. Ce document divise le brainstorming en deux morceaux, mécanique et logiciel /électrique. Ceci est fait pour aider à rendre le processus un peu plus facile à gérer, et parce que ces éléments nécessitent souvent différents ensembles de ressources, ils peuvent souvent être poursuivis en parallèle.

4.2.1 Réfléchir aux améliorations – Mécanique

Pour chacun des domaines d’amélioration potentielle identifiés dans vos tests, réfléchissez à certains changements mécaniques potentiels qui pourraient améliorer les performances. Quelques questions initiales pour chacune des trois catégories sont incluses ci-dessous :

4.2.1.1 Qu’est-ce qui s’est cassé ?

1. Y a-t-il des moyens de changer la géométrie de la pièce qui s’est cassée pour la rendre plus forte ?
2. Y a-t-il d’autres matériaux que nous pourrions utiliser qui renforceraient la pièce sous la même géométrie ?
 - a. N’oubliez pas que plus fort ne signifie pas toujours plus rigide, parfois faire une pièce à partir d’un matériau plus flexible (par exemple, passer de l’aluminium au polycarbonate) peut lui permettre d’absorber l’énergie sans atteindre une déformation permanente.
3. Voulons-nous réellement que la pièce soit plus forte, ou voulons-nous faire des pièces de rechange ?
 - a. Si la cause de l’échec semble anormale et ne peut se produire que rarement dans un tournoi, le cas échéant, vous voudrez peut-être laisser la pièce telle quelle, ou parfois même la rendre légèrement plus faible. Bien que cela puisse sembler contre-intuitif au premier abord, concevoir un maillon faible connu et être prêt à le remplacer si des dommages se produisent peut aider à prévenir des dommages inattendus à des pièces plus coûteuses ou plus difficiles à remplacer. Les goupilles de cisaillement mécanique et les fusibles électriques sont quelques exemples de ces pièces qu’on s’attend à sacrifier.

4.2.1.2 Qu’est-ce qui est incohérent ?

1. Existe-t-il des moyens de modifier la géométrie des pièces pour améliorer la cohérence ?
 - a. Des idées de solutions possibles sont l’ajout ou la suppression de contraintes ou de guides, l’augmentation ou la diminution de la compression sur une pièce de jeu, la suppression de fentes ou d’espaces avec lesquels une pièce de jeu peut interagir par inadvertance, ou l’ajout de roues ou de rouleaux supplémentaires connectés à un système existant.
2. Y a-t-il des façons de modifier les matériaux pour améliorer la cohérence ?

- a. Les réflexions en ce sens tendent généralement à augmenter ou à réduire la flexibilité ou la friction
3. Y a-t-il des façons d'augmenter la cohérence par l'ajout d'actionneurs supplémentaires ?
 - a. À titre d'exemple, mentionnons l'élargissement actif d'un mécanisme d'acquisition de pièces de jeu, l'ajout de loquets ou de portes supplémentaires sur le parcours d'une pièce de jeu, l'ajout d'un actionneur supplémentaire pour contrôler l'orientation des pièces de jeu dans un système, etc.

4.2.1.3 *Qu'est-ce qui prend beaucoup de temps ?*

1. Y a-t-il des guides mécaniques qui peuvent être ajoutés pour accélérer l'exécution de la tâche par les pilotes ?
 - a. Des exemples incluent des coins ou des entonnoirs qui interagissent avec des éléments du terrain, ou un marqueur physique sur le robot qui aide les pilotes à mieux s'aligner avec un élément du terrain.
2. Existe-t-il des moyens de rendre le robot plus robuste face au désalignement ?
 - a. Des exemples incluent l'augmentation de la taille de l'ouverture pour l'acquisition de pièces de jeu ou l'analyse des géométries d'approche pour marquer en fonction de différents alignements.
3. La question est de savoir combien de temps le mécanisme prend pour mener à bien l'action ? Peut-il être accéléré en changeant le rapport de vitesse ?
 - a. Si vous envisagez d'accélérer un mécanisme, vérifiez combien de courant il requiert dans sa configuration actuelle. Le courant évoluera approximativement linéairement avec le ratio de vitesse. Si l'accélération du mécanisme pousse le courant à des niveaux préoccupants, vous devrez peut-être ajouter un autre moteur également.

4.2.1.4 *Idées spécifiques pour le KitBot 2024*

Vous trouverez ci-dessous quelques suggestions spécifiques pour des améliorations mécaniques potentielles au KitBot 2024. Ignorez cette section si vous préférez vous en tenir aux idées émises par votre équipe.

1. **Ajouter un blindage pour éviter la possession de note par inadvertance** - Il existe des endroits sur le KitBot où une note pourrait se coincer par inadvertance (en raison d'un alignement incorrect lors de l'acquisition, d'un rebond sur le haut-parleur, etc.) et qui compterait comme votre seule note autorisée à contrôler. Un écran pourrait être ajouté à ces emplacements pour réduire cette probabilité.
2. **Comblent l'espacement lors de l'acquisition de l'anneau** - Pour garder les choses simples, le KitBot acquiert les notes avec une rampe complètement passive. Cela laisse un espace entre la rampe et le mur car la rampe doit s'arrêter à la limite du cadre périphérique. Un mécanisme pourrait être actionné pour combler cet espace après le début du match, prolongeant la rampe au-dessus des pare-chocs. Vous voudrez probablement vous arrêter

au bord ou légèrement avant le bord du pare-chocs pour permettre au pare-chocs de continuer à absorber les impacts avec le mur.

3. **Réduire la rotation du lanceur.** – Le lanceur du KitBot a besoin d’une seconde pour tourner avant d’être prêt à lancer. Vous pouvez réduire ce temps en augmentant la puissance motrice appliquée à la roue de lancement en passant à un moteur plus puissant ou en ajoutant un deuxième moteur pour mouvoir la roue. Vous pouvez également atténuer cela par un changement au programme modifiant les commandes pour permettre à votre pilote de démarrer la rotation pendant que le robot est encore en prise de position, puis de lancer uniquement lorsqu’il est entièrement prêt.

4.2.2 Réfléchir aux améliorations – Électrique/programmation

Les améliorations électriques et logicielles sont regroupées car elles vont souvent de pair. Apporter des améliorations logicielles à un robot nécessite souvent l’utilisation de signaux de capteurs pour permettre au logiciel de réagir à ce qui se passe. Tout comme le remue-méninges en mécanique, passez en revue chacun de vos domaines d’amélioration potentielle et réfléchissez à la façon dont la rétroaction ou l’automatisation des logiciels pourrait apporter une amélioration.

4.2.2.1 *Qu’est-ce qui s’est cassé ?*

1. L’échec aurait-il pu être évité par l’établissement de limites de déplacement pour un mécanisme?
 - a. Ce sont les solutions les plus souvent implémentées à l’aide d’interrupteurs de limite pour indiquer les limites de course autorisée. Une alternative consiste à utiliser des « limites douces » pour les mécanismes qui ont déjà une rétroaction de position absolue (c’est-à-dire que si vous savez que votre mécanisme est à 30 degrés, vous pouvez limiter par le logiciel le mouvement dans cette direction).
2. L’échec aurait-il pu être évité avec les limites actuelles
 - a. Les limites de courant peuvent aider à réduire la force maximale qu’un mécanisme peut produire et peuvent limiter la surchauffe d’un moteur si un moteur est bloqué en raison d’un blocage inattendu du mécanisme.
3. L’échec aurait-il pu être évité par l’interverrouillage logiciel ?
 - a. Parfois, les robots ont des mécanismes qui ne devraient être actionnés que lorsque le robot est dans un certain état, sinon quelque chose pourrait se casser. Ces exclusions peuvent être appliquées dans le programme pour aider à empêcher les pilotes de commettre une erreur et d’endommager le robot.

4.2.2.2 *Qu’est-ce qui est incohérent ?*

1. Les contrôles sont-ils intuitifs et évidents ?
 - a. Si les pilotes commettent des erreurs, ils peuvent avoir besoin de plus de pratique, mais cela pourrait également signifier que les contrôles ne sont pas intuitifs. Une fois que vous avez sélectionné vos pilotes, assurez-vous de travailler avec eux pour vous assurer

que les contrôles font du sens pour eux. Ce qui peut être évident pour une personne peut être étrange pour une autre. Un exemple de design inadéquat peut être l’utilisation d’un seul bouton pour basculer un état à l’autre alors que les pilotes ne peuvent pas le constater. Explicitement, si le fait d’appuyer sur un seul bouton a activé la roue du lanceur et que le fait d’appuyer dessus à nouveau l’arrête, comment savez-vous si vous avez appuyé et comment se souvenir de l’éteindre ?

2. Y a-t-il des problèmes de synchronisation que l’automatisation logicielle pourrait aider à prévenir ?
 - a. Parfois, l’incohérence vient du fait que les pilotes font quelque chose au mauvais moment. L’automatisation de ce timing par logiciel peut aider en supprimant l’incohérence humaine du processus. Un exemple de ceci est le lanceur de notes. Afin de fonctionner efficacement, la roue avant doit être lancée un peu avant que la roue arrière ne pousse la note vers l’avant. Plutôt que de demander aux pilotes de synchroniser cela, le programme gère cette synchronisation automatiquement en contrôlant l’ensemble de la séquence avec un seul bouton à appuyer et maintenir.
3. Y a-t-il des séquences qui pourraient être gérées automatiquement par le programme ?
 - a. Parfois, pour effectuer une action, un robot doit actionner plusieurs mécanismes dans un ordre spécifique, avec ou sans synchronisation particulière. L’automatisation logicielle peut-elle être utilisée pour appliquer cette séquence ou automatiser l’ensemble afin d’éviter les erreurs ?
4. Existe-t-il des verrouillages logiciels qui pourraient aider à prévenir les problèmes ?
 - a. L’incohérence peut parfois provenir de l’exécution d’une action lorsque le robot n’est pas dans le bon état (non aligné avec le terrain, mécanisme d’interaction mal préparé, etc.). Le programme peut-il être utilisé pour détecter ces états incorrects et empêcher le pilote d’effectuer l’action ?
5. Le comportement du robot est-t-il incohérent même avec des actions vraisemblablement correctes des pilotes ?
 - a. Parfois, cette incohérence provient d’un changement dans le robot ou son environnement tels qu’une tension de batterie différente ou une friction accrue à mesure qu’une pièce s’use. L’ajout de capteurs et de contrôles de conditions dans le programme peut aider à garantir que le robot reproduise les mêmes actions à chaque fois.

4.2.2.3 *Qu’est-ce qui prend beaucoup de temps ?*

1. L’alignement peut-il être automatisé ?
 - a. Le moyen le plus courant est d’utiliser la détection visuelle pour détecter l’emplacement du robot sur le terrain et s’en servir pour diriger le robot vers l’emplacement souhaité, mais en fonction de la tâche, d’autres capteurs tels que les télémètres ou des détecteur de présence peuvent également être utiles.
2. Peut-on fournir plus d’informations aux pilotes ?
 - a. Les pilotes peuvent faire de leur mieux avec leurs propres yeux pour voir ce qui se passe, mais le robot peut également aider en transmettant des signaux

supplémentaires. Le signal d’une caméra vers le poste de pilotage, l’affichage des états du robot sur le tableau de bord, l’utilisation des lumières sur le robot pour signaler son état et l’ajout de rétroaction par vibration sur les contrôleurs (le cas échéant) sont tous des exemples de façons de fournir des informations sur l’état du robot aux pilotes.

4.2.2.4 Idées spécifiques pour le KitBot 2024

Vous trouverez ci-dessous quelques suggestions spécifiques pour une automatisation logicielle potentielle du KitBot 2024. Ignorez cette section si vous préférez vous en tenir aux idées générées par l’équipe.

1. **Contrôle du lanceur en boucle fermée** – En remplaçant le moteur alimentant la roue avant du lanceur pour un moteur avec encodeur intégré (Venom, NEO / Vortex, Falcon, Kraken) ou en ajoutant un encodeur directement ou indirectement sur un moteur existant, vous pouvez passer du contrôle actuel en boucle ouverte (estimation de la vitesse de la roue en fonction de la tension appliquée et du temps) à un contrôle en boucle fermée (contrôle précis de la vitesse de la roue en fonction de la rétroaction). Pour obtenir des informations de vitesse à partir du capteur choisi, vous devrez consulter la documentation du moteur ou du capteur. Pour utiliser ces informations pour contrôler la vitesse des roues, consultez la section [Advanced Controls Introduction](#) de WPILib, en particulier l’article « Tuning a Flywheel Velocity Controller ».
2. **Détection des notes** – Un interrupteur, un capteur de proximité ou un capteur de présence peuvent être ajoutés au robot pour détecter quand une note est dans le robot. Vous pouvez utiliser ces informations pour arrêter automatiquement l’action interne des roues afin de réduire le chauffage du moteur et vous pourriez fournir des informations aux pilotes via des lumières, le tableau de bord ou une vibration de la manette (si vous utilisez un contrôleur avec vibration telle qu’une manette Xbox). Cela peut également réduire la probabilité d’acquiescer plusieurs notes en même temps.
3. **Alignement par la vision** – Le KitBot peut être facilement aligné avec le haut-parleur en raison de la grande ouverture, la proximité de vos pilotes, et la possibilité de presser les pare-chocs contre le caisson des basses pour aligner la profondeur. S’aligner avec la source peut être plus difficile en raison de la distance des pilotes et de l’étroitesse des fentes sur la source par rapport au haut-parleur. L’ajout d’une caméra traitant les [AprilTags](#) peut être utile pour s’aligner sur l’un des deux emplacements extérieurs sur la source. Vous pouvez le faire en ajoutant une webcam branchée sur le roboRIO ou en ajoutant un système de vision externe comme un [Limelight](#), [RaspberryPi with WPILibPi](#) ou [co-processor board with PhotonVision](#). Une fois que vous avez les données de l’[AprilTag](#), vous pouvez les utiliser pour aligner le robot sur la source en :
 - a. En demandant au pilote de positionner le robot près de la source et de pointer dans la bonne direction de sorte que vous soyez sûr que la balise soit en vue.
 - b. En demandant au pilote d’appuyer sur un bouton et de le maintenir enfoncé pour lancer le ciblage par la vision.
 - c. Pendant que le bouton est maintenu, contrôlez la rotation de la plateforme pilotable en utilisant l’emplacement de la balise. Vous pouvez probablement utiliser un contrôle

- proportionnel simple pour vous approcher suffisamment. Comparez l'emplacement de la balise au centre de l'image, multipliez le résultat par une constante que vous calibrez et appliquez le résultat à la rotation de la plateforme pilotable (appliquez le résultat comme valeur de rotation de la fonction d'entraînement arcade ou la moitié de la valeur avec des signes opposés de chaque côté d'une fonction d'entraînement « tank »).
- d. Permettre au pilote de garder le contrôle de la propulsion de la plateforme pilotable en continuant à appliquer la valeur du joystick au mouvement avant / arrière d'une fonction d'entraînement d'arcade, ou en ajoutant la valeur du joystick à la valeur AprilTag pour un entraînement style « tank » (pour un entraînement « tank », vous devrez également détecter si l'une ou l'autre valeur dépasse 1 et réduire les deux côtés si c'est le cas).
 - e. Le pilote peut ensuite diriger le robot vers la source tandis que le robot se maintient orienté vers la balise. Lorsque les pare-chocs atteignent le mur de la source, le centre du robot doit être approximativement aligné avec le centre de la fente. Le robot n'est peut-être pas encore aligné au ras du mur. Continuez à pousser le robot vers le mur de la source et les pare-chocs devraient finalement s'appuyer complètement au mur. Vous pouvez également potentiellement relâcher le bouton d'alignement et appliquer un petit tour dans la direction appropriée pour faciliter cette partie du processus.

Un processus semblable pourrait être utilisé pour s'aligner au haut-parleur. Ou, pour une utilisation plus avancée, vous pouvez utiliser les AprilTags pour obtenir des informations de position complètes pour votre robot et aligner ou planifier un chemin n'importe où sur le terrain !

4.3 Prochaines étapes

Si vous avez déjà envisagé des améliorations avant de construire votre robot, décidez si vous souhaitez faire une mise à jour du remue-méninges sur ce sujet pour ajouter à votre pool d'itérations, ou si vous avez déjà une liste d'idées restantes à reconsidérer. Si vous n'avez pas encore envisagé d'améliorations, rassemblez votre liste complète d'idées des deux exercices et passez à la [Section 5](#) pour obtenir des conseils sur la façon de les prioriser.

5 Choisir – Transformer les idées en un plan

Si vous envisagez des améliorations avant de construire votre KitBot, vous n’avez peut-être pas encore d’idées d’itération. Continuez cet exercice avec seulement les améliorations. Vous pourrez toujours revenir et le refaire avec des idées d’amélioration restantes ainsi que de nouvelles idées d’itération après avoir construit et testé votre robot. Si vous avez déjà construit votre robot, nous vous recommandons de combiner les idées de nouvelles fonctionnalités avec les idées d’itération en un seul exercice de sélection.

Transformer votre liste d’améliorations possibles en une liste d’actions hiérarchisées est certes un exercice difficile qui implique de nombreux facteurs. Il existe de nombreuses façons d’aborder l’équilibre entre les avantages perçus de chaque changement et la difficulté de le mettre en œuvre et les ressources de l’équipe qui peuvent être nécessaires pour le mettre en œuvre. Une méthode possible est présentée ci-dessous.

5.1 Établissement des priorités

En vous appuyant sur votre analyse du jeu et votre vision du rôle de votre robot dans l’alliance, priorisez vos améliorations potentielles de la plus pertinente à la moins pertinente. Ne vous inquiétez pas encore de ce qui est facile et de ce qui est difficile, ou de ce qui utilise quelles ressources. Créez simplement une « liste de rêve » ordonnée d’améliorations du robot. Voici un exemple:

1. Nouvelle capacité de fantaisie du robot
2. Chose automatique qui aide à gagner des points de classement
3. Chose en fin de partie qui aide à gagner des points de classement
4. Modification visant à améliorer la précision pour marquer
5. Faire des pièces de rechange d’un composant

5.2 Analyse – Estimation des ressources et de la portée

Dans cette section, vous allez examiner votre liste et évaluer les ressources nécessaires pour exécuter chaque amélioration. Plutôt que de simplement attribuer un concept ou un nombre avec la difficulté d’une idée, essayez de créer quelques catégories représentant les diverses ressources qui pourront être nécessaires. Certaines de ces catégories peuvent représenter des limites de ressources physiques telles que la disponibilité du matériel, la disponibilité d’une machine ou le budget pour acheter de nouveaux composants / matériaux et certaines peuvent représenter des ressources plus abstraites comme le temps total / la capacité des élèves et des mentors dans un aspect spécifique de l’équipe.

Un exemple de catégories serait : Conception, Fabrication, Logiciels, et Coût des matériaux. Pour chaque idée, évaluez le coût des ressources pour chacune de ces catégories et attribuez une valeur 0-5 (n’hésitez pas à utiliser l’échelle de votre choix), 0 n’étant pas une ressource requise et 5 étant un

besoin de ressources très important. Pour prioriser, utilisez le rang que vous avez créé à l'[Étape 5.1](#), 1 à N (ce n’est pas grave si c’est une échelle différente des valeurs de catégorie).

Pour un nouveau mécanisme, idéalement, vous pourriez trouver des chiffres approximatifs sans avoir besoin de réfléchir et sélectionner un mécanisme spécifique que vous utiliseriez pour terminer la tâche, mais pour certaines équipes, cela pourrait être difficile. Vous devrez peut-être faire un détour à ce stade pour faire un remue-méninges ou prototyper des idées de mécanismes spécifiques afin d’évaluer leurs besoins en ressources, consultez la [Section 5.5](#) pour des ressources supplémentaires.

Un exemple fictif est fourni dans le Tableau 1.

Tableau 1: Exemple de ressources pour l'amélioration

Amélioration	Priorité	Design	Fabrication	Programmation	Matériaux
Nouvelle capacité de fantaisie du robot	1	5	3	3	3
Chose automatique qui aide à gagner des points de classement	2	0	1	3	0
Chose en fin de partie qui aide à gagner des points de classement	3	3	2	1	1
Modification visant à améliorer la précision pour marquer	4	1	3	0	1
Faire des pièces de rechange d’un composant X	5	0	2	0	2

5.3 Analyse – Définissez votre « budget »

Maintenant que vous avez défini les coûts relatifs des ressources liées à vos idées, vous devez établir un « budget » décrivant les capacités de votre équipe dans chaque catégorie. Pour chacune de vos catégories, en utilisant les coûts attribués comme référence pour l’échelle, tenez compte des ressources de votre équipe d’ici votre tournoi. Assurez-vous de considérer le temps dont vous aurez besoin juste avant l’événement pour tester et pratiquer avec toutes les améliorations et préparer votre équipement pour l’événement. Affectez un budget global pour chaque catégorie qui représente vos ressources disponibles dans cette catégorie d’ici votre prochain événement.

À titre d’exemple, considérons une équipe avec 2 jeunes et un mentor en CAO, seulement 1-2 jeunes sans mentor en programmation, une entreprise qui aide à la fabrication de pièces en plus de l’équipement de l’atelier de l’équipe, et un stock substantiel de matières premières et de pièces COTS. Cette équipe pourrait décrire son budget pour les catégories comme suit :

- Design – 5
- Programmation – 3
- Fabrication – 10
- Matériaux – 10

5.4 Faire des plans

Maintenant que vous avez un budget et des coûts, il est temps de les rassembler dans un plan des idées à poursuivre. Une approche consiste simplement à parcourir la liste des priorités jusqu’à ce que le budget de n’importe quelle catégorie soit épuisé, puis à continuer d’ajouter les idées qui n’affectent que les catégories avec du budget restant.

Une autre approche, plus complexe, consiste à demander à des individus ou à de petits groupes de préparer des exemples de plans, puis d’évaluer ces plans en équipe et de sélectionner celui qui, selon l’équipe, donnera le meilleur robot. Cela peut parfois se traduire par une utilisation plus optimale des ressources.

Revenons sur notre exemple fictif ci-dessus avec notre équipe qui a fixé son budget de ressources de conception à 5 points (nous allons ignorer leur budget dans les autres catégories pour plus de simplicité). En utilisant la première approche, cette équipe choisirait d’ajouter une « nouvelle capacité de fantaisie », épuisant tout son budget de conception, puis « Faire quelques pièces de rechange du composant X ».

Tableau 2: Exemple de plan d’amélioration – Option 1

Amélioration	Priorité	Design	Fabrication	Programmation	Matériaux
Nouvelle capacité de fantaisie du robot	1	5	3	3	3
Faire des pièces de rechange d’un composant X	5	0	2	0	2

Ce plan apporte deux améliorations et utilise 5 points de ressources de fabrication, 3 de programmation et 5 de matériaux.

En utilisant la deuxième approche, quelqu’un pourrait suggérer de répartir les points de budget de conception sur les trois améliorations suivantes.

Tableau 3: Exemple de plan d’amélioration – Option 2

Amélioration	Priorité	Design	Fabrication	Programmation	Matériaux
Chose automatique qui aide à gagner des points de classement	2	1	1	3	0

Chose en fin de partie qui aide à gagner des points de classement	3	3	2	1	1
Modification visant à améliorer la précision pour marquer	4	1	3	0	1
Faire des pièces de rechange d’un composant X	5	0	2	0	2

Bien que ce plan passe à côté de l’élément avec la priorité individuelle la plus élevée, il apporte 4 améliorations au lieu de deux et peut mieux utiliser les ressources de fabrication (8 vs 5) et programmation disponibles (4 vs 1). L’équipe devrait alors peser la valeur de ces 4 améliorations par rapport aux 2 de l’autre plan et décider quelle approche produira le meilleur robot pour atteindre les objectifs de l’équipe.

5.5 Prochaines étapes

5.5.1 Améliorations – Nouveaux mécanismes

Pour ajouter de nouveaux mécanismes, l’étape suivante consiste à commencer à réfléchir aux implémentations possibles, puis à prototyper pour déterminer une orientation. Les ressources suivantes de la page Web [Technical Resources](#), sous « Mechanical Resources->General », peuvent être un point de départ utile:

- “Prototyping 101”
- “Prototyping Worksheet”
- “Mechanisms Worksheet”
- “NASA RAP Robotics Design Guide”

5.5.2 Améliorations – Capacités logicielles

Pour ajouter de nouvelles fonctionnalités dans le code, consultez le document « Control System and Programming Documentation » ainsi que la « Programming 101 » de la section « Software/Electrical Resources->Software » de la page Web [Technical Resources](#).

5.5.3 Plan du projet

La dernière chose que vous voudrez faire avant de vous lancer et de commencer le travail est d’esquisser un plan de projet approximatif. Cela peut être aussi simple ou aussi détaillé que vous le souhaitez, mais vous voulez probablement au moins quelques jalons d’ici votre prochain événement. Ceux-ci vous aideront à vérifier si vous êtes sur la bonne voie et vous permettront d’ajouter des ressources (telles que le temps de rencontre) ou de réduire les fonctionnalités avant d’atteindre un

point où vous vous présentez en tournoi avec un robot inachevé ou non testé. Coupez une fonctionnalité ou deux pour vous assurer que vous avez le temps de tester, et la pratique sera probablement rentable à la fin.

6 Annexe A – Exemple d’analyse de jeu de base, 2023

6.1 Exemple - Liste des tâches du robot

Vous trouverez ci-dessous un exemple de liste de tâches du robot (voir [Section 3.1.1](#)) pour le jeu CHARGÉ À BLOC présenté par Haas, de la Compétition de robotique *FIRST* 2023. Cette liste a été faite a posteriori, les équipes peuvent avoir eu plus ou moins de tâches sur leurs propres listes.

Façons pour un robot de marquer :

- Cône niveau supérieur
- Cône niveau médian
- Cône niveau inférieur
- Cube niveau supérieur
- Cube niveau médian
- Cube niveau inférieur
- Pont balancé

Scores détaillés :

- Pont balancé
 - Atteindre l’équilibre en conduisant sur le pont à partir du large bord
 - Atteindre l’équilibre du côté étroit d’un pont déjà équilibré

Façons d’acquérir des pièces de jeu :

- Sous-station double
 - Cubes
 - Cônes renversés
 - Cônes debout
- Sous-station simple directe
 - Cubes
 - Cônes renversés
 - Cônes debout
- Tapis
 - Cubes
 - Cônes renversés
 - Pointe première
 - Bride première
 - Cônes debout

L’acquisition de pièces de jeu du côté opposé du robot à partir de l’endroit où elles sont marquées peut accélérer les parcours, particulièrement important pendant la période Auto (peut avoir moins

d'importance pour les plateformes qui peuvent tourner tout en se déplaçant comme mécanum ou swerve).

Autres tâches :

- Donner des pièces de jeu aux partenaires ?

Tâches défensives :

- Acquérir des pièces de jeu de l'adversaire ? (Possibilité d'acquérir sans extension pour saisir légalement des pièces depuis la zone de chargement adverse)

6.2 Exemple - Comment bien se classer

Vous trouverez ci-dessous un exemple de réflexion sur la façon de bien se classer dans le jeu CHARGÉ À BLOC présenté par Haas en Compétition robotique *FIRST* 2023.

1. Outre remporter des matchs, comment les alliances peuvent-elles gagner des points de classement (RP) ?
 - a. RP de lien – Marquer au moins 5 liens ou au moins 4 liens avec co-op (cela fait référence aux règles originales, pas aux valeurs modifiées pour le Championnat)
 - b. RP de recharge – Marquer au moins 26 points à la station de recharge
2. Pour chacun de ces points de classement quelles capacités du robot sont nécessaires pour...
 - a. Maximiser votre capacité dans notre alliance à gagner le RP ?
 - i. Marquer 15 pièces de jeu mélangées au total sur au moins les rangées inférieure et médiane pour créer 5 liens
 - ii. Équilibrer la station de recharge en période Auto afin que seul un équilibre de 2 robots soit nécessaire dans la fin de partie pour sécuriser le RP
 - b. Obtenir le RP si vous jouez avec 2 autres robots comme le vôtre ?
 - i. Marquer ~ 5 pièces de jeu mélangées au total sur au moins les rangées inférieure et médiane pour créer 5 liens
 - ii. Équilibrer la station de recharge en période Auto afin que seul un équilibre de 2 robots soit nécessaire dans la fin de partie pour sécuriser le RP
 - c. Faire le minimum nécessaire pour que votre alliance gagne le RP ?
 - i. Marquez ~ 3-5 pièces de jeu au total au moins sur la rangée inférieure pour contribuer aux liens.
 - ii. Aidez à équilibrer la station de recharge en fin de partie pour contribuer aux points de la station de recharge.
3. Quels sont les 1^{er} et 2^e critères de bris d'égalité après les points de classement (il est généralement peu probable d'avoir une égalité au-delà de cela et très peu probable qu'une telle égalité décide de plus d'un rang ou deux) ? Ils ne devraient pas influencer grandement la sélection des capacités, mais ils peuvent fournir un argument dans le cas d'une décision difficile entre deux capacités lors de la priorisation.

- a. Points de station de recharge
- b. Points en Auto

6.3 Exemple - Comment gagner des matchs

Vous trouverez ci-dessous un exemple d’analyse des points/temps pour le jeu 2023. Pour simplifier légèrement, cette analyse ne concerne que la période Teleop et ignore Auto. On suppose également que les pièces de jeu marquées contribuent aux liens, ce qui peut ne pas être une hypothèse valide si vous choisissez une stratégie de cubes uniquement. Le temps et le taux de réussite utilisés sont des approximations et sont typiques d’un robot essayant de participer aux éliminatoires (~ 50^e centile) à un événement régional ou de district moyen.

Tableau 4: Exemple d’analyse des points au fil du temps

Action pour marquer	Points	Temps	Taux de réussite	Points/Sec
Cube - inférieur	2 + 5/3 (Lien)	22	95	.158
Cône - inférieur	2 + 5/3 (Lien)	30	95	.116
Cube - médian	3 + 5/3 (Lien)	27	90	.122
Cône - médian	3 + 5/3 (Lien)	35	85	.088
Cube - supérieur	5 + 5/3 (Lien)	30	85	.160
Cône - supérieur	5 + 5/3 (Lien)	38	80	.119
Balance 1 robot	10	15	90	.6
Balance 2 robots	20	20	85	.85
Balance 3 robots	30	25	80	.96

Ensuite, voici quelques réponses approximatives aux questions sur les alliances gagnantes.

1. À quoi ressemble un match pour une alliance gagnante moyenne en qualifications ?
 - 3x Mobilité = 9 points
 - Arrimage et Connexion Auto = 10 points
 - 3x Pièces de jeu en Auto = +3 Points
 - 2x Liens inférieurs = $2*5 + 6*2 = 22$ Points
 - 2x Liens supérieurs = $2*5 + 6*5 = 40$ Points
 - Arrimage et Connexion x2 Robots = 20 Points
 - Stationnement = 2 Points

Total = 106 Points

2. À quoi ressemble un match pour une alliance qui gagnerait environ 80% des matchs de qualification ?

3x Mobilité = 9 points
Arrimage et Connexion Auto = 10 points
4x Pièces de jeu en Auto = +4 Points
3x Liens inférieurs = $3*5 + 9*2 = 33$ Points
2x Liens supérieurs = $2*5 + 6*5 = 40$ Points
Arrimage et Connexion x2 Robots = 20 Points
Stationnement = 2 Points
Total = 118 Points

3. À quoi ressemble un match pour une alliance qui est en mesure de gagner un match éliminatoire ?

3x Mobilité = 9 points
Arrimage et Connexion Auto = 10 points
4x Pièces de jeu en Auto = +4 Points
3x Liens inférieurs = $3*5 + 9*2 = 33$ Points
2x Liens supérieurs = $2*5 + 6*5 = 40$ Points
Arrimage et Connexion x2 Robots = 20 Points
Stationnement = 2 Points
Total = 118 Points

4. À quoi ressemble un match pour une alliance qui gagne le tournoi ?

3x Mobilité = 9 points
Arrimage et Connexion Auto = 10 points
5x Pièces de jeu en Auto = +5 Points
3x Liens inférieurs = $3*5 + 9*2 = 33$ Points
1x Liens médians = $5 + 3*3 = 14$ Points
3x Liens supérieurs = $3*5 + 9*5 = 60$ Points
Arrimage et Connexion x3 Robots = 30 Points
Stationnement = 2 Points
Total = 162 Points

Enfin, en considérant une équipe qui essaie de faire des séries éliminatoires comme objectif, voici quelques réponses approximatives sur les objectifs potentiels de l'équipe :

1. Combien de points essayez-vous d'obtenir pour l'alliance ?
~30-40 points
2. Y a-t-il des tâches spécifiques que vous pensez devoir contribuer au sein de vos alliances?
S'arrimer et se connecter en mode Auto

3. Y a-t-il des tâches spécifiques que vous souhaitez laisser à vos partenaires d’alliance?
Non

6.4 Exemple - Améliorations

Supposons qu’un KitBot 2023 ait été conçu pour fournir des cubes en bas uniquement. En regardant le total de points et les tâches identifiées ci-dessus, l’ajout d’une routine en mode Auto : Dock and Engage Autonomous permettrait au robot de répondre aux tâches et aux objectifs de points identifiés. En regardant le tableau points / temps, la prochaine capacité à envisager d’ajouter serait probablement les cubes au niveau médian. C’est presque aussi efficace que les cubes au niveau bas (et ça pourrait être mieux, si ça pouvait être fait aussi rapidement que les cubes du bas) et donne à l’équipe 3 opportunités de score supplémentaires.