



A Study of the Computer Programming Language Implementation in Computer Engineering Students using the Flowgorithm Platform versus Common Programming

Somchit Bounpasert^{1*}, Kalaphath Kounlaxay², Vialakone Manivanh³, Southeva Simmavong³

¹Central Library, Souphanouvong University, Lao PDR

²Department of Computer Engineering, Faculty of Engineering, Souphanouvong University, Lao PDR

¹Correspondence:

Somchit Bounpasert, ¹Central Library,
Souphanouvong, Souphanouvong
University, Laos, 020 98009306,
somchitbounpasert@gmail.com

Abstract

The evaluation of students' programming abilities and performance using the Flowgorithm Platform revealed that the participants were first assessed on their basic programming knowledge through pre-tests and exercises. Afterward, instruction on the use of Flowgorithm was introduced, including examples of problem analysis and the design of flowcharts in a step-by-step process. Following the training sessions, students completed actual post-tests to evaluate their performance before and after the training.

Article Info:

Submitted: March 16, 2026

Revised: May 07, 2026

Accepted: May 27, 2026

The results showed that post-training test scores improved significantly compared to pre-training scores. Individual performance analysis was also conducted through exercises and tests involving problem analysis and flowchart design using the Flowgorithm software. The comparison of students' programming abilities and performance using the Flowgorithm Platform showed the following results the Sequence Flowchart problem-solving test, the average pre-training score was $X = 70.13$, while the post-training average score increased to $X = 87.23$. the Selection Flowchart problem-solving test, the average pre-training score was $X = 63.47$, while the post-training average score increased to $X = 81.17$. Loop Flowchart problem-solving test, the average pre-training score was $X = 64.77$, while the post-training average score increased to $X = 80.00$.

Keywords: Algorithm, Flowchart, Flowgorithm, Programming Language, Source Code

1. ພາກສະໜີ

ການຂຽນໂປຼແກຼມເບື້ອງຕົ້ນເປັນສ່ວນໜຶ່ງຂອງວິທະຍາສາດຄອມພິວເຕີ (Computer Science), ເປັນ ພື້ນຖານເພື່ອພັດທະນາທັກສະດ້ານຂໍ້ມູນຂ່າວສານເຕັກໂນໂລຊີ (Information Technology : IT), ໜຶ່ງໃນນັ້ນແມ່ນວິສະວະກຳຂໍ້ມູນຂ່າວສານ (Information Engineering) ທີ່ສະໜອງຄວາມສາມາດກ່ຽວກັບແນວຄວາມຄິດພື້ນຖານຂອງການຂຽນໂປຼແກຼມຄອມພິວເຕີເພື່ອພັດທະນາໂປຼແກຼມນຳໃຊ້ (Application) ໃນຂັ້ນຕໍ່ໄປ. ສິ່ງເລົ່ານີ້ເປັນໄປຕາມຄວາມຕ້ອງການຂອງ

ການປະຕິວັດອຸດສາຫະກຳ 4.0 ທີ່ສຸມໃສ່ການຮຽນຮູ້ເຕັກໂນໂລຊີຂໍ້ມູນຂ່າວສານ, ໂດຍສະເພາະແມ່ນຄວາມຊຳນານໃນການຂຽນໂປຼແກຼມເພື່ອຮອງຮັບ IoT (Internet of Things) ຫຼື ລະບົບດິຈິຕອນ (Digital) ເພື່ອໃຫ້ສາມາດກ້າວໄປສູ່ການແຂ່ງຂັນໃນສະຕະວັດທີ 21 (Arik et al., 2018). ເຖິງຢ່າງໃດກໍຕາມ, ການສອນ ແລະ ການຮຽນຂຽນໂປຼແກຼມກໍບໍ່ແມ່ນເລື່ອງງ່າຍ ແລະ ເປັນສິ່ງທ້າທາຍຫຼາຍ, ເຊິ່ງບາງຄັ້ງກໍອາດມີຄວາມຫຍຸ້ງຍາກໃນການຮຽນຮູ້ ຫຼື ການສອນການຂຽນໂປຼແກຼມໃນໄລຍະຕົ້ນ (Alajmi & Alkhatib, 2015).

ໃນການຂຽນໂປຼແກຼມຄອມພິວເຕີອາດຈະຂຽນໃນຮູບແບບພື້ນຖານຄຳສັບ (Text-based) ຫຼື ແຖວຄຳສັ່ງ (Command line) ໃນພາສານັ້ນໆໂດຍກົງ ເຊິ່ງອາດຈະແຕກຕ່າງຈາກ Flowgorithm ທີ່ເປັນໂປຼແກຼມຊ່ວຍໃນການສ້າງແຜນຜັງແບບ Flowchart ແລ້ວສາມາດປ່ຽນ Flowchart ຫຼື Algorithm ນັ້ນເປັນ Code ພາສາໂປຼແກຼມໄດ້ຢ່າງງ່າຍດາຍ. ວິທີການສອນ Algorithms ແລະ ການຂຽນໂປຼແກຼມເປັນສ່ວນຫນຶ່ງຂອງການຄິດ ແລະ ການຄຳນວນຖືວ່າຍັງເປັນບັນຫາທີ່ເປີດກວ້າງຢູ່ (Wolfram, 2016). ການນຳໃຊ້ Flowgorithm, ແມ່ນມີຮູບຮ່າງທີ່ສະແດງເຖິງການປະຕິບັດງານທີ່ແຕກຕ່າງກັນ ເພື່ອທຳຄວາມເຂົ້າໃຈກ່ຽວກັບ Algorithms ຫຼາຍກວ່າພາສາໂປຼແກຼມທົ່ວໄປທີ່ຈຳເປັນຕ້ອງເຂົ້າໃຈໂຄງສ້າງ ແລະ ຫຼັກການຂອງແຕ່ລະພາສາ. Flowgorithm ມີສະພາບແວດລ້ອມການຂຽນໂປຼແກຼມຕາມ Flowchart ຈຳນວນຫຼາຍສຳລັບການປັບປຸງຄວາມເຂົ້າໃຈ ແລະ ທັກສະການແກ້ໄຂບັນຫາຂອງນັກຂຽນໂປຼແກຼມ (Hooshyar et al., 2015, 2015). Algorithm ແມ່ນຂັ້ນຕອນໂດຍທີ່ຂັ້ນຕອນ ຫຼື ຊຸດຂອງກົດລະບຽບທີ່ຖືກອອກແບບເພື່ອປະຕິບັດວຽກງານສະເພາະໃດຫນຶ່ງ ຫຼື ແກ້ໄຂບັນຫາສະເພາະໃດຫນຶ່ງ (Zaretska, 2015). ມັນເປັນການກຳນົດໄວ້ຢ່າງດີ, ລຳດັບຈຳກັດຂອງຄຳແນະນຳ ຫຼື ການດຳເນີນງານທີ່ໃນເວລາທີ່ປະຕິບັດ ແລະ ໃຫ້ຜົນໄດ້ຮັບທີ່ຕ້ອງການ. Algorithms ເປັນພື້ນຖານສຳລັບໂຄງການຄອມພິວເຕີ ແລະ ຍັງຖືກນຳໃຊ້ໃນຂົງເຂດຕ່າງໆນອກເໜືອຈາກວິທະຍາສາດຄອມພິວເຕີ.

Flowgorithm ແມ່ນເຄື່ອງມືການຂຽນໂປຼແກຼມແບບກາຟິກທີ່ຊ່ວຍໃຫ້ຜູ້ໃຊ້ສ້າງຕາຕະລາງການໄຫຼເຂົ້າ ແລະ ປ່ຽນຟອກມັນໂດຍກົງເປັນພາສາການຂຽນໂປຼແກຼມຕ່າງໆ ເຊິ່ງຖືກອອກແບບມາເພື່ອຊ່ວຍໃຫ້ຜູ້ເລີ່ມຕົ້ນຮຽນຮູ້ແນວຄວາມຄິດການຂຽນໂປຼແກຼມໂດຍການເບິ່ງເຫັນການໄຫຼເຂົ້າຂອງການຄວບຄຸມໃນແຜນຜັງງານໂດຍຜ່ານການນຳໃຊ້ຕາຕະລາງການໄຫຼ. ຄຸນສົມບັດທີ່ສຳຄັນຂອງ Flowgorithm ປະກອບມີ: Graphical Interface ທີ່ສະໜອງການໂຕ້ຕອບກາຟິກ (Graphic) ທີ່ເປັນມິດກັບຜູ້ໃຊ້ທີ່ຜູ້ໃຊ້ສາມາດສ້າງ Flowcharts ໂດຍການລາກ ແລະ ວາງອົງປະກອບ. ການສະໜັບສະໜູນຫຼາຍພາສາ Flowgorithm ອະນຸຍາດໃຫ້ຜູ້ໃຊ້ສາມາດປ່ຽນຕາຕະລາງການໄຫຼຂອງເຂົາເຈົ້າເຂົ້າໄປໃນຫຼາຍພາສາການຂຽນໂປຼແກຼມ, ລວມທັງ Python, C, C++, Java, JavaScript ແລະອື່ນໆ. ຄຸນສົມບັດນີ້ຊ່ວຍໃຫ້ຜູ້ໃຊ້ປ່ຽນຈາກການສະແດງພາບໄປ Code ຕົວຈິງໃນພາສາການຂຽນໂປຼແກຼມທີ່ເຂົາເຈົ້າເລືອກ. ເຄື່ອງມືການສຶກສາ Flowgorithm ມັກຈະຖືກນຳໃຊ້ເປັນເຄື່ອງມືການສຶກສາໃນຫຼັກສູດການຂຽນໂປຼແກຼມເພື່ອສອນພື້ນຖານຂອງການອອກແບບ Algorithm ແລະ ເຫດຜົນການຂຽນໂປຼແກຼມ. ມັນເປັນທີ່ນິຍົມໂດຍສະເພາະໃນຫຼັກສູດການຂຽນໂປຼແກຼມໃນຮູບແບບ Visual Representation ຕາຕະລາງຂັ້ນຕອນທີ່ສ້າງຂຶ້ນໃນ Flowgorithm ເປັນຕົວແທນການໄຫຼເຂົ້າຂອງການຄວບຄຸມໃນຂັ້ນຕອນໂດຍຂັ້ນຕອນ, ເຮັດໃຫ້ມັນງ່າຍຂຶ້ນສຳລັບຜູ້ໃຊ້ທີ່ຈະເຂົ້າໃຈເຫດຜົນ ແລະ ໂຄງສ້າງຂອງ code. ຜັງຊັ້ນ (Function) ໃນຕົວ Flowgorithm ຕອບສະໜອງຊຸດຂອງຫນ້າທີ່ສ້າງໃນ ແລະ ການດຳເນີນງານທີ່ຜູ້ໃຊ້ສາມາດນຳໃຊ້ໃນຕາຕະລາງການໄຫຼຂອງມັນ, ເຮັດໃຫ້ມັນງ່າຍຕໍ່ການສ້າງ Algorithms

ແລະ ແກ້ໄຂບັນຫາ. ໃນຂະນະທີ່ Flowgorithm ເປັນເຄື່ອງມືທີ່ເປັນປະໂຫຍດສຳລັບການຮຽນຮູ້ ແລະ ການສອນແນວຄວາມຄິດການຂຽນໂປຼແກຼມ ເພາະເປັນສິ່ງສຳຄັນທີ່ຈະສັງເກດວ່າມັນມີຈຸດປະສົງຕົ້ນຕໍສຳລັບການສຶກສາ ແຕ່ອາດຈະບໍ່ເໝາະສົມສຳລັບການພັດທະນາຊອບແວ (Software) ຂະໜາດໃຫຍ່ (Kanis & Somkiat, 2006). ປົກກະຕິແລ້ວນັກຂຽນໂປຼແກຼມຍ້າຍໄປໃຊ້ສະພາບແວດລ້ອມການພັດທະນາປະສົມປະສານ (IDEs) ແລະ ການເຂົ້າລະຫັດຂໍ້ຄວາມຍ້ອນວ່າພວກເຂົາກ້າວຫນ້າໃນທັກສະການຂຽນໂປຼແກຼມແລ້ວ. ໂຄງສ້າງຊອບແວແມ່ນໄດ້ຮັບການຂຽນໂດຍນຳໃຊ້ IDE (Integrated Development Environment). ອີງຕາມພາສາການຂຽນໂປລແກລມ, ນີ້ຈະງ່າຍ ຫຼື ຍາກຫຼາຍສຳລັບຜູ້ເລີ່ມຕົ້ນ. Flowgorithm ແມ່ນພາສາການຂຽນໂປຼແກຼມທີ່ມີການເບິ່ງເຫັນ (Views) ທີ່ອະນຸຍາດໃຫ້ນັກຂຽນໂປຼແກຼມສາມາດເບິ່ງເຫັນຮູບສັນຍາລັກ Flowchart ເຂົ້າໃນຕາຕະລາງການໄຫຼເຂົ້າແລະຫຼັງຈາກນັ້ນປະຕິບັດພວກມັນເພື່ອກວດສອບຜົນໄດ້ຮັບ. ນອກຈາກນີ້ Flowgorithm ຍັງຮອງຮັບໄດ້ຫຼາຍພາສາ i18n. ພວກເຮົາສາມາດສ້າງ Flowcharts ໃນພາສາຕ່າງໆໄດ້.

ດັ່ງນັ້ນ, ການຂຽນໂປຼແກຼມເປັນລະບົບທາງການຂຽນຄຳສັ່ງເພື່ອສັ່ງໃຫ້ຄອມພິວເຕີເຮັດການຄຳນວນ. ມັນປະກອບດ້ວຍກົດລະບຽບວິທີ, ສັນຍາລັກ ແລະ ສົນທິສັນຍາທີ່ອະນຸຍາດໃຫ້ນັກຂຽນໂປຼແກຼມຂຽນ Code ແລະ ສ້າງຊອບແວ ໂດຍຈຸດປະສົງຕົ້ນຕໍຂອງການຂຽນໂປຼແກຼມແມ່ນເພື່ອໃຫ້ມະນຸດສາມາດສື່ສານຄຳແນະນຳກັບຄອມພິວເຕີໃນແບບທີ່ຄອມພິວເຕີທີ່ສາມາດເຂົ້າໃຈ ແລະ ປະຕິບັດຮ່ວມກັນໄດ້ (Ghazala, 2016). ພວກເຂົາເຈົ້າເຮັດຫນ້າທີ່ເປັນວິທີການສຳລັບນັກພັດທະນາໃນການຂຽນລະຫັດທີ່ສາມາດປະຕິບັດໄດ້ໂດຍຄອມພິວເຕີເພື່ອປະຕິບັດວຽກງານສະເພາະ (Robert, 2018). ມີພາສາການຂຽນໂປຼແກຼມຈຳນວນຫລາຍ, ແຕ່ລະຄົນມີ Syntax, Semantics ແລະ ກໍລະນີການນຳໃຊ້ຂອງຕົນເອງ. ນອກຈາກນີ້, ທາງເລືອກຂອງພາສາແມ່ນຂຶ້ນກັບຄວາມຕ້ອງການຂອງໂຄງການ, ຄວາມຊຳນານຂອງທີມງານພັດທະນາ ແລະ ປັດໃຈອື່ນໆ ເຊັ່ນ: ການປະຕິບັດ, ຂະໜາດ, ແລະ ການບຳລຸງຮັກສາ.

ຈາກການສຶກສາສະພາບບັນຫາການຂຽນໂປຼແກຼມຄອມພິວເຕີຂອງນັກສຶກສາຄະນະວິສະວະກຳເຫັນວ່າ ຍັງພົບບັນຫາຫຼາຍດ້ານ ໂດຍສະເພາະ ການຂຽນໂປຼແກຼມຄອມພິວເຕີເພື່ອການປະມວນຜົນ, ແນະນຳວິທີການ Flowgorithm ເພື່ອແປງ Flowcharts ຫຼື Algorithm. ເຊິ່ງຊ່ວຍໃຫ້ນັກຮຽນສາມາດຮຽນຮູ້ແນວຄວາມຄິດການຂຽນໂປຼແກຼມກ່ອນ ແລະ ນຳໃຊ້ການຈັດຕັ້ງການຮຽນການສອນໃຫ້ແກ່ນັກຮຽນໂດຍນຳໃຊ້ Flowgorithm ໃນການວິເຄາະ ແລະ ອອກແບບແຜນຜັງໂປຼແກຼມນັກຮຽນມີຄວາມຮູ້, ທັກສະ ແລະ ຜົນຄະແນນການທົດສອບພາຍຫຼັງເຂົ້າຝຶກ ເຊິ່ງການທົດສອບໄດ້ຮັບຜົນດີຂຶ້ນ.

ໃນໄລຍະທີ່ຜ່ານມາ ມີນັກຄົ້ນຄວ້າຫຼາຍທ່ານໃຫ້ຄວາມສົນໃຈສຶກສາຄົ້ນຄວ້າກ່ຽວກັບການພັດທະນາໂປຼແກຼມຄອມພິວເຕີ ເຊັ່ນ Jatupak, (2016) ໄດ້ດຳເນີນການຄົ້ນຄວ້າກ່ຽວກັບ ການພັດທະນາເກມຄອມພິວເຕີໂດຍໃຊ້ຕົວແບບ Abstract code Model (AMD) ພື້ນຖານສຳລັບການເພີ່ມທັກສະການແກ້ໄຂບັນຫາທາງຄະນິດສາດຂອງນັກ ຮຽນຊັ້ນ

ມັດທະຍົມຕອນຕົ້ນ. ເກມຄອມພິວເຕີຊ່ວຍເພີ່ມຄວາມສົນໃຈ ແລະ ສາມາດປັບປຸງທັກສະການຄິດຂອງ ຄົນ. ການຄົ້ນຄວ້ານີ້ມີຈຸດປະສົງເພື່ອ ພັດທະນາເກມຄອມພິວເຕີສໍາລັບການເພີ່ມທັກສະການແກ້ໄຂບັນຫາທາງ ຄະນິດສາດຂອງນັກຮຽນຊັ້ນມັດທະຍົມຕອນຕົ້ນ. ການນໍາໃຊ້ຕົວແບບ abstract ເປັນ code ພື້ນຖານ. Tipparat, (2017) ໄດ້ດໍາເນີນການ ຄົ້ນຄວ້າກ່ຽວກັບການພັດທະນາເກມ simulation ຄອມພິວເຕີເພື່ອ ສິ່ງເສີມຄວາມສາມາດໃນການຄິດຍຸດທະສາດຂອງນັກສຶກສາລະດັບ ມັດທະຍົມປາຍ ເຊິ່ງຈຸດປະສົງທົ່ວໄປຂອງການ ຄົ້ນຄວ້ານີ້ແມ່ນເພື່ອ ພັດທະນາເກມຄອມພິວເຕີຈໍາລອງ ສະຖານະການເພື່ອສິ່ງເສີມຄວາມ ສາມາດໃນການຄິດທາງ ຍຸດທະວິທີ ໂດຍມີຈຸດປະສົງສະເພາະ 4 ດ້ານຄື: 1) ເພື່ອສ້າງ ແລະ ຊອກຫາປະສິດທິພາບຂອງເກມ ຄອມພິວເຕີຈໍາລອງ ສະຖານະການ, 2) ເພື່ອປຽບທຽບຜົນການຮຽນກ່ອນ ແລະ ຫຼັງຮຽນກັບ ເກມ ຄອມພິວເຕີຈໍາລອງສະຖານະ ການ, 3) ປຽບທຽບຄວາມສາມາດໃນ ການຄິດຍຸດທະວິທີກ່ອນຮຽນ ແລະ ຫຼັງຮຽນດ້ວຍເກມຈໍາລອງຄອມພິວ ເຕີ, ແລະ 4) ເພື່ອສຶກສາຄວາມພໍໃຈຂອງນັກຮຽນທີ່ມີການຮຽນຮູ້ກັບ ເກມຄອມພິວເຕີຈໍາລອງສະຖານະການ. Kittipong, (2019) ໄດ້ ຄົ້ນຄວ້າວິໄຈເລື່ອງການພັດທະນາເກມເພື່ອການສຶກສາ ໃນວິຊາຄອມພິວ ເຕີ ສໍາລັບນັກຮຽນຊັ້ນມັດທະຍົມສຶກສາຕອນປາຍ, ການຄົ້ນຄວ້ານີ້ແມ່ນ ການພັດທະນາເກມການສຶກສາກ່ຽວກັບ ໂຄງການຄອມພິວເຕີສໍາລັບ ນັກຮຽນຊັ້ນຮຽນທີ 6. ຈຸດປະສົງແມ່ນເພື່ອ 1) ຊອກຫາປະສິດທິພາບ ໂດຍການນໍາໃຊ້ ເກມມາດຕະຖານ 80/80; 2) ເພື່ອປຽບທຽບຜົນການຮຽນ ລະຫວ່າງກ່ອນຮຽນ ແລະ ຫຼັງຮຽນ 3) ເພື່ອສຶກສາດັດຊະນີ ປະສິດທິຜົນ 4) ປະເມີນຄວາມພໍໃຈຂອງນັກຮຽນໃນການຮຽນຮູ້ໂດຍການນໍາໃຊ້ເກມ ການສຶກສາ ເຊິ່ງກຸ່ມຕົວຢ່າງ. ສະນັ້ນ, ໃນການຄົ້ນຄວ້າວິໄຈນີ້ແມ່ນສົມ ທຽບການດໍາເນີນການຂຽນພາສາໂປຼແກຼມຄອມພິວເຕີດ້ວຍໂຟຼກໂລດທິມ ເພຼດຟອມກັບການຂຽນໂປຼແກຼມທົ່ວໄປ ຈຸດປະສົງຂອງການຄົ້ນຄວ້າຄັ້ງນີ້ ເພື່ອ: 1) ເພື່ອການປະເມີນຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປ ແກຣມ Flowgorithm Platfrom. 2) ເພື່ອ ການປຽບທຽບຄວາມ ສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platfrom ແລະ ຊ່ວຍໃຫ້ນັກຮຽນສາມາດຮຽນຮູ້ແນວຄວາມຄິດການຂຽນໂປຼແກຼມກ່ອນ ທີ່ຈະລົງມືຂຽນໂປຼແກຼມຈິງ. ທູດຜ່ອນຄວາມສັບສົນກ່ຽວກັບຫຼັກໄວຍາ ກອນ ແລະ ຄວາມໝາຍສະເພາະຂອງພາສາໂປຼແກຼມ ເພື່ອສົມທຽບຄວາມ ແຕກຕ່າງລະຫວ່າງການນໍາໃຊ້ Flowgorithm ແລະ ບໍ່ ໃຊ້ Flowgorithm ເຂົ້າໃນການຂຽນຂຽນໂປຼແກຼມຄອມພິວເຕີຂອງຜູ້ສອນ ແລະ ຜູ້ຮຽນ. ເຊິ່ງຄໍາຖາມຂອງການຄົ້ນຄວ້າຄັ້ງນີ້ແມ່ນ ການປະເມີນ ຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປແກຣມ Flowgorithm Platfrom ສະແດງອອກຄືແນວໃດ ແລະ ເພື່ອການປຽບທຽບຄວາມ ສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platfrom ມີຄື ແນວໃດແດ່?

2. ອຸປະກອນ ແລະ ວິທີການ

2.1 ອຸປະກອນ

ອຸປະກອນທີ່ໃຊ້ໃນການຄົ້ນຄວ້າຄັ້ງນີ້ ປະກອບມີ ກ້ອງຖ່າຍຮູບ, ເຄື່ອງບັນທຶກສຽງ, ຄອມພິວເຕີ, ໂທລະສັບ ແລະ ປຶ້ມປຶກສ໌.

2.2 . ປະຊາກອນ ແລະ ກຸ່ມຕົວຢ່າງ

ປະຊາກອນ ທັງໝົດ 135 ຄົນ ໃນນັ້ນ ນັກສຶກສາພາກວິຊາວິສະ ວະກໍາຄອມພິວເຕີ, ຄະນະວິສະວະກໍາສາດ, ມະຫາວິທະຍາໄລສຸພານຸວົງ ຈໍາ ນວນ 95 ຄົນ. ນັກສຶກສາຈາກສາຂາວິຊາຕ່າງໆໃນວິທະຍາໄລ 2 ແຫ່ງ ຈໍາ ນວນ 40 ຄົນ (ວິທະຍາໄລເຕັກນິກວິຊາຊີບຫຼວງພະບາງ 20 ຄົນ ແລະ ວິ ທະຍາໄລເຕັກນິກວິຊາຊີບແບບປະສົມນໍ້າບາກ 20 ຄົນ). ໃນຈໍານວນ 135 ຄົນ ສາມາດຄັດເລືອກກຸ່ມຕົວຢ່າງໄດ້ ຈໍານວນ 95 ຄົນ.ໂດຍໃຊ້ວິທີສຸ່ມ ຕົວຢ່າງແບບເຈາະຈົງຂອງພິມໄພ ຫຼວງລາດ (2016:62) ມາເປັນ ແນວທາງໃນການຄັດເລືອກ.

2.3 ເຄື່ອງມືທີ່ໃຊ້ໃນການຄົ້ນຄວ້າ

2.3.1 ບົດສອນ ຫຼື ແຜນການສອນ

ເຄື່ອງມືທີ່ໃຊ້ໃນການຄົ້ນຄວ້າແມ່ນບົດສອນ ຫຼື ແຜນການສອນ ເພື່ອໃຊ້ໃນການທົດສອບທັງໝົດໃນແຕ່ລະຊຸດແມ່ນ 8 ຊຸມ (3 ແບບທົດ ສອບ ແລະ 8 ບົດເຝິກຫັດ) ແຕ່ເດືອນ 2 ຫາເດືອນ 4 ປີ 2025

2.3.2 ແບບທົດສອບກ່ອນ ແລະ ຫຼັງການສອນ

ເຄື່ອງມືທີ່ໃຊ້ໃນການຄົ້ນຄວ້າແມ່ນແບບທົດສອບກ່ອນ ແລະ ຫຼັງ ການ ເຊິ່ງກິດຈະກຳການສອນແມ່ນກະກຽມເນື້ອໃນບົດຮຽນ, ປະກອນ ເຄື່ອງມືກ່ຽວຂ້ອງ ແລະ ອອກແບບທົດສອບ, ປະເມີນຜົນການເຮັດແບບ ທົດສອບ ຫຼື ບົດເຝິກຫັດ ເປັນລາຍບຸກຄົນ, ແນະນຳເນື້ອໃນ, ເຄື່ອງມື ນຳ ໃຊ້ ແລະ ທົດສອບນັກຮຽນກ່ອນການເຝິກ, ມອບບົດເຝິກຫັດໃຫ້ຜູ້ເຂົ້າ ຮ່ວມເຝິກເພື່ອເປັນການ ທົດສອບ, ອະທິບາຍ ແລະ ສາທິດຫຼັກ ການຂຽນ ໂປຼແກຼມ. ການສຶກສາຄົ້ນຄວ້າຄັ້ງນີ້ເປັນການສຶກສາກ່ຽວກັບການນໍາໃຊ້ ເຄື່ອງມືເພື່ອຊ່ວຍເຂົ້າໃນການຂຽນໂປຼແກຼມ ແລະ ການປະມວນຜົນດ້ວຍ Platform Flowgorithm ທີ່ຖືວ່າເປັນເຄື່ອງມືທີ່ສາມາດນໍາໃຊ້ເຄື່ອງມື ເຂົ້າຊ່ວຍໃນການດໍາເນີນການຮຽນ-ການສອນວິຊາການຂຽນໂປຼແກຼມຂ ອງອາຈານສອນ ແລະ ນັກຮຽນອາດຈະເປັນທາງເລືອກໜຶ່ງເພື່ອຊ່ວຍຍົກ ລະດັບຄວາມຮັບຮູ້, ຄວາມເຂົ້າໃຈຕໍ່ການຂຽນ code ແລະ ການແປງຜັງງ ານ Flowchart ໃຫ້ຢູ່ໃນຮູບແບບ Source code ພາສາຕ່າງໆຂອງຄອມ ພິວເຕີ. ເຄື່ອງມືທີ່ໃຊ້ເຂົ້າໃນການຄົ້ນຄວ້າມີດັ່ງນີ້:

1. ແຜນການຮຽນຮູ້ກ່ຽວກັບການອອກແບບແຜນຜັງໂປຼແກຼມ (Program Flowchart)
2. ໂປຼແກຼມ Flowgorithm
3. ບົດເຝິກຫັດການປະເມີນທັກສະການຂຽນໂປຼແກຼ ແບບຟອມ ການປະເມີນທັກສະການອອກແບບ Program Flowchart

2.3.3 ການການຫາຄຸນນະພາບຂອງເຄື່ອງມື

ການທົດສອບຄວາມຖືກຕ້ອງທາງດ້ານເນື້ອໃນຂອງແບບ ທົດ ສອບ ຄະນະຜູ້ວິໄຈໄດ້ສົ່ງແບບ-ທົດສອບທີ່ສ້າງຂຶ້ນໃຫ້ຜູ້ຊ່ຽວຊານ 3 ທ່ານ ເພື່ອພິຈາລະນາ ແລະ ກວດສອບຄວາມຖືກຕ້ອງຂອງມັນ. ຄວາມຖືກຕ້ອງ ຂອງເນື້ອລວມທັງຄວາມເຫມາະສົມຂອງພາສາ ຂອງແຕ່ລະຄໍາຖາມ ທີ່ໃຊ້ ໃນແບບ ທົດສອບເພື່ອພິຈາລະນາຄວາມສອດຄ່ອງລະຫວ່າງຄໍາຖາມ ແລະ ຈຸດປະສົງນີ້ແມ່ນເຮັດໄດ້ໂດຍໃຊ້ດັດຊະນີຄວາມສອດຄ່ອງ ລະຫວ່າງຄໍາ ຖາມ ແລະ ຈຸດປະສົງ (Index of Item Objective Congruence: IOC). ຈາກການພິຈາລະນາຂອງຜູ້ຊ່ຽວຊານທັງ 3 ທ່ານ ມາລວບລວມຫາ

ຄຳດັດສະນີຄວາມສອດຄ່ອງລະຫວ່າງຄຳຖາມ ແລະ ຈຸດປະສົງ (Index of liers Objective) (Index of Item Objective Congruence: IOC)

ຈາກການຫາຄຸນນະພາບຂອງເຄື່ອງມື ຜູ້ວິໄຈນຳເອົາແບບສອບຖາມທີ່ສ້າງຂຶ້ນໃຫ້ຜູ້ຊ່ຽວຊານ 3 ທ່ານ ເພື່ອພິຈາລະນາ ແລະ ກວດສອບຄວາມຖືກຕ້ອງຂອງມັນ ສາມາດສະຫຼຸບຜົນໄດ້ດັ່ງນີ້:

- **ຂໍ້ມູນສ່ວນຕົວ** ກ່ຽວກັບພະນັກງານ ນັກສຶກສາ ພົບວ່າ ມີ ຄ່າ IOC ເທົ່າກັບ 1.00 ພົບວ່າ ແບບສອບຖາມທີ່ນຳໃຊ້ການຄົ້ນຄວ້າຄັ້ງນີ້ຖືກຕ້ອງ ສາມາດນຳໄປໃຊ້ໄດ້;

- ການປະເມີນຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປແກຣມ Flowgorithm Platform ພົບວ່າ ມີ ຄ່າ IOC ເທົ່າກັບ 1.00 ພົບວ່າ ແບບສອບຖາມທີ່ນຳໃຊ້ການຄົ້ນຄວ້າຄັ້ງນີ້ຖືກຕ້ອງ ສາມາດນຳໄປໃຊ້ໄດ້;

ການປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platform ພົບວ່າ ມີ ຄ່າ IOC ເທົ່າກັບ 1.00 ພົບວ່າ ແບບສອບຖາມທີ່ນຳໃຊ້ການຄົ້ນຄວ້າຄັ້ງນີ້ຖືກຕ້ອງ ສາມາດນຳໄປໃຊ້ໄດ້

2.4 ການເກັບກຳຂໍ້ມູນ

ການເກັບກຳຂໍ້ມູນຄັ້ງນີ້ ກຸ່ມຜູ້ວິໄຈໄດ້ລົງເກັບຂໍ້ມູນຕາມກຸ່ມເປົ້າໝາຍ 3 ແຫ່ງຄື: ນັກສຶກສາພາກວິຊາວິສະວະກຳຄອມພິວເຕີ, ຄະນະວິສະວະກຳສາດ, ມະຫາວິທະຍາໄລສຸພານຸວົງ ຈຳນວນ 95 ຄົນ. ນັກສຶກສາຈາກສາຂາວິຊາຕ່າງໆໃນວິທະຍາໄລ 2 ແຫ່ງ ຈຳນວນ 40 ຄົນ (ວິທະຍາໄລເຕັກນິກວິຊາຊີບຫຼວງພະບາງ 20 ຄົນ ແລະ ວິທະຍາໄລເຕັກນິກວິຊາຊີບແບບປະສົມນ້ຳບາກ 20 ຄົນ). ໂດຍສະເພາະແມ່ນຂໍຊົ່ວໂມງສອນນຳອາຈານປະຈຳວິຊາໃນຊົ່ວໂມງນັ້ນ ເຊິ່ງໃນການທົດສອບແມ່ນການນຳໃຊ້ເຄື່ອງມືເພື່ອຊ່ວຍເຂົ້າໃນການຂຽນໂປແກຣມ ແລະ ການປະມວນຜົນດ້ວຍ Platform Flowgorithm ທີ່ຖືວ່າເປັນເຄື່ອງມືທີ່ສາມາດນຳໃຊ້ເຄື່ອງມືເຂົ້າຊ່ວຍໃນການດຳເນີນການຮຽນ-ການສອນວິຊາການຂຽນໂປແກຣມຂອງອາຈານສອນ ແລະ ນັກຮຽນອາດຈະເປັນທາງເລືອກໜຶ່ງເພື່ອຊ່ວຍຍົກລະດັບຄວາມຮັບຮູ້, ຄວາມເຂົ້າໃຈຕໍ່ການຂຽນ code ແລະ ການແປງຜັງງານ Flowchart ໃຫ້ຢູ່ໃນຮູບແບບ Source code ພາສາຕ່າງໆຂອງຄອມພິວເຕີ ກ່ອນ ແລະ ຫຼັງການທົດສອບໃນຊົ່ວໂມງນັ້ນ.

2.5 ການວິເຄາະຂໍ້ມູນ

ການວິເຄາະຂໍ້ມູນ ຄັ້ງນີ້ແມ່ນນຳໃຊ້ Program SPSS (2022) ເພື່ອຊອກຫາຄ່າສະເລ່ຍ Mean ແລະ ຄ່າຜັນປ່ຽນມາດຕະຖານ (SD). ເຊິ່ງໄດ້ວິເຄາະຜົນໄປຕາມ 5 ລະດັບ ຕາມເກນຂອງ Patton, (1990) ຄື:

5. ຄະແນນສະເລ່ຍ 4.50-5.00 ໝາຍເຖິງ ລະດັບຫຼາຍທີ່ສຸດ
4. ຄະແນນສະເລ່ຍ 3.50-4.49 ໝາຍເຖິງ ລະດັບຫຼາຍ
3. ຄະແນນສະເລ່ຍ 2.50-3.49 ໝາຍເຖິງ ລະດັບປານກາງ
2. ຄະແນນສະເລ່ຍ 1.50-2.49 ໝາຍເຖິງ ລະດັບໜ້ອຍ
1. ຄະແນນສະເລ່ຍ 1.00-1.49 ໝາຍເຖິງ ລະດັບໜ້ອຍທີ່ສຸດ

1) ການປະເມີນຜົນຂໍ້ມູນ

ການປະເມີນຜົນຂໍ້ມູນໄດ້ແບ່ງອອກເປັນ 2 ພາກສ່ວນຫຼັກຄື: 1) ການປະເມີນຜົນການເຮັດແບບເຝິກຫັດຕົວຈິງຂອງນັກສຶກສາ ແລະ 2) ການປະເມີນກ່ອນ ແລະ ຫຼັງການເຝິກ

2) ຜົນໄດ້ຮັບຈາກການປະເມີນການເຮັດແບບທົດສອບຕົວຈິງ

ການວິເຄາະຂໍ້ມູນແມ່ນໃຊ້ Microsoft Excel ໂດຍໃຊ້ສະຖິຕິເຊັ່ນ: ຄ່າສະເລ່ຍ ແລະ ເປີເຊັນ. ໂດຍການນຳເອົາຄະແນນຈາກການປະເມີນຜົນຕາມມາດຖານໃນແຕ່ລະການສອບທົດສອບ, ໃນແຕ່ລະຄັ້ງແມ່ນຄິດໄລ່ຄ່າສະເລ່ຍແລ້ວຄິດໄລ່ເປັນເປີເຊັນ ໂດຍສົມທຽບກັບເງື່ອນໄຂການປະເມີນຕາມເກນການໃຫ້ຄະແນນເຕັມ 100% (ຕາຕະລາງ 1).

- ຄະແນນ 80 ຂຶ້ນໄປ ໝາຍຄວາມວ່າ ດີຫຼາຍ.
- ຄະແນນ 70 ຂຶ້ນໄປ ໝາຍຄວາມວ່າ ດີ.
- ຄະແນນ 60 ຂຶ້ນໄປ ໝາຍຄວາມວ່າ ພໍໃຊ້.
- ຄະແນນຕໍ່າກວ່າ 60 ໝາຍຄວາມວ່າ ຕ້ອງໄດ້ປັບປຸງ.

3) ການປະເມີນກ່ອນ ແລະ ຫຼັງການເຝິກ

ການວິເຄາະຂໍ້ມູນການທົດສອບກ່ອນການເຝິກ (Pre-test) ແລະ ຫຼັງຈາກການເຝິກ (Post-test) ໂດຍໃຊ້ສະຖິຕິອັດຕາສ່ວນຮ້ອຍ. ຄ່າຜັນປ່ຽນມາດຖານ (Standard Deviation: S.D.) ຄ່າສະເລ່ຍມາດຕະຖານ (Mean) ແລະ ການທົດສອບຄ່າ T-Test ແບບ 2 ກຸ່ມທີ່ບໍ່ເປັນອິດສະຫຼະຕໍ່ກັນ ຫຼື ກຸ່ມຕົວຢ່າງທັງສອງມີຄວາມສຳພັນກັນ (Paired T-test) ດ້ວຍໂປຼແກຣມຄອມພິວເຕີ.

ການຄັດເລືອກປະຊາກອນສຸ່ມການຊຸ່ມເອົາຈາກຈຳນວນທັງໝົດ 135 ຄົນ ເລືອກເອົາ ຊຸດລະ 30 ຄົນ ລວມເປັນ 90 ຄົນ ແບບງາຍ (Simple random mapping) ເພື່ອເຂົ້າຮ່ວມການປະເມີນກ່ອນ ແລະ ຫຼັງການເຝິກ.

❖ ສູດການຊອກຫາຄ່າ t (Paired T-test):

$$t = \frac{\sum d}{\sqrt{\frac{n(\sum d^2) - (\sum d)^2}{n - 1}}}$$

d ແມ່ນຄ່າຄວາມແຕກຕ່າງທັງຕໍ່ຄູ່
 n ຈຳນວນຕົວຢ່າງ

3. ຜົນໄດ້ຮັບ

3.1 ການປະເມີນຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປແກຣມ Flowgorithm Platform

ການປະເມີນຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປແກຣມ Flowgorithm Platform ຜົນການສຶກສາ ພົບວ່າ ທຳອິດແມ່ນໄດ້ທົດສອບຄວາມຮູ້ພື້ນຖານກ່ຽວກັບການການຂຽນໂປແກຣມຂອງນັກສຶກສາກ່ອນດ້ວຍການເຮັດແບບທົດສອບ ຫຼື ບົດເຝິກຫັດກ່ອນ ຫຼັງຈາກນັ້ນ, ຈຶ່ງເລີ່ມເຂົ້າສູ່ການສະເໜີເນື້ອໃນການນຳໃຊ້ Flowgorithm ພ້ອມຍົກຕົວຢ່າງໃນການວິເຄາະບັນຫາ ແລະ ການອອກແບບແຜນຜັງ Flowchart ຕາມລຳດັບ. ພາຍຫຼັງທີ່ນັກສຶກສາໄດ້ເຂົ້າເຝິກແລ້ວ, ຈະເປັນຂັ້ນຕອນການທົດສອບຕົວຈິງເພື່ອປະເມີນຜົນວ່າກ່ອນ ແລະ ຫຼັງການເຝິກຜົນຂອງການ

ທົດສອບຈະເປັນແນວໃດ ເຊິ່ງຈາກການປະເມີນແລ້ວເຫັນວ່າຜົນຄະແນນການທົດສອບພາຍຫຼັງເຂົ້າເຝິກແມ່ນມີແນວໂນມດີຂຶ້ນຫຼາຍຖ້າທຽບໃສ່ກ່ອນເຂົ້າເຝິກ. ຈາກການວິເຄາະຄະແນນຄວາມສາມາດຂອງຜູ້ເຂົ້າຮ່ວມເຝິກອົບຮົມເປັນລາຍບຸກຄົນດ້ວຍການເຮັດແຕ່ລະບົດເຝິກຫັດໃນແຕ່ລະແບບທົດສອບທີ່ມີການວິເຄາະບັນຫາ ແລະ ແຕ້ມແຜນຜັງ Flowchart ດ້ວຍໂປຼແກຼມ Flowgorithm ເພື່ອແກ້ໄຂບັນຫາ ເຊິ່ງສາມາດສະຫຼຸບຜົນໄດ້ຮັບສະແດງໃນຕາຕະລາງ 2 ແລະ 3.

ຈາກຕາຕະລາງ 2 ການປະເມີນຜົນການປະຕິບັດຕົວຈິງ ໂດຍອີງຕາມເກນການປະເມີນແລ້ວຊື່ໃຫ້ເຫັນວ່າ: ນັກສຶກສາຈຳນວນ 41 ຄົນ ໃນອັດຕາ ສ່ວນ 30.38% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບໄດ້ດີຫຼາຍເນື່ອງຈາກມີຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີ, ສາມາດວິເຄາະບັນຫາ ແລະ ອອກແບບແຜນຜັງ Flowchart ໄດ້ຖືກຕ້ອງ; ນັກສຶກສາຈຳນວນ 37 ຄົນ ໃນອັດຕາສ່ວນ 27.40% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບໄດ້ດີເນື່ອງຈາກມີຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີ, ສາມາດວິເຄາະບັນຫາ ແລະ ອອກແບບແຜນຜັງ Flowchart ໄດ້ຖືກຕ້ອງແຕ່ຍັງເກີດຂໍ້ຜິດພາດໃນບາງຈຸດ; ນັກສຶກສາຈຳນວນ 32 ຄົນ ໃນອັດຕາສ່ວນ 23.70% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບຜິດໃຊ້ໄດ້ເນື່ອງຈາກຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີຍັງບໍ່ທັນໄດ້ດີເທົ່າທີ່ຄວນ, ການວິເຄາະບັນຫາຍັງບໍ່ທັນຖືກຕ້ອງຊັດເຈນ ແລະ ການອອກແບບແຜນຜັງ Flowchart ຍັງເກີດຂໍ້ຜິດພາດຫຼາຍຈຸດ; ແລະ ນັກສຶກສາຈຳນວນ 25 ຄົນ ໃນອັດຕາສ່ວນ 18.52% ຂອງນັກສຶກສາທັງໝົດເຮັດແບບທົດສອບຍັງບໍ່ທັນໄດ້ດີຕ້ອງໄດ້ປັບປຸງຄືນໃໝ່ ເນື່ອງຈາກຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີຍັງບໍ່ທັນເລິກເຊິ່ງ, ການວິເຄາະບັນຫາຍັງບໍ່ທັນຖືກຕ້ອງຊັດເຈນ ແລະ ການອອກແບບແຜນຜັງ Flowchart ຍັງເກີດຂໍ້ຜິດພາດຫຼາຍຈຸດຕ້ອງໄດ້ປັບປຸງຄືນໃໝ່.

ຈາກຕາຕະລາງ 3 ຂ້າງເທິງ ການປະເມີນຜົນການປະຕິບັດຕົວຈິງ ໂດຍອີງຕາມເກນການປະເມີນແລ້ວຊື່ໃຫ້ເຫັນວ່າ: ນັກສຶກສາຈຳນວນ 68 ຄົນ ໃນອັດຕາ ສ່ວນ 50.37% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບໄດ້ດີຫຼາຍເນື່ອງຈາກມີຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີ, ສາມາດວິເຄາະບັນຫາ ແລະ ອອກແບບແຜນຜັງ Flowchart ໄດ້ຖືກຕ້ອງ; ນັກສຶກສາຈຳນວນ 32 ຄົນ ໃນອັດຕາສ່ວນ 23.70% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບໄດ້ດີເນື່ອງຈາກມີຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີ, ສາມາດວິເຄາະບັນຫາ ແລະ ອອກແບບແຜນຜັງ Flowchart ໄດ້ຖືກຕ້ອງແຕ່ຍັງເກີດຂໍ້ຜິດພາດໃນບາງຈຸດ; ນັກສຶກສາຈຳນວນ 24 ຄົນ ໃນອັດຕາສ່ວນ 17.78% ຂອງນັກສຶກສາທັງໝົດສາມາດເຮັດແບບທົດສອບຜິດໃຊ້ໄດ້ເນື່ອງຈາກຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີຍັງບໍ່ທັນໄດ້ດີເທົ່າທີ່ຄວນ, ການວິເຄາະບັນຫາຍັງບໍ່ທັນຖືກຕ້ອງຊັດເຈນ ແລະ ການອອກແບບແຜນຜັງ Flowchart ຍັງເກີດຂໍ້ຜິດພາດຫຼາຍຈຸດ; ແລະ ນັກສຶກສາຈຳນວນ 11 ຄົນ ໃນອັດຕາສ່ວນ 8.52% ຂອງນັກສຶກສາທັງໝົດເຮັດແບບທົດສອບຍັງບໍ່ທັນໄດ້ດີຕ້ອງໄດ້ປັບປຸງຄືນໃໝ່ ເນື່ອງຈາກຄວາມເຂົ້າໃຈເນື້ອໃນບົດຮຽນດີຍັງບໍ່ທັນເລິກເຊິ່ງ, ການວິເຄາະບັນຫາຍັງບໍ່ທັນຖືກຕ້ອງຊັດເຈນ ແລະ ການອອກແບບແຜນຜັງ Flowchart ຍັງເກີດຂໍ້ຜິດພາດຫຼາຍຈຸດຕ້ອງໄດ້ປັບປຸງຄືນໃໝ່.

3.2 ການປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platform

ການປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platform ຜົນ ການປະເມີນກ່ອນ ແລະ ຫຼັງການເຝິກ ຄື ຂັ້ນຕອນນີ້ເປັນການປະເມີນຄະແນນກ່ອນ ແລະ ຫຼັງການເຂົ້າເຝິກ ເຊິ່ງການວິເຄາະຂໍ້ມູນການຄົ້ນຄວ້າກ່ຽວກັບຜົນສຳເລັດໃນການແກ້ໄຂບັນຫາ ແລະ ການຂຽນແຜນຜັງ Flowchart ໂດຍນຳໃຊ້ ໂປຼແກຼມ Flowgorithm ດັ່ງທີ່ສະແດງຢູ່ໃນຕາຕະລາງ 4 ແລະ ຕາຕະລາງ 5.

ຜົນການສຶກສາໃນຕາຕະລາງ 4 ສົມທຽບຜົນການທົດສອບກ່ອນ ແລະ ຫຼັງການເຝິກ. ຈາກການຂັດເລືອກນັກສຶກສາໃນແຕ່ລະຊຸດຮຽນ ຈຳນວນ 30 ຄົນ ($n=30$) ເພື່ອເຂົ້າຮ່ວມທົດສອບໂດຍການເຮັດແບບທົດສອບທັງໝົດ 3 ແບບ (8 ບົດເຝິກຫັດ) ດ້ວຍການສະສົມຄະແນນແຕ່ລະແບບທົດສອບດ້ວຍເກນການໃຫ້ຄະແນນເຕັມ 100 ສະແດງໃຫ້ເຫັນວ່າ: ສຳລັບແບບທົດສອບທີ 1) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຕາມລຳດັບ (Sequence Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 70.13$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 87.23$; ສຳລັບແບບທົດສອບທີ 2) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຂັ້ນຕອນທາງເລືອກ (Selection Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 63.47$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 81.17$; ສຳລັບແບບທົດສອບທີ 3) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງການກະທຳຊ້ຳ (Loop Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 64.77$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 80.33$. ຈາກຜົນການທົດສອບໃນຕາຕະລາງ 8 ສາມາດສະຫຼຸບໄດ້ວ່າການຈັດຕັ້ງການຮຽນການສອນໃຫ້ແກ່ນັກຮຽນໂດຍນຳໃຊ້ Flowgorithm ໃນການວິເຄາະ ແລະ ອອກແບບແຜນຜັງໂປຼແກຼມນັກຮຽນມີຄວາມຮູ້, ທັກສະ ແລະ ຜົນຄະແນນການທົດສອບພາຍຫຼັງເຂົ້າເຝິກ ເຊິ່ງການທົດສອບໄດ້ຮັບຜົນດີຂຶ້ນ ມີຄ່າສະຖິຕິທີ່ຄວາມສຳຄັນໃນລະດັບ 0.05.

ຜົນການທົດລອງໃນຕາຕະລາງ 5. ການທົດສອບປະສິດຕິພາບຂອງເຄື່ອງມືວິໄຈ (E_1/E_2) ເມື່ອທຽບໃສ່ມາດຕານທີ່ກຳນົດຢູ່ທີ່ 80/80 ຈາກຜົນການຄົ້ນຄວ້າທີ່ເກີດຈາກການທົດລອງກັບກຸ່ມຕົວຢ່າງແມ່ນ 83.33/82.91, ສະແດງໃຫ້ເຫັນວ່າເຄື່ອງມືຄົ້ນຄວ້າມີປະສິດທິພາບສູງກວ່າການວິໄຈຂັ້ນພື້ນຖານ ເປັນການວິໄຈທີ່ສາມາດໃຊ້ເປັນເກນໃນການຕັດສິນໃຈໄດ້, ສະນັ້ນ ເນື້ອໃນຂອງແບບສອບຖາມແມ່ນມີຄວາມໜ້າເຊື່ອຖືສູງ ແລະ ມີຄວາມເໝາະສົມທີ່ຈະນຳໄປໃຊ້ເປັນເຄື່ອງມືໃຫ້ແກ່ການສຶກສາວິເຄາະໄດ້.

ການສົມທຽບການຂຽນໂປຼແກຼມລະຫວ່າງການນຳໃຊ້ Flowgorithm ແລະ ການຂຽນແບບທົ່ວໄປ ຜົນການປຽບທຽບລະຫວ່າງການນຳໃຊ້ໂປຼແກຼມ Flowgorithm ເຂົ້າໃນການຂຽນໂປຼແກຼມ ແລະ ການຂຽນໂປຼແກຼມດ້ວຍ ຄຳສັ່ງ (Code) ໂດຍກົງຂອງພາສາໂປຼແກຼມຄອມພິວເຕີ (Command line) ຖ້າສົມທຽບການພັດທະນາທັກສະການຂຽນໂປຼແກຼມສຳລັບນັກສຶກສາທີ່ກຳລັງສຶກສາການຂຽນໂປຼແກຼມຄອມພິວເຕີໃນຂັ້ນເລີ່ມຕົ້ນສັງເກດເຫັນວ່າ ການນຳໃຊ້ໂປຼແກຼມເຂົ້າມາຊ່ວຍ

ວຍໃນການວິເຄາະບັນຫາ ແລະ ການອອກແບບແຜນຜັງ Flowchart ເພື່ອແກ້ໄຂບັນຫາແມ່ນມີຄວາມວ່ອງໄວກວ່າການຂຽນໂປຼແກຼມ ຫຼື ການຂຽນ code ຕົວຈິງໃນພາສາໂປຼແກຼມສະເພາະໃດໜຶ່ງ ເພາະວ່າສໍາລັບຜູ້ເລີ່ມຕົ້ນຮຽນຂຽນໂປຼແກຼມຄອມພິວເຕີ ກ່ອນອື່ນຕ້ອງທຳຄວາມເຂົ້າໃຈ ແລະ ຈະຕ້ອງໄດ້ສຶກສາກ່ຽວກັບຫຼັກການຂຽນ Pseudo Code ກ່ອນ ເນື່ອງຈາກວ່າແມ່ນ code ທີ່ໃຊ້ເພື່ອເປັນຕົວແທນຂອງ Algorithm ດ້ວຍຄໍາສັບ ຫຼື ຄໍາສັ່ງທີ່ຂຽນເປັນພາສາອັງກິດທີ່ບໍ່ຂຶ້ນກັບພາສາຄອມພິວເຕີໃດພາສາໜຶ່ງ, ໝາຍເຖິງການສະແດງຂັ້ນຕອນ, ວິທີການມທີ່ໃຊ້ພາສາຂຽນທີ່ເຂົ້າໃຈງ່າຍ ແລະ ກົດຈະກຳທີ່ຈະນຳສະເໜີມັກໃຊ້ຮູບແບບຄ້າຍຄືປະໂຫຍກຂອງພາສາອັງກິດເພື່ອອະທິບາຍລາຍລະອຽດຂອງ Algorithm.

ສະຫຼຸບ ການປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platform ມີບ ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຕາມລຳດັບ (Sequence Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 70.13$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 87.23$; ສໍາລັບແບບທົດສອບທີ 2) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຂັ້ນຕອນທາງເລືອກ (Selection Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 63.47$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 81.17$; ສໍາລັບແບບທົດສອບທີ 3) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງການກະທຳຊ້ຳ (Loop Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 64.77$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 80.33$.

4. ວິພາກຜົນ

ໃນການການປະເມີນຄວາມສາມາດ ຫຼື ຜົນການໃນການຂຽນໂປແກຣມ Flowgorithm Platform ມີບວ່າ ທຳອິດ ແມ່ນໄດ້ທົດສອບຄວາມຮູ້ພື້ນຖານກ່ຽວກັບການການຂຽນໂປຼແກຼມຂອງນັກສຶກສາກ່ອນດ້ວຍການເຮັດແບບທົດສອບ ຫຼື ບົດເຝິກຫັດກ່ອນ ຫຼັງຈາກນັ້ນ, ຈຶ່ງເລີ່ມເຂົ້າສູ່ການນຳສະເໜີເນື້ອໃນການນຳໃຊ້ Flowgorithm ພ້ອມຍົກຕົວຢ່າງໃນການວິເຄາະບັນຫາ ແລະການອອກແບບແຜນຜັງ Flowchart ຕາມລຳດັບ. ພາຍຫຼັງທີ່ນັກສຶກສາໄດ້ເຂົ້າເຝິກແລ້ວ, ຈະເປັນຂັ້ນຕອນການທົດສອບຕົວຈິງເພື່ອປະເມີນຜົນວ່າກ່ອນ ແລະ ຫຼັງການເຝິກຜົນຂອງການທົດສອບຈະເປັນແນວໃດ ເຊິ່ງຈາກການປະເມີນແລ້ວເຫັນວ່າຜົນຄະແນນການທົດສອບພາຍຫຼັງເຂົ້າເຝິກແມ່ນມີແນວໂນມດີຂຶ້ນຫຼາຍຖ້າທຽບໃສ່ກ່ອນເຂົ້າເຝິກ. ຈາກການວິເຄາະຄະແນນຄວາມສາມາດຂອງຜູ້ເຂົ້າຮ່ວມເຝິກອົບຮົມເປັນລາຍບຸກຄົນດ້ວຍການເຮັດແຕ່ລະບົດເຝິກຫັດໃນແຕ່ລະແບບທົດສອບທີ່ມີການວິເຄາະບັນຫາ ແລະ ແຕ່ມແຜນຜັງ Flowchart ດ້ວຍໂປຼແກຼມ Flowgorithm

ການປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການຂຽນໂປແກຣມ Flowgorithm Platform ມີບ ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຕາມລຳດັບ (Sequence Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 70.13$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 87.23$; ສໍາລັບແບບທົດສອບທີ 2) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຂັ້ນຕອນທາງເລືອກ (Selection Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 63.47$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} =$

81.17 ; ສໍາລັບແບບທົດສອບທີ 3) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງການກະທຳຊ້ຳ (Loop Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 64.77$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 80.33$.

ສົມທຽບການພັດທະນາທັກສະການຂຽນໂປຼແກຼມສໍາລັບນັກສຶກສາ ທີ່ກຳລັງສຶກສາການຂຽນໂປຼແກຼມຄອມພິວເຕີໃນຂັ້ນເລີ່ມຕົ້ນຈະສັງເກດເຫັນວ່າ ການນຳໃຊ້ໂປຼແກຼມເຂົ້າມາຊ່ວຍໃນການວິເຄາະບັນຫາ ແລະ ການອອກແບບແຜນຜັງ Flowchart ເພື່ອແກ້ໄຂບັນຫາແມ່ນມີຄວາມວ່ອງໄວກວ່າການຂຽນໂປຼແກຼມ ຫຼື ການຂຽນ code ຕົວຈິງໃນພາສາໂປຼແກຼມສະເພາະໃດໜຶ່ງ ເພາະວ່າສໍາລັບຜູ້ເລີ່ມຕົ້ນຮຽນຂຽນໂປຼແກຼມຄອມພິວເຕີກ່ອນອື່ນຕ້ອງທຳຄວາມເຂົ້າໃຈ ແລະ ຈະຕ້ອງໄດ້ສຶກສາກ່ຽວກັບຫຼັກການຂຽນ Pseudo Code ກ່ອນ ເນື່ອງຈາກວ່າແມ່ນລະຫັດຈຳລອງທີ່ໃຊ້ເພື່ອເປັນຕົວແທນຂອງ Algorithm ດ້ວຍຄໍາສັບ ຫຼື ຄໍາສັ່ງທີ່ຂຽນເປັນພາສາອັງກິດທີ່ບໍ່ຂຶ້ນກັບພາສາຄອມພິວເຕີໃດພາສາໜຶ່ງ, ໝາຍເຖິງການສະແດງຂັ້ນຕອນ, ວິທີການມທີ່ໃຊ້ພາສາຂຽນທີ່ເຂົ້າໃຈງ່າຍ ອາດໃຊ້ພາສາລາວ ຫຼື ພາສາອັງກິດກໍໄດ້ ຂຶ້ນກັບຄວາມສະດວກສະບາຍຂອງຜູ້ຂຽນ ແລະ ກົດຈະກຳທີ່ຈະນຳສະເໜີມັກໃຊ້ຮູບແບບຄ້າຍຄືປະໂຫຍກຂອງພາສາອັງກິດເພື່ອອະທິບາຍລາຍລະອຽດຂອງ Algorithm. ແຕ່ Flowgorithm ກໍຍັງມີຂໍ້ຈຳກັດຢູ່ຫຼາຍດ້ານ ເຊັ່ນ ບໍ່ເໝາະສົມສໍາລັບໂຄງການຂະໜາດໃຫຍ່ (Not Suitable for Large-scale Projects) ເພາະ Flowgorithm ບໍ່ໄດ້ຖືກອອກແບບມາສໍາລັບການຄຸ້ມຄອງໂຄງການຊອບແວຂະໜາດໃຫຍ່ ຫຼື ການພັດທະນາຊອບແວແບບມີອາຊີບທີ່ມີການອອກແບບແຜນຜັງງານທີ່ສະຫຼັບຊັບຊ້ອນ, ການຄວບຄຸມລະບົບ ແລະ ການເຊື່ອມໂຍງກັບເຄື່ອງມືຕ່າງໆ ເຊິ່ງເກີນຂອບເຂດຄວາມສາມາດຂອງ Flowgorithm. ການວິເຄາະອີງໃສ່ຜົນການປຽບທຽບຄ່າ T-Test ແບບ 2 ກຸ່ມທີ່ບໍ່ເປັນອິດສະຫຼະຕໍ່ກັນ ຫຼື ກຸ່ມຕົວຢ່າງທັງສອງມີຄວາມສຳພັນກັນ (Paired T-test) ດ້ວຍໂປຼແກຼມຄອມພິວເຕີ ເພື່ອປຽບທຽບຄວາມສາມາດໃນການຂຽນໂປແກຣມ Flowgorithm Platform ມີບວ່າ ສໍາລັບແບບທົດສອບທີ 1) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຕາມລຳດັບ (Sequence Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 70.13$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 87.23$. ສໍາລັບແບບທົດສອບທີ 2) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຂັ້ນຕອນທາງເລືອກ (Selection Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 63.47$ Type equation here. ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 81.17$. ສໍາລັບແບບທົດສອບທີ 3) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງການກະທຳຊ້ຳ (Loop Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 64.77$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 80.33$.

ການສຶກສາກ່ຽວກັບການສົມທຽບການຂຽນໂປຼແກຼມດ້ວຍ Flowgorithm ແລະ ການຂຽນໂປຼແກຼມດ້ວຍ ຄໍາສັ່ງໂດຍກົງຂອງພາສາໂປຼແກຼມຄອມພິວເຕີ (Command line) ເຫັນວ່າການຂຽນໂປຼແກຼມດ້ວຍການອອກແບບແຜນຜັງ Flowchart ແລ້ວມາສ້າງໃນ Flowgorithm ແລະ ສັ່ງໃຫ້ປະມວນຜົນເຊິ່ງຜູ້ຂຽນໂປຼແກຼມ ຫຼື

Programmer ຈະສາມາດສັງເກດເຫັນການຄຳນວນ ຫຼື ການໄຫຼຂອງຂໍ້ມູນ (Data Flow) ໃນແຕ່ລະຊັ້ນ (Processing Layer) ໄດ້ດ້ວຍຕາເປົ່າ ແຕ່ການຂຽນໂປຼແກຼມດ້ວຍຄຳສັ່ງໂດຍກົງໃນພາສາໂປຼແກຼມຄອມພິວເຕີ ສະເພາະແມ່ນບໍ່ສາມາດເບິ່ງເຫັນຂະບວນການປະມວນຜົນ ເພາະວ່າໜ່ວຍປົມວນຜົນ (Processor) ຂອງຄອມພິວເຕີຈະບີດີບັດການຄຳນວນແລ້ວ ຈຶ່ງຈະສະແດງຜົນຮັບອອກມາ. ສະນັ້ນ, ຖ້າຈະສົມທຽບລະຫວ່າງສອງວິທີການຂຽນໂປຼແກຼມຄອມພິວເຕີ ແນ່ນອນວ່າ Flowgorithm ແມ່ນທາງເລືອກໜຶ່ງທີ່ເໝາະສົມສຳລັບຜູ້ເລີ່ມຕົ້ນຮຽນຂຽນໂປຼແກຼມໃໝ່ (Beginner programmer) ແລະ ໃນເມື່ອເຂົາເຈົ້າຫາກມີຄວາມຊຳນານງານ (Intermediate Programmer) ຫຼື ຊ່ຽວຊານແລ້ວ (Advance Programmer) ເຂົາເຈົ້າກໍ່ຈະຂຽນໂປຼແກຼມໃນຄອມພິວເຕີດ້ວຍພາສາທີ່ທົດລາຍຖະນັດເລີຍ ເພາະວ່າ Flowgorithm ຍັງມີຂໍ້ຈຳກັດຫຼາຍຢ່າງ ເຊັ່ນ ບໍ່ເໝາະສົມສຳລັບໂຄງການຂະໜາດໃຫຍ່ທີ່ມີຄວາມສະຫຼັບຊັບຊ້ອນຫຼາຍ ແຕ່ເຖິງຢ່າງໃດກໍ່ຕາມການສຶກສາຄົ້ນຄວ້ານີ້ ແມ່ນເຈາະຈົງໃສ່ກຸ່ມນັກຮຽນໃນລະດັບວິທະຍາໄລ ຫຼື ມະຫາວິທະຍາໄລ ທີ່ກຳລັງສຶກສາການຂຽນໂປຼແກຼມເບື້ອງຕົ້ນ ແລະ ຈາກຜົນການເກັບກຳຂໍ້ມູນຈາກການປະເມີນຜົນການເຮັດແບບທົດສອບກ່ອນການເຝິກ (Pre-test) ແລະ ຫຼັງການເຝິກ (Post-test) ຊຶ່ງໃຫ້ເຫັນວ່າຜົນການທົດສອບຂອງນັກສຶກສາໄດ້ຄະແນນດີຂຶ້ນ ກວ່າກ່ອນເຝິກ ເຊິ່ງສອດຄ່ອງກັບຜົນຄົ້ນຄວ້າໄຈຂອງ (Malat, 2019) ໄດ້ທຳການຄົ້ນຄວ້າວິໄຈເລື່ອງການພັດທະນາເກມເພື່ອການສຶກສາ ໃນວິຊາຄອມພິວເຕີ ສຳລັບນັກຮຽນຊັ້ນມັດທະຍົມສຶກສາຕອນປາຍ ເຊິ່ງຜົນຂອງການຄົ້ນຄວ້າປະກົດວ່າຜົນສຳເລັດທາງການສຶກສາເມື່ອປຽບທຽບຜົນໄດ້ຮັບ, ການສຶກສາຫຼັງຈາກຮຽນແມ່ນສູງກວ່າກ່ອນຮຽນ ແລະ ຄວາມເຝິງພໍໃຈຂອງນັກຮຽນມີຜົນການປະເມີນຜົນຢູ່ໃນລະດັບດີຫຼາຍ. ການນຳໃຊ້ເຄື່ອງມືທາງ IT ເຂົ້າຊ່ວຍການສຶກສາຮຽນຮູ້ແມ່ນສາມາດສ້າງແຮງກົຕຸ້ນ ແລະ ເຮັດໃຫ້ຮັກຮຽນມີຄວາມກະຕືລືລົ້ນທີ່ຈະຮຽນຮູ້ເພີ່ມເຕີມ ເພາະມັນດຶງດູດນັກຮຽນດ້ວຍການຈຳລອງທາງກຼາບຟິກທີ່ສວຍງາມຈຳແນກຂໍ້ມູນໄດ້ງ່າຍດ້ວຍຂັ້ນຕອນແຜນຜັງ Flowchart ແລະ ຈາກການຄົ້ນຄວ້າວິໄຈຂອງ (Sittiwong, 2017) ໄດ້ດຳເນີນການຄົ້ນຄວ້າກ່ຽວກັບການພັດທະນາເກມຄອມພິວເຕີທີ່ເປັນແບບຈຳລອງເພື່ອສົ່ງເສີມຄວາມສາມາດໃນການຄິດຍຸດທະວິທີຂອງນັກສຶກສາລະດັບມັດທະຍົມປາຍເຊິ່ງຈຸດປະສົງທົ່ວໄປຂອງການຄົ້ນຄວ້ານີ້ແມ່ນເພື່ອພັດທະນາເກມຄອມພິວເຕີຈຳລອງສະຖານະການ ເພື່ອສົ່ງເສີມຄວາມສາມາດໃນການຄິດທາງຍຸດທະວິທີເຮັດໃຫ້ນັກຮຽນເຂົ້າໃຈ ແລະ ສາມາດວິເຄາະບັນຫາໄດ້ດີຂຶ້ນ. ດັ່ງນັ້ນ, ນັກສຶກສາຈະສາມາດອອກແບບວິທີການແກ້ໄຂບັນຫາຕ່າງໆ ແລະ ແປເປັນ Flowcharts ມີປະສິດທິພາບຫຼາຍຂຶ້ນ. ນອກຈາກນີ້, ນັກສຶກສາທີ່ມີຜື້ນຖານທີ່ແຕກຕ່າງກັນພວກເຂົາເຈົ້າຈະມີຮູບແບບການຮຽນຮູ້ທີ່ແຕກຕ່າງກັນ ເຊິ່ງສາມາດສຶກສາ ແລະ ຄົ້ນຄວ້າໃນແບບທີ່ເຂົາເຈົ້າຕ້ອງການ ໂດຍມີເພື່ອນຊ່ວຍແລກປ່ຽນຄວາມຄິດ ແລະ ມີຄູສອນລຳຖ້າໃຫ້ຄຳແນະນຳ (Güzeldere Aydin & Ocak Karabay, 2019). ດັ່ງນັ້ນ, ການປະເມີນຜົນຂອງນັກຮຽນຈະສູງຂຶ້ນ ແລະ ຂະບວນການເຮັດວຽກເປັນກຸ່ມຈະກະຕຸ້ນເຊິ່ງກັນແລະກັນໃຫ້ມີຄວາມຮັບຜິດຊອບຫຼາຍຂຶ້ນ ເຊິ່ງສາມາດເຮັດວຽກຊ່ວຍເຫຼືອເຊິ່ງກັນແລະ

ກັນ ເຮັດໃຫ້ວຽກງານອອກມາເປັນລະອຽດແລະຖືກຕ້ອງຫຼາຍຂຶ້ນ. ຜົນການຄົ້ນຄວ້າກ່ຽວກັບ ຜົນສຳເລັດໃນການແກ້ໄຂບັນຫາແລະການຂຽນແຜນຜັງໂດຍນຳໃຊ້ໂປຼແກຼມ ກັບນັກສຶກສາທີ່ຮຽນວິຊາການຂຽນໂປຼແກຼມຂອງນັກສຶກສາສົ່ງຜົນໃຫ້ນັກຮຽນມີຜົນການຮຽນສູງຂຶ້ນ ແລະ ເຄື່ອງມືທີ່ໃຊ້ໃນການຄົ້ນຄວ້າແມ່ນມີປະສິດທິພາບ ແລະ ເໝາະສົມເພື່ອນຳໃຊ້ເຂົ້າໃນການຮຽນ-ການສອນກ່ຽວກັບການຂຽນໂປຼແກຼມ Flowgorithm ແມ່ນກ່ຽວຂ້ອງກັບການຂຽນແຜນຜັງແບບຕາມລຳດັບ (Sequence Flowchart), ແບບມີທາງການຄັດເລືອກ (Selection Flowchart) ແລະ ການກະວົນຮອບ (Loop Flowchart) ພາຍຫຼັງທີ່ນັກຮຽນໄດ້ອອກແບບແຜນຜັງແລ້ວ ນັກສຶກສາສາມາດທົດສອບດ້ວຍຂໍ້ມູນຕົວຈິງໄດ້ທັນທີ ແລະ ໃຫ້ນັກສຶກສາກວດກາຂະບວນການຄິດດ້ວຍຕົນເອງວ່າ ການຄຳນວນເປັນໄປຕາມທີ່ໄດ້ອອກແບບໄວ້ແລ້ວໂດຍການກວດສອບແຕ່ລະຂັ້ນຕອນວ່າຖືກຕ້ອງແລ້ວບໍ່, ຖ້າຫາກເກີດຂໍ້ຜິດພາດຂຶ້ນຢູ່ຈຸດໃດກໍ່ຕ້ອງໄດ້ນຳມາປັບປຸງແກ້ໄຂຄືນ ດັ່ງນັ້ນ, ການປະຕິບັດການທົດລອງຕົວຈິງກັບເຄື່ອງມື (Tools) ຂອງໂປຼແກຼມ Flowgorithm ນັກສຶກສາຈະສາມາດສ້າງຄວາມຮູ້ດ້ວຍຕົວເອງເຊິ່ງສອດຄ່ອງກັບ (Panaphong, 2016) ໄດ້ດຳເນີນການຄົ້ນຄວ້າກ່ຽວກັບ ການພັດທະນາເກມຄອມພິວເຕີໂດຍໃຊ້ຕົວແບບ Abstract code Model (AMD) ຜື້ນຖານສຳລັບການເພີ່ມທັກສະການແກ້ໄຂບັນຫາທາງຄະນິດສາດຂອງນັກຮຽນຊັ້ນມັດທະຍົມຕອນຕົ້ນ ເຊິ່ງເກມການຈຳລອງດ້ວຍຄອມພິວເຕີຊ່ວຍເພີ່ມຄວາມສົນໃຈ ແລະ ສາມາດປັບປຸງທັກສະການຄິດຂອງຄົນ ນີ້ສະແດງໃຫ້ເຫັນວ່າ ເກມຄອມພິວເຕີ ທີ່ພັດທະນາສາມາດເພີ່ມທັກສະການແກ້ໄຂບັນຫາທາງຄະນິດສາດຂອງນັກຮຽນໄດ້. ສະນັ້ນ, ຈຶ່ງເປັນອີກທາງເລືອກໜຶ່ງໃນການພັດທະນາທັກສະການແກ້ບັນຫາທາງຄະນິດສາດຂອງນັກຮຽນ.

5. ສະຫຼຸບ

ໃນການຄົ້ນຄວ້າວິໄຈນີ້ ໄດ້ສຶກສາກ່ຽວກັບການສຶກສາຜົນການຂຽນໂປຼແກຼມຄອມພິວເຕີໃນວິຊາວິສະວະກຳຄອມພິວເຕີ ຂອງນັກສຶກສາດ້ວຍເພີດພອມໂຟຼກໂລດທິມກັບການຂຽນໂປຼແກຼມທົ່ວໄປ ໂດຍມີຈຸດປະສົງ ເພື່ອປະເມີນຄວາມສາມາດຫຼື ຜົນການໃນການຂຽນໂປແກຼມ ແລະ ເພື່ອປຽບທຽບຄວາມສາມາດ ຫຼື ຜົນໃນການ. ທິມງານຂອງພວກເຮົາໄດ້ສັງລວມເນື້ອໃນບົດຮຽນເພື່ອນຳສະເໜີການນຳໃຊ້ເຄື່ອງມືຜື້ນຖານໃນການອອກແບບແຜນຜັງງານ ແລະ ໄດ້ອອກແບບທົດສອບໂດຍປະກອບມີ 3 ແບບທົດສອບຫຼັກ ໂດຍປະກອບມີ 8 ບົດເຝິກຫັດຢ່າງທີ່ເປັນການນຳໃຊ້ຄຳສັ່ງຜື້ນຖານໃນການຂຽນໂປຼແກຼມຈາກງ່າຍຫຍາຍຕາມລຳດັບເພື່ອປະເມີນຜົນຄວາມຮັບຮູ້ ແລະ ເຂົ້າໃຈຂອງນັກສຶກສາ. ທຳອິດແມ່ນໄດ້ທົດສອບຄວາມຮູ້ຜື້ນຖານກ່ຽວກັບການການຂຽນໂປຼແກຼມຂອງນັກສຶກສາກ່ອນດ້ວຍການເຮັດແບບທົດສອບ ຫຼື ບົດເຝິກຫັດກ່ອນ ຫຼັງຈາກນັ້ນ, ຈຶ່ງເລີ່ມເຂົ້າສູ່ການນຳສະເໜີເນື້ອໃນການນຳໃຊ້ Flowgorithm ພ້ອມຍົກຕົວຢ່າງໃນການວິເຄາະບັນຫາ ແລະການອອກແບບແຜນຜັງ Flowchart ຕາມລຳດັບ. ພາຍຫຼັງທີ່ນັກສຶກສາໄດ້ເຂົ້າເຝິກແລ້ວ, ຈະເປັນຂັ້ນຕອນການທົດສອບຕົວຈິງເພື່ອປະເມີນຜົນວ່າກ່ອນ ແລະ ຫຼັງການເຝິກຜົນຂອງການທົດສອບຈະເປັນແນວໃດ ເຊິ່ງຈາກການປະເມີນແລ້ວເຫັນວ່າຜົນຄະແນນ

ການທົດສອບພາຍຫຼັງເຂົ້າເຝິກແມ່ນມີແນວໂນມດີຂຶ້ນຫຼາຍຖ້າທຽບໃສ່ກ່ອນເຂົ້າເຝິກ.

ສົມທຽບການພັດທະນາທັກສະການຂຽນໂປຼແກຼມສໍາລັບນັກສຶກສາ ທີ່ກໍາລັງສຶກສາການຂຽນໂປຼແກຼມຄອມພິວເຕີໃນຂັ້ນເລີ່ມຕົ້ນຈະສັງເກດເຫັນວ່າ ການນໍາໃຊ້ໂປຼແກຼມເຂົ້າມາຊ່ວຍໃນການວິເຄາະບັນຫາ ແລະ ການອອກແບບແຜນຜັງ Flowchart ເພື່ອແກ້ໄຂບັນຫາແມ່ນມີຄວາມວ່ອງໄວກວ່າການຂຽນໂປຼແກຼມ ຫຼື ການຂຽນ code ຕົວຈິງໃນພາສາໂປຼແກຼມສະເພາະໃດໜຶ່ງ ເພາະວ່າສໍາລັບຜູ້ເລີ່ມຕົ້ນຮຽນຂຽນໂປຼແກຼມຄອມພິວເຕີ ກ່ອນອື່ນຕ້ອງທໍາຄວາມເຂົ້າໃຈ ແລະ ຈະຕ້ອງໄດ້ສຶກສາກ່ຽວກັບຫຼັກການຂຽນ Pseudo Code ກ່ອນ ເນື່ອງຈາກວ່າແມ່ນລະຫັດຈໍາລອງທີ່ໃຊ້ເພື່ອເປັນຕົວແທນຂອງ Algorithm ດ້ວຍຄໍາສັບ ຫຼື ຄໍາສັ່ງທີ່ຂຽນເປັນພາສາອັງກິດທີ່ບໍ່ຂຶ້ນກັບພາສາຄອມພິວເຕີໃດພາສາໜຶ່ງ, ໝາຍເຖິງການສະແດງຂັ້ນຕອນ, ວິທີການມາທີ່ໃຊ້ພາສາຂຽນທີ່ເຂົ້າໃຈງ່າຍ ອາດໃຊ້ພາສາລາວ ຫຼື ພາສາອັງກິດກໍໄດ້ ຂຶ້ນກັບຄວາມສະດວກສະບາຍຂອງຜູ້ຂຽນ ແລະ ກິດຈະກຳທີ່ຈະນໍາສະ ເໜີມັກໃຊ້ຮູບແບບຄ້າຍຄືປະໂຫຍກຂອງພາສາອັງກິດເພື່ອອະທິບາຍລາຍລະອຽດຂອງ Algorithm. ແຕ່ Flowgorithm ກໍຍັງມີຂໍ້ຈຳກັດຢູ່ຫຼາຍດ້ານ ເຊັ່ນ ບໍ່ເໝາະສົມສໍາລັບໂຄງການຂະໜາດໃຫຍ່ (Not Suitable for Large-scale Projects) ເພາະ Flowgorithm ບໍ່ໄດ້ຖືກອອກແບບມາສໍາລັບການຄຸ້ມຄອງໂຄງການຊອບແວຂະໜາດໃຫຍ່ ຫຼື ການພັດທະນາຊອບແວແບບມີອາຊີບທີ່ມີການອອກແບບແຜນຜັງງານທີ່ສະຫຼັບຊັບຊ້ອນ, ການຄວບຄຸມລະບົບ ແລະ ການເຊື່ອມໂຍງກັບເຄື່ອງມືຕ່າງໆ ເຊິ່ງເກີນຂອບເຂດຄວາມສາມາດຂອງ Flowgorithm. ການວິເຄາະອີງໃສ່ຜົນການປຽບທຽບຄ່າ T-Test ແບບ 2 ກຸ່ມທີ່ບໍ່ເປັນອິດສະຫຼະຕໍ່ກັນ ຫຼື ກຸ່ມຕົວຢ່າງທັງສອງມີຄວາມສໍາພັນກັນ (Paired T-test) ດ້ວຍໂປຼແກຼມຄອມພິວເຕີ ເພື່ອປຽບທຽບຄວາມສາມາດໃນການຂຽນໂປຼແກຼມ Flowgorithm Platform ພົບວ່າ ສໍາລັບແບບທົດສອບທີ 1) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຕາມລຳດັບ (Sequence Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 70.13$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 87.23$. ສໍາລັບແບບທົດສອບທີ 2) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງຂັ້ນຕອນທາງເລືອກ (Selection Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 63.47$ Type equation here. ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 81.17$. ສໍາລັບແບບທົດສອບທີ 3) ການແກ້ບັນຫາແບບທົດສອບແຜນຜັງການກວດກາທໍາຊ້າ (Loop Flowchart) ຄະແນນສະເຫຼ່ຍສະສົມກ່ອນການເຂົ້າເຝິກແມ່ນ $\bar{X} = 64.77$ ແຕ່ພາຍຫຼັງການເຂົ້າເຝິກແມ່ນ $\bar{X} = 80.33$

ຈາກຜົນການທົດສອບສາມາດສະຫຼຸບໄດ້ວ່າການຈັດຕັ້ງການຮຽນການສອນໃຫ້ແກ່ນັກຮຽນໂດຍນໍາໃຊ້ Flowgorithm ໃນການວິເຄາະ ແລະ ອອກແບບແຜນຜັງໂປຼແກຼມນັກຮຽນມີຄວາມຮູ້, ທັກສະ ແລະ ຜົນຄະແນນການທົດສອບພາຍຫຼັງເຂົ້າເຝິກ ເຊິ່ງການທົດສອບໄດ້ຮັບຜົນດີຂຶ້ນໂດຍມີຄ່າສະຖິຕິທີ່ຄວາມສໍາຄັນໃນລະດັບ 0.05.

6. ຂໍ້ຂັດແຍ້ງ

ຂ້າພະເຈົ້າໃນນາມຜູ້ຄົນຄວາວິທະຍາສາດ ຂໍປະຕິຍານຕົນວ່າ ຂໍ້ມູນທັງໝົດທີ່ມີໃນບົດຄວາມວິຊາການດັ່ງກ່າວນີ້ແມ່ນບໍ່ມີຂໍ້ຂັດແຍ້ງທາງຜົນປະໂຫຍດກັບພາກສ່ວນໃດ ແລະ ບໍ່ໄດ້ເອື້ອປະໂຫຍດໃຫ້ກັບພາກສ່ວນໃດພາກສ່ວນໜຶ່ງ, ກໍລະນີມີການລະເມີດໃນຮູບການໃດໜຶ່ງ ຂ້າພະເຈົ້າມີຄວາມຍິນດີທີ່ຈະຮັບຜິດຊອບແຕ່ພຽງຜູ້ດຽວ.

7. ເອກະສານອ້າງອີງ

- Arik, K., Ari, K. (2018). Visualization Code Tools for Teaching and Learning Introductory Programming. *The 2nd International Conference on Informatics for Development 2018*. ISBN 978-602-53524-0-9. C1-100.
<https://icid.sunankalijaga.org/downloads/Proceeding%20ICID%202018.pdf>
- F. Alajmi. (2015). Enhanced Teaching Model (Etm) For Teaching Programming Languages. *International Journal of Computer Applications* (0975 – 8887) Volume 121 – No.20.
- Alajmi, F. & Alkhatib, A. (2015). Enhanced Teaching Model (ETM) for Teaching Programming Languages. *International Journal of Computer Applications*, 121. 13-24. 10.5120/21655-5026.
- Wolfram, S. (2016). *How to Teach Computational Thinking*. Stephen Wolfram Blog. Retrieved from <http://blog.stephenwolfram.com/2016/09/how-to-teachcomputational-thinking>.
- Hooshyar, D., Ahmad, R. B., Nasir, M. H. N. M., Shamshirband, S., & Horng, S. J. (2015). Flowchart-based programming environments for improving comprehension and problem-solving skill of novice programmers: a survey. *International Journal of Advanced Intelligence Paradigms*, 7(1), 24– 56.
- Ghazala, S. S., Noman, I. (2016). A qualitative study of major programming languages: teaching programming languages to computer science

- students. *Journal of Information & Communication Technology*, Vol. 10, No. 1, pp: 24-34.
- Robert, R. G. (2018). Algorithms, Programming, Flowcharts and Flowgorithm. <https://us.edu.pl/wydzial/wsne/wp-content/uploads/sites/20/Bez-kategorii/algorithms-programming-flowcharts-and-flowgorithm.pdf>
- Devin, D. Cook. (2015). Flowgorithm: Principle for Teaching Introductory Programming Using Flowcharts. Proceedings of the 2015 American Society for Engineering Education/Pacific South West Conference.
- Zaretska, I., Zholtkevych, et al. (2015). Algorithm Constructor. <https://ceur-ws.org/Vol-2387/20190501.pdf>
- Matheu, K. K. (2016). *Capabilities and Features of Raptor, Visual Logic, and Flowgorithm for Program Logic and Design*. <https://www.researchgate.net/publication/309354785>
- K. A. Rahman, M. J. Nordin and W. S. B. C. W. A. Che. (2008). Automated programming assessment using the pseudocode comparison technique: Does it really work?. *International Symposium on Information Technology*, Kuala Lumpur, Malaysia, 2008, pp. 1-4, doi: 10.1109/ITSIM.2008.4632071.
- Kanis, C. & Somkiat, W. (2006). Visual Programming using Flowchart. *IEEE, International Symposium on Communications and Information Technologies (ISCIT)*, pp. 1062-1065, doi: 10.1109/ISCIT.2006.339940.
- Jatupak, P. (2016). *Development of computer games based on the abstract code model for increasing mathematics problem solving skills for lower secondary school students*. Burapha University Library, https://digital_collect.lib.buu.ac.th/dcms/files/51810905.pdf
- Tipparat, S. (2017). Development of a computer simulation games for enhancing strategic thinking ability of higher education students. vol. 19, no. 3
- Sittiwong, T. (2017). Development of a computer simulation games for enhancing strategic thinking ability of higher education students. *Journal of Education and Innovation*, 19(3), 16–33. retrieved from https://so06.tci-thaijo.org/index.php/edujournal_nu/article/view/122873
- Kittipong, M. (2019). Educational Game Development in Computer Course on Computer Project for Students in Matthayomsuksa 6 Yang Talat Wittayakhan School. *Journal of Educational Technology and Communications*, 2(4).
- Damla, G. A. & Sakire, O. (2019). Improvement of Classroom Management Skills of Teachers Leads to Creating Positive Classroom Climate. *IJERE, International Journal of Educational Research Review*, 5(1), 10-25.
- Güzeldere Aydın, D., & Ocak Karabay, Ş. (2020). Improvement of Classroom Management Skills of Teachers Leads to Creating Positive Classroom Climate. *International Journal of Educational Research Review*, 5(1), 10-25. <https://doi.org/10.24331/ijere.646832>