挑战者基础班教学手册 I

V 1.0

Unit 1 Getting Started!

Overview

This is a preparatory unit that establishes the foundation for creative computing. This unit introduces students to their programming environment, ensuring they can reasonably imagine possibilities for their own mBlock creation. Students must also create their own cloud management accounts so that their future projects can be easily tracked, shared, modified, or evaluated. Design journals and critique groups are essential tools that help organize learning activities throughout the course, so it is quite important for mentors to provide clear guidelines to students regarding the purpose, creation, usage, and value of their design journals and respective critique groups.

Concepts and Practices

Since this is a preparatory unit, none of the course related concepts or practices is intentioned as learning objective. However, most practices are patterns of problem solving that are applicable to various aspects of life rather than being strictly topic related. Therefore, despite sometimes not explicitly specified, most practices are implicitly coupled with the design philosophy of all learning activities. For instance, the activity that guides students with making design journals encourages students to employ design cycle which is the process of planning, making, sharing, and reflecting.

Unit At a Glance

Topics	**Keywords**	**Concepts**	**Practices**
1.1	Creative Computing mBlock Cloud Account Design Journal	N/A	N/A
1.2	Open Exploration Critique Group	N/A	N/A

Suggested Learning Activities

ID	**Activity**	**Worksheet / Handout**	**Extra Materials**
LA 1.1-1	Intro to mBlock	N	N
LA 1.1-2	Cloud Account	Y	N
LA 1.1-3	Design Journal	N	Y
LA 1.2-1	Brave Explorer	Y	N
LA 1.2-2	Critique Group	Y	N

Topic 1.1

Big Ideas

Creative Computing
Cyber Security
Design Cycle

Learning Objectives (Students will be able to...)

LO 1.1-A
Describe what creative computing is in their own words
LO 1.1-B
Become familiar with mBlock software environment
LO 1.1-C
Perform account sign-up independently
LO 1.1-D
Start a personalized design journal for documenting their design process and reflections

Essential Knowledge

EK 1.1-A-1
Exposure to various use cases of computers
EK 1.1-A-2
Exposure to a wide range of creative possibilities using computers
EK 1.1-A-3
Generation of reasonable creative ideas within the scope of mBlock
EK 1.1-B-1
Identification of various sections of mBlock interface, including but not limited to scene, blocks, sprite, device, backdrop, and scripting area
EK 1.1-B-2
Naming, saving, and retrieving mBlock projects in local environment
EK 1.1-B-3
Naming, saving, and retrieving mBlock projects in the cloud
EK 1.1-C-1
Description of the two essential components of an online account - username and password
EK 1.1-C-2
Realization of common options for usernames, including email, phone number, and custom literals
EK 1.1-C-3
Identification of a weak passwords and its potential risks
EK 1.1-C-4
Generation of secure yet memorable passwords
EK 1.1-C-5
Logging in and out of cloud account in mBlock
EK 1.1-D-1
Description of design cycle and the scope of each of its processes - planning, making, sharing, and reflecting
EK 1.1-D-2
Description of the range of contents that go into design journals
EK 1.1-D-3
Personalization of design journals using provided materials creatively

Intro to mBlock

LA 1.1-1

Overview

In simple words, creative computing is about a high level view of computing devices where users see these devices as tools for releasing creativity as opposed to the more traditional way where users undergo the process of mastering a number of detailed set of skills from the ground up. In short, the main idea is to have young students get going with computers and programming as fast as possible without worrying too much about the nitty gritty or stunned by too much dry and boring content.

This activity is intended both as an ice-breaker and an informal introduction to computers, programming, and creative possibilities in the digital realm. It's important to encourage students' interest and passion through the idea that computers and programming could be seen as tools for creative expression. The activity will start with a general discussion about the various use cases of computers, then zoom in on the creative use cases, followed by a broad overview of mBlock.

Activity Description

1. Ask students about their experiences with computers
    a. How do you usually interact with computers?
    b. What do you usually do on computers?
    c. How many of those things involve being creative with computers?

2. Introduce students to creative computing with mBlock and the range of projects they will be able to create by showing some videos and sample projects. Explain that over the next several sessions they will be creating their own interactive computational media with mBlock

3. What will you create? Ask students to imagine what type of projects they want to create with mBlock

Resources

1. mBlock overview video - <Feishu Folder Location>
2. mBlock sample projects - <Feishu Folder Location>

Notes

1. Instead of writing out their answers to the discussion questions, encourage students to get creative by drawing their responses

Mentor Reflection

1. Were students able to brainstorm a diverse range of project ideas that are reasonably within the scope of mBlock? If not, try adding more diversity to the sample projects to give students a sense of the possibilities. Also, it is also useful to build some concept of the limitations of mBlock. For instance, mBlock versus photo editing softwares, music production softwares, film making softwares, or professional game engines

Cloud Account

LA 1.1-2

Overview

mBlock cloud accounts allow students to easily preserve, track, and share their work with peers and mentors across all supported devices and platforms. Moreover, having access to student projects is valuable to internal R&D sessions and teaching quality discussions. Therefore, setting up all student accounts and managing them systematically is crucial to this course and various other internal procedures.

In this day and age, as more and more aspects of life are serviced and managed by numerous apps ranging from banking to personal health, the ability to create and manage sufficiently secured accounts has almost become a skill. Young students should have the foundational knowledge of how accounts are created, how they are made personal, and how they are kept safe. In this activity, students will gather the essential information for account registration and practice the generation of secure passwords. Also, students will attempt several password cracking exercises to reinforce their idea about account security.

Activity Description

1. mBlock cloud accounts require an email address or a mobile phone number capable of receiving text verification. If students cannot provide a personal email address or have a phone with them during class, mentors should prepare in advance a sufficient number of email addresses and properly distribute and manage them

2. Help students navigate to the account signup page in mBlock. Be sure the account is registered to mBlock China instead of mBlock global. Encourage students to sign in and out of their accounts

3. Have students create mock projects to practice saving and locating their projects

4. To make it easier for members of the class to follow one another's mBlock profile, consider creating a class list of usernames (email/phone number) and passwords

5. Construct, as a class, a management guideline for account sharing, project naming, and other rules to better organize and secure all student works

Resources

1. Password management sheet - <Feishu Folder Location>

Notes

1. To remember passwords while maintaining privacy, have students write down their username and password in individually sealed envelopes or folders. Mentors are personally responsible for keeping these documents organized and secured

Mentor Reflection

1. Were students able to successfully create mBlock cloud accounts and sign in and out of them without looking at their sealed passwords?

Design Journal
LA 1.1-3

Overview

Although it is still quite a mystery how creative ideas are generated, the process of bringing an idea into reality and eventually perfecting it is necessarily systematic. Creative ideas are not only precious but also spontaneous, incomplete, and fragmented. So it is a good practice to record all these ideas, reflect on their tensions, potential, difficulty of implementation, and sketch out outlines of the final product. From planning to making to sharing, the design cycle is worth keeping a journal and often more valuable than the final result.

A common problem of note taking, either physically or digitally, is that it usually becomes more functional than personal. The difference between a journal and a regular notebook is that journals usually have more sentimental value which encourages a positive cycle of addition, reflection, and preservation. Therefore, this activity is designed to create a bond between the students and their design journals from the get go. Students are expected to deeply personalize their design journals using provided materials and share the joy of some good old handcrafting.

Materials

1. A variety of papers, markers, and craft materials commonly seen in Explorer classes

Activity Description

1. Introduce students to the idea of the design journal, a physical notebook where they can brainstorm ideas and share personal reflections, similar to a personal journal or diary. Explain that students will be prompted to update their design journals throughout their mBlock programming adventures, but encourage them to add to their journals anytime during the process of designing projects to capture ideas, inspiration, notes, sketches, questions, frustration, triumphs, and etc.

2. Browse through a gallery of sample design journals to get ideas for what type of design journals students would like to create. Give students time to start and personalize their design journals using provided materials

3. Ask students to create their first design journal post by answering some questions related to LA 1.1-1
   a. How would you describe mBlock to a friend?
   b. Write or sketch ideas for three different mBlock projects you are interested in creating

4. Encourage students to share their design journals and initial reflections with a neighbor

Resources

1. Sample design journals - <Feishu Folder Location>

Notes

1. Deign journals should be used often in future learning activities
2. Design journals could be used in private and public settings including one on one diagnosis and peer review

Mentor Reflection

1. What do the reflection responses tell you about students' interests?

Topic 1.2

Big Ideas

Creative Computing
Feedback

Learning Objectives (Students will be able to…)

LO 1.2-A
Reasonably describe some creative possibilities in mBlock
LO 1.2-B
Create a mBlock project that demonstrates some basic effects
LO 1.2-C
Describe the purpose of a critique group
LO 1.2-D
Give and receive feedback on their/others' mBlock project in their critique group

Essential Knowledge

EK 1.2-A-1
mBlock provides blocks that produce sound effects
EK 1.2-A-2
mBlock provides blocks that produce motion effects
EK 1.2-A-3
mBlock provides a library of built-in design elements (sprites/backdrops)
EK 1.2-A-4
Design elements in mBlock have properties such as size, position, and orientation
EK 1.2-B-1
The green flag in mBlock can be the trigger to start a program
EK 1.2-B-2
When green flag clicked block corresponds to the green flag under the stage
EK 1.2-B-3
Various blocks can be snapped together
EK 1.2-B-4
Projects in mBlock could be put into presentation mode
EK 1.2-C-1
Critique groups involve getting feedback on the negative, confusing, and positive aspects of a project
EK 1.2-C-2
Providing feedback in critique groups is based on a solid understanding of someone else's project
EK 1.2-C-3
Negative feedback is not about criticizing or personal attacks
EK 1.2-C-4
Feedback should be valuable and valued
EK 1.2-C-5
One can learn from others' projects in a critique group
EK 1.2-D-1
Understanding the feedback received in critique group
EK 1.2-D-2
Knowing how to act on the feedback received in critique group
EK 1.2-D-3
Utilize critique group handout to give and receive feedback

Brave Explorer

LA 1.2-1

Overview

It is in the nature of young students to explore and play around with things that they have never encountered before. Often times, this process could seem to be orderless and has no obvious underlying logic. At the end of the day, however, doing outweighs not doing. If students are restricted at the very beginning, some of them could become conditioned to following orders. This is certainly not a negative behavior is some circumstances. But since this is an introductory course, it is essential to keep the candle of curiosity lit and have all students be driven by their spirt of wonder and adventure.

Like in most future learning activities, mentors are mainly responsible for guiding students through a series of learning steps instead of having a strong presence within each step. Mentors should provide the structures for exploration, overcoming challenges, reflecting on mistakes, and sharing triumphs. In this activity, students will first play around in mBlock, then be challenged to make happen certain effects, and eventually enter a sharing session about what they have discovered.

Activity Description

1. Make sure all students are signed in to their mBlock accounts

2. Give students sufficient time to explore the mBlock interface in an open-ended way. Prompt students with keywords such as "fearlessly" and "surprising". Encourage students to work together, ask each other for help, and share what they are figuring out

3. Ask for volunteers to share with the entire class one thing they discovered

4. After the volunteers have shared, offer several challenges to the students
    a. Did anyone figure out how to add sound?
    b. Did anyone figure out how to change the background?
    c. Did anyone figure out how to change the size of a sprite?
    d. Dis anyone figure out how to start a new project?

Resources

1. Brave explorer handout - <Feishu Folder Location>

Notes

1. A major goal of this activity is to establish a culture of fearlessness, exploration, and peer collaboration. It is expected that students will not know much ahead of time - and the classroom environment becomes a space where everyone is learning together

Mentor Reflection

1. Do students know how to initiate a new project?
2. Do students understand the basic mechanism of snapping blocks together?

Critique Group

LA 1.2-2

Overview

Even though the bond between mentors and students is crucial to a successful learning environment, the dynamic between peers is unique and irreplaceable. Students are often more willing to share, voice concerns, and raise question with one another. Therefore, having a classroom culture for critique groups and regularly utilizing critique groups is invaluable to mentors.

Inside a critique group is a place where students give feedback to others' works. This process is about learning from others, praising others' brilliance, receiving others' suggestions, and bravely facing confusions. For this first critique group session, the subject matter will be the final result each student has finished from LA 1.2-1. Mentors are encouraged to provide critique group handouts to all students as discussion guideline.

Activity Description

1. Introduce students to the idea of a critique group, a small group of designers who share ideas and projects with one another in order to get feedback and suggestions for further development

2. Provide critique group handouts to all students

3. Divide students in small groups. In these critique groups, ask students to take turns sharing their ideas, drafts, or prototypes, for example, LA 1.2-1 projects

4. Let students gather feedback by having their critique group members respond to Red, Yellow, Green categories on their handouts. Encourage students to record other notes, feedback, and ideas in their design journals

Resources

1. Critique group handout - <Feishu Folder Location>

Notes

1. It can be valuable to have a dedicated group of peers to give you encouragement and feedback on your design iterations. Provide opportunities for students to continue meeting with their critique groups during all units
2. Some students could be sensitive to receiving too little positive comments or view suggestions in a negative light. Mentors should be aware of students' emotional responses during critique group sessions and manage student behaviors wisely

Mentor Reflection

1. Did all students have a chance to share their work and get feedback?
2. Was any student too shy or withdrawn during the process?
3. Did all student understand the feedback they received?

Unit 2 Wondering

Overview

Now that students have entered the world of mBlock and creative computing, what is best way of helping them get started? Well, the answer must lie between students being told what to do in a highly structured environment and student being left totally alone to explore under their own direction. For this unit, at least, students are expected to take on more of a self-directed route of open exploration. Students will develop a grasp on the most fundamental ideas of programming on their own by participating in and reflecting from various fun and exiting activities. A final project for this unit will be in interactive collage.

Concepts and Practices

The key computational concept in this unit is the concept of sequence - identifying and specifying an ordered series of instructions. Sequence forms the basis of program execution. In many ways, most other computational concepts in this course are about altering the sequence of a program. Sequence is also a powerful concept as students can now feel empowered by translating their ideas into blocks of computer code that tell the computer what to do. The practice of incremental iteration and testing and debugging are also lightly touched on in this unit. Young learners get easily carried away in programming by dragging and snapping blocks together without ever testing in the process. This is understandable but as the program grows larger, the difficulty of locating an error becomes harder as well. So It's best to catch bugs as early as possible in the development process.

Unit At a Glance

Topics	Keywords	Concepts	Practices
2.1	Decomposition Instruction Precision	SEQ	N/A
2.2	Simple Sequence Limited Scope	SQE, LOO	ICI
2.3	Debugging Analysis	SEQ, LOO	TND
2.4	Interactive Collage	SEQ, LOO	ICI, TND

Suggested Learning Activities

ID	Activity	Worksheet / Handout	Extra Materials
LA 2.1-1	Let's Dance	N	N
LA 2.2-1	Step-By-Step	Y	N
LA 2.2-2	10 Blocks	Y	N
LA 2.3-1	Debug It	Y	N
LA 2.4-1	About Me	Y	N

Topic 2.1

Big Ideas

SEQ
Decomposition
Atomic Operation
Communication Model

Learning Objectives (Students will be able to…)

LO 2.1-A
Express a complex activity using a sequence of instructions
LO 2.1-B
Appreciate the value of precise instructions
LO 2.1-C
Describe the drawbacks of being precise in terms of communication model
LO 2.1-D
Describe the difference between computers and humans in terms of communication model

Essential Knowledge

EK 2.1-A-1
Recognition of conceivable steps in a complex activity
EK 2.1-A-2
Factors that affect weather a single step in a complex activity should be decomposed further
EK 2.1-A-3
Organize the instructions for the steps in a complex activity in a concise and efficient manner
EK 2.1-A-4
The decomposed steps of a complex activity should take into account the effects of previous steps
EK 2.1-B-1
Precision implies the lack of confusion
EK 2.1-B-2
Precise instructions ensure the certainty and limit the scope of possible outcomes
EK 2.1-B-3
A series of precise instructions is more likely to produce the desired outcome compared to vague ones
EK 2.1-C-1
Precise instructions are difficult to construct in human communication
EK 2.1-C-2
Precision usually increases the total number of instructions for expressing an activity
EK 2.1-C-3
Precise instructions usually take more time to process for humans
EK 2.1-D-1
Computers rely heavily on precise instructions to produce a desired outcome which also needs to be precisely defined
EK 2.1-D-2
Computers do no more and no less than what they have been instructed
EK 2.1-D-3
Humans rely heavily on assumptions, emotions, expressions, body languages, and context to reason about an instruction

Let's Dance

LA 2.1-1

Overview

Computers are extremely good at following orders, too good perhaps that it comes at a cost. Being extremely good at following orders implies that instructions are carried out in the a certain sequence, doing nothing more or less than what's specified. This is drastically different from the communication model people engage in on a daily basis. For starters, communication among people involves a complex set of behaviors such as verbal language, body language, facial expressions, etc.. On top of this, meanings and intentions are usually implicit where the interpretations rely on assumptions, context, and emotions.

In this activity, students will experience the difficulty of carrying out precise behaviors under the human communication model. Through this experience, students are expected to understand the importance of sequence and precision to digital devices.

Activity Description

1. Ask students to get into director/follower pairs. Have a display ready to present the dance videos

2. For each pairs:
    a. Have the following partner facing away from the display and the directing partner (and the rest of the group) facing the display
    b. Show the video to the director and the group, but not to the follower
    c. Ask the director to describe to their partner (using only words) how to perform the sequence of dance moves shown in the video

3. Use this activity to start a discussion about the importance of sequence in specifying a set of instructions. You can let students reflect individually in their design journals or facilitate a group discussion by inviting different pairs to share their thoughts
    a. What was easy/difficult about being the director?
    b. What was easy/difficult about being the follower?
    c. What was easy/difficult about watching?
    d. How does this activity relate to what we're doing with mBlock?

Resources

1. Dance videos - <Feishu Folder Location>

Notes

1. This is one of several activities in this guide that are computer-free. Stepping back from the computer can support fresh perspectives on and new understandings of computational concepts, practices, and perspectives.
2. After the activity, have students reflect and write down step-by-step instructions for one the dances. In programming, this is sometimes called "pseudocode"

Mentor Reflection

1. Can students explain what is important about sequence when specifying instructions?
2. Do students realize the value of being specific and precise when giving instructions?

Topic 2.2

Big Ideas

SEQ
LOO
ICI

Learning Objectives (Students will be able to…)

LO 2.2-A
Develop understanding of an unknown block in a scientific way
LO 2.2-B
Describe the natural sequence of execution of a number of blocks snapped together
LO 2.2-C
Describe the workings of a counting repeat block
LO 2.2-D
Create a mBlock project with some sequential logic
LO 2.2-E
Appreciate the value of incremental iteration

Essential Knowledge

EK 2.2-A-1
The scientific method involves forming a hypothesis, testing the hypothesis, and iterating based on the observed outcome
EK 2.2-A-2
The written description of a block provides valuable information
EK 2.2-A-3
A hypothesis on the effects of an unknown block should be formed based on its written description
EK 2.2-A-4
A hypothesis on the effects of an unknown block should be tested and adjusted according to the result
EK 2.2-B-1
Blocks snapped together in mBlock are executed from top to bottom
EK 2.2-C-1
A counting repeat block executes all sandwiched blocks in their natural sequence for the specified number of times
EK 2.2-C-2
A counting repeat block can have other blocks snapped before or after it
EK 2.2-C-3
The number of times a counting repeat block repeats is a specifiable integer
EK 2.2-D-1
A storyline is useful when building a creative project
EK 2.2-D-2
A project plan or storyline should be decomposed into instructions that could be translated into available blocks
EK 2.2-D-3
Blocks should be snapped together in a sequence that is reflective of the desired steps
EK 2.2-E-1
As the complexity of a project increases, the number of blocks needed increases
EK 2.2-E-2
As the number of blocks in a project increases, the probability of error increases
EK 2.2-E-3
It is harder to locate an error among a large number of blocks than a small number of them
EK 2.2-E-4
Incremental iteration reduces the scope of errors by ensuring each step along the way behaves as expected

Step-By-Step
LA 2.2-1

Overview

In light of LA 2.1-1, this activity will ask students to apply their experience and understandings about sequence and instructions in mBlock by creating a dancing sprite animation. Mentors are encouraged to either implicitly or explicitly employ the practices of design cycle. In other words, students should start by generating a rough idea about what dance they want their sprite to do. Then deconstruct the idea into a sequence of reasonably detailed and actionable instructions. During the implementation process, mentors should encourage students to practice incremental iteration through experimenting and debugging.

Once students have a reasonably completed project, they are expected to expand on top of it with additional blocks. Using the step-by-step handout is recommended.

Activity Description

1. Make sure students are signed into their mBlock accounts and start a new project. Have the step-by-step handout available to guide students during the activity

2. Have students follow the handout to create an animated program. Encourage students to add other blocks and experiment with motion, sprites, looks, costumes, sound, or backdrops to make the project more polished

3. Let students share their first mBlock creations with one another.

4. Ask students to think back on the design process and reflect in their design journals or as a group discussion
    a. What was surprising about the activity?
    b. How did it feel to be led step-by-step through the activity?
    c. When do you feel most creative?

Resources

1. Step-by-step handout - <Feishu Folder Location>

Notes

1. Some disconnects might exist between the dance instructions during the planning phase and the available instructions/blocks during implementation. This is a common confusion among students that mentors should be aware of

Mentor Reflection

1. Were students fluent in signing in and starting a new mBlock project?
2. Were students able to create a dancing sprite?
3. Were students able to save and share projects?

10 Blocks

LA 2.2-2

Overview

For young students, it is important to avoid being overwhelmed by too much new information. Especially in programming, when there is an endless number of different devices, blocks, and use cases, limiting the scope of a learning activity could be surprisingly effective sometimes. A limited scope motivates students to develop deeper understanding about the content they are working with. Moreover, creativity really takes the center stage when the available resource is constrained.

In this activity, students are expected to realize the advantages and disadvantages of using only 10 blocks to create a project. Students are encouraged to use all of the 10 blocks at least once and get feedback from their critique groups. Using the 10 blocks handout is recommended.

Activity Description

1. Make sure students are signed into their mBlock accounts and start a new project. Have the 10 blocks handout available to guide students during the activity

2. Give students time to create a project with only these 10 mBlock blocks: go to, glide, say, show, hide, set size to, play sound until done, when this sprite clicked, wait, and repeat. Remind students to use each block at least once in their project and encourage them to do more with different sprites, costumes, or backdrops

3. Invite students to share their projects in their critique groups

4. Ask students to think back on the design process in their design journals or in a group discussion
    a. What was difficult about being able to use only 10 blocks?
    b. What was easy about being able to use only 10 blocks?
    c. How did it make you think of things differently?

Resources

1. 10 blocks handout - <Feishu Folder Location>
2. Critique group handout - <Feishu Folder Location>

Notes

1. It's surprising how much one can do with just 10 blocks. Take this opportunity to encourage different ideas and celebrate creativity by inviting students to present their projects in front of the class

Mentor Reflection

1. Do projects include all 10 blocks?
2. How do different students react to the idea of creating with constraints? What might this tell you about how this student learns?

Topic 2.3

Big Ideas

SEQ
LOO
TND

Learning Objectives (Students will be able to…)

LO 2.3-A
Describe what a programming bug is
LO 2.3-B
Implement fixes to a bug
LO 2.3-C
Realize there could be numerous programming solutions to the same problem
LO 2.3-D
Describe a number of debugging strategies

Essential Knowledge

EK 2.3-A-1
A bug is a term used in the programming world to describe an error in a program
EK 2.3-A-2
There are many types of bugs that can exist in a program such as syntax and logic
EK 2.3-A-3
Debugging is the process of identifying, locating, and fixing a bug
EK 2.3-B-1
A bug must be identified and located through testing and reasoning
EK 2.3-B-2
Solutions to a bug must also be tested to avoid bugs in themselves
EK 2.3-B-3
The successfulness of a bug solution depends on the specification of desired output
EK 2.3-C-1
Computational concepts are used and implemented differently by different people
EK 2.3-C-2
Just like the same message could be communicated in various ways in natural language, programs can achieve the same output through various implementations
EK 2.3-C-3
All programming solutions are not created equal. They differ in efficiency and reasonability
EK 2.3-D-1
Tracing in debugging is largely about generating a sequence of logical descriptions of what's happening in a program at every moment in time
EK 2.3-D-2
Logging in debugging is about explicitly displaying certain information during program execution that are otherwise implicit in order to aim program analysis
EK 2.3-D-3
Problem simplification in debugging is about analyzing programming instructions in sections and generating a high level description for each section. This is particularly useful for catching logic errors

Debug It

LA 2.3-1

Overview

Testing and debugging is a crucial ability to develop in programming. Testing and debugging requires you to first understand a given program, reason about its expected output, compare the expected output to the actual output, identify the problem, and come up with effective solutions. As you can see, a debugging exercise involves or tests a lot of useful perspectives of programming, and this why there will be a number of learning activities related to testing and debugging in the future.

Students will be given a series of either completed programs or code segments to "fix". Weather a "fix" is successful is judged according to specific output criteria. As students improve their testing and debugging practices, they will generally become more fluent in understanding others' programs and modifying or expanding on others' works.

Activity Description

1. Have the debug it handout available to guide students during the activity

2. Guide students to implement the given programs in mBlock and perform testing and debugging on their computers. Make sure students save each corrected program as a new project.

3. Ask students to reflect back on their testing and debugging experiences in their design journals or in a group discussion
    a. What was the problem/bug?
    b. How did you identify the problem/bug?
    c. Did others have alternative approaches to fixing the problem/bug?

4. Generate a list of debugging strategies by collecting students' problem finding and problem solving approaches

Resources

1. Debug it handout - <Feishu Folder Location>

Notes

1. This activity can also work in groups. If mentors decide to put students in groups for this activity, please be aware of passive students who don't contribute as much as expected. Pairing strategies should also be considered
2. Testing and debugging is probably the most common activity of programmers. Things rarely work as planned, so developing a set of testing and debugging strategies will be beneficial to any computational creator
3. Some common strategies involved in testing and debugging are information logging, tracing, and problem simplification

Mentor Reflection

1. Were students able to solve all five bugs? If not, how might you clarify the concepts expressed in the unsolved programs?
2. What different testing and debugging strategies did students employ?

Topic 2.4

Big Ideas

SEQ
LOO
ICI
TND

Learning Objectives (Students will be able to…)

LO 2.4-A
Become familiar with a wider range of mBlock blocks
LO 2.4-B
Create an interactive mBlock project that has a solid logical context
LO 2.4-C
Polish their project through testing and debugging

Essential Knowledge

EK 2.4-A-1
When key pressed block can be triggered through keyboard
EK 2.4-A-2
When this sprite clicked block can be triggered by mouse clicks
EK 2.4-A-3
Blocks in the looks category can change properties of sprites and have them say things in a chat bubble next to them
EK 2.4-A-4
Wait blocks can pause the execution of a program
EK 2.4-A-5
Forever blocks act just like counting repeat blocks with the number of repeats set to infinity
EK 2.4-B-1
An interactive project is a project where users can trigger certain effects through additional input such as key presses or mouse clicks
EK 2.4-B-2
A creative computing project with solid logical context implies a clear theme that is expanded through reasonable and relatable design elements and effects
EK 2.4-C-1
A completed project should go through thorough testing
EK 2.4-C-2
As the complexity of a project increases, the debugging process could produce ripple effects
EK 2.4-C-3
Ripple effects occur when the solution to a particular bug is successful but causes errors to appear other parts of the program
EK 2.4-C-4
A project is considered polished after it has gone through multiple iterations of thorough testing and debugging with no more error found

About Me

LA 2.4-1

Overview

This activity is about giving students the opportunity to create their first deliverable project. As a deliverable project, it should be grounded in a relatively logical context and implemented with a relatively complete function set. And being the first project of this kind, working with a context that is deeply relatable could be helpful to students to overcome their fear of failure. In other words, the "about me" context doesn't invoke too much creative pressure so that students can focus on the actual iterations of the project.

Depending on students' progress, a good starting point for this project could be an interactive collage, which is simply a collection of a number of clickable sprites representing things in students' lives. Mentors should make plans in advance to have students be ready to present their projects to their parents.

Activity Description

1. Introduce students to the concept of the interactive collage, a mBlock project that represents aspects of themselves through clickable sprites. Optionally, show some sample projects to students to invoke their creativity

2. Make sure students are signed into their mBlock accounts and start a new project. Have the about e handout available to provide guidance.

3. Give students time to create an about me interactive collage mBlock project, encouraging them to build up their programs by experimenting and iterating

4. Allow students to share their works-in-progress with others

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. What are you most proud of? Why?
    b. What did you get stuck on? How did you get unstuck?
    c. What might you want to do next?
    d. What did you discover from looking at others' about me projects?

Resources

About me handout - <Feishu Folder Location>

Notes

Sample projects can simultaneously inspire and intimidate, open the creative space and constrain it. Encourage a wide range of creations; diversity is great
Students can further personalize projects by using a camera or webcam to bring images into the project
Don't forget to make real use of students' design journals. Instead of constantly providing direct answers to questions from students, ask them to record questions in their design journals and ask help from their peers

Mentor Reflection

Do projects make creative use of sprites, costumes, looks, backdrops, or sound?
Are projects interactive? Can users interact with various elements within the project?

Unit 3 Bringing Life

Overview

Coming from unit 2, students have experienced just how much they can do with just a few sprites and a small number of programming blocks creatively sequenced. Nevertheless, in order to bring more liveliness into a project, its sprites must first come alive. Animation is a beautiful idea because to the surprise of many, its principles are quite simple. Quickly cycling through a series of static pictures, a perfect job description for computers. This is partly why students can achieve animated effects quite easily in mBlock. This unit will focus on helping students complete a final project that is both interactive and animated.

Concepts and Practices

As mentioned in unit 2, a lot of computational concepts are closely related to sequence. Loop, for example, is about altering the sequence to go into a cycle. Parallelism, on the other hand, is about the execution of multiple sequences at the same time or creating the illusion of it. And event is often used to define the starting condition of program sequences. Parallelism and event are useful concepts to developing an animated and interactive project. Incremental iteration and testing and debugging are still the practices students should focus on in this unit.

Unit At a Glance

Topics	Keywords	Concepts	Practices
3.1	Acting Two Things At Once Sounds	PRL, EVN	ICI
3.2	Paint Editor Visual Effects	SEQ	ICI, TND
3.3	Flipbook Costumes	SEQ, LOO, PRL	ICI
3.4	Debugging	SEQ, LOO	TND
3.5	Animated Interactive	SEQ, LOO, EVN, PRL	ICI, TND

Suggested Learning Activities

ID	Activity	Worksheet / Handout	Extra Materials
LA 3.1-1	Act It Out	N	N
LA 3.1-2	Build-A-Band	Y	N
LA 3.2-1	Colors & Shapes	N	N
LA 3.3-1	It’s Alive	N	N
LA 3.4-1	Debug It	Y	N
LA 3.5-1	Music Video	Y	N

Topic 3.1

Big Ideas

PRL
EVN
ICI

Learning Objectives (Students will be able to…)

LO 3.1-A
Describe what events are and how they work in mBlock
LO 3.1-B
Describe what parallelism is and how it works in mBlock
LO 3.1-C
Describe the notion of “reset” in programming
LO 3.1-D
Create an interactive mBlock project that involves complex sequencing

Essential Knowledge

EK 3.1-A-1
An event is an action or the occurrence of a set of well defined conditions
EK 3.1-A-2
Events are often used as triggers for program execution
EK 3.1-A-3
Typical events in mBlock include when green flag clicked, when key pressed, when this sprite clicked, and when I receive message
EK 3.1-B-1
A single sprite in mBlock can have multiple event triggers of the same or different events
EK 3.1-B-2
As long as the event takes place or the defined conditions are met, programming instructions under the event blocks will execute in their natural sequence
EK 3.1-B-3
Parallelism is the effect of more than one set of instructions executing at the same time or multiple actions taking place at the same time
EK 3.1-B-4
Parallelism can be achieved by having multiple event blocks of the same event in a sprite
EK 3.1-B-5
Parallelism can also be achieved by careful sequencing of independent programming blocks
EK 3.1-C-1
The states of various design elements including sprites are usually altered during program execution. These states might include position, orientation, size, and more
EK 3.1-C-2
Altered states of design elements during program execution are not automatically undone at the end of the execution
EK 3.1-C-3
Reset is the notion of putting design elements back into their original states usually at the beginning of a program
EK 3.1-D-1
Wait blocks can pause the execution of a set of instructions. Combined with parallelism, multiple design elements in a project can appear to be coordinated
EK 3.1-D-2
Wait blocks also take effect inside loops
EK 3.1-D-3
Complex sequencing involves incremental iteration of detailed analysis of the sequential flow of a program that might involve parallelism and coordination of instructions across various design elements

Act It Out

LA 3.1-1

Overview

Similar to LA 2.1-1, this activity asks students to perform certain instructions. However, instead of verbal instructions, students will act out actual scripts in mBlock. Mentors and/or students will write short and simple programs for the actors to read and perform. The key concept in this activity is parallelism and events. Script writers should aim to have the actors perform two actions simultaneously both independently and reactively.

Parallelism is about more than one action happening at the same time. The key understanding about parallelism is that the set of actions must usually be unrelated or independent. For instance, walking to the left and right and the same time is obviously unrealistic. But walking and talking at the same time is completely natural.

Events, on the other hand, are triggers that decides the timing of certain instructions and aids the logical sequencing of a program.

The success of activities like this relies heavily on detailed preparation and planning from mentors.

Activity Description

1. Have a display ready to display which blocks and scripts will be performed

2. Ask for pairs of volunteers

3. Prompt the two volunteers to act out a series of instructions
   a. Have one person do one thing
   b. Have that person "reset"
   c. Have one person do two things simultaneously
   d. Have the second person do an independent task
   e. Have the second person do a dependent task (responding to first person)

4. Reflect on the experience as a group to discuss the concepts of events and parallelism
   a. What are the different was that actions were triggered?
   b. What are the mechanisms for events in mBlock?
   c. What were the different ways in which things were happening at the same time?
   d. What are the mechanisms that enable parallelism in mBlock?

Resources

1. Sample scripts - <Feishu Folder Location>

Notes

1. The activity involves the notion of "reset", which is something beginners often overlook. If they want things to start in a particular location, with a particular look, etc., students need to understand that they are completely responsible for programming those setup steps
2. This activity can be useful for demonstrating the broadcast and when I receive block pair

Mentor Reflection

1. Can students explain what events and parallelism are and how they work in mBlock?

Build-A-Band

LA 3.1-2

Overview

It wouldn't be right for creative computing to leave music out. At the point, students should already be comfortable exploring in mBlock and piecing blocks together in some logical order. Since the concepts of parallelism and events were just introduced in LA 3.1-1, this is a great opportunity to have students apply and polish their understandings.

In this activity, students will create an upgraded interactive collage of a music band. Students are expected to create various interactive instruments, possibly along with musicians on a stage. Each instrument should be able to be played independently and optionally be orchestrated as a whole.

This activity is extremely extensible. Mentors can even have the entire class work together and create a class band. Using the build-a-band handout is recommended.

Activity Description

1. Make sure students are signed into their mBlock accounts and have the build-a-band handout available to guide students

2. Give students time to create interactive instruments by pairing sprites with sounds. Encourage them to experiment with different ways to express sounds in mBlock by exploring other blocks in the sound category or using the editing tools within the sounds tab

3. Allow students to demonstrate their bands to one another or let students walk around to interact with others' bands. A gallery walk is recommended where students put their projects in presentation mode and then invite them to walk around and explore each other's projects

4. Ask students to think back on the design process in their design journals or in a group discussion
    a. What did you do first?
    b. What did you do next?
    c. What did you do last?

Resources

1. Build-a-band handout - <Feishu Folder Location>

Notes

1. To share as a whole group, have students perform their mBlock instruments together to form a class band
2. Encourage use of events and parallelism

Mentor Reflection

1. Do projects make creative use of sounds?
2. Are the sprites in the projects interactive?
3. Are events utilized?
4. Is parallelism demonstrated?

Topic 3.2

Big Ideas

SEQ
ICI
TND

Learning Objectives (Students will be able to…)

LO 3.2-A
Become familiar with a wider range of mBlock blocks
LO 3.2-B
Import and incorporate custom design elements into a mBlock project
LO 3.2-C
Creative use of the paint editor in mBlock
LO 3.2-D
Create a mBlock project with custom design elements and various visual effects

Essential Knowledge

EK 3.2-A-1
The change color effect by block changes the hue of the sprite
EK 3.2-A-2
The change fisheye effect by block gives the impression of a sprite being seen through a wide-angle lens
EK 3.2-A-3
The change whirl effect by block twists the sprite around its center point, therefore distorting the sprite
EK 3.2-A-4
The change ghost effect by block modifies the transparency of the sprite
EK 3.2-A-5
The change mosaic effect by block shows multiple smaller images of the sprite, therefore creating a “mosaic” effect
EK 3.2-B-1
Getting design elements from the internet involves searching, saving, and format considerations
EK 3.2-B-2
Location of the upload option in mBlock
EK 3.2-B-3
Disk and file system structure for common operating systems like Mac and Windows
EK 3.2-C-1
When different drawing tools are selected in the paint editor, the effect of clicking and dragging on the canvas is different as well
EK 3.2-C-2
Blank fills can be used creatively to produce cutout effect on shapes
EK 3.2-C-3
Regular shapes like circles and rectangles can be modified into irregular shapes by using the reshape tool
EK 3.2-C-4
When shapes are selected by the selection tool, their orientation and size could be modified
EK 3.2-D-1
Custom design elements react just the same as built-in sprites to programming blocks
EK 3.2-D-2
Combining loops and looks effect blocks often produce surprising results
EK 3.2-D-3
Incrementally test each design element and make sure they behave as expected will reduce the difficulty of debugging when the project is complete

Colors & Shapes
LA 3.2-1

Overview

LA 3.1-2 largely covers sound related blocks and concepts in mBlock. Now it's the paint editor's turn. Although mBlock's built-in sprite library is quite comprehensive and students can always utilize the internet for design elements, nothing can replace the value of a custom drawn sprite.

The paint editor in mBlock is relatively primitive. But with simple shaping tools and coloring technique, amazing designs could still be achieved. It is quite often that students produce better results than their mentors in an activity like this.

Students are expected to experiment with all tools in the paint editor and combine different shapes, fills, outlines to get surprising results. Students can then animate their designs using blocks in the looks category.

Activity Description

1. Show sample projects to inspire students

2. Make sure students are signed into their mBlock account and start a new project

3. Give students time to create a project that include various colored shapes. Invite students to experiment with looks blocks and the paint editor to explore their artistic abilities

4. Encourage students to share their creative work with others. A gallery walk is recommended (LA 1.1-3)

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. How did you incorporate different shapes and colors into your project? Where did this idea come from?
    b. What was challenging about this activity?
    c. What was surprising about this activity?

Resources

1. Sample colors and shapes projects - <Feishu Folder Location>

Notes

1. mBlock supports both bitmap and vector graphics. Help students navigate to the vector mode or bitmap mode button in the paint editor to design and manipulate different types of images and text. Mentors should avoid getting into too much technical details

Mentor Reflection

1. Do projects include various shapes and colors?
2. Do students utilize effect blocks in looks category?

Topic 3.3

Big Ideas

SEQ
LOO
PAR
ICI

Learning Objectives (Students will be able to…)

LO 3.3-A
Describe the principles of animation
LO 3.3-B
Explain the difference between sprites and costumes
LO 3.3-C
Create an animated mBlock project with custom design elements

Essential Knowledge

EK 3.3-A-1
A flipbook is a series of pictures on a stack of paper, bound together. When you flip the paper rapidly with your thumb, it looks like the images are moving
EK 3.3-A-2
Just like a flipbook, the effect of animation is achieved by a rapid succession of sequential images that minimally differ from each other.
EK 3.3-A-3
The smoothness of an animation is mainly determined by how much each image differ from one another and how fast they are being played
EK 3.3-B-1
A single sprite in mBlock can have multiple costumes
EK 3.3-B-2
A costume of a sprite can be thought of as one of many poses the sprite has
EK 3.3-B-3
A costume of a sprite can also be thought of as one of many costumes an actor might wear in a movie
EK 3.3-C-1
When designing the costumes for an animation effect, the positioning of each costume in the paint editor should be carefully considered
EK 3.3-C-2
The starting and finishing costumes in an animation are helpful in deciding what costumes should come in-between
EK 3.3-C-3
mBlock does not automatically cycle through the costumes of a sprite
EK 3.3-C-4
Combining loops and next costume block is a common way to achieve an animated effect
EK 3.3-C-5
Loop frequency could be decreased by incorporating wait blocks inside a loop
EK 3.3-C-6
Costume designs should be incrementally tested to ensure each costume produces the desired result
EK 3.3-C-7
The code segment that is responsible for cycling through various costumes can be separated out from other actions through parallelism

It's Alive

LA 3.3-1

Overview

If designing a sprite is exciting, then bringing a sprite alive would be thrilling. To a lot of young students, the magic behind animations might still be a mystery. The concept of creating a series of different poses/costumes of the same character and quickly cycling through them can generate an animated effect is simple, beautiful, and profound.

Animations are useful in creating a more realistic, engaging, and entertaining interactive experience. Therefore, knowing how to efficiently create a set of costumes for a sprite to achieve a desired animated effect is valuable in creative computing. This activity will ask students to either pick or create a sprite, then have them imagine an animated effect to be achieved by adding a number of costumes to their sprite.

Activity Description

1. Introduce the concept of an animation as looping through a series of incrementally different pictures, such as in a flipbook or a claymation film

2. Show sample projects to inspire students

3. Make sure students are signed into their mBlock accounts and start a new project

4. Encourage students to explore loops by changing costumes or backdrops to create an animation

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. What is the different between a sprite and a costume?
    b. What is an animation?
    c. List three ways you experience loops in real life

Resources

1. Sample it's alive projects - <Feishu Folder Location>

Notes

1. The different between sprites and costumes is often a source of confusion. The metaphor of actors wearing multiple costumes or performing different poses can help clarify the difference
2. Students can animate their own image by taking pictures of themselves using a camera or webcam

Mentor Reflection

1. Can students distinguish sprites and costumes?
2. Some students are particularly interested in developing animation projects and prefer to spend their time drawing and designing sprites, costumes, or backdrops. How might you engage students in both the aesthetic and technical aspects of projects?

Topic 3.4

Big Ideas

SQE
LOO
TND

Learning Objectives (Students will be able to…)

LO 3.4-A
Describe the difference between timed and untimed blocks
LO 3.4-B
Describe the workings of repeat until blocks
LO 3.4-C
Employ problem simplification in debugging

Essential Knowledge

EK 3.4-A-1
Some blocks in mBlock have two versions. One without a specified time and the other with a certain time limit
EK 3.4-A-2
An untimed block doesn't halt the execution of the program. But the effects of an untimed block do not end until explicitly specified
EK 3.4-A-3
A timed block will halt the execution of the program and carry out its effects until the specified time is met. The effects of a time block will end once the time limit is up
EK 3.4-B-1
A repeat until block is similar to a forever loop but with an exit condition. In other words, a repeat until block repeats the sandwiched blocks until a certain condition is met
EK 3.4-B-2
Repeat until blocks can also have other blocks snapped before or after them
EK 3.4-B-3
Repeat until blocks behave the same as forever loops if the exist conditions are never met
EK 3.4-C-1
When a program is lengthy, a good way to perform debugging is to first test the program in sections.
EK 3.4-C-2
Obverse how sections of a program behave and generate a clear description for the behaviors of each section
EK 3.4-C-3
Based on the descriptions of each section of a program, observe any logical error
EK 3.4-C-4
If no logical error exists in a lengthy program, zoom in on the potential sections of codes that have matching descriptions of the buggy behavior

Debug It

LA 3.4-1

Overview

Same as LA 2.3-1, this activity iterates on a different set of testing and bugging problems. The general purpose and strategies are unchanged.

Activity Description

1. Have the debug it handout available to guide students during the activity

2. Guide students to implement the given programs in mBlock and perform testing and debugging on their computers. Make sure students save each corrected program as a new project.

3. Ask students to reflect back on their testing and debugging experiences in their design journals or in a group discussion
    a. What was the problem/bug?
    b. How did you identify the problem/bug?
    c. Did others have alternative approaches to fixing the problem/bug?

4. Generate a list of debugging strategies by collecting students' problem finding and problem solving approaches

Resources

1. Debug it handout - <Feishu Folder Location>

Notes

1. This activity can also work in groups. If mentors decide to put students in groups for this activity, please be aware of passive students who don't contribute as much as expected. Pairing strategies should also be considered
2. Testing and debugging is probably the most common activity of programmers. Things rarely work as planned, so developing a set of testing and debugging strategies will be beneficial to any computational creator
3. Some common strategies involved in testing and debugging are information logging, back tracking, problem simplification, and simulation
4. Facilitate this activity in a whole group by having students act out the problematic programs in a similar way to LA 3.1-1

Mentor Reflection

1. Were students able to solve all five bugs? If not, how might you clarify the concepts expressed in the unsolved programs?
2. What different testing and debugging strategies did students employ?

Topic 3.5

Big Ideas

Copyright
SEQ
LOO
EVN
PAR
ICI
TND

Learning Objectives (Students will be able to…)

LO 3.5-A
Explain the purpose of copyrights
LO 3.5-B
Record songs or sound effects in mBlock and incorporate them in a project
LO 3.5-C
Create an animated interactive mBlock project that utilizes all concepts covered in unit 3
LO 3.5-D
Iterate on their projects based on feedback from peers

Essential Knowledge

EK 3.5-A-1
Creativity is valuable and should be valued
EK 3.5-A-2
Creating and remixing others' work is different from stealing others' work
EK 3.5-A-3
Using others' creative ideas without giving credit or greatly modifying them is usually a violation of their copyrights
EK 3.5-B-1
Location of the record option under sound tab
EK 3.5-B-2
mBlock provides basic sound editing tools such as trimming, tempo changing, volume changing, and sounds effects
EK 3.5-B-3
Recorded sound effects behave just the same as built-in sound effects
EK 3.5-C-1
Code segments that are responsible for producing independent results can usually be separated into numerous parallelized sequences
EK 3.5-C-2
Motion, animation, and sound are three aspects of a project that could usually be implemented independently.
EK 3.5-C-3
When a project has interactive features during program execution, it's important to consider weather the triggered actions will cause errors or conflicts to normal program execution
EK 3.5-D-1
Consider negative feedbacks as debugging challenges
EK 3.5-D-2
When something is confusing in your project to others, consider its presentation style and necessity
EK 3.5-D-3
The feedback based iteration process involves continuous back and forth communication with peers

Music Video

LA 3.5-1

Overview

Now that students have experience with sounds, sprites, and animations, a natural step moving forward would be to have the students direct their own music videos. A music video would involve everything students have learned in unit 3 and be a great deliverable project.

With their current toolkit, students can get real creative with designing the stage, the artist, instruments, sounds, and animations. Mentors should aim to provide sufficient time for students to realize their creativity while manage to have enough time for project delivery to parents. As this is a summarizing activity/project for unit 3, mentors should also be extra aware of potential weaknesses in students' learning progress in order to address them in a timely manner.

Activity Description

1. Introduce students to the idea of creating a music video in mBlock that combines music with animation and custom drawn sprites. Optionally, show a few sample projects to inspire them

2. Make sure students are signed into their mBlock account and start a new project

3. Give students open-ended time to work on their projects, optionally with the music video handout available to provide guidance

4. Help students give and receive feedback while developing their projects. Utilizing critique groups is recommended

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. What was a challenge you overcame? How did you overcome it?
    b. What is something you still want to figure out?
    c. How did you give credit for ideas, music, or code that you borrowed to use in your project?

Resources

1. Music video handout - <Feishu Folder Location>
2. Sample music video projects - <Feishu Folder Location>
3. Critique group handout - <Feishu Folder Location>

Notes

1. To further personalize projects, help students include a favorite song or record themselves singing or playing an instrument, using features under the sounds tab
2. This activity provides an opportunity for mentors to start a discussion about copyrights, giving credit and attribution

Mentor Reflection

1. Do the projects combine sprites and sound?
2. What parts of the projects did students choose to animate?
3. Are there certain blocks or concepts introduced up until now that students might still be struggling with? How might you help?

Unit 4 Story Time

Overview

Getting things to move, dance, sing, and animate in mBlock is good and all. But without a guiding theme, creative elements are just disconnected pieces of gems. An engaging storyline is the string that can often turn pieces of gems into a dazzling necklace. A storyline is aided by its characters, conversations, and environment. Therefore, programming blocks that are potentially useful in creating those aspects of a story are covered in this unit. All in all, students will develop their abilities to plan, create, debug, and reflect on a more comprehensive project than what they have done before. The final project of this unit will be a collaborative storytelling project.

Concepts and Practices

Building on other people's work has been a longstanding practice in programming, and has only been amplified by the internet that provides access to a wide range of other people's work. An important goal of creative computing is to support connections between students through reusing and remixing. Therefore, reusing and remixing is the main computational practice students should focus on. As for concepts, sequence, event, and parallelism will still take the center stage in each topic.

Unit At a Glance

Topics	Keywords	Concepts	Practices
4.1	Modularizing Make A Block Actions	SEQ, EVN, PRL	ANM
4.2	Synchronization Wait Broadcast	EVN, PRL	RNR
4.3	Backdrop	SEQ, EVN, PRL	ICI
4.4	Debugging	EVN, PRL	ANM, TND
4.5	Collaboration Storytelling	EVN, PRL	RNR

Suggested Learning Activities

ID	Activity	Worksheet / Handout	Extra Materials
LA 4.1-1	Characters	Y	N
LA 4.2-1	Conversations	Y	N
LA 4.3-1	Scenes	N	N
LA 4.4-1	Debug It	Y	N
LA 4.5-1	Our Creature	N	Y
LA 4.5-2	Pass It On	Y	N

Topic 4.1

Big Ideas

SEQ
EVN
PRL
ANM

Learning Objectives (Students will be able to…)

LO 4.1-A
Design their own mBlock blocks that achieve well defined behaviors
LO 4.1-B
Describe the drawbacks of a lengthy program
LO 4.1-C
Appreciate the value of abstraction and modularizing
LO 4.1-D
Create interactive sprites in mBlock with custom designed behaviors using make a block feature

Essential Knowledge

EK 4.1-A-1
Location of make a block feature
EK 4.1-A-2
Naming a custom block is as important as its implementation. Block names should be meaningful and precise
EK 4.1-A-3
The definition of a custom block is different from the actual custom block. There can only be one definition of a custom block, but once defined a custom block could be used as many times as needed
EK 4.1-A-4
A custom block could be thought of as one block containing many other blocks
EK 4.1-B-1
A lengthy program is difficult to analyze, comprehend, and debug
EK 4.1-C-1
A custom block defines a set of instructions that could be used over and over again
EK 4.1-C-2
A custom block provides the opportunity to use naming as description for a set of instructions
EK 4.1-C-3
Custom blocks can usually reduce the length of a program
EK 4.1-C-4
A well designed program with custom blocks often reads like a recipe
EK 4.1-D-1
Custom blocks are tied to specific sprites. A custom block created for one sprite is not available for another sprite
EK 4.1-D-2
Custom blocks can be triggered by events

Characters

LA 4.1-1

Overview

Since unit 4 is largely a preparatory unit for unit 5 which is about game design, most activities in this unit help students develop computational concepts and practices in the context that is helpful for designing games. In this activity, for example, students will learn about creating their own blocks which represent various actions their sprites can potentially carry out.

The exact actions student are expected to implement through make a block is not specified in this activity. However, mentors should prompt students with possible routes they can take to better prepare them for unit 5.

Custom blocks have some profound impact on programming. Just on the surface level, custom blocks make code segments or entire programs more readable and reusable. Mentors should explicitly explain these implications to students. Using the characters handout is recommended.

Activity Description

1. Optionally, show sample projects and have the characters handout available to guide students

2. Give students time to create their own mBlock blocks using the Make a Block feature found in the My Blocks category. Instruct them to design two sprites/characters that each have two actions/behaviors. Conduct a walkthrough of the Make a Block feature together as a class, if needed

3. Allow students to share their characters and behaviors with one another. Invite students to present their work to the class and demonstrate how they implemented the Make a Block feature

4. Ask students to think back on the design process in their design journals or in a group discussion
    a. How would you explain Make a Block to someone else?
    b. When might you use Make a Block?

Resources

1. Characters handout - <Feishu Folder Location>
2. Sample characters projects - <Feishu Folder Location>

Notes

1. If students are struggling with figuring out how to use the Make a Block feature, invite the entire class to do a walkthrough of the feature together

Mentor Reflection

1. Do projects include two sprites that each have two behaviors using the Make a Block feature?
2. Can students explain how to use the Make a Block feature to each other and to you?

Topic 4.2

Big Ideas

Synchronization
EVN
PRL
RNR

Learning Objectives (Students will be able to…)

LO 4.2-A
Organize advanced instruction sequences using wait blocks
LO 4.2-B
Organize advanced instruction sequences using broadcast blocks
LO 4.2-C
Remix a given project to meet a different set of criteria
LO 4.2-D
Describe the advantages and disadvantages of remixing a project

Essential Knowledge

EK 4.2-A-1
When the actions of more than one sprite need to be coordinated or synchronized, wait blocks can be weaved into the instructions of both sprites to have one sprite wait for the actions of the other
EK 4.2-B-1
Synchronization becomes exponentially harder as the number of design elements increases. Broadcast blocks are useful as broadcasts could be received by all other sprites at once. And sprites can react to different broadcast messages accordingly
EK 4.2-C-1
Just like debugging, the basis for a successful remix is a comprehensive understanding of the original project
EK 4.2-C-2
Redesigning parts of the original project is sometimes more efficient than modifying existing codes
EK 4.2-C-3
Design elements can be added to or subtracted from the original project during a remix
EK 4.2-C-4
A remixed project should also be made public if the original project is publicly available
EK 4.2-D-1
Remixing a project involves a program analysis overhead
EK 4.2-D-2
If the original project provides good infrastructure for building the new project, then remixing can be fun and efficient
EK 4.2-D-3
If the original project is not a good fit as the foundation for the new project, then remixing can be challenging
EK 4.2-D-4
The idea of a new project is not always necessary when remixing a project. Inspirations generated from the original project can be natural and a good basis for remix

Conversations
LA 4.2-1

Overview

This activity builds on top of the key concepts introduced in LA 3.1-1, which is events and parallelism. Events and parallelism are extremely useful in game design and are heavily used to coordinate or synchronize instruction sequences.

Students will first explore a given starter program that involves two characters talking a joke. They will then analyze how the joke is being synchronized in the given program. Mentors will then challenge students to use broadcasting as an alternative to the discovered approach. Students are encouraged to reuse and remix the given program as much as possible, combining knowledge from previous topics. Using the conversations handout is recommended.

Activity Description

1. Ask students to explore the Penguin Joke starter project as a group and have the conversations handout available to guide students

2. Make sure students are signed into their mBlock accounts and start a new project

3. Invite students to see inside the Penguin Joke starter project to observe how the conversation is animated and synchronized using wait blocks. Have students redesign the project to coordinate the conversation using the broadcast, broadcast and wait, and when I receive blocks

4. Encourage students to share their joke projects with one another. Invite students to present their work to the class and demonstrate how they implemented broadcast

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. How would you describe broadcast to someone else?
    b. When would you use timing in a project? When would you use broadcasting?

Resources

1. Conversations handout - <Feishu Folder Location>
2. Penguin joke starter project - <Feishu Folder Location>
3. Broadcast example project - <Feishu Folder Location>

Notes

1. If students are having trouble understanding how to use the broadcast and when I receive block pair, invite them to explore the broad cast example project in groups

Mentor Reflection

1. Do projects use broadcast and when I receive blocks?
2. Can students explain how to use the broadcast, broadcast and wait, and when I receive blocks in their own words?

Topic 4.3

Big Ideas

SEQ
EVN
PRL
ICI

Learning Objectives (Students will be able to…)

LO 4.3-A
Compare and contrast backdrops and sprites
LO 4.3-B
Initiate a sprite's actions based on scenes
LO 4.3-C
Create an animated project with multiple scene changes and actions that are scene specific

Essential Knowledge

EK 4.3-A-1
Sprites are dynamic design elements that can have movement and react to other sprites or their environment
EK 4.3-A-2
Backdrops are static design elements that usually represent background environment or set the mood for a scene
EK 4.3-A-3
Sprites and backdrops can both respond to certain events
EK 4.3-A-4
Most sound blocks are available to both backdrops and sprites
EK 4.3-A-5
Both sprites and backdrops can have multiple costumes
EK 4.3-B-1
When backdrop switches to block is an event that can trigger backdrop/scene specific actions
EK 4.3-B-2
A mBlock project can contain multiple backdrops
EK 4.3-B-3
Backdrop switching is initiated by block instructions
EK 4.3-C-1
The switch backdrop to and wait and when backdrop switches to blacks can be paired to achieve certain sequencing effects for synchronization purposes
EK 4.3-C-2
All actions in a scene should be tested to ensure their expected behaviors before moving on to other scenes
EK 4.3-C-3
A common error in scene switching is that sprites' behaviors are not well coordinated or synchronized with the scene change

Scenes

LA 4.3-1

Overview

Following LA 4.1-1 and 4.2-1, students now have a foundation of knowledge to create multiple characters with their own behaviors carried out in a synchronized fashion either using waits or broadcasting. In this activity, students will learn about scenes/backdrops.

Scenes and backdrops are not only useful in storytelling, but also essential in game design. mBlock games often use backdrop techniques to simulate different environments, levels, and even motions. Encourage students to fully explore all blocks related to backdrops in mBlock and make use of them creatively.

Activity Description

1. Show sample projects to inspire students

2. Make sure students are signed into their mBlock accounts and start a new project

3. Give students time to develop a project that includes multiple scene changes using different backdrops, such as in a slideshow. Challenge students to explore and manipulate scripts in the scene to initiate backdrop changes

4. Allow students to share their projects with one another. Invite students to present their work to the class and demonstrate how they implemented switching backdrops

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. What does the backdrop have in common with sprites?
    b. How is the backdrop different from sprites?
    c. What other types of projects (beyond animations) use scene changes?

Resources

1. Sample scenes project - <Feishu Folder Location>

Notes

1. If students are having trouble figuring out how to switch backdrops, encourage them to tinker with blocks under the looks category, especially the switch backdrop to, switch backdrop to and wait, and next backdrop blocks

Mentor Reflection

1. Do projects successfully coordinate multiple scenes using changing backdrops?

Topic 4.4

Big Ideas

EVN
PRL
ANM
TND

Learning Objectives (Students will be able to…)

LO 4.4-A
Find solutions to a series of bugs related to events, parallelism, and custom blocks
LO 4.4-B
Describe what a parameter of a custom block is
LO 4.4-C
Capture typed input from users
LO 4.4-D
Incorporate broadcast and wait blocks when organizing instruction sequences

Essential Knowledge

EK 4.4-A-1
In mBlock, bugs related to parallelism and events are tedious to debug because it involves switching back and forth among a number of sprites. Working in pairs can greatly improve debugging efficiency
EK 4.4-A-2
Simulating the execution of a program is a great way to identify bugs given the program is reasonably simple and short
EK 4.4-B-1
Parameters allow information be passed into a custom block
EK 4.4-B-2
Parameters further increases the flexibility of a given custom block
EK 4.4-B-3
Parameters affect the output of a custom block but not the logic definition
EK 4.4-C-1
The ask and wait block will prompt the question on stage and wait for the user to input and confirm
EK 4.4-C-2
In some cases, keyboards' input settings might affect whether the information is properly captured in mBlock.
EK 4.4-C-3
The answer block stores the user input captured from the ask and wait block
EK 4.4-C-4
Even though mBlock allows text as input, it is a common bug when programmers use the answer block without considering if it's text or numbers
EK 4.4-D-1
broadcast and wait blocks are extremely useful in roll call scenarios or when numerous sprites must take action in order

Debug It

LA 4.4-1

Overview

Same as LA 3.4-1, this activity iterates on a different set of testing and bugging problems. The general purpose and strategies are unchanged.

Activity Description

1. Have the debug it handout available to guide students during the activity

2. Guide students to implement the given programs in mBlock and perform testing and debugging on their computers. Make sure students save each corrected program as a new project.

3. Ask students to reflect back on their testing and debugging experiences in their design journals or in a group discussion
    a. What was the problem/bug?
    b. How did you identify the problem/bug?
    c. Did others have alternative approaches to fixing the problem/bug?

4. Generate a list of debugging strategies by collecting students' problem finding and problem solving approaches

Resources

1. Debug it handout - <Feishu Folder Location>

Notes

1. Being able to read others' code is a valuable skill and is critical for being able to engage in the practices of reusing and remixing
2. This activity is a great opportunity for pair programming. Divide students into pairs to work on the debugging challenges
3. Students can explain their code revision by right-clicking on mBlock blocks to insert code comments

Mentor Reflection

1. Were students able to solve all five bugs? If not, how might you clarify the concepts expressed in the unsolved programs?
2. What different testing and debugging strategies did students employ?

Topic 4.5

Big Ideas

EVN
PRL
RNR

Learning Objectives (Students will be able to…)

LO 4.5-A
Describe what a handshake process is in collaboration
LO 4.5-B
Appreciate the value of reusing and remixing
LO 4.5-C
Create an open-ended collaborative storytelling mBlock project

Essential Knowledge

EK 4.5-A-1
During a collaboration, people are often in charge of different aspects of a project that must be eventually pieced together
EK 4.5-A-2
Some tasks in a collaboration are dependent on other tasks
EK 4.5-A-3
When a dependent relationship occurs in collaboration, a handshake process is essential
EK 4.5-A-3
A handshake process is largely about exchanging tips and notes about things that ought to be taken into consideration but might not be due to limited scopes of expertise and knowledge
EK 4.5-B-1
Reusing is about efficiency and not reinventing the wheels
EK 4.5-B-2
Remixing is about creativity and achieving surprising results
EK 4.5-B-3
When a project's reusability matches a remixer's creative idea, the best of both worlds become one
EK 4.5-C-1
A storyline is a powerful tool to hold many creative ideas together and help make sense of them
EK 4.5-C-2
An open-ended collaborative project does not have a clear goal or criteria
EK 4.5-C-3
Participants should respect each others' creative ideas in an open-ended collaborative environment
EK 4.5-C-4
Each member or group should be responsible for debugging the code they implemented

Our Creature

LA 4.5-1

Overview

This is a computer-free activity that utilizes drawing to help students develop a deeper understanding about the concept and process of reusing and remixing. Students will be each responsible for drawing parts of an unknown creature.

One key takeaway from the activity is to realize the surprising creative power of collaboration. Another one is to realize the importance of a good handover process. A handover process is crucial in collaboration because the person responsible for one task might not be familiar with other tasks. Therefore, during the process of task to task transition, high quality communication and information exchange should take place either explicitly or implicitly.

Materials

1. Blank drawing paper, folded into third
2. Things to sketch with (pencils, pens, markers, etc.)

Activity Description

1. In this activity, students will draw a "creature" in three parts

2. Give each student a tri-folded sheet of blank drawing paper and one minute to draw a "head" for their creature. Next, have them fold the paper over so that the head is hidden, with little prompts for where to continue the drawing. After the head is hidden, students will pass the creature to another student. Then, give students time to draw a "middle" for their creature, using the guides from the head, but without peeking. After the middles are hidden (and prompts drawn), pass the creatures. Finally, give students time to draw a "bottom" for their creature. When finished, unfold the papers to reveal the collaboratively constructed creatures

3. Post drawings on a wall or board and let students explore the outcome of their creative contributions

4. Facilitate a group discussion about co-authorship, collaboration, and reusing and remixing work
    a. What is your definition of remixing?
    b. Think about the creature you started. How did your ideas become extended or enhanced by others' contributions?
    c. Consider the creatures you extended, how did your contributions extend or enhance others' ideas?

Resources

N/A

Notes

1. This activity is a warm-up activity for LA 4.5-2
2. Have students sign their names at the bottom of each creature drawing they worked on to identify the contributing artists

Mentor Reflection

1. Can students explain remixing and collaboration and their benefits?

Pass It On

LA 4.5-2

Overview

By completing this activity, students will produce their third deliverable project. Students are expected to utilize everything they have learned up until this point, with an emphasis on the topics covered in unit 4.

This activity will be about collaboration and storytelling. Very much like LA 4.5-1, students will work in pairs to complete parts a storytelling project in mBlock and remix others' projects. Using the pass it on handout is recommended.

Activity Description

1. Divide the class into pairs. Introduce students to the concept of a pass-it-on-story, a mBlock project that is started by a pair of people, and then passed on to other pairs to extend and reimagine

2. Encourage students to start in whatever way they want - focusing on characters, scenes, plot, animations, sounds, or whatever element excites them. Give each pair time to work on their collaborative story before having them rotate to extend another story by remixing the project

3. After a number of rotations, allow students to revisit story projects with their contributions. Invite students to present the story projects with students gathered around to watch

4. Ask students to rethink the design process in their design journals or in a group discussion
    a. How did it feel to remix and build on others' work? How did it feel to be remixed?
    b. Where else in your life have you seen or experienced reusing and remixing?
    c. How was working with someone else different from your prior experiences of designing your mBlock projects?

Resources

1. Pass it on handout - <Feishu Folder Location>

Notes

1. As this is a relatively comprehensive deliverable project that might involve all prior knowledge, mentors should focus on picking up confusions among students
2. Encourage students to draw inspirations from the projects they completed in LA 4.1-1, 4.2-1, and 4.3-1
3. A screening event is recommended during project delivery to parents. Invite parents to the viewing and offer snack and drinks

Mentor Reflection

1. What parts of projects did students contribute to?
2. Do students seem comfortable with the concepts of events and parallelism and practices of reusing and remixing? If not, in what ways can these be further clarified?

Unit 5 Arcade Party

Overview

Personalization is an important guiding principle in the design of creative computing experience. Personalization means both connecting to personal interests and acknowledging that personal interests can vary considerably. There are many ways of knowing and doing - and exploring these multiple ways can help support interest, motivation, and persistence among young students. In this unit, students explore some of the advanced concepts and challenging problems associated with game design. An advanced concept or challenging problem can be made more accessible if rooted in activities that are personally meaningful. The final project of this unit is a complex game which involves all previous concepts covered in this course.

Concepts and Practices

The three relatively more advanced computational concepts that will be covered in this unit are conditional, operator, and data. Game design is a perfect theme for developing these concepts. As for computational practices, students should already be relatively familiar with most of them at this point. However, as projects grow larger in size and complexity, the role of testing and debugging also becomes larger.

Unit At a Glance

Topics	Keywords	Concepts	Practices
5.1	Game Design Starter Game	PRL, CON, OPT, DAT	RNR
5.2	Variable Score	CON, DAT	TND, RNR
5.3	Game Mechanisms Extension	CON, OPT, DAT	TND
5.4	Interactive Puzzles Debugging	CON, OPT, DAT	TND

Suggested Learning Activities

ID	Activity	Worksheet / Handout	Extra Materials
LA 5.1-1	Dream Games	N	Y
LA 5.1-2	Starter Games	Y	N
LA 5.2-1	Score	Y	N
LA 5.3-1	Extensions	Y	N
LA 5.4-1	Interactions	Y	N
LA 5.4-2	Debug It	Y	N

Topic 5.1

Big Ideas

PRL
CON
OPT
DAT
RNR

Learning Objectives (Students will be able to…)

LO 5.1-A
Become familiar with a rider range of blocks in mBlock
LO 5.1-B
Express their definitions of a game
LO 5.1-C
Describe common game mechanics in mBlock
LO 5.1-D
Create a self-directed game project that will be iterated on and extended throughout this unit

Essential Knowledge

EK 5.1-A-1
If-then and If-then-else blocks create splits in program execution. Different paths of execution is taken based on whether the conditions specified by these blocks are met or not
EK 5.1-A-2
Conditions in mBlock come in the shape of pointy rectangles
EK 5.1-A-3
Operator blocks are green in mBlock and they mostly perform arithmetic calculations and comparison
EK 5.1-A-3
Variables can be set to a certain number or increased by a certain number
EK 5.1-B-1
Some common elements in a game are objectives, rewards, challenges, rules, and i interactions
EK 5.1-C-1
When key pressed events combined with motion blocks controls sprites' movement
EK 5.1-C-2
Collisions could be captured by having an if-then block checking for a touching condition in a loop
EK 5.1-C-3
Parallelism is used pervasively in game design
EK 5.1-C-4
Pick random number block can be used to randomize actions and effects in games
EK 5.1-D-1
A starter project doesn't need to be feature complete or polished in any way
EK 5.1-D-2
Game design is about a lot of trial and error
EK 5.1-D-3
Design journals should be utilized heavily for a self-directed game project

Dream Games

LA 5.1-1

Overview

When talking about games, most young students now days think of computer games and mobile games. While in reality, the definition of a game is flexible. Some key components that are common in most games are goals, rules, challenges, and interactions. These components may or may not exist all at once in a game. And they might exist in various forms. Minecraft, for example, doesn't seem to set a clear goal for players. However, it provides a sandbox where players can act on their imaginations which are goals set by themselves.

What makes a good game is an entirely different discussion. Graphics, sounds, mechanics, storyline, and difficulty can all make a huge difference in how the final product is perceived. However, a key takeaway is that a good game doesn't always need to be complicated or designed by experienced professionals. Sometimes a good idea outshines everything else.

This activity will engage students in an open discussion about games, game designs, and beyond.

Materials

1. Paper and sticky notes
2. Things to sketch with (pencils, pens, markers, etc.)

Activity Description

1. Divide students into small groups

2. In their groups, ask students to generate a list of games that they enjoy playing. They can compose the list using their design journals. Give students a short time period to write down as many games as they can. Then have students narrow down their favorites ones

3. After some time, ask groups about their list of games
    a. What do the games have in common?
    b. What features of their design make them a game?

4. Facilitate a class discussion about what characteristics make up a game and generate a class list of common game mechanics. Next, ask students to imagine their dream game and write a list of design elements for that game

5. Invite students to share their dream game lists in their small groups to get feedback and suggestions

Resources

N/A

Notes

1. Invite students to refer back to this dream game list while programming games in other related activities in this unit

Mentor Reflection

1. Do the dream game lists include features of games?

Starter Games
LA 5.1-2

Overview

Even though students have now developed a lot of computational concepts and practices, computational game design in mBlock has its own unique challenges. This activity provides a number of starter game projects for students to choose from as a foundation to build on in future activities.

Students will receive handouts for Maze, Pong, and Flappy Pig. These handouts provide guidelines or code segments for various important game mechanics. Mentors should realize these are just three options aimed at students who lack inspiration or prior experience with game design. If some students have a clear idea and the motivation to implement their own starter game project, it is completely acceptable.

Designing games is a great way to learn about computational concepts like conditionals, operators, and data. And it is also incredibly related to practices like experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing. Mentors should focus on these aspects during class.

Activity Description

1. In the activity, students will create a starter game project that can be revisited and extended during LA 5.3-1, 5.4-1, and 5.5-1. Have the starter games handouts available to guide students

2. Choose one game project to facilitate as a class or let students choose which game they want to create. Give students time to start building their games or let them remix one of the starter projects

3. Encourage students to get feedback on their game-in-progress. Have half of the students stay in their seats with their projects open while the other half walks around exploring projects, asking questions, and giving feedback, then switch sides

4. Ask students to think back on their design process in their design journals or in a group discussion
    a. What was challenging about designing your game?
    b. What are you proud of?

Resources

1. Starter games handouts - < Feishu Folder Location>
2. Sample starter games projects - <Feishu Folder Location>

Notes

1. At the end of unit 5, during project delivery, an arcade event is recommended. Have students place their projects in presentation mode and invite parents and students to walk around and play others' games
2. The flappy pig game introduces cloning. Explain this idea in simple terms if needed as it will be covered more in unit 5

Mentor Reflection

1. Do games include conditionals, operators, and data?

Topic 5.2

Big Ideas

CON
DAT
TND
RNR

Learning Objectives (Students will be able to…)

LO 5.2-A
Describe what a variable is and why variables are useful
LO 5.2-B
Construct a scoring system by remixing a given program

Essential Knowledge

EK 5.2-A-1
Variables are created to store information in a program
EK 5.2-A-2
Variables names should be concise and meaningful, representative of the information stored
EK 5.2-A-3
Without variables it is nearly impossible for a mBlock program to memorize information
EK 5.2-A-4
A variable can be set to a specific number or increased by a specific number. They can only be integers in mBlock
EK 5.2-B-1
To implement a scoring system for a game, the condition for scoring must be clearly defined
EK 5.2-B-2
The scoring condition must be constructed and continuously checked throughout the play of the game
EK 5.2-B-3
At least one variable is needed for the program to memorize what the score is at any moment of time and increment it accordingly

Score

LA 5.2-1

Overview

Scoring system is a crucial part of game design. It is the most direct way of reflecting progress, reward, or objective. Its necessity to game design will drive students to pay close attention to concepts such as variables and data which are traditionally dry and daunting subject for students. Variables and data form the basis for many computational models and are used everywhere in programming. Therefore, mentors should aim to make sure all students have a solid understanding of these concepts.

In this activity, students will be introduced to a sample game project which is lacking a scoring mechanism. Students will first understand its current implementation and explore the concepts of variables and data to remix the project. Using the score handout is optional.

Activity Description

1. Explore the sample game project as a group and optionally have the score handout available to guide students

2. Make sure students are signed into their mBlock accounts and start a new project

3. Give students time to explore variables by remixing the sample project to add score to the game. Once student have finished remixing the sample project, give students time to incorporate score into their previously started game project in LA 5.2-1

4. Allow students to share their sample project remixes or game projects with added score. Invite students to present their projects to the group and demonstrate how they implemented score using variables

5. Ask students to think back on the design process in their design journals or in a group discussion
    a. How would you explain variables to someone else?
    b. What are variables good for?

Resources

1. Score handout - <Feishu Folder Location>
2. Sample game project - <Feishu Folder Location>

Notes

1. Variables are an important mathematical and computational concept. Students are taught about variables in their math classes, but many students have a difficult time learning them. Games are one way to make the usefulness of variables more concrete
2. Without variables, computer programs have no way to memorize anything. In other words, computers use variables to memorize information

Mentor Reflection

1. Can students explain what a variable is and what variables are good for?

Topic 5.3

Big Ideas

CON
OPT
DAT
TND

Learning Objectives (Students will be able to…)

LO 5.3-A
Describe the workings of multiple game mechanisms in mBlock
LO 5.3-B
Extend their game project with more common game mechanisms

Essential Knowledge

EK 5.3-A-1
Levels can be achieved by having a level variable increased only when certain conditions are met. It is very much like a scoring system with consequences
EK 5.3-A-2
Timer effects can be achieved by utilizing the reset timer and timer blocks
EK 5.3-A-3
A restart button could be implemented with a custom drawn button and when this sprite clicked block, in addition to some broadcasting to reset other sprites
EK 5.3-A-4
A restart button could be implemented with a custom drawn button and when this sprite clicked block, in addition to some broadcasting to reset other sprites
EK 5.3-A-5
A multiplayer game can be achieved by having two sets of independent controls that corresponds to different sprites and actions
EK 5.3-B-1
As an extension is added to an original project, testing and debugging is crucial to ensure no bugs are introduced that can cause catastrophe
EK 5.3-B-2
Inviting peers to review on each revision of a program can result in incredibly useful feedback

Extensions

LA 5.3-1

Overview

Aside from the scoring system introduced in LA 5.3-1, there are many other desirable game design elements that students might want to add to their own projects. In this activity, students will have the chance to explore a series of extensions including levels, timer, enemies, rewards, mouse, restart, menu, and multiplayer. Each of these extensions has its own sample project demonstrating the effects. Students are expected to analyze these sample projects to understand their inner workings and implement them in their own game projects.

As for computational concepts and practices, the running theme throughout unit 5 is conditionals, operators, and data. The different extensions introduced in this activity are just various ways of applying these concepts. Therefore, if students are struggling with adding extensions to their projects, it might be an indication that these concepts should be revisited.

Activity Description

1. Show sample projects and have the extensions handout available to guide students

2. Give students time to explore the code of the sample programs to investigate different ways games can be increased in difficulty or extended. Ask students to select one or more extensions to add to their previously started game projects. Give students time to experiment and incorporate the extension(s) into their games

3. Allow students to share their extended game projects with one another. Invite students to present their projects to the group and demonstrate how they implemented various extensions

4. Ask students to think back on the design process in their design journals or in a group discussion
    a. What are different ways of increasing difficulty in a game?
    b. Which extensions did you add to your game project?
    c. Describe your process for including the extension(s) in your game

Resources

1. Extensions handout - <Feishu Folder Location>
2. Sample extensions projects - <Feishu Folder Location>

Notes

1. To provide more scaffolding for students needing extra support, we suggest walking through one extension sample project as a class and helping students add the extension to their game projects

Mentor Reflection

1. Were students able to incorporate extensions into their original game projects?

Topic 5.4

Big Ideas

CON
OPT
DAT
TND

Learning Objectives (Students will be able to…)

LO 5.4-A
Incorporate interactive mechanisms into their game project
LO 5.4-B
Receive and give feedback on projects in their critique groups
LO 5.4-C
Perform testing and debugging for a complex project

Essential Knowledge

EK 5.4-A-1
Sound levels can be triggers for certain actions
EK 5.4-A-2
There are 3 main types of collision detection - color based, sprite based, and border based
EK 5.4-A-3
Sprites can appear to be following the mouse pointer with some simple looped blocks

Interactions

LA 5.4-1

Overview

This activity consists of 9 puzzles where each puzzle corresponds to an interactive effect that student can directly implement in their game projects or draw inspirations from. These interactive effects covers results of key presses, collisions, distance changes, mouse positioning, mouse clicks, and etc..

The puzzle form factor encourages students to compete, share, and collaborate. Mentors should focus on keeping a friendly and relatively organized environment for this activity. Mentors should also be aware of whether there are certain blocks or concepts students are still struggling with.

Activity Description

1. On their own or in small groups, challenge students to further explore mBlock by creating mBlock programs that solve each of the 9 interactions programming puzzles. These interactions puzzles explore sensing blocks, engaging some of the more advanced concepts in mBlock related to interactivity

2. Have the interactions handout available to guide students during the activity

3. Make sure all students are signed into their mBlock accounts and start a new project for each of the 9 puzzles

4. Each puzzle can have several possible solutions. Invite students or groups to share different solutions and strategies

5. Ask students to think back on the challenge in their design journals or in a group discussion
    a. Which puzzles did you work on?
    b. What was your strategy for solving these puzzles?
    c. Which puzzles helped you think about your game project?

Resources

1. Interactions handout - <Feishu Folder Location>

Notes

1. Choose particular challenges that highlight new blocks or concepts you would like students to have deeper understanding to go over together as a class

Mentor Reflection

1. Are the puzzles solved?
2. Did students explore other approaches for solving the puzzles?
3. Are there certain blocks or concepts students are still struggling with? How might you help?

Debug It

LA 5.4-2

Overview

Same as LA 4.4-1, this activity iterates on a different set of testing and bugging problems. The general purpose and strategies are unchanged.

Activity Description

1. Have the debug it handout available to guide students during the activity

2. Guide students to implement the given programs in mBlock and perform testing and debugging on their computers. Make sure students save each corrected program as a new project.

3. Ask students to reflect back on their testing and debugging experiences in their design journals or in a group discussion
   a. What was the problem/bug?
   b. How did you identify the problem/bug?
   c. Did others have alternative approaches to fixing the problem/bug?

4. Generate a list of debugging strategies by collecting students' problem finding and problem solving approaches

Resources

1. Debug it handout - <Feishu Folder Location>

Notes

1. This activity provides an opportunity to check in with students who might need some additional attention or support, particularly around the concepts of conditionals, operators, and data

Mentor Reflection

1. Were students able to solve all five bugs? If not, how might you clarify the concepts expressed in the unsolved programs?
2. What different testing and debugging strategies did students employ?

单元 1 预备，开始！

概述

这是一个为创意计算奠定基础的准备单元。本单元向学生介绍他们的编程环境，确保他们能够合理地想象mBlock给予自己的创造可能性。学生还必须创建自己的云管理帐户，以便能够轻松跟踪、共享、修改或评估他们未来的项目。设计日志和评论小组是帮助组织整个课程学习活动的基本工具，因此导师向学生提供关于设计日志和各自评论小组的目的、创建、使用和价值的明确指导非常重要。

概念与习惯

由于这是一个预备单元，所以课程相关的概念和习惯都不打算作为学习目标。然而，大多数习惯都是解决问题的模式，适用于生活的各个方面，而不是严格地与主题相关。因此，尽管有时没有明确说明，但大多数习惯都隐含地与所有课堂活动的设计理念相结合。例如，指导学生制作设计日志的活动鼓励学生采用设计思维，即计划、制作、分享和反思的过程。

单元一览

课题	关键词	概念	习惯
1.1	创意计算 mBlock 云账户 设计日志	N/A	N/A
1.2	自由探索 评论小组	N/A	N/A

推荐课堂活动

编号	活动	活动说明页	额外物料
LA 1.1-1	初始mBlock	N	N
LA 1.1-2	云账户	Y	N
LA 1.1-3	设计日志	N	Y
LA 1.2-1	勇敢探索者	Y	N
LA 1.2-2	评论小组	Y	N

课题 1.1

主要思想

创意计算
网络安全
设计思维

学习目标 (学生可以...)

LO 1.1-A
用他们自己的话描述什么是创意计算
LO 1.1-B
熟悉mBlock软件环境
LO 1.1-C
独立完成账户注册
LO 1.1-D
创建个性化的设计日志，记录他们的设计过程和思考

必要知识

EK 1.1-A-1
计算机的各种使用场景
EK 1.1-A-2
利用计算机进行创意的可能性
EK 1.1-A-3
合理的想象mBlock中存在的创意可能性
EK 1.1-B-1
认识mBlock界面的各个部分，包括但不限于舞台、模块、角色、设备、背景和程序区域
EK 1.1-B-2
在本地环境中命名、保存和检索mBlock项目
EK 1.1-B-3
命名、保存和检索云中的mBlock项目
EK 1.1-C-1
描述在线帐户的两个基本组件-用户名和密码
EK 1.1-C-2
注册用户名的常见选项，包括电子邮件、电话号码和自定义文字
EK 1.1-C-3
弱密码的识别及其潜在风险
EK 1.1-C-4
生成安全而难忘的密码
EK 1.1-C-5
在mBlock中登录和退出云帐户
EK 1.1-D-1
描述设计日志及其每个过程的范畴-规划、制作、共享和反省
EK 1.1-D-2
描述设计日志中应该包含的内容
EK 1.1-D-3
利用提供的材料努力个性化自己的设计日志

初识mBlock

LA 1.1-1

概述

简而言之，创意计算是一种高层次看待电子设备的角度，学生将这些设备视为释放创造力的工具，而不是更传统的方式，即学生从一开始就掌握一系列详细的技能。简言之，创意计算的主要思想之一是让年轻学生尽快开始使用计算机和编程，而不必太担心细节，也不必被太多枯燥乏味的内容所阻拦。

此活动旨在破冰，并对计算机、编程和数字领域的创新可能性进行非正式介绍。通过将计算机和编程视为创造性表达的工具，鼓励学生的兴趣和热情是很重要的。本次活动将首先对计算机的各种应用场景进行讨论，然后聚焦于富有创造性的场景。最后对mBlock进行全面概述。

活动描述

1. 询问学生他们使用计算机的经验
    a. 你通常如何与电脑互动?
    b. 你通常在电脑上做什么?
    c. 这些事情中有多少涉及到利用计算机实现自己的创造性?

2. 通过播放一些视频和示例项目，向学生介绍mBlock如何用于创意计算以及他们能够创建的项目范围。说明在接下来的几节课中，他们将使用mBlock创建自己的交互式创意计算项目

3. 你将创造什么？让学生想象他们想用mBlock创建什么类型的项目

辅助资源

1. mBlock概述视频 - <Feishu文件夹位置>
2. mBlock示例项目 - <Feishu文件夹位置>

提示

1. 不需要学生写下他们对讨论问题的答案，而是鼓励通过画出他们的答案

导师反省

1. 学生们是否能够在mBlock的范围内头脑风暴出各种各样的项目想法？如果不是，试着在样本项目中增加更多的多样性，让学生感受到可能性。此外，建立对mBlock的局限性的认知也很重要。例如，mBlock与照片编辑软件、音乐制作软件、电影制作软件或专业游戏引擎相比

云账户

LA 1.1-2

概述

mBlock云帐户允许学生在所有受支持的设备和平台上轻松保存、跟踪并与同学和导师分享他们的项目。此外，拥有学生项目的访问权对于内部研发会议和教学质量讨论非常有价值。因此，建立所有学生账户并对其进行系统管理对于本课程和其他各种内部程序至关重要。

在当今时代，从银行到个人健康，随着越来越多的应用程序为生活的方方面面提供服务和管理，创建和管理足够安全的在线账户几乎已成为一项技能。年轻学生应该具备如何创建账户、如何将账户个人化以及如何确保账户安全的基础知识。在本活动中，学生将收集帐户注册的基本信息，并练习生成安全密码。此外，学生们还将尝试一些破解密码的练习，以强化他们对帐户安全的认识。

活动描述

1. mBlock云帐户需要能够接收文本验证的电子邮件地址或移动电话号码。如果学生在课堂上无法提供个人电子邮件地址或电话，导师应提前准备足够数量的电子邮件地址，并妥善分发和管理
2. 帮助学生导航到mBlock中的帐户注册页面。确保该帐户已注册到mBlock China而不是mBlock global。鼓励学生练习登录和注销他们的帐户
3. 让学生创建模拟项目来练习保存和定位他们的项目
4. 为使全班同学更容易相互共享项目，请考虑创建用户名（电子邮件/电话号码）和密码的类列表。
5. 作为一个班级，为帐户共享、项目命名和其他规则制定管理指南，以更好地组织和保护所有学生作品

辅助资源

1. 密码管理表 - <Feishu文件夹位置>

提示

1. 为了在维护隐私的同时记住密码，让学生在单独密封的信封或文件夹中写下用户名和密码。导师个人负责组织和保护这些文件

导师反省

1. 学生是否能够成功创建mBlock云帐户并在不查看其密封密码的情况下登录和注销?

设计日志

LA 1.1-3

概述

尽管创意是如何产生的仍然是一个谜，但将创意付诸实践并最终完善它的过程必然是系统性的。创意不仅珍贵，而且是自发的、不完整的、支离破碎的。因此，记录所有这些想法，反思它们的紧张关系、潜力、实施难度，并勾勒出最终产品的轮廓，这是一个很好的做法。从计划到制作再到分享，设计周期值得记日记，而且往往比最终结果更有价值。

无论是实体还是电子，记笔记的一个常见问题是，它通常功能性高于情感价值。日记本和普通笔记本的区别在于，日记本通常具有更多的情感价值，这鼓励了进行添加、反思和保存的一个积极循环。因此，本活动旨在从一开始就在学生和他们的设计日志之间建立联系。学生们被期望使用提供的材料对他们的设计期刊进行深度个性化，并享受一些好的手工制作的乐趣。

额外物料

1. 各种各样的纸张、记号笔和工艺材料，常见于探索者课堂和手工制作中。

活动描述

1. 向学生介绍设计日志的想法，这是一个物理笔记本，他们可以在其中集思广益，分享个人想法，类似于个人日志或日记。说明在mBlock编程过程中，将提示学生更新其设计日志，但鼓励他们在设计项目过程中随时添加日志，以捕捉想法、灵感、笔记、草图、问题、挫折、胜利等。

2. 浏览样本设计期刊库，了解学生希望创建哪种类型的设计期刊。让学生有时间使用提供的材料开始并个性化他们的设计期刊

3. 通过回答与LA 1.1-1相关的问题，让学生创建他们的第一篇设计日志
    a. 你会如何向朋友描述mBlock?
    b. 为您感兴趣创建的三个不同mBlock项目编写或勾勒想法

4. 鼓励学生与邻居分享他们的设计日志和初步思考

辅助资源

1. 样本设计期刊 - <Feishu文件夹位置>

提示

1. 设计日志应在未来的学习活动中经常使用
2. 设计日志可用于私人和公共场合，包括一对一诊断和同行评审

导师反省

1. 关于学生的兴趣，反思回答告诉了你什么?

课题 1.2

主要思想

创造性计算

反馈

学习目标 (学生可以...)

LO 1.2-A
合理描述mBlock中的一些创造性可能性

LO 1.2-B
创建一个mBlock项目，演示一些基本效果

LO 1.2-C
描述批评小组的目的

LO 1.2-D
在他们的批评小组中就他们/他人的mBlock项目提供并接受反馈

必要知识

EK 1.2-A-1
mBlock提供产生声音效果的块

EK 1.2-A-2
mBlock提供产生运动效果的块

EK 1.2-A-3
mBlock提供了一个内置设计元素库（角色/背景）

EK 1.2-A-4
mBlock中的设计元素具有大小、位置和方向等属性

EK 1.2-B-1
mBlock中的绿色标志可以作为启动程序的触发器

EK 1.2-B-2
单击绿色标志时，块对应于舞台下的绿色标志

EK 1.2-B-3
可以将各种块捕捉在一起

EK 1.2-B-4
mBlock中的项目可以置于演示模式

EK 1.2-C-1
批评小组涉及获得项目消极、混乱和积极方面的反馈

EK 1.2-C-2
在评论小组中提供反馈是基于对他人项目的深入了解

EK 1.2-C-3
负面反馈与批评或人身攻击无关

EK 1.2-C-4
反馈应该是有价值和有价值的

EK 1.2-C-5
一个人可以从批评小组中的其他项目中学习

EK 1.2-D-1
理解批评小组收到的反馈

EK 1.2-D-2
了解如何根据批评小组收到的反馈采取行动

EK 1.2-D-3
利用批评小组讲义给予和接受反馈

勇敢探索者

LA 1.2-1

概述

年轻学生的天性是探索和玩弄他们从未遇到过的东西。通常情况下，这个过程看起来是无序的，没有明显的内在逻辑。然而，归根结底，做比不做更重要。如果学生一开始就受到限制，他们中的一些人可能会习惯于服从命令。在某些情况下，这肯定不是一种消极的行为。但由于这是一门入门课程，所以必须点燃好奇心的蜡烛，让所有学生都被他们的好奇心和冒险精神所驱使。

与大多数未来的学习活动一样，导师主要负责指导学生完成一系列的学习步骤，而不是在每个步骤中都有很强的影响力。导师应提供探索、克服挑战、反思错误和分享成功的结构。在这项活动中，学生们将首先在mBlock中玩耍，然后接受挑战，使其产生一定的效果，并最终进入一个分享环节，分享他们所发现的东西。

活动描述

1. 确保所有学生都已登录到他们的mBlock帐户

2. 给学生足够的时间以开放的方式探索mBlock接口。用“无畏”和“惊奇”等关键词提示学生。鼓励学生一起工作，相互寻求帮助，并分享他们的想法

3. 请志愿者与全班分享他们发现的一件事

4. 志愿者分享后，向学生们提出一些挑战
    a. 有人知道如何添加声音吗?
    b. 有人知道如何改变背景吗?
    c. 有人知道如何改变角色的大小吗?
    d. 有人知道如何开始一个新项目吗?

辅助资源

1. 勇敢的探索者讲义-<Feishu文件夹位置>

提示

1. 这项活动的一个主要目标是建立一种无畏、探索和同侪协作的文化。预计学生不会提前知道太多，教室环境变成了每个人都在一起学习的空间

导师反省

1. 学生们知道如何启动一个新项目吗?
2. 学生是否理解将块对齐的基本机制?

评论小组

LA 1.2-2

概述

尽管导师和学生之间的联系对于成功的学习环境至关重要，但同龄人之间的互动是独一无二的，不可替代的。学生们通常更愿意彼此分享、表达关切和提出问题。因此，为批评小组建立课堂文化并定期利用批评小组对导师来说是非常宝贵的。

在评论小组里，学生们会对他人的作品给予反馈。这个过程就是向别人学习，表扬别人的才华，接受别人的建议，勇敢地面对困惑。对于第一次评论小组会议，主题将是每个学生从LA 1.2-1中完成的最终结果。鼓励导师向所有学生提供评论小组说明页，作为讨论指南。

活动说明

1. 向学生介绍批评小组的想法，这是一个由设计师组成的小组，他们彼此分享想法和项目，以便获得进一步发展的反馈和建议

2. 向所有学生提供批评小组说明页

3. 把学生分成小组。在这些评论小组中，要求学生轮流分享他们的想法、草稿或原型，例如LA 1.2-1项目

4. 让学生通过让他们的批评小组成员对讲义上的红色、黄色和绿色类别做出回应来收集反馈。鼓励学生在设计日志中记录其他笔记、反馈和想法

辅助资源

1. 评论小组讲义 - <Feishu文件夹位置>

提示

1. 在设计迭代过程中，有一个专门的同事小组给你鼓励和反馈是很有价值的。在所有单元中，为学生提供继续与批评小组会面的机会
2. 一些学生可能会敏感地接受太少的正面评论，或者从负面的角度看待建议。导师应在小组讨论中了解学生的情绪反应，并明智地管理学生行为

导师反省

1. 是否所有学生都有机会分享他们的工作并获得反馈?
2. 在这个过程中，是否有学生过于害羞或孤僻?
3. 所有学生都理解他们收到的反馈吗?

单元 2 奇思妙想

概述

既然学生们已经进入了mBlock和创造性计算的世界，那么帮助他们入门的最佳方式是什么？答案必须在学生被告知在一个高度结构化的环境中该做什么和让学生在自己的指导下独自探索之间。至少在本单元中，学生们被期望采取更多的自我导向的开放式探索路线。学生将通过参与和反思各种有趣和激动人心的活动，掌握自己编程的最基本理念。本单元的最后一个项目是互动拼贴。

概念与习惯

本单元中的关键计算概念是序列识别和指定有序指令序列的概念。顺序是程序执行的基础。在许多方面，本课程中的大多数其他计算概念都是关于改变程序顺序的。序列也是一个强大的概念，因为学生现在可以通过将他们的想法翻译成计算机代码块来告诉计算机该做什么，从而感觉到自己的能力。在本单元中，增量迭代、测试和调试的实践也被轻描淡写地提到。年轻的学习者在编程过程中很容易沉迷于将块拖到一起，而无需在过程中进行测试。这是可以理解的，但随着程序越来越大，定位错误的难度也越来越大。因此，最好在开发过程中尽早捕获bug.

单元一览

课题	关键词	概念	习惯
2.1	分解 指令 精确性	SEQ	N/A
2.2	简易序列 限制范围	SQE, LOO	ICI
2.3	调试 分析	SEQ, LOO	TND
2.4	互动大集	SEQ, LOO	ICI, TND

推荐课堂活动

编号	活动	活动说明页	额外物料
LA 2.1-1	跳舞吧	N	N
LA 2.2-1	一步一步	Y	N
LA 2.2-2	10个模块	Y	N
LA 2.3-1	调试	Y	N
LA 2.4-1	关于我	Y	N

课题 2.1

主要思想

SEQ
分解
原子操作
传播模式

学习目标 (学生可以...)

LO 2.1-A
使用一系列指令表示复杂的活动
LO 2.1-B
欣赏精确指示的价值
LO 2.1-C
描述在沟通模式方面精确的缺点
LO 2.1-D
描述计算机和人类在沟通模式方面的差异

必要知识

EK 2.1-A-1
识别复杂活动中可想象的步骤
EK 2.1-A-2
应进一步分解影响复杂活动中单个步骤的因素
EK 2.1-A-3
以简洁高效的方式组织复杂活动中的步骤说明
EK 2.1-A-4
复杂活动的分解步骤应考虑前面步骤的影响
EK 2.1-B-1
精确意味着没有混乱
EK 2.1-B-2
精确的指示确保了确定性，并限制了可能结果的范围
EK 2.1-B-3
与模糊的指令相比，一系列精确的指令更有可能产生预期的结果
EK 2.1-C-1
在人类交流中很难构建精确的指令
EK 2.1-C-2
精度通常会增加表示活动的指令总数
EK 2.1-C-3
对于人类来说，精确的指令通常需要更多的时间来处理
EK 2.1-D-1
计算机在很大程度上依赖于精确的指令来产生期望的结果，这也需要精确的定义
EK 2.1-D-2
计算机所做的不多也不少
EK 2.1-D-3
人类在很大程度上依赖于假设、情感、表情、肢体语言和上下文来对指令进行推理

跳舞吧

LA 2.1-1

概述

计算机非常善于遵循命令，也许这太好了，因为它需要付出代价。非常善于遵循命令意味着指令是按照一定的顺序执行的，不做比指定的多或少的事情。这与人们每天从事的交流模式截然不同。首先，人与人之间的交流涉及一系列复杂的行为，如言语、肢体语言、面部表情等。。除此之外，当解释依赖于假设、上下文和情感时，含义和意图通常是隐含的。

在本次活动中，学生将体验到在人际沟通模式下执行精确行为的困难。通过本次体验，学生应了解序列和精度对数字设备的重要性。

活动说明

1. 让学生分成指导者/表演者两组。准备好展示舞蹈视频

2. 每组:
   a. 让表演者背对着显示器，指挥者（和其他学生）面对显示器
   b. 向指导者和其他学生播放视频，但不要向表演者播放
   c. 请指导者向他们的搭档描述（仅使用文字）如何执行视频中显示的舞蹈动作序列

3. 使用此活动开始讨论序列在指定一组指令时的重要性。你可以让学生在他们的设计日志中单独思考，或者通过邀请不同的学生分享他们的想法来促进小组讨论
   a. 当指导者有什么容易/困难?
   b. 成为表演者的容易/困难是什么?
   c. 看视频有什么容易/难的地方?
   d. 此活动与我们使用mBlock所做的工作有何关系?

辅助资源

1. 舞蹈视频 - <Feishu文件夹位置>

提示

1. 这是本指南中与计算机无关的几个活动之一。从计算机后退可以支持对计算概念、实践和观点的新观点和新理解。
2. 活动结束后，让学生反思并写下一步一步的舞蹈指导。在编程中，这有时被称为“伪代码”

导师反省

1. 学生在指定说明时，能否解释顺序的重要方面?
2. 学生是否意识到在给出指示时具体而准确的价值?

课题 2.2

主要细想

SEQ
LOO
ICI

学习目标 (学生可以...)

LO 2.2-A
以科学的方式发展对未知区块的理解
LO 2.2-B
描述多个块的自然执行顺序
LO 2.2-C
描述计数重复块的工作原理
LO 2.2-D
使用某些顺序逻辑创建mBlock项目
LO 2.2-E
欣赏增量迭代的价值

必要知识

EK 2.2-A-1
科学方法包括形成一个假设，检验该假设，并根据观察到的结果进行迭代
EK 2.2-A-2
块的书面描述提供了有价值的信息
EK 2.2-A-3
关于未知区块影响的假设应基于其书面描述形成
EK 2.2-A-4
关于未知块效应的假设应根据结果进行检验和调整
EK 2.2-B-1
在mBlock中对齐的块从上到下执行
EK 2.2-C-1
计数重复块按其自然顺序执行所有夹在中间的模块，执行指定次数
EK 2.2-C-2
计数重复块可以在其之前或之后捕捉其他块
EK 2.2-C-3
计数重复块重复的次数是可指定的整数
EK 2.2-D-1
在构建创造性项目时，故事情节非常有用
EK 2.2-D-2
项目计划或故事情节应分解为可转换为可用块的说明
EK 2.2-D-3
块应按反映所需步骤的顺序捕捉在一起
EK 2.2-E-1
随着项目复杂性的增加，所需的块数也会增加
EK 2.2-E-2
随着项目中块数的增加，出错的概率也会增加
EK 2.2-E-3
在大量块中定位错误比在少量块中定位错误更困难
EK 2.2-E-4
增量迭代通过确保过程中的每一步都按预期进行，从而减少了错误的范围

一步一步

LA 2.2-1

概述

根据LA 2.1-1，本活动将要求学生通过创建舞蹈精灵动画，将他们对mBlock中序列和指令的经验和理解应用于mBlock。鼓励导师隐式或显式地采用设计周期的实践。换句话说，学生们应该首先对他们想要他们的精灵跳什么舞产生一个大致的想法。然后将想法分解成一系列合理详细且可操作的说明。在实现过程中，导师应该鼓励学生通过实验和调试来练习增量迭代。

一旦学生有了一个合理完成的项目，他们就需要在项目的基础上增加额外的模块。建议使用分步讲义。

活动描述

1. 确保学生已登录他们的mBlock帐户并开始一个新项目。在活动期间提供分步讲义以指导学生

2. 让学生按照讲义制作动画节目。鼓励学生添加其他块，并尝试运动、角色、外观、服装、声音或背景，使项目更加完美

3. 让学生彼此分享他们的第一个mBlock创作。

4. 要求学生回顾设计过程，并在他们的设计日志或小组讨论中进行反思
    a. 这项活动有什么令人惊讶的地方？
    b. 在活动中一步一步地被引导，感觉如何？
    c. 什么时候你觉得最有创造力？

辅助资源

1. 分步讲义 - <Feishu文件夹位置>

提示

1. 计划阶段的舞蹈指令与实施阶段的可用指令/块之间可能存在一些断连。这是学生中普遍存在的困惑，导师应该意识到这一点

导师反省

1. 学生在登录和启动新mBlock项目方面是否熟练？
2. 学生们能创造一个跳舞的角色吗？
3. 学生是否能够保存和共享项目？

10个模块

LA 2.2-2

概述

对于年轻学生来说，避免被太多的新信息淹没是很重要的。特别是在编程中，当有无数不同的设备、块和用例时，限制学习活动的范围有时会出人意料地有效。有限的范围会促使学生对他们正在使用的内容有更深的理解。此外，当可用资源受到限制时，创造力确实占据了中心地位。

在这项活动中，学生们将认识到仅使用10个区块创建项目的优点和缺点。鼓励学生至少使用10个模块一次，并从他们的批评小组获得反馈。建议使用10块讲义。

活动描述

1. 确保学生已登录他们的mBlock帐户并开始一个新项目。在活动期间提供10个街区的讲义以指导学生
2. 给学生时间创建一个只有10 MB块的项目：转到、滑动、说、显示、隐藏、设置大小、播放声音直到完成，当角色点击时，等待并重复。提醒学生在他们的项目中至少使用每个区块一次，并鼓励他们用不同的角色、服装或背景做更多的事情
3. 邀请学生在他们的评论小组中分享他们的项目
4. 让学生在设计日志或小组讨论中回顾设计过程
    a. 仅仅使用10个模块有什么困难?
    b. 只使用10个模块有什么容易的?
    c. 它是如何让你改变想法的?

辅助资源

5. 10个活动说明页-<Feishu文件夹位置>
6. 评论小组说明页-<Feishu文件夹位置>

提示

1. 令人惊讶的是，一个人只用10个模块就能完成多少任务。借此机会鼓励不同的想法，并通过邀请学生在全班面前展示他们的项目来庆祝创造力

导师反省

1. 项目是否包括所有10个模块?
2. 不同的学生对约束创造的想法有何反应？这会告诉你这个学生是如何学习的?

课题 2.3

主要思想

SEQ
LOO
TND

学习目标 (学生可以...)

LO 2.3-A
描述什么是编程错误
LO 2.3-B
实现对bug的修复
LO 2.3-C
认识到同一个问题可能有许多编程解决方案
LO 2.3-D
描述一些调试策略

必要知识

EK 2.3-A-1
bug是编程界用来描述程序错误的术语
EK 2.3-A-2
程序中可能存在许多类型的bug，例如语法和逻辑
EK 2.3-A-3
调试是识别、定位和修复bug的过程
EK 2.3-B-1
必须通过测试和推理来识别和定位缺陷
EK 2.3-B-2
错误的解决方案也必须经过测试，以避免自身出现错误
EK 2.3-B-3
错误解决方案的成功与否取决于所需输出的规格
EK 2.3-C-1
计算概念被不同的人使用和实现
EK 2.3-C-2
就像相同的信息可以用自然语言以不同的方式进行交流一样，程序也可以通过不同的实现实现相同的输出
EK 2.3-C-3
并非所有编程解决方案都是平等的。它们在效率和合理性上有所不同
EK 2.3-D-1
调试中的跟踪在很大程度上是关于生成程序中每时每刻发生的事情的逻辑描述序列
EK 2.3-D-2
登录调试是指在程序执行期间显式显示某些信息，这些信息在其他情况下是隐式的，以便进行程序分析
EK 2.3-D-3
调试中的问题简化是关于分段分析编程指令，并为每个分段生成高级描述。这对于捕捉逻辑错误特别有用

调试

LA 2.3-1

概述

测试和调试是在编程中开发的关键能力。测试和调试要求您首先了解给定的程序，了解其预期输出的原因，将预期输出与实际输出进行比较，确定问题，并提出有效的解决方案。如您所见，调试练习涉及或测试许多有用的编程透视图，这就是为什么将来会有许多与测试和调试相关的学习活动。

学生将获得一系列完成的程序或代码段来“修复”。根据特定的输出标准判断“修复”是否成功。随着学生对测试和调试实践的改进，他们通常会更流利地理解他人的程序，修改或扩展他人的作品。

活动描述

1. 在活动期间提供调试it讲义以指导学生

2. 指导学生在mBlock中实现给定的程序，并在其计算机上执行测试和调试。确保学生将每个更正的程序保存为新项目。

3. 要求学生在设计日志或小组讨论中回顾他们的测试和调试经验
   a. 问题/错误是什么?
   b. 你是如何识别问题/bug的?
   c. 其他人是否有其他解决问题/bug的方法?

4. 收集学生发现问题和解决问题的方法，生成调试策略列表

辅助资源

1. 调试活动说明页 - <Feishu文件夹位置>

提示

1. 此活动也可以分组进行。如果导师决定让学生分组参加此活动，请注意被动的学生，他们的贡献没有预期的多。还应考虑配对策略
2. 测试和调试可能是程序员最常见的活动。事情很少按计划进行，因此开发一套测试和调试策略对任何计算创建者都是有益的
3. 测试和调试中涉及的一些常见策略是信息记录、跟踪和问题简化

导师反省

1. 学生们能解决所有五个错误吗？如果没有，你如何解释未解决错误中表达的概念?
2. 学生们采用了哪些不同的测试和调试策略?

课题 2.4

主要思想

SEQ
LOO
ICI
TND

学习目标 (学生可以...)

LO 2.4-A
熟悉更广泛的mBlock块
LO 2.4-B
创建具有可靠逻辑上下文的交互式mBlock项目
LO 2.4-C
通过测试和调试完善他们的项目

必要知识

EK 2.4-A-1
按键时，可通过键盘触发块
EK 2.4-A-2
单击此角色时，可通过鼠标单击触发块
EK 2.4-A-3
“外观”类别中的块可以更改角色的属性，并让角色在旁边的聊天泡泡中说话
EK 2.4-A-4
等待块可以暂停程序的执行
EK 2.4-A-5
永久块的作用就像计数重复块，重复数设置为无穷大
EK 2.4-B-1
交互式项目是一个用户可以通过其他输入（如按键或鼠标单击）触发某些效果的项目
EK 2.4-B-2
一个具有坚实逻辑背景的创造性计算项目意味着一个清晰的主题，通过合理和相关的设计元素和效果进行扩展
EK 2.4-C-1
一个完成的项目应该经过彻底的测试
EK 2.4-C-2
随着项目复杂性的增加，调试过程可能会产生连锁反应
EK 2.4-C-3
当某个特定bug的解决方案成功，但在程序的其他部分出现错误时，就会产生连锁反应
EK 2.4-C-4
一个项目在经过多次迭代的彻底测试和调试之后被认为是完美的，没有发现更多的错误

关于我
LA 2.4-1

概述

本活动旨在为学生提供创建第一个可交付项目的机会。作为一个可交付的项目，它应该建立在一个相对逻辑的环境中，并用一个相对完整的功能集来实现。作为这类项目的第一个，在密切相关的环境中工作可能有助于学生克服对失败的恐惧。换句话说，“关于我”的上下文不会引起太多的创造性压力，因此学生可以专注于项目的实际迭代。

根据学生的进度，这个项目的一个好的起点可能是一个交互式拼贴，它只是一些可点击的角色的集合，代表学生生活中的事物。导师应提前制定计划，让学生准备好向家长展示他们的项目。

活动描述

1. 向学生介绍交互式拼贴的概念，这是一个mBlock项目，通过可点击的角色展现自己的各个方面。或者，向学生展示一些示例项目以激发他们的创造力

2. 确保学生已登录他们的mBlock帐户并开始一个新项目。提供关于活动说明页以提供指导。

3. 给学生时间创建一个关于我的交互式拼贴mBlock项目，鼓励他们通过实验和迭代来构建自己的程序

4. 允许学生与他人分享他们正在进行的工作

5. 让学生在设计日志或小组讨论中回顾设计过程
    a. 你最自豪的是什么？为什么？
    b.你被什么卡住了？你是怎么脱钩的？
    c. 下一步你想做什么？
    d. 你从别人的关于我的项目中发现了什么？

辅助资源

关于我活动说明页 - <Feishu文件夹位置>

提示

样本项目可以同时激发和恐吓，打开创意空间并约束它。鼓励广泛的创作；多样性是伟大的
学生可以通过使用摄像头或网络摄像头将图像带入项目，进一步个性化项目
别忘了真正利用学生的设计期刊。不要总是直接回答学生的问题，而是让他们在设计日志中记录问题，并向同龄人寻求帮助

导师反省

项目是否创造性地使用精灵、服装、外观、背景或声音？
项目是互动的吗？用户能否与项目中的各种元素进行交互？

单元 3 带来生命

概述

从第二单元开始，学生们已经体验到，仅仅用几个角色和少量创造性地排序的编程块，他们就能做多少事情。然而，为了给一个项目带来更多的活力，它的角色必须首先活跃起来。动画是一个美丽的想法，因为令许多人惊讶的是，它的原理非常简单。快速浏览一系列静态图片，这是计算机的完美工作描述。这就是为什么学生可以很容易地在mBlock中实现动画效果的部分原因。本单元将重点帮助学生完成交互式和动画的最终项目。

概念与习惯

如第2单元所述，许多计算概念与序列密切相关。例如，循环就是改变序列进入一个循环。另一方面，并行性是指在同一时间执行多个序列，或者产生一种错觉。事件通常用于定义程序序列的启动条件。并行性和事件是开发动画和交互式项目的有用概念。增量迭代、测试和调试仍然是学生在本单元中应该关注的实践。

单元一览

课题	关键词	概念	习惯
3.1	表演 两件事一起 声音	PRL, EVN	ICI
3.2	画图 视觉效果	SEQ	ICI, TND
3.3	翻页书 造型	SEQ, LOO, PRL	ICI
3.4	测试	SEQ, LOO	TND
3.5	动画 互动	SEQ, LOO, EVN, PRL	ICI, TND

推荐课堂活动

编号	活动	活动说明页	额外物料
LA 3.1-1	演出来	N	N
LA 3.1-2	组个乐队	Y	N
LA 3.2-1	色彩与形状	N	N
LA 3.3-1	活过来了	N	N
LA 3.4-1	测试	Y	N
LA 3.5-1	音乐MV	Y	N

课题 3.1

主要思想

PRL
EVN
ICI

学习目标 (学生可以…)

LO 3.1-A
描述mBlock中的事件及其工作方式
LO 3.1-B
描述什么是并行性，以及它如何在mBlock中工作
LO 3.1-C
描述编程中“重置”的概念
LO 3.1-D
创建涉及复杂排序的交互式mBlock项目

必要知识

EK 3.1-A-1
事件是一个动作或一组定义明确的条件的发生
EK 3.1-A-2
事件通常用作程序执行的触发器
EK 3.1-A-3
TmBlock中的典型事件包括单击绿色标志时、按键时、此精灵单击时以及我收到消息时
EK 3.1-B-1
mBlock中的单个精灵可以具有相同或不同事件的多个事件触发器
EK 3.1-B-2
只要事件发生或满足定义的条件，事件块下的编程指令将按其自然顺序执行
EK 3.1-B-3
并行性是指同时执行多个指令集或同时执行多个操作的效果
EK 3.1-B-4
并行性可以通过在一个sprite中具有同一事件的多个事件块来实现
EK 3.1-B-5
并行性也可以通过仔细排序独立的编程块来实现
EK 3.1-C-1
包括角色在内的各种设计元素的状态通常在程序执行过程中改变。这些状态可能包括位置、方向、大小等
EK 3.1-C-2
程序执行期间设计元素的更改状态不会在执行结束时自动撤消
EK 3.1-C-3
重置是通常在程序开始时将设计元素恢复到其原始状态的概念
EK 3.1-D-1
等待块可以暂停一组指令的执行。与并行性相结合，项目中的多个设计元素看起来可以协调
EK 3.1-D-2
等待块也在循环内生效
EK 3.1-D-3
复杂排序涉及对程序顺序流进行详细分析的增量迭代，这可能涉及不同设计元素之间指令的并行性和协调

演出来

LA 3.1-1

概述
与LA 2.1-1类似，本活动要求学生执行某些说明。然而，学生们将用mBlock来表演实际的脚本，而不是口头指令。导师和/或学生将编写简短的节目供演员阅读和表演。此活动的关键概念是并行性和事件。剧本作者的目标应该是让演员同时独立和反应地执行两个动作。

并行性是指同时发生多个动作。关于并行性的关键理解是，操作集通常必须是无关的或独立的。例如，同时向左和向右行走显然是不现实的。但同时走路和说话是完全自然的。

另一方面，事件是决定某些指令时间的触发器，有助于程序的逻辑排序。

此类活动的成功在很大程度上取决于导师的详细准备和计划。

活动说明

1. 准备好显示器，以显示将执行哪些块和脚本

2. 请几对志愿者

3. 提示两名志愿者执行一系列指令
   a. 让一个人做一件事
   b. 让那个人“重置”
   c. 让一个人同时做两件事
   d. 让第二个人独立完成任务
   e. 让第二个人执行相关任务（响应第一个人）

4. 作为一个小组反思经验，讨论事件和并行性的概念
   a. 触发行动有什么不同?
   b. mBlock中的事件机制是什么?
   c. 同时发生的事情有哪些不同的方式?
   d. 在mBlock中实现并行性的机制是什么?

辅助资源

1. 示例脚本-<Feishu文件夹位置>

提示
1. 该活动涉及“重置”的概念，这是初学者经常忽略的。如果他们想在特定的位置、特定的外观等开始工作，学生需要理解他们完全负责编程这些设置步骤
2. 此活动对于演示广播和接收块对时非常有用

导师反省

1. 学生能否解释什么是事件和并行性以及它们如何在mBlock中工作?

组个乐队

LA 3.1-2

概述

创造性计算不考虑音乐是不对的。在这一点上，学生们应该已经习惯于在mBlock中进行探索，并以某种逻辑顺序将块拼接在一起。由于并行性和事件的概念刚刚在LA 3.1-1中介绍，这是一个让学生应用和完善其理解的绝佳机会。

在这项活动中，学生们将创建一个升级的乐队互动拼贴。学生们被要求创作各种互动乐器，可能与舞台上的音乐家一起创作。每种乐器都应该能够独立演奏，也可以作为一个整体进行编排。

这个活动是非常可扩展的。导师甚至可以让全班同学一起工作，组建班级乐队。建议使用内置a波段说明页。

活动说明

1. 确保学生已登录到他们的mBlock帐户，并提供build-a-band说明页以指导学生
2. 让学生有时间通过将角色与声音配对来创建交互式乐器。鼓励他们通过探索“声音”类别中的其他块或使用“声音”选项卡中的编辑工具，尝试用mBlock表达声音的不同方式
3. 允许学生相互展示他们的乐队，或让学生四处走动与其他乐队互动。建议学生在画廊散步，将他们的项目置于演示模式，然后邀请他们四处走动，探索彼此的项目
4. 让学生在设计日志或小组讨论中回顾设计过程
    a.你先做什么？
    b.接下来你做了什么？
    c.你昨天做了什么？

辅助资源

1. 构建a波段说明页-<Feishu文件夹位置>

提示

1. 作为一个整体分享，让学生一起表演他们的mBlock乐器，组成班级乐队
2. 鼓励使用事件和并行

导师反省

1. 项目是否创造性地利用声音？
2. 项目中的精灵是交互式的吗？
3. 活动是否被利用？
4. 是否演示了并行性？

课题 3.2

主要思想

SEQ
ICI
TND

学习目标 (学生可以...)

LO 3.2-A
熟悉更广泛的mBlock块
LO 3.2-B
将自定义设计元素导入并合并到mBlock项目中
LO 3.2-C
mBlock中绘画编辑器的创造性使用
LO 3.2-D
使用自定义设计元素和各种视觉效果创建mBlock项目

必要知识

EK 3.2-A-1
“按块更改颜色”效果会更改角色的色调
EK 3.2-A-2
通过块改变鱼眼效果，给人一种通过广角镜头看到角色的印象
EK 3.2-A-3
块的“更改旋转”效果会围绕精灵的中心点扭曲角色，从而扭曲角色
EK 3.2-A-4
“按块更改重影效果”修改精灵的透明度
EK 3.2-A-5
“按块更改马赛克效果”（change mosaic effect by block）显示精灵的多个较小图像，因此创建了“马赛克”效果
EK 3.2-B-1
从internet获取设计元素涉及到搜索、保存和格式考虑
EK 3.2-B-2
上传选项在mBlock中的位置
EK 3.2-B-3
Mac和Windows等常见操作系统的磁盘和文件系统结构
EK 3.2-C-1
在绘制编辑器中选择不同的绘制工具时，在画布上单击和拖动的效果也不同
EK 3.2-C-2
空白填充可以创造性地用于在形状上产生剪切效果
EK 3.2-C-3
可以使用“重塑”工具将圆形和矩形等规则形状修改为不规则形状
EK 3.2-C-4
使用选择工具选择形状时，可以修改形状的方向和大小
EK 3.2-D-1
自定义设计元素对编程块的反应与内置角色一样
EK 3.2-D-2
组合循环和外观效果块通常会产生令人惊讶的结果
EK 3.2-D-3
增量测试每个设计元素，确保它们的行为符合预期，这将减少项目完成后调试的难度

色彩与形状

LA 3.2-1

概述

LA 3.1-2主要涵盖mBlock中与声音相关的块和概念。现在轮到油漆编辑了。虽然mBlock的内置sprite库非常全面，学生可以随时利用互联网进行设计元素，但没有什么可以取代定制绘制的sprite的价值。

mBlock中的绘制编辑器相对原始。但通过简单的造型工具和着色技术，仍然可以实现令人惊叹的设计。在这样的活动中，学生往往比导师产生更好的结果。

学生们需要在paint editor中使用所有工具进行实验，并结合不同的形状、填充和轮廓以获得令人惊讶的结果。然后，学生可以使用“外观”类别中的块为其设计制作动画。

活动说明

1. 展示激发学生灵感的示例项目

2. 确保学生已登录他们的mBlock帐户并开始一个新项目

3. 给学生时间创建一个包含各种颜色形状的项目。邀请学生试验外观块和绘画编辑器，探索他们的艺术能力

4. 鼓励学生与他人分享他们的创造性工作。建议进行画廊步行（LA 1.1-3）

5. 让学生在设计日志或小组讨论中回顾设计过程
    a.你是如何将不同的形状和颜色融入到你的项目中的？这个想法是从哪里来的？
    b.这项活动有什么挑战性？
    c.这项活动有什么令人惊讶的地方？

辅助资源

1. 示例颜色和形状项目-<Feishu文件夹位置>

提示

1. mBlock同时支持位图和矢量图形。帮助学生导航到绘制编辑器中的矢量模式或位图模式按钮，以设计和操作不同类型的图像和文本。导师应该避免涉及太多的技术细节

导师反省

1. 项目是否包括各种形状和颜色？
2. 学生是否在外观类别中使用效果块？

课题 3.3

主要思想

SEQ
LOO
PAR
ICI

学习目标 (学生可以...)

LO 3.3-A
描述动画的原理
LO 3.3-B
解释角色和造型的区别
LO 3.3-C
使用自定义设计元素创建动画mBlock项目

必要知识

EK 3.3-A-1
动画书是一堆纸上的一系列图片，装订在一起。当你用拇指快速翻转纸张时，看起来图像在移动
EK 3.3-A-2
就像动画书一样，动画效果是通过快速连续的顺序图像来实现的，这些图像之间的差异最小。
EK 3.3-A-3
动画的平滑度主要取决于每个图像之间的差异程度以及播放速度
EK 3.3-B-1
mBlock中的一个角色可以有多种造型
EK 3.3-B-2
一个角色的造型可以被认为是角色的许多姿势之一
EK 3.3-B-3
一个角色的造型也可以被认为是一个演员在电影中可能穿的许多服装之一
EK 3.3-C-1
为动画效果设计造型时，应仔细考虑每个造型在绘制编辑器中的位置
EK 3.3-C-2
动画中的起始服装和结束服装有助于确定服装之间的位置
EK 3.3-C-3
mBlock不会自动循环穿过精灵的造型
EK 3.3-C-4
组合循环和下一个服装块是实现动画效果的常用方法
EK 3.3-C-5
通过在环路中加入等待块，可以降低环路频率
EK 3.3-C-6
服装设计应逐步进行测试，以确保每件服装都能产生预期的效果
EK 3.3-C-7
负责循环各种服装的代码段可以通过并行性从其他动作中分离出来

活过来了

LA 3.3-1

概述

如果设计一个角色是令人兴奋的，那么让一个角色活着将是令人兴奋的。对许多年轻学生来说，动画背后的魔力可能仍然是个谜。创造同一角色的一系列不同姿势/服装，并在其中快速循环，可以产生动画效果，这一概念简单、美丽、深刻。

动画有助于创造更真实、更吸引人、更有趣的互动体验。因此，了解如何高效地为角色创建一套服装，以实现所需的动画效果在创造性计算中是很有价值的。本活动将要求学生选择或创建一个精灵，然后让他们想象一个动画效果，通过在他们的角色中添加一些服装来实现。

活动说明

1. 介绍动画的概念，即循环播放一系列递增的不同图片，如动画书或动画电影
2. 展示激发学生灵感的示例项目
3. 确保学生已登录他们的mBlock帐户并开始一个新项目
4. 通过改变服装或背景来创造动画，鼓励学生探索循环
5. 让学生在设计日志或小组讨论中回顾设计过程
    a. 角色和造型有什么不同?
    b.什么是动画?
    c.列出你在现实生活中体验循环的三种方式

辅助资源

1. “活过来了”活动示例项目 - <Feishu文件夹位置>

提示

1. 角色和造型之间的差异往往是造成混淆的原因。演员们穿着多种服装或摆出不同姿势的比喻有助于澄清这种差异
2. 学生可以通过使用相机或网络摄像头拍摄自己的照片来制作自己的图像动画

导师反省

1. 学生们能区分角色和造型吗?
2. 一些学生对开发动画项目特别感兴趣，他们更喜欢花时间绘制和设计角色、造型或背景。您如何让学生参与项目的美学和技术方面?

课题 3.4

主要思想

SQE
LOO
TND

学习目标 (学生可以...)

LO 3.4-A
描述定时块和不定时块之间的区别
LO 3.4-B
描述重复块的工作原理
LO 3.4-C
在调试中使用问题简化

必要知识

EK 3.4-A-1
mBlock中的某些块有两个版本。一个没有规定的时间，另一个有一定的时间限制
EK 3.4-A-2
未计时的块不会停止程序的执行。但是，除非明确指定，否则未计时块的效果不会结束
EK 3.4-A-3
定时块将停止程序的执行并执行其效果，直到达到指定的时间。一旦时间限制达到，时间块的效果将结束
EK 3.4-B-1
重复直到块类似于永久循环，但具有退出条件。换句话说，重复直到块重复夹在中间的模块直到满足特定条件
EK 3.4-B-2
重复此操作，直到块也可以在其之前或之后捕捉其他块
EK 3.4-B-3
重复此操作，直到始终不满足exist条件时，块的行为与永久循环相同
EK 3.4-C-1
当一个程序很长时，执行调试的一个好方法是首先分段测试该程序。
EK 3.4-C-2
正面是程序各部分的行为，并为每个部分的行为生成清晰的描述
EK 3.4-C-3
根据对程序每个部分的描述，观察任何逻辑错误
EK 3.4-C-4
如果在一个冗长的程序中不存在逻辑错误，那么放大代码中对错误行为具有匹配描述的潜在部分

调试

LA 3.4-1

概述

与LA 2.3-1相同，此活动会迭代一组不同的测试和调试问题。一般目的和战略不变。

活动说明

1. 在活动期间提供调试活动说明页以指导学生

2. 指导学生在mBlock中实现给定的程序，并在其计算机上执行测试和调试。确保学生将每个更正的程序保存为新项目。

3. 要求学生在设计日志或小组讨论中回顾他们的测试和调试经验
    a.问题/错误是什么?
    b.您是如何识别问题/bug的?
    c.其他人是否有其他解决问题/bug的方法?

4. 收集学生发现问题和解决问题的方法，生成调试策略列表

辅助资源

1. 调试活动说明页-<Feishu文件夹位置>

提示

1. 此活动也可以分组进行。如果导师决定让学生分组参加此活动，请注意被动的学生，他们的贡献没有预期的多。还应考虑配对策略
2. 测试和调试可能是程序员最常见的活动。事情很少按计划进行，因此开发一套测试和调试策略对任何计算创建者都是有益的
3. 测试和调试中涉及的一些常见策略是信息记录、回溯、问题简化和模拟
4. 通过让学生以类似于LA 3.1-1的方式表演有问题的节目，在整个小组中促进这项活动

导师反省

1. 学生们能解决所有五个错误吗?如果没有，您如何澄清未解决方案中表达的概念?
2. 学生们采用了哪些不同的测试和调试策略?

课题 3.5

主要思想

Copyright
SEQ
LOO
EVN
PAR
ICI
TND

学习目标 (学生可以...)

LO 3.5-A
解释版权的目的
LO 3.5-B
在mBlock中录制歌曲或音效，并将其合并到项目中
LO 3.5-C
创建一个动画交互式mBlock项目，利用第3单元中涵盖的所有概念
LO 3.5-D
根据同行的反馈迭代他们的项目

必要知识

EK 3.5-A-1
创造力是有价值的，应该受到重视
EK 3.5-A-2
创造和混合他人的作品与窃取他人的作品是不同的
EK 3.5-A-3
使用他人的创意而不给予赞扬或对其进行重大修改通常是对其版权的侵犯
EK 3.5-B-1
“声音”选项卡下“录制”选项的位置
EK 3.5-B-2
mBlock提供基本的声音编辑工具，如修剪、改变节奏、改变音量和声音效果
EK 3.5-B-3
录制的音效与内置音效的表现完全相同
EK 3.5-C-1
负责产生独立结果的代码段通常可以分成许多并行序列
EK 3.5-C-2
运动、动画和声音是项目的三个方面，通常可以独立实现
EK 3.5-C-3
当项目在执行过程中具有交互特性时，考虑触发触发的动作会对程序执行造成错误或冲突是很重要的
EK 3.5-D-1
考虑负反馈作为调试挑战
EK 3.5-D-2
当某事在你的项目中混淆到他人时，考虑它的呈现方式和必要性
EK 3.5-D-3
基于反馈的迭代过程涉及与对等方的连续来回通信

音乐MV

LA 3.5-1

概述

现在，学生们已经有了声音、精灵和动画的经验，一个自然的进步就是让学生们自己导演音乐视频。一个音乐视频将包含学生在第三单元学到的所有内容，是一个很好的可交付项目。

通过他们目前的工具包，学生们可以通过设计舞台、艺术家、乐器、声音和动画获得真正的创造力。导师应致力于为学生提供足够的时间来实现他们的创造力，同时设法有足够的时间将项目交付给家长。由于这是第3单元的总结活动/项目，导师还应特别注意学生学习进度中的潜在弱点，以便及时解决这些弱点。

活动说明

1. 向学生介绍在mBlock中创建音乐视频的想法，该视频将音乐与动画和自定义绘制的精灵相结合。（可选）显示几个示例项目以激发他们的灵感

2. 确保学生已登录他们的mBlock帐户并开始一个新项目

3. 给学生开放的时间来完成他们的项目，可以选择使用音乐视频讲义来提供指导

4. 帮助学生在开发项目时给予和接受反馈。建议使用批评小组

5. 让学生在设计日志或小组讨论中回顾设计过程
    a. 你克服了什么挑战？你是怎么克服的？
    b.你还想弄清楚什么？
    c.你是如何对借来用于项目的想法、音乐或代码给予赞扬的？

辅助资源

1. 音乐视频活动说明页-<Feishu文件夹位置>
2. 音乐视频项目示例-<Feishu文件夹位置>
3. 评论小组活动说明页-<Feishu文件夹位置>

提示

1. 要进一步个性化项目，请使用“声音”选项卡下的功能，帮助学生加入喜爱的歌曲或录制自己唱歌或演奏乐器的声音
2. 这项活动为导师们提供了一个机会，让他们开始讨论版权、给予信任和归属

导师反省

1. 项目是否结合了精灵和声音？
2. 学生们选择了项目的哪些部分制作动画？
3. 到目前为止，学生们是否仍在努力克服某些障碍或概念？你能帮忙吗？

单元 4 故事时间

概述

在mBlock中移动、跳舞、唱歌和制作动画都很好。但如果没有一个指导性的主题，创意元素只是一块块不相连的宝石。引人入胜的故事情节是一根绳子，它经常能把宝石变成耀眼的项链。故事情节由人物、对话和环境辅助。因此，本单元将介绍在创建故事的这些方面可能有用的编程块。总之，学生们将发展他们的能力，计划、创建、调试和反思一个比以前更全面的项目。本单元的最终项目将是一个协作讲故事项目。

概念与习惯

在编程中，以他人的工作为基础一直是一种长期的做法，只有互联网提供了广泛的他人工作的访问，这种做法才得以扩大。创造性计算的一个重要目标是通过重用和混合来支持学生之间的联系。因此，重用和重新混合是学生应该关注的主要计算实践。至于概念，顺序、事件和并行性仍将占据每个主题的中心位置。

单元一览

课题	关键词	概念	习惯
4.1	模组化 自定义模块 行为	SEQ, EVN, PRL	ANM
4.2	同步 等待 广播	EVN, PRL	RNR
4.3	背景	SEQ, EVN, PRL	ICI
4.4	调试	EVN, PRL	ANM, TND
4.5	合作 讲故事	EVN, PRL	RNR

推荐课堂活动

编号	活动	活动说明页	额外物料
LA 4.1-1	人物	Y	N
LA 4.2-1	对话	Y	N
LA 4.3-1	场景	N	N
LA 4.4-1	调试	Y	N
LA 4.5-1	我们的怪兽	N	Y
LA 4.5-2	混烧	Y	N

课题 4.1

主要思想

SEQ
EVN
PRL
ANM

学习目标 (学生可以...)

LO 4.1-A
设计他们自己的mBlock块，以实现定义良好的行为
LO 4.1-B
描述冗长程序的缺点
LO 4.1-C
欣赏抽象和模块化的价值
LO 4.1-D
使用“生成块”功能在mBlock中创建具有自定义设计行为的交互式角色

必要知识

EK 4.1-A-1
创建块特征的位置
EK 4.1-A-2
命名自定义块与其实现一样重要。块名称应该有意义且精确
EK 4.1-A-3
自定义块的定义不同于实际的自定义块。自定义块只能有一个定义，但定义后，可以根据需要多次使用自定义块
EK 4.1-A-4
可以将自定义块视为包含许多其他块的一个块
EK 4.1-B-1
冗长的程序很难分析、理解和调试
EK 4.1-C-1
自定义块定义了一组可以反复使用的指令
EK 4.1-C-2
自定义块提供了使用命名作为一组指令描述的机会
EK 4.1-C-3
自定义块通常可以缩短程序的长度
EK 4.1-C-4
带有自定义块的精心设计的程序通常读起来像一个食谱
EK 4.1-D-1
自定义块绑定到特定角色。为一个角色创建的自定义块不能用于另一个角色
EK 4.1-D-2
自定义块可以由事件触发

人物

LA 4.1-1

概述

由于第四单元主要是第五单元游戏设计的预备单元，本单元中的大多数活动都有助于学生在有助于设计游戏的环境中发展计算概念和实践。例如，在本活动中，学生将学习如何创建自己的块，这些块表示角色可能执行的各种动作。

此活动中未指定学生通过生成块应执行的确切操作。然而，导师应该提示学生他们可以采取的可能路线，以便更好地为第五单元做好准备。

自定义块对编程有着深远的影响。仅在表面层次上，自定义块使代码段或整个程序更具可读性和可重用性。导师应向学生明确解释这些含义。建议使用字符讲义。

活动描述

1. （可选）显示示例项目，并提供角色讲义以指导学生
2. 让学生有时间使用“我的块”类别中的“生成块”功能创建自己的mBlock块。指导他们设计两个精灵/角色，每个精灵/角色有两个动作/行为。如果需要，将“生成块要素”作为一个类一起进行演练
3. 允许学生相互分享他们的性格和行为。邀请学生向全班展示他们的作品，并演示他们如何实现“生成块”功能
4. 让学生在设计日志或小组讨论中回顾设计过程
   a.你会如何向别人解释制造障碍？
   b.你什么时候可以用它做街区？

辅助资源

1. 字符活动说明页-<Feishu文件夹位置>
2. 示例字符项目-<Feishu文件夹位置>

提示

1. 如果学生正在努力弄清楚如何使用“生成块”功能，请全班一起演练该功能

导师反省

1. 项目是否包含两个精灵，每个精灵使用“生成块”功能具有两种行为？
2. 学生能否向彼此和您解释如何使用“生成块”功能？

课题 4.2

主要思想

Synchronization
EVN
PRL
RNR

学生目标 (学生可以...)

LO 4.2-A
使用等待块组织高级指令序列
LO 4.2-B
使用广播块组织高级指令序列
LO 4.2-C
重新混合给定项目以满足不同的标准集
LO 4.2-D
描述重新组合项目的优点和缺点

必要知识

EK 4.2-A-1
当需要协调或同步多个角色的动作时，可以将等待块编入两个角色的指令中，让一个角色等待另一个角色的动作
EK 4.2-B-1
随着设计元素数量的增加，同步变得越来越困难。广播块很有用，因为所有其他角色可以同时接收广播。角色可以对不同的广播信息做出相应的反应
EK 4.2-C-1
就像调试一样，成功的混音的基础是对原始项目的全面理解
EK 4.2-C-2
重新设计原始项目的某些部分有时比修改现有代码更有效
EK 4.2-C-3
在混音过程中，可以向原始项目添加或从原始项目中减去设计元素
EK 4.2-C-4
如果原始项目是公开的，那么重新混合的项目也应该公开
EK 4.2-D-1
重新组合项目涉及到程序分析开销
EK 4.2-D-2
如果原始项目为构建新项目提供了良好的基础设施，那么重新组合将是有趣和高效的
EK 4.2-D-3
如果原始项目不适合作为新项目的基础，那么重新混合可能是具有挑战性的。
EK 4.2-D-4
在重新组合项目时，新项目的想法并不总是必要的。从原始项目中产生的灵感可以是自然的，并且是混音的良好基础

对话

LA 4.2-1

概述

此活动建立在LA 3.1-1中介绍的关键概念之上，即事件和并行性。事件和并行性在游戏设计中非常有用，大量用于协调或同步指令序列。

学生将首先探索一个给定的入门课程，其中包括两个角色讲笑话。然后，他们将分析笑话在给定程序中是如何同步的。然后，导师将挑战学生使用广播作为发现方法的替代方法。鼓励学生尽可能地重用和重新混合给定的程序，并结合以前主题的知识。建议使用对话讲义。

活动描述

1. 要求学生以小组的形式探索企鹅笑话入门项目，并提供对话讲义以指导学生
2. 确保学生已登录他们的mBlock帐户并开始一个新项目
3. 邀请学生观看企鹅笑话入门项目，观察对话是如何使用等待块进行动画和同步的。让学生重新设计项目，以协调对话，使用广播、广播和等待，以及当我收到块时
4. 鼓励学生彼此分享他们的笑话项目。邀请学生向全班展示他们的作品，并演示他们如何实施广播
5. 让学生在设计日志或小组讨论中回顾设计过程
    a. 你会如何向其他人描述广播?
    b.你什么时候会在项目中使用计时?你什么时候使用广播?

辅助资源

1. 对话活动说明页-<Feishu文件夹位置>
2. 企鹅笑话入门项目-<Feishu文件夹位置>
3. 广播示例项目-<Feishu文件夹位置>

提示

1. 如果学生在理解如何使用广播以及当我收到块对时有困难，请他们以小组的形式探索广泛的演员示例项目

导师反省

1. 项目是否使用广播和接收块?
2. 学生能否用自己的话解释如何使用广播、广播和等待，以及何时我收到块?

课题 4.3

中心思想

SEQ
EVN
PRL
ICI

学习目标 (学生可以...)

LO 4.3-A
比较和对比背景和角色
LO 4.3-B
根据场景启动角色的动作
LO 4.3-C
创建具有多个场景更改和场景特定动作的动画项目

必要知识

EK 4.3-A-1
角色是动态设计元素，可以移动并对其他角色或其环境作出反应
EK 4.3-A-2
背景是静态设计元素，通常表示背景环境或设置场景的气氛
EK 4.3-A-3
角色和背景都可以对某些事件做出反应
EK 4.3-A-4
大多数声音块可用于背景和角色
EK 4.3-A-5
角色和背景都可以有多种造型
EK 4.3-B-1
当背景切换到“块”时，是一个可以触发背景/场景特定动作的事件
EK 4.3-B-2
mBlock项目可以包含多个背景
EK 4.3-B-3
背景切换由块指令启动
EK 4.3-C-1
背景切换到和等待以及背景切换到黑色的时间可以配对，以实现某些同步效果
EK 4.3-C-2
在转移到其他场景之前，应测试场景中的所有动作，以确保其预期行为
EK 4.3-C-3
场景切换中的一个常见错误是角色的行为与场景更改没有很好地协调或同步

场景

LA 4.3-1

概述

在La4.1-1和4.2-1之后，学生现在有了建立多个字符的基础知识，他们自己的行为以同步方式进行，或者使用等待或广播。在本活动中，学生将学习场景/背景。

场景和背景不仅在讲故事中很有用，在游戏设计中也很重要。mBlock游戏通常使用背景技术来模拟不同的环境、级别甚至动作。鼓励学生充分探索mBlock中与背景相关的所有模块，并创造性地利用它们。

活动描述

1. 展示激发学生灵感的示例项目
2. 确保学生已登录他们的mBlock帐户并开始一个新项目
3. 给学生时间开发一个项目，包括使用不同背景的多个场景变化，例如幻灯片。挑战学生探索和操作场景中的脚本，以启动背景变化
4. 允许学生彼此分享他们的项目。邀请学生向全班展示他们的作品，并演示他们是如何实现背景切换的
5. 让学生在设计日志或小组讨论中回顾设计过程
   a.背景与精灵有什么共同之处?
   b.背景与精灵有何不同?
   c.哪些其他类型的项目（动画以外）使用场景更改?

辅助资源

1. 示例场景项目-<Feishu文件夹位置>

提示

1. 如果学生在了解如何切换背景时遇到困难，鼓励他们修补“外观”类别下的块，尤其是切换背景到、切换背景到并等待以及下一个背景块

导师反省

1. 项目是否使用不断变化的背景成功地协调多个场景?

课题 4.4

中心思想

EVN
PRL
ANM
TND

学习目标 (学生可以...)

LO 4.4-A
找到与事件、并行性和自定义块相关的一系列错误的解决方案
LO 4.4-B
描述自定义块的参数是什么
LO 4.4-C
捕获用户输入的类型
LO 4.4-D
在组织指令序列时合并广播和等待块

必要知识

EK 4.4-A-1
在mBlock中，与并行性和事件相关的bug对于调试来说是乏味的，因为它涉及到在多个精灵之间来回切换。成对工作可以大大提高调试效率
EK 4.4-A-2
如果程序相当简单和简短，那么模拟程序的执行是识别错误的一种很好的方法
EK 4.4-B-1
参数允许将信息传递到自定义块中
EK 4.4-B-2
参数进一步增加了给定自定义块的灵活性
EK 4.4-B-3
参数影响自定义块的输出，但不影响逻辑定义
EK 4.4-C-1
ask and wait块将在后台提示问题并等待用户输入确认
EK 4.4-C-2
在某些情况下，键盘的输入设置可能会影响mBlock中是否正确捕获信息
EK 4.4-C-3
应答块存储从询问和等待块捕获的用户输入
EK 4.4-C-4
尽管mBlock允许文本作为输入，但当程序员使用应答块而不考虑它是文本还是数字时，这是一个常见的错误
EK 4.4-D-1
广播和等待块在点名场景中非常有用，或者当许多角色必须按顺序采取行动时

调试

LA 4.4-1

概述

与LA 3.4-1相同，此活动将迭代一组不同的测试和调试问题。一般目的和战略不变。

活动描述

1. 在活动期间提供调试it讲义以指导学生

2. 指导学生在mBlock中实现给定的程序，并在其计算机上执行测试和调试。确保学生将每个更正的程序保存为新项目。

3. 要求学生在设计日志或小组讨论中回顾他们的测试和调试经验
    a. 问题/错误是什么?
    b.您是如何识别问题/bug的?
    c.其他人是否有其他解决问题/bug的方法?

4. 收集学生发现问题和解决问题的方法，生成调试策略列表

辅助资源

1. 调试活动说明页-<Feishu文件夹位置>

提示

1. 能够阅读他人的代码是一项宝贵的技能，对于能够参与重用和重新组合的实践至关重要
2. 这项活动是结对编程的绝佳机会。将学生分成两人进行调试挑战
3. 学生可以通过右键单击mBlock块插入代码注释来解释他们的代码修订

导师反省

1. 学生们能解决所有五个错误吗？如果没有，您如何澄清未解决方案中表达的概念?
2. 学生们采用了哪些不同的测试和调试策略?

课题 4.5

主要思想

EVN
PRL
RNR

学习目标 (学生可以...)

LO 4.5-A
描述合作中的握手过程
LO 4.5-B
欣赏重用和重新混合的价值
LO 4.5-C
创建一个开放式协作讲故事mBlock项目

必要知识

EK 4.5-A-1
在协作过程中，人们通常负责项目的不同方面，这些方面最终必须拼凑在一起
EK 4.5-A-2
协作中的某些任务依赖于其他任务
EK 4.5-A-3
当协作中出现依赖关系时，握手过程至关重要
EK 4.5-A-3
握手过程主要是就应该考虑但可能不是由于专业知识和知识范围有限的事情交换提示和笔记
EK 4.5-B-1
重复使用是为了提高效率，而不是重新发明轮子
EK 4.5-B-2
混音是关于创造力和取得令人惊讶的结果
EK 4.5-B-3
当一个项目的可重用性与remixer的创意相匹配时，两个世界的最佳结合点就会合二为一
EK 4.5-C-1
故事情节是一个强大的工具，可以将许多创造性的想法结合在一起，并帮助理解它们
EK 4.5-C-2
开放式协作项目没有明确的目标或标准
EK 4.5-C-3
参与者应在开放式合作环境中尊重彼此的创意
EK 4.5-C-4
每个成员或组都应该负责调试他们实现的代码

我们的怪物

LA 4.5-1

概述

这是一项无需电脑的活动，利用绘画帮助学生加深对再利用和混合的概念和过程的理解。学生们将各自负责绘制未知生物的部分。

这项活动的一个关键收获是认识到合作的惊人创造力。另一个是认识到良好的移交过程的重要性。交接流程在协作中至关重要，因为负责一项任务的人员可能不熟悉其他任务。因此，在任务到任务的转换过程中，高质量的沟通和信息交换应该以明示或暗示的方式进行。

额外物料

1. 空白画纸，折叠成第三张
2. 素描用的东西（铅笔、钢笔、记号笔等）

活动说明

1. 在这项活动中，学生们将画一个“生物”，分为三个部分

2. 给每个学生一张三折的空白画纸和一分钟的时间为他们的生物画一个“头”。接下来，让他们把纸折叠起来，这样头部就被隐藏起来了，并且几乎不需要提示在哪里继续绘图。头部隐藏后，学生将该生物传递给另一名学生。然后，给学生时间为他们的生物画一个“中间”，使用头部的引导，但不要偷看。隐藏中间点（并绘制提示）后，传递生物。最后，给学生时间为他们的生物画一个“底部”。完成后，展开纸张，展示协作构建的生物

3. 将图纸张贴在墙上或黑板上，让学生探索其创造性贡献的结果

4. 促进关于合著、协作、重用和混合作品的小组讨论
    a. 你对混合的定义是什么?
    b.想想你创造的生物。你的想法是如何通过他人的贡献得到扩展或增强的?
    c.考虑你扩展的生物，你的贡献是如何延伸或增强他人的想法的?

辅助资源

N/A

提示

1. 本活动是LA 4.5-2的热身活动
2. 让学生在他们绘制的每一幅生物画的底部签名，以确定贡献的艺术家

导师反省

1. 学生能否解释混合和协作及其好处？艺术家

混烧

LA 4.5-2

概述

通过完成此活动，学生将产生他们的第三个可交付项目。学生们应该利用到目前为止所学的一切，重点放在第四单元所涵盖的主题上。

这项活动将是关于协作和讲故事的。与《洛杉矶4.5-1》非常相似，学生将两人一组完成mBlock中的一个故事讲述项目，并将其他人的项目进行混音。建议使用“传递信息”讲义。

活动说明

1. 把全班分成两人一组。向学生介绍传递故事的概念，这是一个由两人启动的mBlock项目，然后传递给其他两人进行扩展和重新构思

2. 鼓励学生以任何他们想要的方式开始——关注人物、场景、情节、动画、声音或任何让他们兴奋的元素。在让他们轮换通过混合项目来扩展另一个故事之前，给每一对时间来完成他们的合作故事

3. 经过多次轮换后，允许学生重新阅读故事项目及其贡献。邀请学生展示故事项目，让学生围观

4. 要求学生在设计日志或小组讨论中重新思考设计过程
    a.在他人作品的基础上进行混音和构建感觉如何？混音的感觉如何？
    b.在你的生活中，你在哪里见过或经历过重复使用和混合？
    c.与其他人合作与您以前设计mBlock项目的经验有什么不同？

辅助资源

1. 在讲义上传递-<Feishu文件夹位置>

提示

1. 由于这是一个相对全面的可交付项目，可能涉及到所有先前的知识，导师应该关注学生之间的困惑
2. 鼓励学生从他们在LA 4.1-1、4.2-1和4.3-1中完成的项目中汲取灵感
3. 在项目交付给家长期间，建议进行筛选活动。邀请家长观看，并提供点心和饮料

导师反省

1. 学生对项目的哪些部分做出了贡献？
2. 学生们对事件和并行性的概念以及重用和混合的实践是否感到满意？如果没有，可以通过哪些方式进一步澄清？

单元 5 游戏厅派对

概述

个性化是创造性计算体验设计的重要指导原则。个性化意味着既要联系个人利益，又要承认个人利益可能会有很大差异。知道和做的方法有很多，探索这些方法有助于培养年轻学生的兴趣、动机和毅力。在本单元中，学生探索与游戏设计相关的一些高级概念和挑战性问题。如果一个高级概念或具有挑战性的问题植根于对个人有意义的活动中，那么它可以变得更容易理解。本单元的最后一个项目是一个复杂的游戏，涉及本课程所涉及的所有先前概念。

概念与习惯

本单元将介绍的三个相对更高级的计算概念是条件、运算符和数据。游戏设计是发展这些概念的完美主题。至于计算实践，在这一点上，学生应该已经相对熟悉其中的大多数。然而，随着项目规模和复杂性的增加，测试和调试的作用也越来越大。

单元一览

课题	关键词	概念	习惯
5.1	游戏设计 启动项目	PRL, CON, OPT, DAT	RNR
5.2	变量 分数	CON, DAT	TND, RNR
5.3	游戏机制 拓展	CON, OPT, DAT	TND
5.4	互动 解谜 调试	CON, OPT, DAT	TND

推荐课堂活动

编号	活动	活动说明页	额外物料
LA 5.1-1	完美游戏	N	Y
LA 5.1-2	启动项目	Y	N
LA 5.2-1	分数	Y	N
LA 5.3-1	拓展	Y	N
LA 5.4-1	互动	Y	N
LA 5.4-2	调试	Y	N

课题 5.1

主要思想

PRL
CON
OPT
DAT
RNR

学习目标 (学生可以...)

LO 5.1-A
熟悉mBlock中的一系列方块
LO 5.1-B
表达他们对游戏的定义
LO 5.1-C
描述mBlock中常见的游戏机制
LO 5.1-D
创建一个自我指导的游戏项目，该项目将在本单元中迭代和扩展

必要知识

EK 5.1-A-1
If-then和If-then-else块在程序执行中创建拆分。根据这些块指定的条件是否满足，采用不同的执行路径
EK 5.1-A-2
mBlock中的条件以尖矩形的形式出现
EK 5.1-A-3
运算符块在mBlock中为绿色，它们主要执行算术计算和比较
EK 5.1-A-3
变量可以设置为某个数字或增加某个数字
EK 5.1-B-1
游戏中的一些常见元素是目标、奖励、挑战、规则和互动
EK 5.1-C-1
当按下键时，事件与运动块组合控制精灵的移动
EK 5.1-C-2
通过使用if-then块检查循环中的接触条件，可以捕获碰撞
EK 5.1-C-3
并行性在游戏设计中被广泛使用
EK 5.1-C-4
拾取随机数块可用于在游戏中随机化动作和效果
EK 5.1-D-1
初学者项目不需要以任何方式完成或完善功能
EK 5.1-D-2
游戏设计需要大量的尝试和错误
EK 5.1-D-3
设计日志应该大量用于自我指导的游戏项目

完美游戏

LA 5.1-1

概述

现在，当谈论游戏时，大多数年轻学生会想到电脑游戏和手机游戏。而在现实中，游戏的定义是灵活的。大多数游戏中常见的一些关键组件是目标、规则、挑战和交互。这些组件可能在游戏中同时存在，也可能不存在。它们可能以各种形式存在。例如，Minecraft似乎没有为玩家设定明确的目标。然而，它提供了一个沙箱，玩家可以根据自己设定的目标发挥想象力。

一个好的游戏是由一个完全不同的讨论组成的。图形、声音、机械、故事情节和难度都会对最终产品的感知产生巨大的影响。然而，一个关键的收获是，一个好的游戏并不总是需要复杂或由经验丰富的专业人士设计。有时候，一个好主意比其他任何东西都更出色。

本活动将让学生参与有关游戏、游戏设计及其他方面的公开讨论。

额外物料

1. 纸和便签
2. 素描用的东西（铅笔、钢笔、记号笔等）

活动说明

1. 把学生分成小组
2. 在小组中，要求学生列出他们喜欢玩的游戏。他们可以使用自己的设计日志撰写列表。给学生一段短时间写下尽可能多的游戏。然后让学生缩小他们最喜欢的
3. 一段时间后，询问小组他们的游戏列表
   a.这些游戏有什么共同点?
   b.他们的设计有哪些特点使他们成为一款游戏?
4. 促进课堂讨论游戏的特点，并生成常见游戏机制的课堂列表。接下来，让学生想象他们的梦想游戏，并为该游戏编写一份设计元素列表
5. 邀请学生在小组中分享他们的梦想游戏列表，以获得反馈和建议

辅助资源

N/A

提示

1. 在本单元其他相关活动中编写游戏时，请学生参考此梦想游戏列表

导师反省

1. 梦想游戏列表是否包含游戏的功能?

启动项目

LA 5.1-2

概述

尽管学生们现在已经发展了许多计算概念和实践，但mBlock中的计算游戏设计也有其独特的挑战。这项活动提供了一些初学者游戏项目供学生选择，作为未来活动的基础。

学生将收到迷宫、乒乓球和Flappy Pig的讲义。这些讲义为各种重要的游戏机制提供了指南或代码段。导师应该意识到，这只是针对缺乏灵感或游戏设计经验的学生的三种选择。如果一些学生有明确的想法和动机来实施他们自己的初学者游戏项目，这是完全可以接受的。

设计游戏是学习条件、运算符和数据等计算概念的好方法。它还与实验和迭代、测试和调试、重用和混用、抽象和模块化等实践密切相关。导师应在课堂上关注这些方面。

活动说明

1. 在活动中，学生将创建一个入门游戏项目，可在LA 5.3-1、5.4-1和5.5-1期间重新访问和扩展。提供初学者游戏讲义，以指导学生

2. 选择一个游戏项目作为一个课堂，或者让学生选择他们想要创建的游戏。给学生时间开始制作他们的游戏，或者让他们混合一个初始项目

3. 鼓励学生对正在进行的游戏获得反馈。让一半的学生坐在座位上，打开他们的项目，而另一半则四处探索项目，提出问题，给出反馈，然后交换立场

4. 让学生在设计日志或小组讨论中回顾他们的设计过程
    a. 设计游戏有什么挑战性?
    b. 你以什么为荣?

辅助资源

1. 初学者游戏活动说明页-<Feishu文件夹位置>
2. 入门游戏项目示例-<Feishu文件夹位置>

提示

1. 在第5单元结束时，在项目交付期间，建议举行街机活动。让学生将他们的项目置于演示模式，邀请家长和学生四处走动，玩其他人的游戏
2. flappy pig游戏引入了克隆技术。如果需要，请用简单的术语解释这个想法，因为第5单元将对其进行更多介绍

导师反省

1. 游戏是否包含条件、运算符和数据?

课题 5.2

主要思想

CON
DAT
TND
RNR

学习目标(学生可以...)

LO 5.2-A
描述什么是变量以及为什么变量有用
LO 5.2-B
通过混合给定程序构建评分系统

必要知识

EK 5.2-A-1
创建变量是为了在程序中存储信息
EK 5.2-A-2
变量名称应简洁且有意义，代表存储的信息
EK 5.2-A-3
没有变量，mBlock程序几乎不可能记住信息
EK 5.2-A-4
变量可以设置为特定的数字，也可以增加特定的数字。它们只能是mBlock中的整数
EK 5.2-B-1
要为比赛实施计分系统，必须明确规定计分条件
EK 5.2-B-2
在整个比赛过程中，必须建立并持续检查得分条件
EK 5.2-B-3
程序需要至少一个变量来记忆任何时刻的分数并相应地递增

分数

LA 5.2-1

概述

得分系统是游戏设计的关键部分。这是反映进步、回报或目标的最直接方式。它对游戏设计的必要性将促使学生密切关注变量和数据等概念，这些概念对学生来说传统上是枯燥而令人生畏的主题。变量和数据构成了许多计算模型的基础，在编程中无处不在。因此，导师应致力于确保所有学生对这些概念有坚实的理解。

在本次活动中，将向学生介绍一个缺乏评分机制的示例游戏项目。学生将首先了解其当前的实现，并探索变量和数据的概念，以重新混合项目。使用分数讲义是可选的。

活动描述

1. 以小组的形式探索示例游戏项目，并有选择地提供分数讲义以指导学生

2. 确保学生已登录他们的mBlock帐户并开始一个新项目

3. 让学生有时间通过混合样本项目探索变量，为游戏添加分数。一旦学生完成了样本项目的混音，给学生时间将分数纳入他们之前在洛杉矶5.2-1开始的游戏项目中

4. 允许学生分享他们的示例项目混音或添加分数的游戏项目。邀请学生向小组展示他们的项目，并演示他们如何使用变量实施分数

5. 让学生在设计日志或小组讨论中回顾设计过程
    a. 你会如何向其他人解释变量?
    b.变量有什么好处?

辅助资源

1. 分数活动说明页-<Feishu文件夹位置>
2. 示例游戏项目-<Feishu文件夹位置>

提示

1. 变量是一个重要的数学和计算概念。学生们在数学课上学习变量，但许多学生学习变量很困难。游戏是使变量的有用性更具体的一种方法
2. 没有变量，计算机程序就无法记忆任何东西。换句话说，计算机使用变量来记忆信息

导师反省

1. 学生们能解释什么是变量以及变量的用途吗?

课题 5.3

主要思想

CON
OPT
DAT
TND

学习目标 (学生可以…)

LO 5.3-A
描述mBlock中多个游戏机制的工作原理
LO 5.3-B
使用更常见的游戏机制扩展他们的游戏项目

必要知识

EK 5.3-A-1
只有在满足某些条件时，才可以通过增加液位变量来实现液位。这很像一个带有后果的计分系统
EK 5.3-A-2
利用重置计时器和计时器块可以实现计时器效果
EK 5.3-A-3
一个重启按钮可以通过一个自定义绘制的按钮来实现，当这个精灵点击块时，除了一些广播来重置其他精灵
EK 5.3-A-4
多人游戏可以通过两组独立的控件来实现，这些控件对应于不同的精灵和动作
EK 5.3-B-1
当扩展添加到原始项目中时，测试和调试对于确保不会引入可能导致灾难的bug至关重要
EK 5.3-B-2
I邀请同龄人对课程的每一次修订进行审查可以产生非常有用的反馈

拓展

LA 5.3-1

概述

除了在LA 5.3-1中引入的计分系统外，还有许多其他理想的游戏设计元素，学生可能希望添加到他们自己的项目中。在本次活动中，学生将有机会探索一系列扩展，包括关卡、计时器、敌人、奖励、鼠标、重启、菜单和多人游戏。每个扩展都有自己的示例项目来演示其效果。希望学生分析这些示例项目，了解其内部工作原理，并在自己的游戏项目中实施。

至于计算概念和实践，第5单元的主题是条件、运算符和数据。本活动中引入的不同扩展只是应用这些概念的不同方式。因此，如果学生们正在努力为他们的项目添加扩展，这可能表明应该重新审视这些概念。

活动描述

1. 展示示例项目，并提供扩展讲义以指导学生

2. 给学生时间探索示例程序的代码，以研究增加游戏难度或扩展游戏的不同方式。要求学生选择一个或多个扩展以添加到先前开始的游戏项目中。给学生时间进行实验，并将扩展部分融入到他们的游戏中

3. 允许学生彼此分享他们的扩展游戏项目。邀请学生向小组展示他们的项目，并演示他们如何实施各种扩展

4. 让学生在设计日志或小组讨论中回顾设计过程
    a. 在游戏中增加难度的不同方法有哪些?
    b.您在游戏项目中添加了哪些扩展?
    c.描述在游戏中包含扩展的过程

辅助资源

1. 扩展活动说明页-<Feishu文件夹位置>
2. 示例扩展项目-<Feishu文件夹位置>

提示

1. 为了给需要额外支持的学生提供更多的支架，我们建议学生在课堂上浏览一个扩展示例项目，并帮助学生将扩展添加到他们的游戏项目中

导师反省

1. 学生们能够将扩展融入他们最初的游戏项目吗?

课题 5.4

主要思想

CON
OPT
DAT
TND

学习目标 (学生可以...)

LO 5.4-A
将交互机制纳入他们的游戏项目中
LO 5.4-B
在他们的评论小组中接受并给出项目反馈
LO 5.4-C
对复杂项目执行测试和调试

必要知识

EK 5.4-A-1
声级可以触发某些动作
EK 5.4-A-2
碰撞检测有三种主要类型-基于颜色、基于角色和基于边界
EK 5.4-A-3
角色可能会出现在一些简单的循环块的鼠标指针之后

互动

LA 5.4-1

概述

本活动由9个拼图组成，每个拼图对应一个互动效果，学生可以直接在游戏项目中实施或从中获得灵感。这些交互效果包括按键、碰撞、距离更改、鼠标定位、鼠标单击等结果。

拼图形式因素鼓励学生竞争、分享和合作。导师应注重为这项活动保持一个友好和相对有组织的环境。导师还应了解学生是否仍在努力解决某些障碍或概念。

活动描述

1. 通过创建mBlock程序，解决9个交互编程难题中的每一个，让学生自己或在小组中挑战，进一步探索mBlock。这些交互谜题探索感知块，涉及mBlock中与交互相关的一些更高级的概念

2. 在活动期间提供互动讲义以指导学生

3. 确保所有学生都已登录他们的mBlock帐户，并为9个谜题中的每一个启动一个新项目

4. 每个谜题都有几种可能的解决方案。邀请学生或小组分享不同的解决方案和策略

5. 让学生在设计日志或小组讨论中回顾挑战
    a. 你做了哪些拼图?
    b.你解决这些难题的策略是什么?
    c.哪些谜题帮助你思考你的游戏项目?

辅助资源

1. 互动活动说明页-<Feishu文件夹位置>

提示

1. 选择特别的挑战，突出新的模块或概念，你希望学生有更深的理解，作为一个班级一起复习

导师反省

1. Are t难题解决了吗?
2. 学生们是否探索了其他解决谜题的方法?
3. 学生们是否仍在努力解决某些障碍或概念？你能帮忙吗?

Debug It
LA 5.4-2

概述

与LA 4.4-1相同，此活动将迭代一组不同的测试和调试问题。一般目的和战略不变。

活动描述

1. 在活动期间提供调试活动说明页讲义以指导学生
2. 指导学生在mBlock中实现给定的程序，并在其计算机上执行测试和调试。确保学生将每个更正的程序保存为新项目。
3. 要求学生在设计日志或小组讨论中回顾他们的测试和调试经验
    a. 问题/错误是什么?
    b.您是如何识别问题/bug的?
    c.其他人是否有其他解决问题/bug的方法?
4. 收集学生发现问题和解决问题的方法，生成调试策略列表

辅助资源

1. 调试活动说明页-<Feishu文件夹位置>

提示

1. 这项活动提供了一个机会，让学生了解可能需要额外关注或支持的内容，特别是关于条件句、运算符和数据的概念

导师反省

1. 学生们能解决所有五个错误吗？如果没有，您如何澄清未解决方案中表达的概念?
2. 学生们采用了哪些不同的测试和调试策略?

Appendix 附录

Behavior Feedback
行为反馈

	Rarely or Not Observed 几乎不或未察觉	Sometimes 偶尔	Often 经常
Hyperactive 过度兴奋			
Physically restless 肢体			
Verbally restless 口头			
Nervous muscle twitching 肌肉抖动			
Withdrawn 过度消极			
Tried 疲惫			
Unhappy 沮丧			
Stares blankly into space 呆滞			
Doesn’t asks for assistance when needed 不寻求帮助			
Doesn’t attempt work 不尝试课堂活动或任务			
Avoids attention to self 不愿被注意到			
Poor Attention 难以专注			
Can’t follow oral/on board/visual materials/instructions 无法跟从口头/板书/视觉引导			
Doesn’t complete any classwork 无法完整完成任何课堂活动或任务			
Demands repeated instructions 需要不断重复说明			
Distracted by ordinary stimuli 对常规干扰因素过于敏感			
Agressive 过度侵略性			
Physical aggression 肢体			
Verbal aggression 语言			
Gets angry when things don’t go his/her way 不顺其意愿时易怒			
Uncooperative 低配合度			
Blames others for own mistakes 因自身错误责怪他人			
Works only under negative reinforcement 只对负面强化做出反应			
Argues with peers over minor issues 常因无关紧要的事与同学争吵			
Inappropriate Social Behaviors 不恰当社交现象			
Doesn’t initiate play or talk with peers 不主动开始玩耍或交流			
Avoided by peers 被孤立或排挤			
Voices concerns about not being liked by others 表示不被同学或老师喜欢			

Positive Behaviors | Additional Comments 积极行为 | 附加评价

Product Feedback 产品反馈

Fluidity 连贯性

Subjectively, how well does the content of this topic fit into the entire course? If this is the first topic, is the content appropriate as preparatory material? If not, does the content have a fluid and logical connection to the contents from previous topics?
主观感受上，此课题内容在整体课程大纲中是否恰当？如果这是第一个课题，那么课题内容作为预备内容恰当吗？如果这不是第一个课题，此课题内容是否与之前的课题内容是否连贯？从教学感受与逻辑连接层面考虑

Rating 评分 ______ (1 - 5)

Elaboration 阐述

Topic Chuck 课题体量

Setting aside the influences of the learning activities, is the amount of learning objectives and essential knowledge too much or too little?
抛开具体的教学活动安排和进展情况来说，本课题的学习目标和必要知识的体量是否合理？

Too little 太少 ______ Too much 太多 ______ Appropriate 适度 ______

Elaboration 阐述

Learning Activity 教学活动

In terms of integration and engagement, rate all learning activities conducted for this topic
从活动与学习目标的贴合度以及学生在活动进行中的参与度考虑，1-5分评价所有此课题中涉及到的活动

ID 代号	Integration 贴合度	Engagement 参与度

Comments and Suggestions 附加评价与改进建议