

STUDIO K

Studio K: A Game Design Curriculum for Computational Thinking

Wade Berger, Gabriella Anton, Emanuel Rosu, Tyler Banh, Jeremy Dietmeier
Project Leads: Ben Shapiro, David Hatfield, Matthew Berland, and Kurt Squire

Studio K is a game design curriculum, online community, and set of teacher-support tools intended to enable teachers to help learners to make their own video games using Microsoft Kodu. Kodu is a powerful 3D game design tool with visual programming that enables users to focus on creating compelling games for their friends.

The Studio K curriculum, embedded in our online community, is designed to support the development of computational thinking and problem solving skills. Policymakers have identified computational thinking as a critical set of skills that students must be equipped with in order to be successful in STEM careers, as well as to fully participate in today's and tomorrow's creative society.

Studio K is designed with the notion that students' interests should help guide their learning experiences. Within the curriculum, students explore challenges around the mechanics, aesthetics, and dynamics of game design that have them playing, fixing, creating, and reviewing games with Kodu. Studio K supports collaboration and peer-supported learning, asking to interact with each other to facilitate the acquisition of key concepts.

The following document will outline the core curriculum of Studio K based around Kodu, including lessons, alignment to Common Core Standards, and website guides.

Table of Contents

Introduction	5
Curriculum Design	6
Site Tour	8
Researching Game Design.....	15
Technical Requirements	16
Connections to Computational and Systems Thinking	17
STEM Standards Alignment.....	19
Studio K 101	23
General Learning Goal	23
Mission - Joining the Studio.....	23
Mission - Introduction – Game Design 101	23
Mission - Setting the Scene	24
Mission - Getting the Guy	26
Mission - Making Moves.....	27
Mission - In It to Win It.....	29
MECHANICS	31
General Learning Goal	31
Mission - Copycat: Copying, cloning, object control	31
Mission - Now You See Me: Big picture of broadcasting	34
Mission - High Score!: Engineering scoring systems.....	36
AESTHETICS.....	39
General Learning Goal	39
Mission - Landcrafting: Designing the game world.....	39
Mission - Kodu's Keeper: Creative control of assets	41
Mission - omg DRAMA: Narrative and dramatic elements.....	42
Mission - Creative Genius: Mixing all the aesthetic elements together.....	43
DYNAMICS	45
General Learning Goal	45
Mission - Tick Tock BOOM: Mastering timers and scores.....	45
Mission - Pwnd!: Designing the competition	47
Mission - Struttin': Commanding the Pacing	49
Mission - Whole New World	50
BOSS FIGHT	53
General Learning Goal	53
Mission - Like a Boss: The final challenge – do you have what it takes?	53
Mission - Roll Credits: Recognizing your skills.....	54
Appendix.....	55
Appendix A: Entry Research Survey	55
What happens when the Kodu is on land type 1?	57
Appendix B: Exit Research Survey	58
Appendix C: Studio K Vocabulary List.....	60

Introduction

In the Games+Learning+Society at the University of Wisconsin-Madison, we interact with games constantly – designing them, developing them, and researching the learning and activities that occur during and after play. Games can be very meaningful tools for teaching and can motivate students to think perceive content in new ways.

Studio K is designed to be one solution to the question of how to get youth more engaged and interested in programming. Rather focusing on programming, which can be daunting to learn, Studio K harnesses interest in games – providing game design content with scaffolded programming supported by Microsoft Kodu. With this format, we hope to provide a safe, unique environment for students to begin interacting with programming.

Originally run by researchers at GLS as an after school program, Studio K has grown to include an online curriculum hosted with facilitator and student website community features that support interest-driven learning and self-directed study. We strongly believe that students should have the opportunity to participate in a game design club whether or not they have access to an in-person course or club – Studio K offers this opportunity for youth. The curriculum can be applied in a variety of learning spaces – ranging from classroom use to after-school programming to guided study at home or in the library. Students can safely explore their interests and gain expertise in a vast array of game making practices with use of the site.

In order to join the program, an adult facilitator needs to register for an account, make a class, and then invite students to join. Each class will include community features for asynchronous communication and collaboration that are independent from other running classes. The secured format allows students the freedom to interact with their peers while ensuring that they remain safe from any negative interactions online.

Researchers at the Games+Learning+Society Center are studying the activities that occur around the curriculum within this online space and during game design sessions in Microsoft Kodu. Through the study of these activities, we hope to better understand how students are learning programming and computational thinking practices, and how students' interest and trajectory in computer science changes with their experience in an interest-driven game design program.

Curriculum Design

This curriculum is designed to get students to think about games as systems of interrelated parts and components, each of which can be changed or manipulated in order to affect the player experience. In Mario, for example, the goal of the game is simply “get to the end of the level”; however, if you changed the goal of the game to “collect all the coins”, the player will have to interact with and play through the game differently.

We begin by breaking down the understanding of games into three dimensions: Mechanics, Aesthetics, and Dynamics. We build a framework of design based on the following definitions:

Mechanics: the base components of every game that control how a player acts during the game. These include both the rules of the games and the underlying programming that is controlling the actions.

Aesthetics: the desirable emotional responses evoked by the player, controlled through the manipulation of world design, narrative, and dramatic elements of the game.

Dynamics: the outcome of mechanics and aesthetics interacting to create a player experience. Exploration of dynamics considers altering mechanics and aesthetics in order to achieve balance in a game and engagement for players.

The Mechanics, Aesthetics, and Dynamics framework is broken down into a series of missions, each focusing on a key game design or programming concept. Within each mission, students are presented with a number of challenges. While the curricular units can be completed in any order, we suggest that students start first with either Mechanics or Aesthetics before continuing onto Dynamics. The structure of the curriculum encourages a multi-faceted interaction with the content, so each mission will have a mixture of the following challenge types.

Watch: The students will view a video that introduces game design or programming content.

Play: The students will play a selected game that highlights either good or bad game design elements. After playing, students will be asked to answer critical questions about the game.

Create: The students will be challenged to make a game that focuses on the game design or programming concept from the mission.

Playtest: Emphasizing a crucial phase of game design, students will be asked to play and review peer created games from the mission, leaving critical feedback for classmates.

At the end of the curriculum, the students will be asked to complete a Boss fight – or the final create challenge that combines all the game design and programming concepts together. This challenge is more open than the others, encouraging students to really explore their personal interests in games.

Curriculum facilitators are encouraged to interact with their students as they complete the online curriculum. While all the content is included on the website, critical thinking about game design is improved through conversation about designs and games. Another important part of this curriculum is allowing students to call on their own experiences with games, which will help them to understand the lessons in terms of those experiences. For example, they will be able to more concretely think about the Rules of their own racing game if they can relate them to the Rules of Mario Kart or Need For Speed.

It's important to keep in mind is that, with respect to the students' game designs, there are almost never any right or wrong answers. Your students will most likely have very different experiences with video games and will have different ways of thinking about video games, especially what they find fun and why. So, in addition to the game design and computational thinking skills, this curriculum should also provide students a safe arena in which to discuss this shared interest and learn how to focus their ideas in order to think more like professional game designers.

In each mission, we have provided some teacher tools to help get you thinking critically about the concepts and to guide the curriculum facilitation. These will also appear on the website within each mission. You are encouraged to use the 'Teacher Lounge', or facilitator's forum page, on the website to interact with other facilitators to share experience, receive help and feedback, and build a strong Studio K community.

Site Tour

The Studio K website has been designed to provide students with the necessary support to explore their interests in game development, by both supplying information on how to make games and a safe environment to interact with peer budding game developers. The features that enable this experience are the homepage, gallery, mission page, community page, help and resources, and the ‘teacher’s lounge’

Homepage

The homepage holds different information for facilitators and for students. For students, the homepage includes a personalized avatar, a newsfeed of activities of students from the class, the levels that the student has uploaded on the site, the reviews that the student has left for others, and an uploading feature for free-design games created outside of the curriculum (see Figure 1.1).

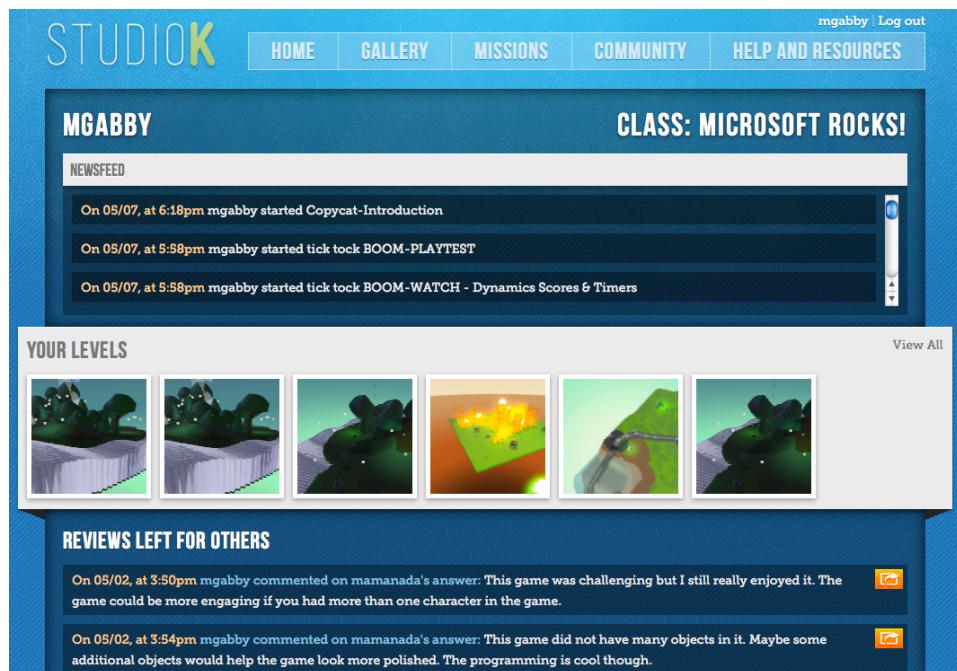


Figure 1.1 the homepage for students.

For facilitators, the homepage is the resource to track all students’ progress in the curriculum. Here, teachers can add multiple teachers to the class, view the newsfeed with progress updates from students in the

class (see Figure 1.2), and see the progress for each student on all challenges within the curriculum (see Figure 1.3).

The screenshot shows the STUDIO K homepage for facilitators. At the top, there is a navigation bar with links for HOME, GALLERY, MISSIONS, COMMUNITY, and HELP AND RESOURCES. A user 'tom' is logged in, with a 'Log out' option. Below the navigation bar, the title 'MICROSOFT ROCKS!' is displayed, along with 'MAIN TEACHER: TOM' and 'Co-teacher: teacher Co-teacher: teacher_tr Co-teacher: tom'. There is a search bar for 'Teacher name' and an 'ADD' button. A 'NEWSFEED' section shows recent activity: 'On 05/08, at 1:18am mgabby started Copycat-Introduction', 'On 05/08, at 12:58am mgabby started tick tock BOOM-PLAYTEST', and 'On 05/08, at 12:58am mgabby started tick tock BOOM-WATCH - Dynamics Scores & Timers'. A scroll bar is visible on the right side of the newsfeed.

Figure 1.2 the homepage for facilitators, showing the teacher access tools and newsfeed

The screenshot shows the COMET44 homepage for facilitators, specifically for student 'comet44'. At the top, there is a dropdown menu for the student and a 'REFRESH' button. The main area is divided into several sections: 'STUDIO K 101' (with items like 'Joining the Studio', 'Introduction', 'Setting the Scene', 'Getting the Guy', 'Making Moves', and 'In It to Win It'), 'COPYCAT' (with items like 'Introduction', 'WATCH - Cloning and Creatables Tutorial', 'CREATE - Cloning & Creatables', 'WATCH - Teleportation Tutorial', and 'CREATE - Teleportation'), 'NOW YOU SEE ME' (with items like 'PLAY - Stay out of my yard!', 'WATCH - Broadcasting Tutorial', 'CREATE - Option #1 Red Light - Green Light', and 'CREATE - Option #2 Musical Chairs'), 'HIGH SCORE!' (with items like 'PLAY - Pong' and 'PLAY - Survival Game'), 'LANDCRAFTING' (with items like 'PLAY - Water World' and 'WATCH - Land Design Intro'), and 'KODU'S KEEPER' (with items like 'PLAY - Water World 2.0' and 'WATCH - Using Objects'). Each item has a green checkmark or a red X next to it, indicating completion status.

Figure 1.3 the homepage for facilitators, showing an individual student's progress through the curriculum.

Gallery

The gallery holds all the games that students have uploaded to the site. Students can upload games outside of the curriculum and then share them with their class with a click of a button.

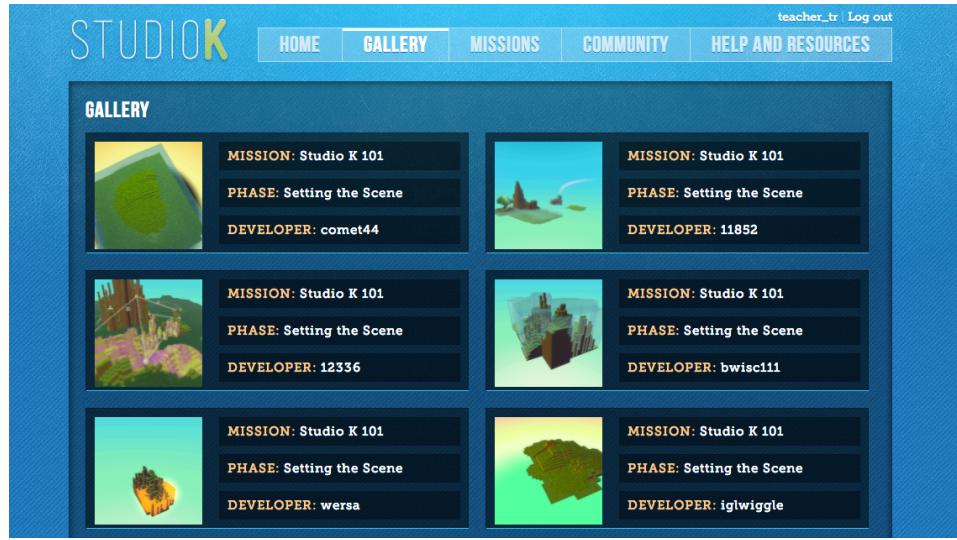


Figure 1.4 the gallery, housing all student games.

By clicking on a game in the gallery, one can see a description of the game, reviews left on the game, and similar games made by other students.

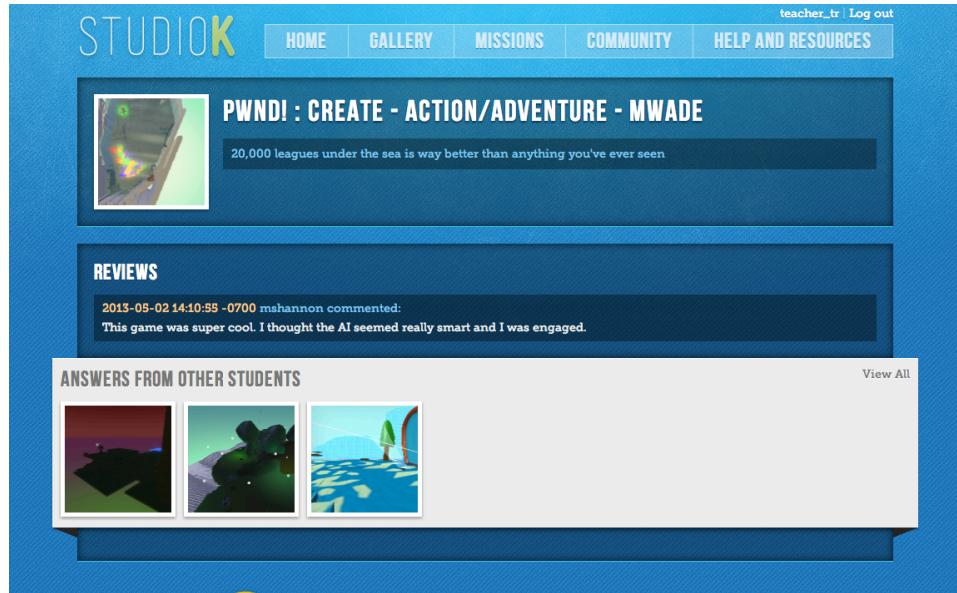


Figure 1.5 the individual game page available for each submitted game, including reviews and answers from other students.

Mission page

The mission page houses the curriculum map, broken down into smaller units with focused learning goals called ‘missions’ (see Figure 1.6).

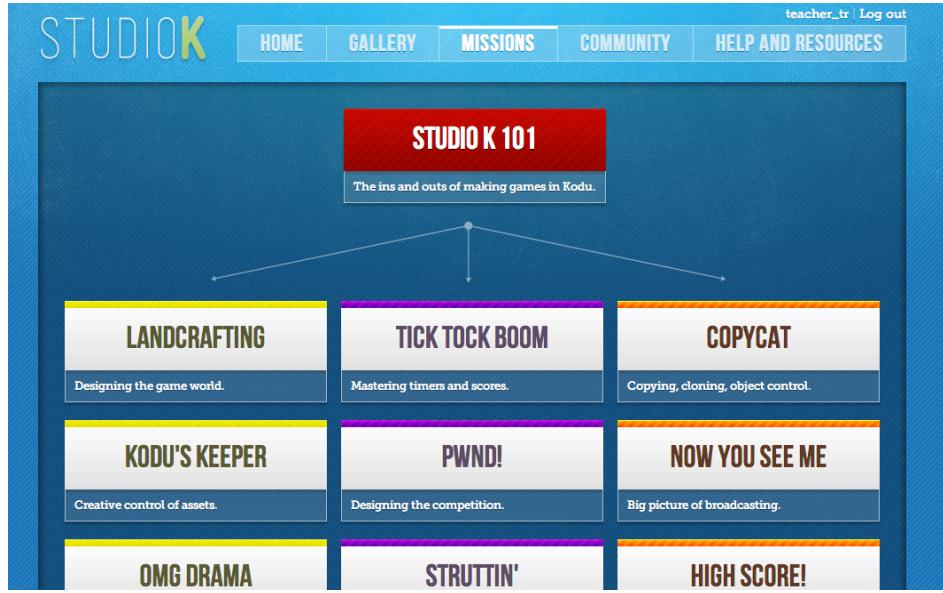


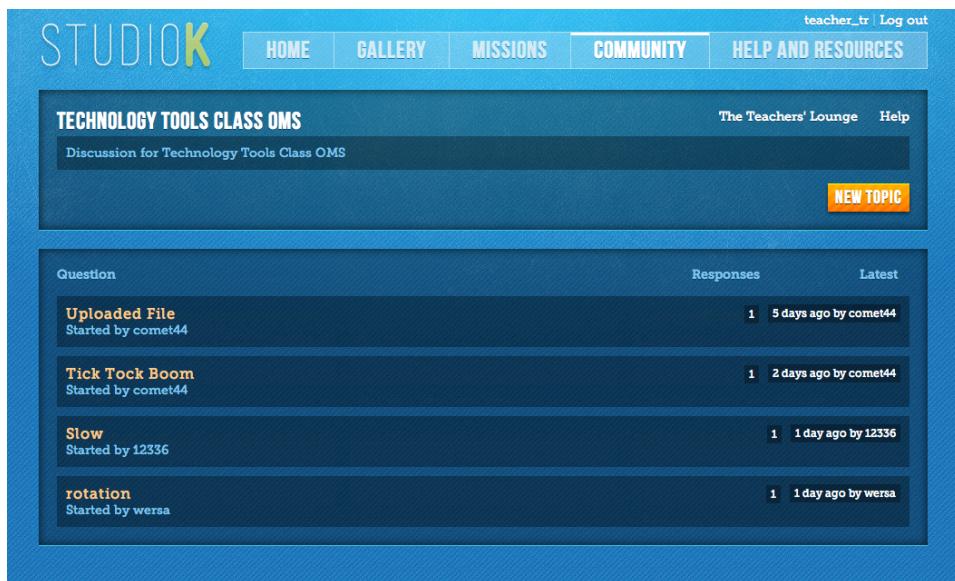
Figure 1.6 the Mission map

Clicking on one of the mission boxes brings the student to the individual mission content pages that include the associated challenges (see Figure 1.7). Students will be prompted with watch, play, create, and playtest challenges.

Figure 1.7 the Mission challenge page, describing the tasks students must complete in each mission.

Community page

The community is a forum that both students and facilitators can use to interact during the program (see Figure 1.8). Students can submit questions to be answered and facilitators are given the control to monitor and approve new forum topics and responses before they're visible to the entire classroom. This forum can be used as a space to get help or to start discussions around the content.



The screenshot shows the Studio K website's Community page. At the top, there is a navigation bar with links for HOME, GALLERY, MISSIONS, COMMUNITY (which is highlighted in blue), and HELP AND RESOURCES. On the far right of the top bar, it says "teacher_tr | Log out". Below the navigation bar, there is a dark header bar with the text "TECHNOLOGY TOOLS CLASS OMS" on the left and "The Teachers' Lounge Help" on the right. A "NEW TOPIC" button is located in the bottom right corner of this header. The main content area displays a list of forum topics in a table format. The columns are labeled "Question", "Responses", and "Latest". There are four topics listed:

Question	Responses	Latest
Uploaded File Started by comet44	1	5 days ago by comet44
Tick Tock Boom Started by comet44	1	2 days ago by comet44
Slow Started by 12336	1	1 day ago by 12336
rotation Started by wersa	1	1 day ago by wersa

Figure 1.8 the Community page, a forum space for students and facilitators to interact during the program.

Teacher's Lounge

As a forum page, the teacher's lounge is similar to the Community page, however only facilitators have access to this page (see Figure 1.9). The teacher's lounge is a forum that is open for all facilitators on the website to post discussion topics or questions. These will be seen and answered by other teachers using the program as well as the research and outreach teams for the Studio K project at the Games+Learning+Society Center at the University of Wisconsin-Madison.

Question	Responses	Latest
Scoring Systems Mission: Designing a System Started by wade	1	7 months ago by wade
Narrative Mission: Grand Visions in Narrative Started by wade	1	7 months ago by wade
Narrative Mission: Telling Stories in the Medium Started by wade	1	7 months ago by wade
Spaces Mission: Thematic Spaces Started by wade	1	7 months ago by wade
Assets Mission: Balancing Assets in World Started by wade	1	7 months ago by wade
Scoring Systems Mission: General Suggestions Started by wade	1	7 months ago by wade

Figure 1.9 the Teacher's Lounge, a forum space for program discussion with all facilitators using the program.

Help and Resources

The Help and Resources page is the go to page for guidance in learning how to use Kodu (see Figure 1.10). This page hosts tutorial videos explaining simple tasks like adding ground to more complex feats like using invisible objects to control world states. The Help and Resources page also hosts transcripts of all tutorial and content videos for Studio K.

The screenshot shows the 'HELP' section of the StudioK website. At the top, there's a navigation bar with links for 'HOME', 'GALLERY', 'MISSIONS', 'COMMUNITY', and 'HELP AND RESOURCES'. On the far right of the top bar, it says 'teacher_tr | Log out'. Below the navigation, a sub-menu for 'HELP' is shown with the sub-link 'Helpful information about using Studio K'. In the center, there's a 'NEW TOPIC' button. Below these, a 'Pages:' dropdown menu has '1' and '2' selected. A 'Latest' link is also present. The main content area lists six video tutorials, each with a title, a 'Started by wade' note, a timestamp ('23 days ago'), and a 'by wade' author link. The titles are: 'Video Tutorial: Camouflage and Uncamouflage', 'Video Tutorial: Changing Sky Color', 'Video Tutorial: Changing Size Scale State', 'Video Tutorial: Raise and Lower Water State', 'Video Tutorial: Changing Water Type State', and 'Video Tutorial - Linking Levels in Kodu'.

Question	Responses	Latest
Video Tutorial: Camouflage and Uncamouflage Started by wade	1	23 days ago by wade
Video Tutorial: Changing Sky Color Started by wade	1	23 days ago by wade
Video Tutorial: Changing Size Scale State Started by wade	1	23 days ago by wade
Video Tutorial: Raise and Lower Water State Started by wade	1	23 days ago by wade
Video Tutorial: Changing Water Type State Started by wade	1	23 days ago by wade
Video Tutorial - Linking Levels in Kodu Started by wade	1	23 days ago by wade

Figure 1.10 the Help and Resources page, hosting Kodu tutorial videos.

Researching Game Design

The primary goal for Studio K is to provide a meaningful learning experience around game design that will increase the interest that youth have in programming and computer science. One way to support this goal long term is to research the learning that occurs throughout the Studio K curriculum to argue for the increased use of similar programs in schools.

We seek to understand how participation in the community and with the curriculum facilitates the understanding and growth of game design and programming practices – expressed through curricular products and through use of Kodu. Research will also focus on how computational thinking practices are expressed and emerge through ongoing game design practices.

To meet these goals, researchers at the Games+Learning+Society Center at the University of Wisconsin-Madison are collecting data from the activity that occurs on the website, in the curriculum, and during design sessions in Microsoft Kodu. When registering for an account, users will be prompted with a consent form asking for their approval or disapproval for participating in the study. Research on the curriculum includes: two fifteen minute surveys on interest and knowledge of game design and programming, collection of curriculum materials, and event logs saved from design sessions using Kodu.

In order to better capture the learning that occurs throughout the curriculum, we ask that facilitators plan approximately fifteen minutes time at the beginning and end of the unit with Studio K so students can take the entry survey (Appendix A) and exit survey (Appendix B). We ask that the exit survey is completed at the conclusion of the unit, regardless of whether students have fully completed the curriculum.

For additional information about the research, please contact
support.studiok@learninggamesnetwork.org.

Technical Requirements

Use of Studio K requires an Internet connection and access to a PC computer. The curriculum was designed to support a one to one ratio of students to computers.

The specific requirements for Kodu and Studio K are listed below:

Operating Systems: Windows 7, Windows Vista, Windows XP (latest updates required)

Additional requirements: A graphics card that supports DirectX 9.0c and Shader Model 2.0 or higher .NET Framework 3.5 or higher XNA Framework 3.1 Redistributable Internet connection (compatible with any web browser).

Connections to Computational and Systems Thinking

In addition to examining games and game design, this curriculum supports the development of computational thinking skills. Computational thinking is argued to be a method of problem solving integral to success in computer science, STEM careers, with additional application in the soft sciences and humanities (NRC, 2009; NRC, 2011; Wing, 2006). In summary, computational thinking is a way of ‘problem solving, designing systems, and understanding human behavior that draws on concepts fundamental to computing’ (Kolodner, NRC, 2011). In this curriculum, we will divide the process of computational thinking into three constructs: debugging, logic, and feedback. The general curricular framework facilitates the development of computational thinking practices, while some missions will target the development of specific practices and thinking. The constructs are defined below. Their specific application to game design is detailed in the learning objectives for the given mission.

Debugging

Definition: Debugging is a process of identifying and solving errors during design.

Example: During design, a student tries to program his character to eat apples when it hits them. When the student plays the game after changing the code, she discovers the character throws the apples instead of eating them. After noticing that the game does not run the correct way, the student returns to the programming to see what her code says and fixes the problems leading to the incorrect behavior during play.

Logic

Definition: Using if-then-else logic structure to describe a potential chain of events, based on a given set of rules.

Example: Players in chess can move a piece around the board without releasing it from their hand. In doing so, they are examining the potential set of outcomes from that move.
Example: If I land on a property that is owned by another player in Monopoly, then I will have to pay them rent.

Feedback

Definition: Feedback is a sequence of player actions that apply individual or community experience during a design session.

Example: A student is making a race game with a set time to beat. After playing the game a few times, the student determines it is complete and asks a friend to play it. During play, the friend comments on how difficult the game is and then angrily quits

before winning. The student realizes that some components of the game need to be altered to make the game fun *and* challenging for its players.

STEM Standards Alignment

The Studio K curriculum not only helps students learn about coding and game development, but also meets many Common Core Standards both for Science and Technical Literacy and English Language Arts. Additionally, Studio K aligns with the Science and Engineering Practices set out by the Next Generation Science Standards that were developed by The National Research Council, the National Science Teachers Association, and the American Association for the Advancement of Science. They identify eight practices of science and engineering that are essential for all students to learn, several of which are met by Studio K.

Science and Technical Literacy Standards

CCSS.ELA-Literacy.RST.6-8.3 Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.

In Studio K, and computer programming in general, students will need to perform tasks in a specific order to achieve the wanted results. While programming their games, students will follow long procedures, including many specific and complicated steps in order to create games. For example, while creating a game, a student will need to create the game world, the objects within the world, and how these objects interact. Each object and interaction will need to be carefully programmed through procedures learned in the lessons in order to properly interact with one another and make the game playable.

CCSS.ELA-Literacy.RST.6-8.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context

Studio K provides educators with a list of technical vocabulary terms from computer science and integrates them into the lessons. While Kodu is only one of many programming languages, many of the terms are found across computer science and are valuable to know for anyone wishing to continue studying programming. Additionally, students will learn the vocabulary used by game makers and how they think about and talk about games and game mechanics.

CCSS.ELA-Literacy.RST.6-8.7 Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table).

In Studio K, students are challenged to design their games on paper before their implementation in Kodu. Students must learn how to express the relationships between characters, behaviors, and events within their game so that others can understand the processes. Additionally, students can be challenged to apply game design concepts and new computational thinking practices in other areas. For example, students can be tasked with designing a game that teaches a math concept or represents the themes or events within a novel. Game design

principles can be applied to various content areas in order to increase engagement.

CCSS.ELA-Literacy.RST.9-10.5 Analyze the structure of the relationships among concepts in a text, including relationships among key terms (e.g., *force*, *friction*, *reaction force*, *energy*).

In Studio K, students are often asked to think critically about the design components of a game and how they work together to provide a play experience, compare games on a general and specific level, and provide feedback/reviews with evidence to support their claims.

For example, when designing a game, a player must balance the relationships between the scoring mechanics, timers, and difficulty level of the game as well as the aesthetics and story of the game in order to create a pleasurable experience. When a student is creating a game, the students will learn how to think about these relationships, run tests of their design ideas, gather data (informally, and adjust their game based on those data. This learning process is also in line with engineering design (for example see: Massachusetts Science and Technology/Engineering Curriculum), one goal of which is for students to develop an iterative process that involved modeling and optimizing.

English Language Arts Standards

CCSS.ELA-Literacy.W.7.1 Write arguments to support claims with clear reasons and relevant evidence.

As a part of learning how to make games in Studio K, students will also be playing games and writing critical reviews. In these reviews, students will need to argue about the quality of the game and its use of game mechanics, providing examples from the game play in order to bolster their argument.

CCSS.ELA-Literacy.W.7.2 Write informative/explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content.

In Studio K this is found when students write Narratives for their games, including appropriate text bubbles, in-game messages, and overall game story arcs. Additionally, students will create introductions to their games, describing the game, its mechanics, and design goals.

CCSS.ELA-Literacy.W.7.3 Write narratives to develop real or imagined experiences or events using effective technique, relevant descriptive details, and well-structured event sequences.

In Studio K, this occurs when students use design elements to guide player experience. Students learn that narrative, if structured effectively, engages players in the gaming experience and facilitates motivation and determination to

complete the game. Rather than solely identifying a logical flow and interesting content, game designers additionally restructure and apply this narrative so that event sequences unfold naturally and logically.

CCSS.ELA-Literacy.W.7.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

This occurs in Studio K when students create Narratives for their games. Students will learn how to create narratives that support their game with writing and stories that will keep the players of their games interested and engaged. This will require students to think about the ways in which their writing will affect their players and the gameplay, making sure that the students keep their audience in mind when thinking about how to best represent their narrative through the video game medium.

CCSS.ELA-Literacy.W.7.5 With some guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on how well purpose and audience have been addressed.

Studio K provides pathways for students to both give and receive feedback on their games from both peers and adults. Many times, this feedback will be directed at specific elements of gameplay and mechanics and how well the student has integrated different parts of gameplay into their narrative. This feedback will give the students the opportunity to create new iterations of their game based on the feedback, from changing one mechanic, to completely reworking a mechanic system.

Next Generation Science Standards

1. Asking questions and defining problems

In Studio K, students will determine why their game is not working through debugging. They will need to recognize a problem, define what part is not working, and ask themselves how to best solve the issue.

2. Developing and using models

Studio K challenges students to draw models of their game before they implement them within Kodu. Their drawings will help the students realize parts of the game system they are trying to make and identify the ways in which the different parts of the system (moving, score systems, win states) will work together to create a fun and challenging game.

5. Using mathematics and computational thinking

Students in Studio K will engage in computational thinking. They will create strategies for thinking about the data they receive about their games, create sequences of steps to solve problems, and develop new systems within their games.

6. Constructing explanations and designing solutions

Through Studio K, students will learn how to solve problems in their games through designing solutions. This process will include identifying a problem, and then generating, testing and improving solutions. They will learn how to optimize game play through feedback, debugging, and creating multiple iterations of their games.

7. Engaging in argument from evidence

As a part of Studio K, students will be asked to respectfully critique each other's games and provide evidence or examples of ways in which the game could be made better. The students will learn how to receive feedback as well as how to respond, be it by agreeing with the critique and adjust aspects of their game, or by creating a counter argument for why their game works in its current iteration.

Studio K 101

General Learning Goal

This unit is designed to introduce the basic knowledge required to make a simple game in Kodu.

Mission - Joining the Studio

Entry Survey

Students will first be asked to complete an entry survey so that we can capture their interests and knowledge relating to game designing and programming (see Appendix A). This will not capture any personal information about the students.

Mission - Introduction – Game Design 101

Learning Goal

This mission highlights the goals of the curriculum and introduces the concept that games are made up of components.

Kodu Learning Objectives

- None

Relevant Help & Resources Video Tutorials

- Resources page: Game Genres

Watch – Introduction Video

Video Content

You've probably played a lot of games in your life-- anything from soccer to hopscotch, from Monopoly to Mario. Games come in a lot of different forms and can be a lot of fun to play. While it is easy to only think of games as fun, making a game is often very complicated. We want to start thinking about games as a bunch of parts, or components that work together and follow rules. Together, these parts or components make up a system that creates an experience for the players. Game designers need to think about the goals and rules of their game, the story or narrative, and finally how space and assets all fit together in order to create a fun and engaging experience for players. Game development can come in a lot of different forms, from a single independent game designer making a game all by himself to a giant team with members who focus specifically on programming, design, or art. In Studio K, you'll get to take on all these roles and jump right into creating games all on your own.

Facilitator Resources

This mission is primarily intended to provide a starting measure of students' interests and abilities for researchers at the Games+Learning+Society Center at the University of Wisconsin-Madison. More importantly, it provides students with some context for the work they will be completing throughout the curriculum. As a facilitator, you can encourage your students to accurately and seriously answer the questions in the survey. Their answers are important to discovering how youth understand and acquire content during their experience with Studio K.

After finishing the survey, students will watch a short video that briefs them on the content in the curriculum. If the class is held in person, this is a good time to start a discussion about favorite games and the reasoning behind that decision. If this class is online, this discussion can be held online in the Studio K forum. The curriculum encourages students to think more critically about games and the components that make them. Establishing a conversation around games, especially games that the students already play, will help students get in the mindset of thinking of games as more than just consumable products that are "fun".

During your discussions around games, some students may talk about games that are beyond their maturity level. If this occurs, it is helpful to talk about the components that make the game unique rather than focusing on the mature content of the game. For example, students may bring up Resident Evil, Grand Theft Auto, Call of Duty, Halo, or other games that they like to play because they can "beat people up" or "use cool guns", etc. If you, as the facilitator, are not comfortable talking about those topics or themes, discussions can instead focus on things like the design of the game world. Those games often allow for more player choice in what to do and where to go, rather than forcing a linear progression through the game. Talk about how the game world defines the player experience or how it can affect the player's emotions – like increasing the apprehension or fear in a level.

Discussion/Forum Questions:

- What constitutes a game? Why?
- What do you need to make a good game? Why?
- What is your favorite game? Why?

Mission - Setting the Scene

Learning Goal

This mission introduces the Kodu tools required to alter the landscape in a Kodu world. Students should begin to think about how they should design land with a focused thematic goal.

Kodu Learning Objectives

- Accessing the *Kodu menu* and *Help menu*
- Altering the ground with the *Ground Brush*
- Raising and lowering the ground with the *Up/Down* menu item
- Evening out the height of the ground with the *Flatten* menu item
- Creating uneven ground with the *Roughen* menu item

- Adding and personalizing water with the *Water* menu item

Relevant Help & Resources Video Tutorials

- Video Tutorial: Raising the Ground in Kodu
- Video Tutorial: Changing World Settings
- Video Tutorial: How to Add Water to the World

Watch – Setting the Scene

Video Content

What's the first thing you see in a game? The world. Setting the scene for your characters is an important step in making a good game. For this first challenge, make a tropical land for your world. Don't add objects yet-- just focus on the base land design. Here are some tools to help you do this. When you open a new world in Kodu you'll see a menu at the bottom of the screen. Just scroll through. Use the Ground Brush to add, delete, and change the color of the ground. Use the Up/Down menu item to raise and lower the ground. Use the Flatten item to even out the ground you're working on. The Roughen menu item does the opposite, making the ground more uneven. Finally, don't forget to select the color of and add water with the Water menu item. Just remember, with any of these menu items, changing the brush size or playing with other options can get you that special look you're thinking of. If you forget how to use a menu item, go to the upper left side of Kodu for a help menu. Just as a recap, your challenge is to make a tropical land for your game. Visit the help page for extra info on this challenge. Happy designing!

Create – Setting the Scene

Challenge

Make a tropical land for your game. Don't add any objects yet, just focus on designing the land. Make sure to describe your game when you upload it!

Facilitator Resources

This mission is designed to get students to explore and become comfortable with the land editing tools in Kodu. The tropical world theme provides students with a greater range of freedom in their designs while still promising interaction with a wide range of tools and features in Kodu. In this challenge, there are no right or wrong designs. If students feel like the challenge is too open-ended, prompt them to talk about how they imagine a tropical scene – talking through design problems aloud can help create a concrete plan for design.

As students are designing, feel free to ask them to describe the theme of the world or why they made design decisions. Being able to verbally describe their thoughts will help them to become more precise and critical in game reviews. This activity will also allow discrepancies between how they see their world and how the player sees the world to become illuminated. Throughout the whole program, facilitators and students should feel comfortable interacting and talking critically about the games being created. With this in mind, feel free to make suggestions or talk about how you perceive the games they're creating.

In this challenge in particular, encourage students to explore the settings in Kodu. For this challenge, some of the more relevant settings may be changing the color of the ground, water, and sky or altering the lighting of the world. It's easier to figure out how to use the basic tool features on the menu in Kodu by trial and error but often students forget to look through the help menu in the upper left hand of the screen to learn how to change the settings. The menu details how to complete all the actions available with each menu tool and the controls to do so. This help menu is extremely helpful, especially for a Kodu beginner, so remind students to check that menu for extra information if they're stuck in this challenge.

Discussion/Forum Questions:

- Why did you decide to design your world the way you did?
- How do you want players to feel when they're exploring your world? What specifically in your world will help players to feel that way?
- How would you design your tropical world if you wanted your players to feel relaxed? How would you design your tropical world if you wanted your players to feel apprehensive?

Mission - Getting the Guy

Learning Goal

This mission introduces the tools and features associated with adding objects to a world in Kodu. Students should begin to think about how objects can affect the theme of a world.

Kodu Learning Objectives

- Adding an object to a world in Kodu
- Changing the size, color, and location of an object in Kodu

Relevant Help & Resources Video Tutorials

- Video Tutorial: Understanding the Resource Meter
- Video Tutorial: Adding and Removing Objects from the World

Watch – Getting the Guy

Video Content

You've made your tropical world and now you have to fill it with objects in order to complete the scene. For this challenge, add objects to your world that fit the tropical theme. When adding a character to the world, use the Object Tool in the menu. Just move the cursor to where you want to add the object in the world and then click with your mouse or press A on your gamepad to select the object that you want. Change the size and color of your objects to make them better fit in your world. Visit the help page for extra info on this challenge. Happy designing!

Create – Getting the Guy

Challenge

Add objects to your tropical land to complete the tropical theme. Make sure to describe your game when you upload it!

Facilitator Resources

This mission is designed to get students comfortable with the object adding and editing tools in Kodu. Throughout the curriculum, students will be encouraged to think about using object more critically and purposefully in their games. In Kodu, there is an object thermometer measuring how “full” of objects a world is. When the thermometer is full it means there are too many objects in the world and the world is in danger of crashing Kodu or running poorly. You should encourage your students to choose objects carefully, as each should be meaningful for the player experience of the game.

Discussion/Forum Questions:

- What objects did you choose? Why?
- Did you specifically change the settings of any objects to change the feel or theme of your tropical world? If yes, what and why?
- How could changing the color or size of the objects in your tropical world affect the player experience?

Mission - Making Moves

Learning Goal

This mission introduces the concept of programming in Kodu. Students will be shown how to find the programming window and the basic structure of Koding. They'll learn how to program automatic and player controlled actions.

Kodu Learning Objectives

- How to find the programming window on an object
- General structure and how-to of coding in Kodu
- Programming objects to move with use of keyboard or gamepad
- Programming objects to move automatically
- Programming objects to move away from other objects

Relevant Help & Resources Video Tutorials

- Video Tutorial: Opening and Programming an object

Watch – Making Moves

Video Content

You have your world, but you need your characters to move in it. For this challenge, add a player controlled character (PC) and at least one non-player character (NPC) to your game. You should be able to move the PC with the keyboard or the gamepad and the NPC should run away from you. To get started, if you want to program an object then you go to the Object Tool, then mouse over the object, and go to the programming page for it. Just follow the help menu in the upper left corner if you get stuck. Once in the programming

page, you can see that a blank line says “when blank, do blank.” Every time you want an object to do something, then you need to program it to do so. And you need that action to be triggered by something either within the game-- for example, seeing or bumping into something-- or from the other player, using either the keyboard or the gamepad. I’m going to program this turtle to move with the use of the gamepad. I’ll select a gamepad in the When section of the line, but since there is more than one button on the gamepad I’m going to more specifically select the L Stick for movement. Then I’ll go over to the Do section and say what the object will do when the L Stick is used, adding Move and then Quickly to the Do section. Right now, this line of Kode says “When the left stick is used, then the turtle will move quickly.” You can test out your programming in Play to make sure that it works the way you want. Just as a recap, add a player-controlled character and at least one non-player character to your game for this challenge. Program the PC to move and the NPC to move automatically away from the PC. Go to the help page for more info if you need. Happy designing!

Create – Making Moves

Challenge

Add a player controlled character (PC) and at least one non-player character (NPC) to your game. You should be able to move the PC with the keyboard or gamepad and the NPC should run away from you. Make sure to describe your game when you upload it!

Facilitator Resources

For this mission, students will learn how to program their characters to move – both with the use of a gamepad or keyboard as well as automatically.

If students are finding it challenging to write their own Kode for this challenge, encourage them to watch the tutorial video “Opening and Programming an object in Kodu”. Otherwise, talking through the programming with the students can often help clarify what is needed to be Koded. For example:

- “What do you want the character/object to do?”
Students may confuse the Do and When sections in a line of Kode. Asking them this helps them to verbalize the end goal of their line of Kode – whatever action they want the object to perform.
- “Do you want this to be controlled by the player or happen automatically?”
This question will help structure the type of ‘When’ statement required in the line of Kode. If the student wants the player to control the actions, then the programming needs to include either the keyboard or gamepad.
- “Do you want the action to happen all the time or only sometimes?”
If the student wants the action to happen automatically, the next question should prompt whether the action occurs constantly or conditionally. If the action should occur continually, then the ‘Always’ tile should be used. Otherwise, prompt the student with, “what do you want to cause the action?”

The assessment for this is more concrete than other create challenges – with the following programming answers accurately capturing the desired game products in Kodu. First, play the game in order to see if the characters seem to be moving as they should. Then in order to check for accuracy, open the programming windows on the player controlled (PC) character and the non-player character (NPC) to look through the Kode.

PC

When: Gamepad Lstick → Do: Move
 When: Keyboard arrows → Do: Move

The Kode for the When section can also include other parts of both the keyboard and the gamepad (e.g. Rstick or WASD) as long as it is playable. If the game is uncomfortable to play, e.g. using keys 'W', 'P', 'X', and 'M' instead of the arrow keys or WASD for movement, then you should alert the designer. Having a discussion about the player's experiences is helpful for becoming a more aware designer.

The Do section of Kode can include more specific movement controls, e.g. 'Do: Move Quickly' or 'Do: Move Slowly'.

NPC

When: Always → Do: Move
 When: See 'X Character' → Do: Move Away

The movement Kode in the Do section can be more specific. In the NPC line one, the NPC can move 'wander', 'quickly', and 'slowly'. For the NPC line two, the movement needs to be 'Away' but can also include 'quickly' or 'slowly'.

Mission - In It to Win It

Learning Goal

This mission introduces the concept of a win condition and teaches the programming to Kode one in Kodu.

Kodu Learning Objectives

- Programming a win condition

Relevant Help & Resources Video Tutorials

- Koding a Win Condition
- Koding a Lose Condition

Watch – In it to Win it

Video Content

So you have your world, your objects, and your characters – so let's make it a working game. In tag, if you're 'it' you have to tag the other person. For this challenge, Program your game to have a win state that's visible to the player. It's important to do this so that your players know what's going on in the game. While you may think that it's obvious how you lose or win in a game that you make, feedback is still important for the player in case they don't see it as clearly as you do. For more info, go to the help page."

Create – In it to Win it

Challenge

Program your game to have a win state visible to the players. Make sure to describe your game when you upload it!

Facilitator Resources

For this mission, students will learn how to program a win condition in the game. For this create challenge, the correct Kode is simple:

When: Bump 'X object' → Do: Win

The created game is very simple – with the player only needing to bump into the NPC in order to win. Consequently, the game itself is not engaging. If students finish early in the Studio K 101 unit, you can encourage them to add more complexity to the game. Can they make it more difficult for the player to win? Can they make a condition in which the player loses? In a real game of tag, after tagging a player then he/she becomes "it" – can they program this feature in the game as well? By encouraging varying levels of complexity in the game, you can support students who may be speeding through the curriculum and also those that may need more help throughout.

MECHANICS

General Learning Goal

This unit is designed to introduce concepts of Mechanics, or the rules (both applied to the world, the player, and other objects in the world) that are intended to produce a game or gameplay. Throughout, students will learn about goals, rules, broadcasting state changes, and scoring systems in games. Each mission in this unit will focus on a different component - encouraging students to creatively apply concepts in their own game creations. With the conclusion, students will have a basic understanding of use of mechanics in games - a base they'll build on to create increasingly more complex games.

Mission - Copycat: Copying, cloning, object control

Learning Goal

This mission is designed to introduce students to the concept that games are a system of components that work together to affect player experience. Students will learn about goals, rules, and object manipulation. Focus will be on a more complex use of objects in Kodu through the use of cloning and creatables, which can be applied to create more complex features in Kodu: including 'teleportation'.

Kodu Learning Objectives

- Copying objects
- Use of Creatables
- 'Teleportation' in Kodu

Relevant Help & Resources Video Tutorials

- Resources page: Goals in Games
- Video Tutorial: Adding and Removing Objects from the World
- Video Tutorial: Creatable Objects
- Video Tutorial: Koding Multiple Scores and Timers
- Video Tutorial: Page Switching and Inline Switching

Watch – Introduction to Games

Video Content

Games are made up of components that have rules for behavior and interaction with each other. Every system has some Goal, or something that it is trying to complete. In games, the Goal is the win condition, or how the player wins the game. In chess, for example, the goal of the game is to capture the opponent's king, which is done by moving pieces around the board. In basketball or soccer, the goal of the game is to score more points than the opposing team. Two common goals are spatial goals (or when a win condition depends on the player's location) and task-oriented goal (or when the win condition depends on the player completing a task to win), while there are other types of goals as well. When creating a game, the designer must think about the overall goal of the

game and how players will interact with that goal. Different goals will require a different play style in order to complete it. Start to think about goals and how they can change a gaming experience. The goal of the game is mediated by rules that restrict player action during gameplay. Rules are what determine how an agent may behave in a given system. In games, Rules work the same way, except they determine how a player may behave and interact with the game world. Rules can generally be defined by: 1. Behavior for which the player will be punished. For example, in basketball you cannot walk or run without dribbling the ball. This is called traveling, and the other team will get the ball. 2. Rules that restrict behavior. For example, in chess, the player cannot move a pawn sideways or backwards. There is no punishment, as those are not valid moves; the Rules simply define the set of actions available to the players. It's important to design games with both the goal and the rules of the game in mind so that the player can have an engaging experience – this mission will help you to explore how to do this.

Watch – Cloning and Creatables Tutorial

Video Content

Oftentimes when making games, you will want to create the same object, with the same properties, multiple times. Sometimes, you will want these objects to all be there when a game starts: in Kodu, we can achieve this through cloning.

To clone an object, simply hover over the object you wish to duplicate while in programming mode, and then hit the right trigger. Other times, you may want a duplicated object to appear while the game is running. You can see an example in the apple picking game, as apples that all react to being bumped by a Kodu pop out of the tree one after another. We achieve this by using creatables. To make an object creatable, we simply change its settings as shown. You can see that this object now appears differently in the editing mode. In play, this object does not appear as placed in edit. However, we can program other objects to now create instances of the creatable as desired: you can observe how the tree was programmed to do so in the game we just watched.

Create – Cloning and Creatables

Challenge

Make a level in which at least one type of object is automatically generated. No need to make a game or to include a win condition - just practice your new programming skills. If you get stuck, look back at the Objection Creation video for help. Make sure to describe your game when you submit it!

Watch – Teleportation Tutorial

Video Content

Think about the cloning and creatables challenge and videos. with those, you learned how an object can be used to expand the design of a game. While the concept may be basic, using Creatables in kodu opens the door to many more complex designs for games. For example, in this game, the player is able to select the player-controlled character before starting the level. After the player selects the character he or she wants, the character is “teleported” to the start of

the level. That teleported character is actually a Creatable, which is created by an invisible rock on the main piece of land. The creatable object holds a player character code. While the originally bumped object serves only as a trigger to tell the game which creatable-- the red or the blue Kodu-- should be created and when. Looking in the Kode, you'll see that when bumped, the trigger Kodus will change a score based on their color. Meanwhile, an invisible rock on the mainland watches for that score and creates an appropriately colored player character. For the next challenge, build a game that uses a teleportation system. If you get stuck, visit the help page for a hint.

Create – Teleportation

Challenge

Create a game that includes teleportation. If you get stuck, look back at the other videos in Object Creation or visit the help page for hints.

Playtest – Review

Challenge

Play the game and leave a review. Make sure to focus on how the copying, cloning, and object control features are used in the game.

Facilitator Resources

Introduction to Games

This mission will be the first foray into the more technical side of game making. Before delving in the technical side, students must first understand how games are a system of components that all work together to create the play experience. The Introduction to Games video helps orient students to these concepts, while teaching them some basic concepts what goals and rules are in games.

If meeting in-person, one activity that encourages a richer understanding of goals and rules is to have students work together to create a game that can be played in the classroom. For example, you can start out asking for a student to create the goal of the game – tag another player, collect X number of items, etc. You can then set the first rule of the game – players can only walk, players must not leave a certain space, etc. Then, play the game. Once the goal is met, talk with your students about whether it was fun, engaging, challenging. Repeat the cycle of adding rules, playing, and discussing the play experience. This activity can help students understand how goals and rules work interact, and how rules work together within a game to affect play.

Cloning and Creatables

The Cloning and Creatables section of the mission is designed to give students greater mastery over use of objects in Kodu – specifically by teaching them some less obvious techniques, like using creatables and cloning. An object that is set as ‘creatable’ can be infinitely generated by other objects within a game. Cloning is a feature in Kodu that allows the user to copy an object, including the programming, and create infinite copies. These two features can increase efficiency and the complexity of games. For cloning, a common use is to quickly

and easily make a swarm of identically behaving objects. For creatables, the difficulty of the game is often increased by adding automatically generated enemies.

For the create challenge, students' submissions should include two objects in the game. During play, you should notice that there is an object generating another type of object infinitely. The creatable object will be set as "creatable" in the settings menu. The object generating the creatable should be programmed similarly to 'When: Always → Do: Create 'Object X''. Variation on this could include creating the object after a state change or an action.

Teleportation

By applying the object manipulation skills learned earlier in the mission, students can effectively design a teleportation mechanic in their games. This can be appealing for creating games with portals (e.g. the hidden passages in Mario games or concrete mechanics like in Portal and Portal 2) or for adding complexity in user choices (e.g. selecting a character or choosing a world) as well as other applications.

In order to accurately apply the mechanic in Kodu, the teleported object needs to be a creatable object. Behind the scenes, the object will be programmed to disappear after either bumping a certain object (e.g. a hut or a pipe) or being on a certain land type. Then either another object or an invisible object located in the desired teleportation output will generate the creatable object. During gameplay, the object seems to be transported from one location to another.

Discussion/Forum Questions:

- How can cloning be used creatively in a game?
- How can creatables be used creatively in a game?
- How does 'teleportation' work in Kodu?
- How can 'teleportation' be used creatively in a game?

Mission - Now You See Me: Big picture of broadcasting

Learning Goal

This mission is designed to teach state broadcasting in Kodu and how it can be used in games.

Kodu Learning Objectives

- Programming broadcasted states (including audio and visual)
- Designing meaningful reactions to broadcasted state changes

Relevant Help & Resources Video Tutorials

- Video Tutorial: Page Switching and Inline Switching
- Video Tutorial: Koding Characters to Express Emotion
- Video Tutorial: Koding Size State Changes
- Video Tutorial: Koding Sky State Changes
- Video Tutorial: Koding Water Type State Changes
- Video Tutorial: Koding Water Height State Changes

Play – Stay out of my Yard!

Challenge

Play through the Angry Neighbor. After you play through, explain when the player should try to get the ball and how that system works in programming.

Watch – Broadcasting Tutorial

Video Content

In the Game 'Stay out of my Yard', the goal is to reach the ball that has been left in the angry neighbor's yard. The game is more complicated because the player is restricted by the emotions of the neighbor. When he is happy and green, the player can safely move in the line of sight of the neighbor. When the neighbor's mood switches and he turns red, then the player is stunned and loses when in the neighbor's line of sight. This concept is called broadcasting – having an object's state control conditions in the game. In 'Stay out of my Yard', the whole game revolves around the broadcasted color and emotion of the neighbor. This concept can be applied in other ways as well – with objects 'paying attention' to states in the world or broadcasting their own states.

Create – Option #1 Moving Statues

Challenge

Use broadcasting in a level. Make a "Moving Statues" Game in which players react to the visual changes of another object. If you get stuck, visit the help page for hints. Make sure to describe your game when you submit it!

Create – Option #2 Musical Chairs

Challenge

Use broadcasting in a level. Make a "Musical chairs" game in which players react to the sound changes of another object. If you get stuck, visit the help page for hints. Make sure to describe your game when you submit it!

Playtest – Review

Challenge

Play the game and leave a review. Make sure to focus on how the game uses broadcasting.

Facilitator Resources

Broadcasting is a more complex play mechanic but a useful one for creating games in Kodu. In this mission, students will learn the ins and outs of using broadcasting – being prompted with creating a game with either audio or visual broadcasted states.

In order for a successful application of this concept, students will need to have two components in their games: 1. an object that is broadcasting a state (e.g. emotion, sound, etc) and, 2. an object that reacts to the change in that state. Since this is a more technical challenge, encourage your students to browse the Help and Resource section of the website to get helpful video tutorials on Kodu techniques.

Discussion/Forum Questions:

- Explain how broadcasting works in Kodu. How can broadcasting be used in Kodu to make games more engaging?
- What are some examples of broadcasted states in commercial games?
- How does broadcasting interact with dramatic elements in games?

Mission - High Score!: Engineering scoring systems

Learning Goal

This mission is designed to teach students what scoring systems are and how they can be creatively applied in games. The mission also helps students better understand the application of simple scoring systems commonly used in Kodu games.

Kodu Learning Objectives

- Programming timer and score variables
- Using visible and invisible scores
- Using multiple timers or scores in a single game
- Using scores to represent other variables

Relevant Help & Resources Video Tutorials

- Video Tutorial: Invisible Timers
- Video Tutorial: Coding Multiple Scores and Timers
- Video Tutorial: Using Scores to Represent non-Score Variables

Play – Pong

Challenge

Play through Pong. Pay attention to how the scoring system works in the game. After playing, explain when and how scores are used in play.

Play – Survival Game

Challenge

Play through the survival game. Pay attention to how the scoring system works in the game. After playing, explain when and how scores are used in the game.

Play – Health in a Shooter

Challenge

Play through Robot Turtle Shooter. Pay attention to how the scoring system works in the game. After playing, explain when and how scores are used in play.

Playtest – Kodu TD

Challenge

Play through Tower Defense. Pay attention to how the scoring system works in the game. After playing, explain when and how scores are used in play.

Watch – Scores Overview

Video Content

Scoring systems are one form of representing a player's performance in the game, and are usually represented as points or some sort of currency like money, cash, gold, etc. Scores are present in a lot of games, but not all games. They provide a means of engagement for players, motivating them to get the most money, complete challenges in the fastest time, explore all of a world, or be the best player they can be. The scoring system can be used for many purposes within a game as well. Sometimes scores are a part of the goal of the game, like Angry Birds or Tetris. In those games, the goal of the game is to get the highest score. In other games, scoring is not a goal, but acts with rules of the game. In Mario, there are multiple scores -- one score from the gold coins collected in each individual level. Mario has a spatial goal -- to get to the end of the level. The points are to give the player an extra life, which may help achieve the goal but is more related to the rules of the game. In other games, the score represents an amount of money the player has made in the game and can be applied to buy better play mechanics or change the rules of the game, like buying rocket boots which make the player fly. Other games have achievement systems rather than points. There are achievements in Steam and on XBOX Live or Trophies on Playstation Network. This scoring system is intended to motivate players to spend more time in the games and to experience new play styles. When designing your games, keep in mind how and why a scoring system would be used and what that player will experience with the system.

Create – Hungry Hungry Hippos

Challenge

Make a game that uses scores in at least 2 different ways. Make sure to describe your game when you submit it!

Facilitator Resources

Scoring systems are an important part of games – affecting player motivation and engagement, as well as the general flow of the game. This less flashy component of games can often be overlooked, but crafting a well working scoring system can be challenging. In this mission, students will learn the importance of scoring systems in games and gain experience with several Kodu specific applications. In Kodu, Scoring systems or scores can be used in many ways to give players feedback and control the states in the world. The point systems can act as currency, timers, and conventional scores - all either visible or hidden to the player.

It's important to encourage your students to find a scoring system that matches the play mechanics, goals, and rules of the game. The system used affects the player experience – for example, timers can make the player feel more tense or rushed. In an exploration game, a timer may create the wrong play experience. Rather, racing games more commonly feature timers as that tension is often the target emotion for players. The most successful game designs will originate when the score mechanics are created with the player in mind.

During this time, you could start a discussion about the scoring systems used in various popular games and how they affect the player. How could those systems be adapted to fit the Kodu environment?

For the create challenge, encourage your students to get creative with their use of scores and timers. Try to get students to move away from simple responses like having counters for collection of two different objects. Students can incorporate invisible scores and timers in their games for this challenge as well. When talking to your students, it may help to draw from other commercial games as a way to think outside the box.

Discussion/Forum Questions:

- Think of your favorite game - describe the scoring system and how it affects your experience during play?
 - What would be the effect of changing the score system to X?
- What are the benefits of having multiple scoring or timing systems in a game?
- Why are achievements used in games?

AESTHETICS

General Learning Goal

This unit is designed to encourage students to think more critically about object use, designing land, and incorporating narrative and dramatic elements in games. The missions will encourage students to start making their games with intentional aesthetic design choices. In Kodu, land design is often the faster tool set to acquire – so the following missions challenge students to rethink how they use the tools in their games.

Mission - Landcrafting: Designing the game world

Learning Goal

This mission is designed to introduce students to concepts around the space, not including objects, in games.

Kodu Learning Objectives

- None

Relevant Help & Resources Video Tutorials

- None

Introduction – Aesthetics

Video Content

The aesthetic, or visual, design of a game is very important for an engaging experience in a game. While different genres of game focus more or less heavily on the visual design, all have very careful consideration of the assets and the space used in a game. The narrative of a game also becomes important for many types of games as well. In order to become a good game designer, you have to understand how assets, space, and narrative can be used to help shape a player's experience and engagement in a game. This mission will introduce some important concepts for understanding the player experience related to aesthetics.

Play – Water World

Challenge

Explore Water World 1. Think about how the stage is designed. After exploring, explain how you felt as a player while exploring that world. Would you have changed any design of the space?

Watch – Land Design Intro

Video Content

Spaces are where the game takes place. They are carefully designed to create a certain player experience, like feeling tension or awe. For example, the horror game Slender takes place in a dark forest and abandoned buildings at night. The only light for the player is from a flashlight, so the darkness creeping into the center of the screen increases the tension and feeling of dread, which then contributes to the game's scariness. In Slender, the player may have to play more cautiously since sight is limited in the dark environment. In more open games like Shadow of the Colossus, the player is able to quickly move through the world since sight is not limited in such an open environment. In Kodu, specifically, you can use types of land and design to help structure the experience of the player and help reinforce the story you're telling in the game or help reinforce the playing style required to play a game. Space is also reinforced by the genre that you're expecting to play. For example, platform games like Mario or Sonic require the player-controlled character to jump to and from suspended platforms and on and over obstacles. Because of the genre, the design of the land requires platforms and other obstacles for the characters to jump over. By carefully crafting your world, you can help your players have the desired gameplay experience.

Facilitator Resources

Since students will already know how to use the land brushes in Kodu, this mission encourages critical thinking about space. Students will explore an underwater world and are asked to reflect on their experience in the game, the changes they would make if they had been designing. This is a great time to start a conversation about the use of space in Water World and more generally in games.

During the discussion about spaces, some students may talk about games like Spore, Zoo Tycoon, Sim City, and other similar games. These games are sandbox games, which the player is almost never restricted, in terms of where they can go and what they can do. The content of the game may change, but the player freedom stays constant across all sandbox games. This would be a good opportunity to ask about how the game would change if the space was more confined. Similarly, you could ask how games with restricted spaces, like Farmville, would change if you made the space much larger.

In both of these cases, what features of the game would the designers have to change? Could they even call it the same game, or would the play experience be so different that they would have to call it something else?

Discussion/Forum Questions:

- How would Sim City, or another game with a very open space, change if you restricted the space? How would that affect the player experience?
- How does the space change the way you play a game?
- What is the space like in your favorite game? How does the space affect your play experience?

Mission - Kodu's Keeper: Creative control of assets

Learning Goal

This mission is designed to encourage students to think more critically about use of objects, or assets, in games.

Kodu Learning Objectives

- None

Relevant Help & Resources Video Tutorials

- None

Play – Water World 2.0

Challenge

Explore Water World 2. Think about the use of assets in the world. After exploring, explain how assets were used creatively in the world. What would you have done differently with the use of assets?

Watch – Creative Assets

Video Content

Assets are all of the objects that a player sees in the game. They are the trees, the buildings, the sky, the power-ups, the non-player characters, the weapons--all the things that you can see in the game. The design of the assets gives the game atmosphere and personality. Think about what makes the worlds of Mario and Donkey Kong Country. They're both platformers and the general plots are both very similar with spatial goals, but they take place in very different worlds. This is largely due to the assets. Mario collects coins, activates special blocks, and eats mushrooms. Donkey Kong collects bananas and gold statues, breaks barrels and crates, and opens treasure chests. He fights rats, spiders, vultures, and other jungle creatures. He has jungle animal friends who help him get through the world. The assets from both games provide the same benefits for the player. For example, coins and bananas both are used to collect points. In this situation, the assets change the player experience by altering the atmosphere and personality of the game. As you are designing, think about what kind of atmosphere you would want and what assets would well represent that atmosphere. You also want to think about what information is being provided by the game world. In Kodu, specifically, there are the set assets that can be used creatively in space design. Narrative can help your designs reach their full potential by explaining and justifying the assets used in the world.

Facilitator Resources

Assets are very important and can often be overlooked for their utility. It is easy for students to design a level and put objects in their world, but giving those objects utility, like awarding an extra life or giving points, can be more challenging. After this unit, students should be more aware of their design choices regarding assets. In previous challenges, they might have had a world of empty assets, but they should begin to use their assets for additional purposes.

You can check if an asset is used either by observing interactions while playing or by looking through the Kode of objects. If it is used, the Kode will either be on the object that is being used or on the asset that interacts with that object.

Once again, this mission is a great place to start a discussion – especially comparing games and the assets used within to alter player experience. You can start by talking about Water World 2 and the differences, especially for the player, between the two worlds. What changed? What was the effect? Would other design decisions been more successful? How could these worlds be altered to change the emotion the player feels? After covering attitudes on Water World 2, open the discussion for a broader conversation on the assets in games.

Discussion/Forum Questions:

- How do different assets change the way you play a game?
- What are some assets in your favorite game? How do these assets affect *your* play experience?
- How do assets and space interact to affect player experience

Mission - omg DRAMA: Narrative and dramatic elements

Learning Goal

This mission introduces the concepts of narrative and dramatic elements. Students will be expected to think critically about the application of these components in games.

Kodu Learning Objectives

- None

Relevant Help & Resources Video Tutorials

- None

Play – Water World 3.0

Challenge

Explore Water World 3. When you're done, explain how narrative is used in the game. What changes would you make to narrative make the game more engaging for players?

Watch – Storytelling in Games

Video Content

Narrative is the story of the game and provides the structure for the events of the game. In games, building a narrative requires that the designer considers not only the story but how and why events unfold. Narrative appears in a few ways in video games: through action sequences or short videos, through text boxes or speech bubbles. Unlike a book or a movie, the designer needs to carefully space how and when a player receives information contributing to the narrative. For example, it would not be a fun game if the player had to read many really long

text boxes or had to watch 20 minutes of action sequences during play. For Kodu, much of the narrative will appear in text boxes and character actions (like characters changing their emotions or making sounds). Narrative is not always vital to the player's experience, like Pacman, for example. In Pacman, each of the ghosts has a name and a background, but that information is not important to playing the game. Other games, like Zelda, have developed storylines that are important to understanding how to play the game. When designing the narrative, it's also important to think about the base story and consider whether or not it will be fun for players. A good narrative will engage players in the game, playing for longer and enjoying the experience more. Elements of a story are present within a narrative in a game, but appear with different sequencing compared to a book or movie. During your design, think first about the story you want to tell and then how it can be altered to better fit a game environment. Make sure to think about the choices you want to give the player and when they should be available.

Facilitator Resources

Narrative and dramatic elements can be one of the more challenging components for students to add successfully in a game. It requires creating an engaging storyline that can be applied to a game without overloading the player with too many text bubbles, or cut scenes in commercial games. It can also be complicated to incorporate player choice in meaningful ways in the narrative.

Start a discussion encouraging students to describe their experience playing through Water World 3, and how that compared to the first two Water Worlds. This final world more closely resembles a game – but with narrative that is not fully incorporated. What are suggestions for improving the flow of the game? Finally, open the conversation to cover a broader range of games. How do commercial games incorporate narrative and dramatic elements? Do they use tools that are unavailable in Kodu?

Discussion/Forum Questions:

- Out of the games you've played, what one had the best narrative? What were some of the design elements that made it successful?
- What was a game that was very disengaging because of the narrative? What were the elements that created that experience? How could they be changed to improve your engagement?
- How do videogames tell stories differently than books or movies?

Mission - Creative Genius: Mixing all the aesthetic elements together

Learning Goal

This mission will test students on the aesthetic elements they learned throughout the unit. Students will be asked to incorporate

Kodu Learning Objectives

- Incorporating aesthetic design into Kodu

Relevant Help & Resources Video Tutorials

- Video Tutorial: Creating a Game with Starting Instructions
- Video Tutorial: Koding a Textbox
- Video Tutorial: Changing Size Scale
- Video Tutorial: Changing Sky Color
- Video Tutorial: Changing Water Type
- Video Tutorial: Koding Characters to Express Emotions
- Video Tutorial: Understanding the Resource Meter

Create – Kodu is in Another Castle

Challenge

Make a simple game with a spatial goal of reaching a castle. Rather than focusing on programming, the challenge is to make the scariest game space that you can. Make sure to describe your game when you submit it!

Playtest – Review

Challenge

Play the game and leave a review. Make sure to focus on how the game uses land design, creative assets, narrative, and dramatic elements.

Facilitator Resources

This mission is designed to give students the opportunity to show off their critical aesthetic design skills by incorporates all the techniques discussed in the space, assets, narrative, and dramatic elements sections. While the mechanics of the game are quite simple, with the simplest answer to program a win condition when the PC bumps into a castle, designing a player experience is complicated and can be very time consuming.

While students are working, it may be helpful for you to walk about the room asking students to explain their design. Both talking aloud about their concept and speaking to a third party can illuminate areas that should be revised. It may also be helpful for students to get in pairs or small groups to discuss their designs.

The actual assessment of this game is more challenging, as there are only a few required features in the game. First, students should have a win condition in their game. They can expand on the simple goal – but players should have some accomplishment during play. Secondly, students should have represented all aesthetic design elements in their design: space, assets, narrative, and design elements. You should encourage your students to iterate on their designs if you feel they're lacking in one element.

DYNAMICS

General Learning Goal

This unit is designed to facilitate a more complex understanding of mechanics and aesthetics in games, and specifically how these two components interact to influence player engagement. Students will learn about balance, the concept of tuning a game's rules and systems so that they are aligned in terms of effectiveness, challenge, and desirability of use. Within this unit, students will be introduced to concepts of balanced timers and scores, competition, pacing, and effective world design. All of the missions in Dynamics will include playtesting challenges so that students interact and receive feedback from those that play their games.

Mission - Tick Tock BOOM: Mastering timers and scores

Learning Goal

This mission is designed to teach students how to create scoring and timing systems that are balanced – creating a more engaging game for players.

Kodu Learning Objectives

- Using multiple scores in Kodu
- Playtesting games and iterative design

Relevant Help & Resources Video Tutorials

- Video Tutorial: Using Scores to Represent Non-Score Variables
- Video Tutorial: Invisible Timers
- Video Tutorial: Coding Multiple Scores and Timers
- Video Tutorial: Using Invisible Objects

Watch - Introduction to Dynamics

Video Content

Games are complex systems that have to be very carefully balanced to be challenging enough to get players engaged, but not too challenging to cause frustration. Some of the hardest parts of designing games deal with creating a good balance in the game. Balance is required in both the mechanics of the game and also the aesthetics. Getting a balanced game requires a lot of testing from the designer and can also benefit from playtesting with fresh eyes. In this mission, you'll be challenged to make your games more engaging for players by paying close attention to the balance in your games.

Play – Fusionoid

Challenge

Play Fusionoid. When you're done, explain how timers and scores are used in the game. Would you make any changes to make the game more balanced?

Watch – Dynamics scores & times

Video Content

You were asked to think about and use scoring systems in the mechanics challenge. Now, think about how there needs to be balance in the scoring system so that players will not find it too easy or too difficult to win a game. For example, racing games often have a challenge to complete a track before the clock runs out. The time on the clock is carefully planned so that the player is challenged but can feasibly beat the level. If it's too short, the game is impossible, and if it's too long, the game is too easy. This concept is the same across all forms of scoring systems. These scoring systems can become increasingly more complex, as additional types of scoring are added to a game. For example, points from collecting coins and trying to beat the clock in a race can both factor into how a player is ranked against others in the game. In Kodu, specifically, invisible scoring systems can be used to trigger state changes in the world. For example, an invisible score can keep track of objects collected, with the tenth object triggering the player-controlled character to gain the skill to jump. As scores become more complex and more balanced, testing and playtesting becomes vital in making sure that the scoring systems are working as intended.

Create – Fair game

Challenge

Create a Circus or Fair game in which the goal of the game is to get at many points as possible by hitting targets that are worth different values.

Playtest – Review

Challenge

Playtest and comment on a world. Make sure to focus your input on the use of timers and scores in the game. Other suggestions for design or programming are helpful as well.

Facilitator Resources

It's easy to add a scoring system to a game in Kodu, but it becomes more challenging to craft that system to provide an engaging and intuitive experience for the player. Simple systems – like collecting coins in Mario - don't often require any explanation or training from the game in order for the player to understand them. A more complex system, currency or multiple health bars for different talents will require some explanation or training from the game. It's important that the same standards held in commercial games appear in the games students are making throughout Studio K. It takes time and practice to fully realize that as a designer, one has a different perspective on how the game will work than the player. This mission encourages students to think critically about the difference between designing a game and designing a game for a player. They'll need to think about the motivations of the player, the engagement, the understanding and perceptions of the player.

Students may find it helpful in this mission, and the following in Dynamics, to work more closely with a partner or small group so that there are more opportunities to interact with potential players or third parties who can give honest feedback about play experience. Videogame design in itself is extremely iterative, and for sensitive mechanics like scoring systems, designers should be testing and revising their designs to ensure the best player experience. Additionally, you could give students extra time at the end of the mission, after the playtest phase is complete, to go back and revise their submitted games based on the feedback given in the reviews.

Discussion/Forum Questions:

- In the Scoring Systems challenges, what kind of information is best to include in a review or play testing session?
- How do scores provide feedback to players and why is this information important?
- What are some games with really well designed scoring systems? What are some games with poorly designed scoring systems?

Mission - Pwnd!: Designing the competition

Learning Goal

This mission is designed to teach concepts of competition, created through AI behavior and by understanding player engagement.

Kodu Learning Objectives

- Designing AI behavior
- Playtesting games and iterative design

Relevant Help & Resources Video Tutorials

- Video Tutorial: Moving on a Path
- Video Tutorial: Page Switching and Inline Switching
- Video Tutorial: Koding with the Not Tile
- Video Tutorial: Koding a Lose Condition

Play – 1vCPU Hockey (Easy)

Challenge

Play CPU Hockey (Easy). When you're done, explain why the game is easy to beat and what could be changed to make it more difficult.

Play – 1vCPU Hockey (Hard)

Challenge

Play CPU Hockey (Hard). When you're done, explain why CPU Hockey (Hard) harder than CPU Hockey (Easy).

Watch – Competition Overview

Video Content

The difficulty of the game affects how engaging the player finds it. One way to change the difficulty is to create challenging opponents for the player in the game. The player should feel like there is a possibility of losing throughout the whole game. AI or NPC programmed behavior changes the perceived difficulty of the level. You played both CPU Hockey Easy and CPU Hockey Hard. The only difference between these two games was the behavior of the AI or NPC. In Easy, the NPC patrolled a set path in front of the goal. In Hard, the NPC moved on a set path but also was programmed to move towards the puck. Just this one change increases the difficulty of CPU Hockey and makes the game more engaging for players. In other games, difficulty can be increased through increasing the randomness of behavior. For example, having a boss that is able to randomly use either blips or missiles may make the boss fights more challenging, since the player cannot learn and use a pattern to beat the boss. When making games, think about how your NPCs' behavior can help balance out a game to make it more engaging. Remember to test and playtest games to make sure that you get the right balance in your game.

Create – Action/Adventure

Challenge

Design an action/adventure game. Your game must include at least 3 different types of enemies - each with a different attribute. Make sure to describe your game when you submit it!

Playtest – Review

Challenge

Playtest and comment on a world. Make sure to focus on how the AI behavior is programmed in the game.

Facilitator Resources

This mission will orient your students to thinking more critically about how they program the characters in the world to provide a range of difficulty for their players. Like the other concepts in the Dynamics unit, the concepts are best understood when working closely with other students to allow open communication about design.

For competition specifically, students may have designed their games for themselves in the past. At first, a functioning game is more alluring than a well-crafted, engaging game. With this mindset, it's easy to cut corners in designing the play mechanics – the competition may become easier and easier as the designer realizes it's challenging to test and retest new features in 'Play' if there is some chance of losing. While altering the play mechanics in order to check the whether features that appear later in the game are functioning, encourage your students to maintain a level of complexity and difficulty in their games. Otherwise, the players may not find the games entertaining or engaging.

For this mission, pair or group work is encouraged, as is an open discussion about this design component. Start a conversation with your students about what makes it difficult to beat an enemy and what makes it easy. By drawing on commercial games, students may begin to see patterns emerging in how easy, medium, and hard enemies are programmed. The difficulty level is reached by a combination of the attack play mechanics used and the randomness or complexity of the attack patterns. It may be helpful to ask students to think of how they figure out how to win in boss fights in their favorite games.

Discussion/Forum Questions:

- Boss fights test your knowledge and your skills in a game. What is your process for playing in a boss fight? What makes them more challenging? How do you win?
- What makes an enemy easy to beat?
- What makes an enemy hard to beat?

Mission - Struttin': Commanding the Pacing

Learning Goal

This mission is designed to encourage students to think more critically about designing games with good pacing.

Kodu Learning Objectives

- Understanding how pacing is applied in games

Relevant Help & Resources Video Tutorials

- None

Play – Bad Pacing Platformer

Challenge

Play Bad Pacing Platformer. When you're done, explain why the pacing is not good in the game.

Play – Good Pacing Platformer

Challenge

Play through Good Pacing Platformer. When you're done, explain at least 3 reasons why this game has better pacing than the Bad Pacing Platformer game. Would you have made any other changes to improve the pacing?

Watch – Pacing Overview

Video Content

We are going to take some time to investigate two separate platformer worlds. If you are not familiar, platformer games are ones in which the camera follows one character as he/she hops, jumps, runs, or flies from one horizontal platform to another. These types of games are easy to build in Kodu, but they can be tricky

to make into enjoyable and fun games. In order to build a fun platformer in Kodu, you will need to make sure to be considerate of the pacing. Pacing is the flow of the game, basically how smoothly the player moves through an event and then onto the next. Some games are built to really trip up the player but that does not mean the game has bad pacing. In the example world we provided, the pacing is poor because the player has to go backwards through the level at different points and needs to constantly go up and down on the platforms to collect all the coins. This is not a good pacing for a platformer game. In the better example, we simply removed some of the highest platforms and rearranged the placement of the coins, so a player can find the best path to get all the coins. This doesn't make the game easier or harder, it just allows the player to have a more enjoyable and engaging play experience. The concept of pacing is important for all games, not just platformers. While you're designing – make sure to be aware of the timing of events, the placement of object and enemies, and the mechanics used in your game so that the flow is enjoyable for players.

Facilitator Resources

In this mission, students will interact with one of the more abstract concepts in the Studio K curriculum. Pacing, the flow of the game - how smoothly the player moves through an event and then onto the next, is important to making an engaging game but can be very tricky to alter in a game. Rather than testing this skill, this mission encourages a more open understanding and conversation about the issues around designing a well-paced game.

Since pacing is subjective, start a discussion about whether they agreed with how the Bad Pacing Platformer and Good Pacing Platformer games were designed. How would they further change the game to make the experience more enjoyable? After exhausting the conversation around those games, open it to commercial games – what are some games that students get caught up playing and lose track of time? What are some games that were challenging to get through because they got boring in some parts? What were the design features in these games that made the pacing good or bad?

Discussion/Forum Questions:

- How would you change the Bad Pacing Platformer game to make it better? How would you change the Good Pacing Platformer to make it better?
- Will pacing be the same for every player? How can the pacing in games be designed to make the majority of players engaged?
- What game can you lose track of time playing? What parts cause you to do so? How can you apply those parts to a game you make in Kodu?

Mission - Whole New World

Learning Goal

This mission is designed to highlight the importance of intentional space design in games and how this can improve the balance and pacing of the game.

Kodu Learning Objectives

- Crafting space that guides player action
- Playtesting and iterative design

Relevant Help & Resources Video Tutorials

- Video Tutorial: Koding Size State Changes
- Video Tutorial: Koding Sky State Changes
- Video Tutorial: Koding Water Type State Changes
- Video Tutorial: Koding Water Height State Changes

Watch – Designing for your Player

Video Content

Balance and good pacing in a game is also obtained through the world design. When you think good games, the assets and spaces in the game often give the layers hints for how to play and guide them to make certain decisions in the game. For example, in many games, characters can climb the sides of buildings, or walls. Generally the walls look different when they are climbable which provides hints and feedback to the player. By doing this, the designers have effectively highlighted the path for the player. So when you are designing your games in Kodu, think about how your players will navigate the world, how they would make sense of what is happening, what they are supposed to do, and where they are supposed to go. You, as the game designer, can provide clues to the player to help them figure out all of these things. Each genre of game requires a different use of the space in order to get players to do what is expected in the world. It wouldn't make sense to have a racing game that allows cheating by going backwards – by changing the space of the game so that players are unable to move backwards, the space guides player activity thus making the game more engaging.

Create – Making Space Dynamic

Challenge

Design a space that very clearly guides the player through your level. Use creative assets and land design to do so. Make sure to describe your game when you submit it!

Playtest – Save Princess Kodu

Challenge

Play through 'Save Princess Kodu'. When you're done, explain how the space was used in the game and how you would change it to improve it.

Playtest – Review

Challenge

Play the game and leave a review. Make sure to focus on how the game world is designed and what your experience as a player was like because of the world.

Facilitator Resources

This mission is designed to get students to think more critically about world design in their games. Expanding on the concepts covered in the Aesthetics unit, students should learn that spaces in commercial games are designed extremely carefully, with each element contributing to the player experience – both emotionally and mechanically. Designing a space to elicit an emotional space was covered in the Aesthetics unit, so this mission builds on that, covering the mechanical side of space and asset use.

Each genre of game is designed with a unique space that supports how the player interacts with the goals, rules, play mechanics, and narrative. In exploration games, like World of Warcraft, the space is open and unstructured – players traverse the world guided by quest lines but the freedom of exploration is always present. Conversely, a platformer is much more restricted in the space design – with player movement often constrained to forward progression only. If the spaces in an exploration game and platformer game were to be switched, the goals, rules, and play mechanics for each would no longer make sense.

Students should be aware how important space is for their games. Players should be able to make play decisions in the game based information they receive from the space and assets in the world. In order to get students to think more critically about design, start a discussion about the spaces in their favorite games. How would player experience change if the space was more or less linear/open?

Discussion/Forum Questions:

- What is your favorite genre to play? Describe the spatial design of that genre. How does that affect your experience as a player?
- What sort of information can be gained from the space in a world?
- What are the effects of having a more linear or more open space in a game? How do other game components, like goals, rules, or narrative, interact with these spaces?

BOSS FIGHT

General Learning Goal

This unit is designed to test the concepts learned in the Mechanics, Aesthetics, and Dynamics units. Students showcase their skills in a final project, which will be playtested and reviewed by their peers.

Mission - Like a Boss: The final challenge – do you have what it takes?

Learning Goal

This unit is designed for students to showcase their game design skills.

Kodu Learning Objectives

- None

Relevant Help & Resources Video Tutorials

- None

Create – Option 1

Challenge

Create a game which takes at least 5 minutes to play! Make sure to describe your game when you submit it!

Create – Option 2

Challenge

Build a game with at least 3 stages/levels. Make sure to describe your game when you submit it!

Playtest – Review

Challenge

Play the game and leave a review. For this review, focus on all the elements of game design that you have picked up.

Facilitator Resources

Since the Boss Fight is the final challenge to students and incorporates all skills learned throughout Studio K, Students may need additional time for design. Like other create challenges, it may be helpful for students to work in pairs or small groups so they can easily discuss their design ideas. You may also want to encourage students to create a design document prior to opening Kodu. Design documents are frequently used in the creation of commercial games. The

document is a stable reference with a fairly complete description of the features of the game – including goals, rules, mechanics, narrative, and flow. This may be helpful for keeping students on track and also seeing how the design is altered throughout the process.

The assessment of the games is once again subjective, although the challenge objectives should be met each submission. Kodu encourages the exploration of game design based on interest – so you should see a range of games based on your students' interests. You may find that you are receiving submissions that focus heavily on either the mechanics or the aesthetics of the games. Students should explore their expertise and interests within Kodu, but you should also encourage students to make a fully functional and professional game. So while the narrative may be expertly crafted, also ensure that the underlying mechanics are functional and provide support the engaging experience. This final challenge is a great one to give students extra time after receiving feedback from playtesting to iterate on their games. The more time spent interacting with peers around the games and design of the games will lead to better quality games.

Mission - Roll Credits: Recognizing your skills

Survey - One final Challenge

At the end of the curriculum, students be asked to complete an exit survey so that we can determine how their interests and knowledge relating to game designing and programming (see Appendix A) have changed since the start of the program. This will not capture any personal information about the students.

Since this data is vital for understanding the effects of spending time in the curriculum, you should set time to take the exit survey at the end of your unit with Studio K regardless of whether or not the curriculum has been finished in its entirety.

Appendix

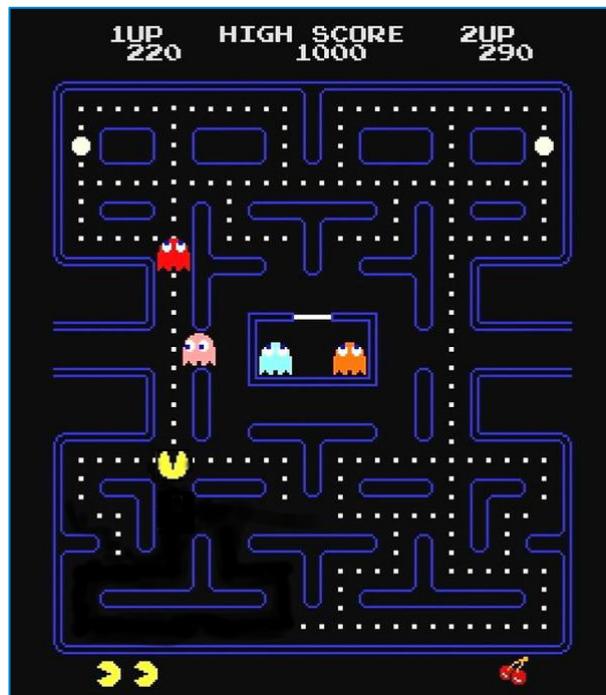
Appendix A: Entry Research Survey

1. What is your Studio K username?
2. What is your sex?
 - a. Male
 - b. Female
3. How old are you?
4. What race do you consider yourself? Select all that apply
 - a. American Indian or Alaska Native
 - b. Asian or Pacific Islander
 - c. Black or African American
 - d. White
 - e. I prefer not to answer
 - f. Other (please specify) _____
5. What ethnicity do you consider yourself?
 - a. Hispanic, Latino, or Spanish origin
 - b. NOT of Hispanic, Latino, or Spanish origin
 - c. I prefer not to answer
6. How well do the following statements describe you?

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I hope to get a job doing something with video games.	<input type="radio"/>				
It's important to me to know how to program.	<input type="radio"/>				
I would be interested in taking more classes on programming.	<input type="radio"/>				
It's important to me to know how to design video games.	<input type="radio"/>				

How well do the following statements describe you?	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I think of myself as a gamer.	<input type="radio"/>				
My peers know I play games.	<input type="radio"/>				
I like to be thought of as a gamer.	<input type="radio"/>				
Gaming is important to who I am.	<input type="radio"/>				

The first challenge is to think about the game Pacman. Think about how Pacman works and how it is run on a computer.



7. In the game of Pacman, why is it important that the ghosts try to kill Pacman?
8. How would do the ghosts decide where to find Pacman?
9. In Pacman, in what situation should Pacman move left?

```
This is the Kode for a Kodu  
When: Always → Do: Move  
When: Sees Black Cycle → Do: Follow  
    When: → Do: Shoot Blips It  
When: On Land 1 → Emote Happy  
    When: Timer Random → Do: Create creatable Heart Once  
    When: See Black Cycle → Do: Move Away It  
    When: See Heart Not → Do: Switch Page 2
```

10. What does the Kodu do when it sees a black cycle?

What happens when the Kodu is on land type 1?

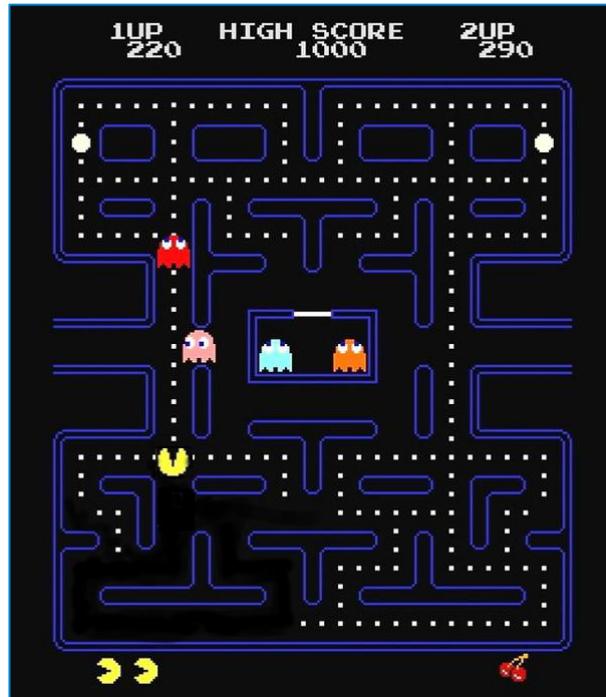
Appendix B: Exit Research Survey

1. What is your Studio K username?
2. How well do the following statements describe you?

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I hope to get a job doing something with video games.	<input type="radio"/>				
It's important to me to know how to program.	<input type="radio"/>				
I would be interested in taking more classes on programming.	<input type="radio"/>				
It's important to me to know how to design video games.	<input type="radio"/>				
I liked the studio k class.	<input type="radio"/>				
I did not like Kodu.	<input type="radio"/>				
I learned game design concepts.	<input type="radio"/>				
I did not learn programming concepts.	<input type="radio"/>				

How well do the following statements describe you?	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
I think of myself as a gamer.	<input type="radio"/>				
My peers know I play games.	<input type="radio"/>				
I like to be thought of as a gamer.	<input type="radio"/>				
Gaming is important to who I am.	<input type="radio"/>				

The first challenge is to think about the game Pacman. Think about how Pacman works and how it is run on a computer.



3. In the game of Pacman, why is it important that the ghosts try to kill Pacman?
4. How would do the ghosts decide where to find Pacman?
5. In Pacman, in what situation should Pacman move left?

```
This is the Kode for a Kodu

When: Always → Do: Move

When: Sees Black Cycle → Do: Follow

    When: → Do: Shoot Blips It

When: On Land 1 → Emote Happy

    When: Timer Random → Do: Create creatable Heart Once

    When: See Black Cycle → Do: Move Away It

    When: See Heart Not → Do: Switch Page 2
```

6. What does the Kodu do when it sees a black cycle?
7. What happens when the Kodu is on land type 1?

Appendix C: Studio K Vocabulary List

Aesthetics

the visual aspects of a game.

(Tick Tock Boom - Introduction to Dynamics)

AI

artificial intelligence; a set of behaviors programmed into NPCs meant to simulate intelligent behavior.

(Pwned! - Competition Overview)

Balance

the concept of tuning a game's rules and systems so that they are aligned in terms of effectiveness, challenge, and desirability of use.

(Tick Tock Boom - Introduction to Dynamics)

Behavior

an action set of a computer controlled characters; NPCs will often have a set of behaviors they cycle through depending on what occurs around them.

(Pwned! - Competition Overview) -

Broadcasting

the concept of one object sending a “message” that can be received by other objects and then reacted to.

(Now You See Me - Broadcasting Tutorial)

Characters

NPC

non-playable character; characters that exist in the world but act as prescribed by how they are programmed.

(Studio K 101 - Making Moves)

PC

player character, manipulated by the player through an input device.

(Studio K 101 - Making Moves)

Cloning

allows objects with the same properties to be made in the game world before run time (during edit mode).

(Copycat - Cloning and Creatables Tutorial) -

Creatables

a setting in Kodu that allows objects with preset properties to be instantiated at run time.

(Copycat - Cloning and Creatables Tutorial)

Feedback

how much information a player receives about their performance (immediate and long term).

(High Score! - Scores Overview)

Gameplay

what a player does in interacting with a game; the interactive experience of the player.

(Landcrafting - Watch - Land Design Intro)

Genre

the category a type of game falls into: platformer, shooting, racing, etc. Often games mix and mash common elements of multiple genres.

(Landcrafting - Land Design Intro)

Line

a line of Kode.

(Studio K 101 - Making Moves)

Mechanics

rules (both applied to the world, the player, etc.), that are intended together to produce a game or gameplay.

(Tick Tock Boom - Introduction to Dynamics)

Narrative

the story, either explicitly authored or emergent, experienced through gameplay.

(Kodu's Keeper - Using Objects)

NPC

See 'Character'.

Objects

'containers' that hold code which give them properties and allow them to perform actions. In Kodu, all objects are physically represented as things or characters with their own set of properties and actions that can be placed in the world.

(Studio K 101 - Getting the Guy)

Object Properties

an objects traits that can be manipulated both during the design phase and at run time through code.

(Copycat - Cloning and Creatables Tutorial)

Pacing

the rate at which events play out inside the game world

(Whole New World - Designing for Your Player)

PC

See 'Character'.

(Studio K 101 - Making Moves)

Platformer

a game in which the camera follows one character as he/she hops, jumps, runs, or flies from one horizontal platform to another.

(Struttin' – Pacing Overview)

Randomness

using random variables to add an element of surprise to gameplay; randomness can also be used to create unique experiences each time a game is played.

(Pwned! - Competition Overview)

Spaces

the spatial design inside a game world; achieved by the arrangement of objects and assets inside a world as well as by land manipulation.

(Landcrafting - Watch - Land Design Intro)

Spatial Goal

a goal that is achieved through physical movement within a game world.

Copycat - Introduction

Systems

well crafted mechanics that together result in desirable play dynamics.

Tick Tock Boom - Introduction to Dynamics

Testing/Playtesting

the practice of playing through one's game or having others play it in order to observe inconsistencies and bugs. Also the concept of analyzing one's code to make sure intended behavior occurs.

(Tick Tock Boom - Introduction to Dynamics) -

Triggers

the 'if this...then that' sequence that games use often to signal changes in areas from game state to player performance. An example: hitting a tree triggers an apple to launch out.

(Studio K 101 - Making Moves)

Win State

(Studio K 101 – In it to Win it)

when the conditions to meet the goal of a game are met.

World

the space of the game (or a part of game). The player might not see the entire world at once but the designer does. Often called a 'scene' in different game development environments.

(Studio K 101 - Setting the Scene)