

A. Discussion: Game Analysis (1)

B. Discussion: Game Design Terms (2)

C. Discussion: Intellectual Property (3)

D. Discussion: Changing Rules (4)

E. Getting into Scratch

Before we get into designing our games, we first need to know how to use our tool! Let's step through a few tutorial projects so you can familiarize yourself with the capabilities of Scratch, before you begin designing your game.

Scratch Super Basics

1. We will be using Scratch, a graphical programming tool, created by MIT for this class. Please go to scratch.mit.edu in your web browser. Before we start designing our games, let's look at the tool and what it can do.
2. Click on the "Explore" link. Spend five minutes looking at existing games and animations, to see some of the capabilities of Scratch.
3. Click on the "Create" link. The first time you start a Create session, you will see a "Step-by-Step Intro" on the right-hand side of the screen. Follow this Step-by-step intro up through step 11. Stop once you get to the step titled, "Explore!"

Note: Some of these steps include playing sound. If you do not have earbuds or headphones with you, use the sound blocks anyway.

Let's Make Pickin' Sticks!

Before we get into designing and implementing your own game, it would be a good idea to have an idea of how to use Scratch to make a game!

Download the "Pickin' Sticks Scratch" guide from:

http://www.moosader.com/content/tutorials/BSA_GameDev/PickinSticksScratch.pdf

and follow the steps to make a simple game!

F. Design your own Game

You've now seen a little bit of what Scratch can do. Let's move on to designing our games, and after you get counselor approval on your design, you can begin implementing it in Scratch.

If you would like, you can build on top of the Pickin' Sticks game, or you can start from scratch... In Scratch!

Keep It Simple: Remember that you can always work on games outside of this class! You can work on bigger projects at home. For now, focus on making old arcade-styled games, and games with only a few gameplay elements.

Following the merit badge requirements, and the Game Design merit badge handbook, consider the following:

What is the vision statement? (5a)

What is the game's medium? - Probably the computer!

What is the player format? (page 25 of handbook) - Single-player? Head-to-head? These two are probably the best choices, since we don't have time to make a large multiplayer game!

What are the objectives? (page 27 of handbook) - Do you want to get a high-score? Capture or destroy opponent assets? Collect all the stars? Solve a puzzle? There are even more ideas in the handbook.

What is the theme? (page 33 of handbook) - What is the setting, or story, or even the characters? What helps make your game have its own living, breathing world?

What is the play value? (5b)

What makes this game worth playing? Is it challenging? Does it tell a story?

What are the rules? (5c)

Make a list of rules for your game. Setup, progression of play, and resolution. (page 29 of handbook)

Draw your game's elements (5d)

Create a new project in Scratch (you might want to open this in a new tab so you don't lose Pickin' Sticks as a reference!). Draw any sprites and backdrops you would like to use. You could also just sketch your ideas for the graphics in your notebook

Once you've come up with a "game plan" (so to speak), get approval from the counselor before continuing. The coordinator will make sure that the **scope** is small enough that it can be completed within the class time.

G. Implement your own Game (6)

Prototype the Game (6a)

Once you have your counselor's approval, begin implementing your game in Scratch.

Test the Prototype (6b, 6c)

Help out another scout by testing their game, and find one or more scouts to test your game as well.

You will need to go through at least three testing iterations. Each iteration:

- Have another scout test your game
- Did their gameplay experience match your intended experience?
- Fix any problems with the rules: unclear, holes, dead ends, exploits, etc.
- Change at least one rule, mechanic, or objective before letting another person test the game.

H. Blind Test & Analyze (7)

Write Instructions (7a)

In Scratch, or in Notepad, write your game's instructions: How to start the game, what the objectives are, how to play, and how to finish the game.

Share the Prototype (7b)

Have a group of people who have not tried your game out test it now. Give them the instructions and the game, and any physical components. Silently watch as they play the game, do not give them feedback or try to alter how they're playing.

After they're done playing, record their feedback in your notebook.

Share your Game Design notebook with the Counselor (7c)

Discuss player reactions to your game, and talk about what you learned while planning, designing, and implementing the game.

Based on how you refined your game through testing, figure out what you like most about your game, and what might need to change.

I. Discussion: Career Opportunities (8b)

Want to keep making games?

Here are some free tools you might look at, if you want to pursue game development in your free time!

Game Creation

Game Maker, yoyogames.com/studio

Has a free version you can use to make games with a graphical interface

Sploder, sploder.com

An online game making program.

Unity3D, unity3d.com

A popular 3D engine used to make 2D and 3D video games!

RPG Maker VX Lite,

tryace.rpgmakerweb.com/download-lite

An engine that makes creating Role Playing Games easy!

Starter Programming

Python with PyGame, pygame.org

Python is a popular scripting language, and PyGame is a library that lets you use graphics, sound, gamepads, and more!

Lua with LÖVE, love2d.org

Lua is also a scripting language, and LÖVE is similar to PyGame!

JavaScript, khanacademy.org/cs

If you have a web browser, you have JavaScript! Khan Academy has lessons on learning JavaScript!

Art Tools

Graphics Gale, humanbalance.net/gale/us

Graphics Gale has a free version you can use to work on Pixel Art and animated game sprites!

Paint.net, getpaint.net

Paint.net is a photo/image editor you can use to draw backgrounds, sprites, and more!

Audio Tools

Audacity, audacity.sourceforge.net

Audacity is an audio recording and editing tool, useful for making your own sound effects!

Anvil Studio, anvilstudio.com

Midi editing software for you to write songs in!

BFXR, bfxr.net

A web-based program that will let you generate retro-style sound effects for your games!