EECS 494 2015.Fall – Project 1 – Classic Game Project

Final Due Date: Tuesday, February 2, 2015 before class begins

Goals / Spec:

Working in pairs, you will faithfully recreate core features of one of four console games using C# inside the Unity game development engine. You will be responsible for recreating a specified “classic level”[1] of the original game in exacting detail as well as implementing a new “custom level” of your own design. Prior to the start of the assignment, you will be provided with a list of approved games and a starting tutorial for each.

There are several goals to this assignment:

1. You will learn C# and the Unity game engine (while not having to simultaneously design a new game from scratch).
2. You will get a chance to examine and recreate the esoteric design details of a gaming classic. There are dozens of tiny details in these games that made them rise above all of the other games of the era, and many of them are not immediately obvious.
3. You will gain experience with the Agile development methodology—including burndown charts and scrums—that will be used throughout the rest of the class.
4. You will take part in the iterative process of design that is the cornerstone to good game design.
5. You will get a chance to see how the game mechanics and technologies that you develop for the recreation of the classic level can be repurposed for your custom level.
6. Remaking or reusing the original graphics is ***not*** a goal for this project. You will be scored on how well your game emulates the feel of the original game but very little score will be given based on how good the graphics look.

Your game will be delivered as **both a zipped Unity project folder and a Unity web build** hosted on a web page, and you are responsible for the hosting. For information about the web space provided for you by U-M, look here:

http://www.umich.edu/~umweb/how-to/homepage.html

I strongly encourage you to take advantage of the free GIT repository service that EECS hosts for you. You can find this at http://gitlab.eecs.umich.edu .

Mindset:

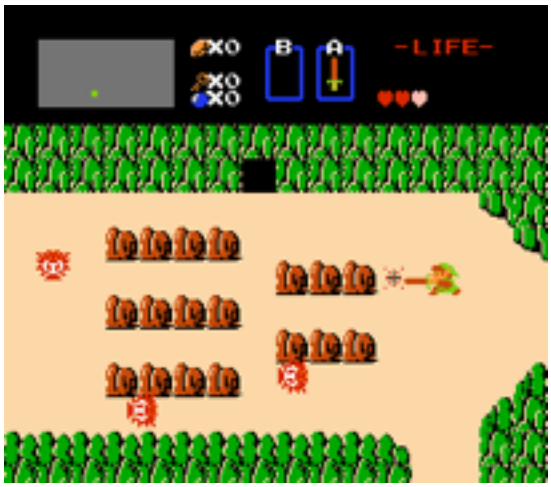
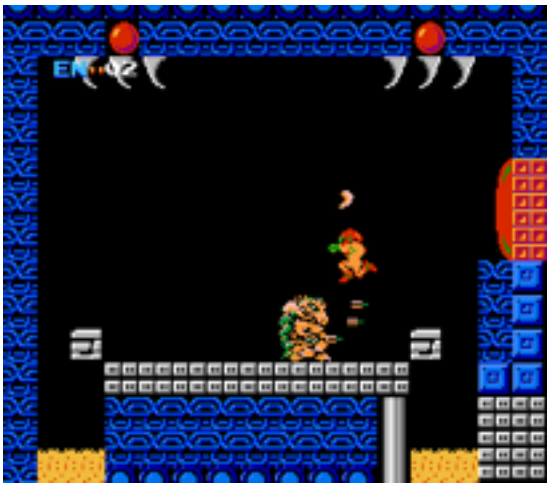
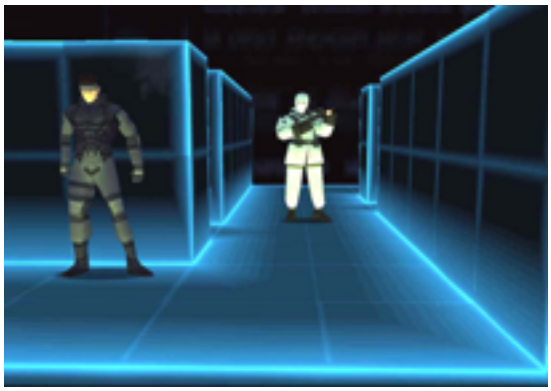
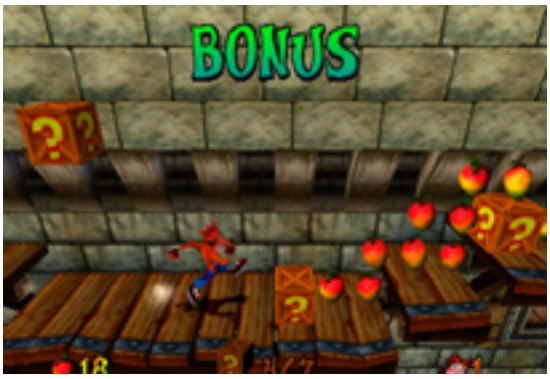
Read the Gamasutra article "How to Prototype a Game in 7 Days" by the Experimental Gameplay Project team from Carnegie Mellon (http://www.gamasutra.com/view/feature/130848/how_to_prototype_a_game_in_under_7_.php?print=1). Then check out their website to see the kind of fantastically creative games that can be made by a single person in just one week if approached with the right mindset and abilities: http://experimentalgameplay.com/ . Remember that in this project, it's much more important to have something that plays properly than something that is coded "properly". The key is capturing the feel of the game and the game mechanics, not fancy coding.

---

[1] A single level of a game would be something like World 1-1 in Super Mario Bros. or a single dungeon in The Legend of Zelda. If you have any question about what would be equivalent to a "level" in your game, please speak with Jeremy or your GSI as soon as possible.

Possible Games

There are 4 possible games from which you'll pick one to make:

The Legend of Zelda (NES)	Metroid (NES)
	
Metal Gear Solid VR Missions (PS1)	**Crash Bandicoot (PS1)**
	

We are expecting different things from each different game.

The Legend of Zelda has sliding blocks, items that confer abilities, inventory management, and hidden rooms, but each room is only one screen in size.

Metroid has scrolling screens (in two directions), lots of changes to movement and abilities, but very simple enemies.

Metal Gear Solid VR Missions has enemy patrol and stealth mechanics as well as hundreds of levels (you will only need to implement a few of them).

Crash Bandicoot is a classic 3D platformer with various character abilities, enemies, and pickups.

Grading:

You will be graded based on the following criteria (totaling to 100%). For all of these, just doing exactly what is required of you will tend to get you a B. If you want an A or higher, you need to go beyond the base requirements.

5% – Tutorials / BDC / Progress – You need to have selected your game and your teammate, set up and shared your burndown chart on time, and progressed beyond the initial tutorial. **Due 1/19**.

10% – Alpha / Custom Sketch – Is your game on-track? Have you completed at least 75% of the required tasks on the burndown chart? Do you have a sketch of what your custom level will look like? **Due 1/26**.

15% – Classic Game Project Turn-in – Did you turn in your project on time? Did you complete all the required tasks on the burndown chart? **Due 2/2**!

15% – Mechanical elements – How faithfully did you recreate the mechanics of the original? Does the game feel complete and internally consistent in the same way as the original? Did you capture the "special sauce" that made the game a classic? **Due 2/2**!

15% – Technical elements – Does the game work? If a game isn't playable, it's not a game. Memory leaks, glitches, and other bugs will count against you for this requirement. **Due 2/2**!

5% – Aesthetic elements – Does the game move and feel like the original? Sprites and actual game art will not count at all for this aspect of the grading, however the “feel” of the game will (see the book *Game Feel* by Steve Swink). Don't forget about music. **Due 2/2!**

20% – Custom Level – Does your custom level take advantage of the various mechanics in the game? Does it find a new way to remix the classic elements of the game while at the same time feeling like it belongs with the other game levels? Is it at an appropriate level of difficulty? You should try to make a beginner-to-medium level of difficulty. This should be something like the second or third dungeon in The Legend of Zelda (definitely NOT the last dungeon). Does it have good progression and guide the player well? **Due 2/2!**

15% – Process: Agile Methodology – This will be the first project for which we will be requiring you to follow the Agile development methodology. You will be creating burndown charts that must be updated before each class scrum, and you will be required to take part in at least one scrum. **Due 2/2!**

Tools

To do this assignment you will need to use Unity 3D, MonoDevelop (or Microsoft Visual Studio) and C#. There is no reliable version of Unity available on the CAEN computers, however, Unity is free and is available for both Windows and Mac operating systems (at http://unity3d.com). There are also music and audio programs (like Audacity and Garage Band) and paint programs (such as Adobe Photoshop, Macromedia Fireworks, & the Windows Paint program) available on CAEN machines. Additionally, there are lots of game development resources available on PixelProspector (http://www.pixelprospector.com/indie-resources/). Also, as a student, you can get all of Adobe Creative Cloud for <$20/month (http://www.adobe.com/products/creativecloud/students.edu.html) and all of AutoDesk's tools (3DSMax, Maya, Mudbox, etc.) for free (http://www.autodesk.com/education/student-software). Unity is also available in the Mac Training Lab on the 3rd floor of the Duderstadt Center.

Agile Progress Reports and Burndown Chart

As described in the video "Intro to Agile Scrum in Under 10 MInutes" (http://www.youtube.com/watch?v=XU0llRltyFM), a burndown chart is a fantastic tool for tracking progress on a project. Look for the instructions on how to use our class's Google Doc-based burndown chart here:

http://bit.ly/494BurndownInstructions2

We may ask you to take part in a scrum on *any* class day during the project: Scrum – This will always happen on Tuesdays, but I could call some people out on Thursdays as well. Be prepared!

As a slight modification on what was shown in the video, participants in a our scrum meetings will answer four basic questions:

1. What are the name and logline of your game?
2. What did you do since the last class day?
3. What will you do before the next class day?
4. What do you need help with?

Our in-class version of scrum will be very similar. On scrum days, if you are randomly chosen to present, we will bring up your burndown chart on the projector screen, and you will tell us the name and logline of your game, what you've done since the last scrum, and what you plan to do before the next one. Also let us know if something is blocking you, and we'll find someone to assist you with that (which could be an instructor or a peer).

If you miss a scrum where you are chosen, you must present at another scrum, but you will lose 10% of your Agile Methodology process grade.

Playtests on Tuesdays

You are required to have your game playable and ready to show on every Playtest day. We may also assign you to play various games made by your peers as homework. When you are not showing, you should try to give meaningful feedback to as many people as possible. You need to have playtests logged on the online system for your game every playtest day (additional playtest logs will also be beneficial). http://feedback.prototools.net/

Final Deliverable

Due February 2 – You must submit two zip files to Canvas: a zip file containing the entire Unity project folder for your game **and** a zip file of a folder containing your compiled web build. Double check to ensure that you can unzip this file and play the game. ***Projects that are unplayable because of a bad submission (due to corruption, incomplete submission, or any other reason) will be counted late until they are playable***.

Naming Convention: 494W16-P1-GameName-FirstnameLastinitial_FirstnameLastinitial (e.g. 494W16-P1-Metroid-JeremyB_AustinY). Be certain to **name your folder before zipping**!

Setting-Up Your Burndown Chart

1. Log in to Google. These burndown charts require a Google account.
2. Read the instructions at: http://bit.ly/494BurndownInstructions2
3. Go to the burndown chart template for your chosen game and then choose File > Make a Copy… from the menu to make your own copy.

4. Name the new file W16-P1BD-GameName-FirsnameLastinitial_FirsnameLastinitial (e.g. W16-P1BD-Metroid-JeremyB_AustinY) and click Ok. This will create a new copy of the template in your Google drive.
5. Share the new document with the GSIs and me:
    1. Click [Share] in the top, right corner.
    2. Add <ayarger@umich.edu>, <medstone@umich.edu>, <ryanwawr@umich.edu>, <gameprof@umich.edu> in the Invite People text area.
    3. Choose "Can Edit" from the popup to the right of that text area.
    4. Click [Share & save].

Late Policy as Stated in the Syllabus

Late assignments will be assessed a penalty of 2.5% for each 6 hours it is late (10% per day). We will not grant extensions except for extended sicknesses.

Questions

The class forum is available through the Canvas Piazza and is by far the best place to ask questions.