EECS 494 – Assignment 2 – 2D Game

Due Tuesday, October 8, 2013 at 11:59pm

Goals / Spec:

Working individually, create an original 2D game using Zenilib that will run under Windows on the CAEN computers in the CSE lab. We will provide you with a framework that handles most of the low-level messy stuff (initializing rendering devices, setting up back buffering and vertical sync, etc.). You have to design a 2D game, create the graphics, and program the inner workings of your game. You will also be responsible for implementing a Burndown Chart and using the AGILE project methodology to get your game done on time.

We expect a game on the order of Centipede (or more!). The game should not be a reimplementation (or scaled down version) of an existing game. The more original the game is the better. Be sure to attend the classes September 16th & 20th; these sessions will help you get to know Zenilib and the requirements of the assignment.

Your game must run under Windows on the CAEN computers in the CSE computer labs.

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 in just one week if you approach it with the right mindset and abilities: http://experimentalgameplay.com/ .

Also, be aware that if your game is challenging or long, you need to give us a way to be able to experience the whole game in a reasonable amount of time. For instance, if your game is exceedingly difficult, you need to give us the ability to not fail so that we can see the final boss that you spent 20% of your development working on (e.g. invincibility cheat, etc.). If the game grows gradually over a week, you need to give us several save games to load, each of which places us at a different point in the progression of the game. This is **absolutely** something you should also keep in mind whenever you submit a game to a contest. Judges generally only have 5 minutes or less to decide whether the game deserves more of their time (and you should definitely not make 4 of those 5 minutes be waiting on an installer!).

Grading:

You will be graded based on the following criteria (totaling to 100%):

- **40% – Formal & Dynamic Mechanical elements** – How good is your game design? This is determined by how well your game executes its goals as stated in your postmortem document. Is the game fun to play? Does the design feel complete and internally consistent? Does it achieve your experiential goals?

- **15% – Formal & Dynamic 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.

10% – 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.

10% – Process: Postmortem Documentation – Your postmortem document will describe the process that you went through to create this game. Does it accurately demonstrate your design and development process? Does it demonstrate a good progression in the development of your game design and game.

7.5% – Originality – Is this something different from what we've seen before in class? Are you approaching a different kind of design problem? Are you approaching a common design goal in a novel way?

7.5% – Complexity of Implementation – Did you try something difficulty when making this game?

5% – Formal & Dynamic Aesthetic elements – I don't care much about how pretty the game is. However, I **DO** care about whether the game feels good to play. Take a look at the Gamasutra article above and think about "juiciness" when making your games.

5% – In-Game Documentation – Does your game include in-game instructions. We don't want you to have a manual, and a tutorial is not critical, but the player must have somewhere to go to be able to learn what various buttons and controls do in your game.

Tools

To do this assignment you will need to use Microsoft Visual Studio 2010 (Express) (or Visual Studio/C++ 2012), zenilib (see below), and a paint program (such as Adobe Photoshop, Macromedia Fireworks, the Windows Paint program available on CAEN machines, the GIMP, etc.).

C/C++ Compiler

You need Microsoft Visual Studio 2010 (or Visual Studio/C++ 2012). Visual C++ 2010 Express is a free download. http://www.microsoft.com/visualstudio/eng/downloads

The Zenipex Library (zenilib)

It handles all of the low level stuff that is such a pain in game programming so you can concentrate more on creating the game. You can find information and downloads for the Zenipex Library at http://zenilib.com. Follow the instructions and get the latest version of the zenilib and its supporting files.

Paint Program

You will have to create graphics for each of your game objects. You can use any paint program that creates BMP and/or PNG files. The Paint program in windows, the shareware program Paint Shop Pro, Macromedia Fireworks, Adobe Photoshop and the GIMP are all appropriate choices. GIMP is free, and as a student, you can get all of Adobe Creative Cloud for <$20/month.

You must test your program on the Windows machines in the CSE computer labs to make sure that it runs correctly. If you are doing your development elsewhere, you should test your game periodically on the CSE machines to avoid an unpleasant surprise on the day that your game is due!

Progress Turnins

September 20, 2013: 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.

Any class day during the project: Scrum – 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 scrum?
3. What will you do before the next scrum?
4. What is blocking you?

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.

September 27 & October 4: Playtests – You are required to show your game on both Playtest days. On those days, we will play half of your games for the first half of the class time and then swap to the other half. 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://prototools.net/

Final Deliverable

October 8, 2013 @ 11:59pm: Playable Game & Postmortem Doc –
You must submit a zip file to CTools containing:

1. A playable build of your game for Windows PCs that will definitely run on the Windows machines in the CSE lab.
2. A folder containing all of your code.
3. A PDF of your postmortem document. An example postmortem document will be sent out next week. Just like the playtest notes for Assignment 1, you want to keep a journal of your progress on the project while working on it, and your postmortem will be compiled from that.

Naming Convention: EECS494-A2-LastnameFirstname (e.g. EECS494-A2-GibsonJeremy). 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. Go to the template at: http://bit.ly/EECS494_A2_Burndown
3. Choose File > Make a Copy...
4. Name the new file EECS494-A2BD-LastnameFirstname (e.g. EECS494-A2BD-GibsonJeremy) and click Ok. This will create a new copy of the template in your Google drive.
5. Share the new document with Mitchell and me:
    1. Click [Share] in the top, right corner.
    2. Add <bazald@umich.edu> and <gameprof@umich.edu> in the Invite People text area.
    3. Choose "Can Comment" from the popup to the right of that text area.
    4. Click [Share & save].
6. Enter your name and the game name on the first row.
7. Add features (along with rank and estimated hours to complete) in columns A-C. As your project progresses, modify the number of hours remaining in columns I-P.

Late Policy as Stated in the Syllabus

To be considered on time, assignments must be turned in at the specified time on the due date. Each student has a total of **three** late days that can be used throughout the semester. Thus, during the semester a student can turn in assignments so that the total number of days late does not exceed three without any penalty. For example, a student could turn in three different assignments, each one day late, or all assignments on time except one that is three days late. For the final project, late days cannot be shared with other team members.

Once the number of late days exceeds a total of three, late assignments will be assessed a penalty of 1% for each 6 hours it is late (4% per day). Since we provide the late days, we will not grant extensions except for extended sicknesses.

Questions

The class forum url is: https://phorum.eecs.umich.edu/list.php?13. The forum is the best place to ask questions about this assignment.