386 — Game Dev — Simple New VG in Pygame

Project #3: Simple New Video Game in Pygame

This project is to design, prototype, and build a new (simple) video game in **Python + Pygame**. You'll build it in several phases.

Introduction

This is the final project for the course. The objective of this assignment is to create a non-trivial video game of your own choosing. The final game must be a well conceived and implemented game that aims to be challenging and engaging for the target audience.

A premium is placed on having a well documented and well conceived design. The final project is judged not only on the completeness of the solution but also on the quality of the student's understanding of the project as demonstrated by a written project summary, and an in class presentation and demonstration of a working video game program. Also, include a bunny.

Teams

The final project may be built either individually or in a team of up to 4 class students. If it is a team, then itemize what each member was tasked with doing/building in your project design report.

You should consider using Agile "Stand-ups" to ensure progress. Less than 3 minutes time, progress board, and the 3 questions answered (sometimes embarrassingly).

Rough Concept Spec

Please submit a concept-phase rough speculative (spec) game summary for your game, 1 week after the start of this project. The rough spec should be submitted as a 1-page pdf file. (Scan the drawings, if needed.) The contents of each rough spec is as follows:

1. Developer Team name, and team members.
2. Game Name: in 7 words or less.
3. Game Tag-Line: in 1 sentence. What does it do? What is it about?
4. Game Theme/F-World: in 1 sentence. What is the key context? What is the background?
5. Key Type of Action/Mechanics: in 1 sentence. What is the player's go-to action move/play?
   EX: take a piece, jump to platform, leap to next moving log, swing sword, shoot a missile, roll a bunny.
6. Technology/VG-Style: in 1-2 phrases. What type of VG is it? What type of play-area view?
   a. Type EX: RPG, RTS, Board-game, Sim-no-Oppo, FPS, spaceship shooter, etc, etc
   b. View EX: 2D vs 3D, scroller vs still, side vs top vs isometric, level vs endless scroll, etc, etc
7. Aesthetics: What should the game look like? 1-2 sentences **for each** of the following:
   a. Lg: Best super-frills version (not likely to be implemented, but a useful strawman)
   b. Md: Simple nice-frills version
   c. Sm: Spare mini-frills version (the bare minimum you'd be happy with)
8. Rough Artwork: 2-3 rough hand drawings with 1-2 sentence captions **each**. (See lecture pix examples.)
   a. Of main backdrop, or still of action including backdrop
   b. Of key action in game, isolated from backdrop

Game Sample

A 10-30-second rough very-simplified animated sample of the game in action. This game sample must be suitable for demonstration in class to the instructor. It is due 2 weeks after the start of this project.

Steps:

1. Build a Proof-of-Concept doc containing:
   a. A mini-level design.
   b. Describe the key feature you want to focus on in the Sample.
   c. How will it look? Backgroud, foreground objects, avatars, movement, actions/moves
      i. In the rough simplified Sample; ii. In the scaled-up Game
   d. What development steps will you need to build the Sample.
2. Dev Tasks to be addressed:
   a. Rough background (eg, hand-drawn, scanned)
   b. Rough foreground & objects

c. Rough avatars/NPCs: look (eg, maybe only 1 moving limb, etc) (if the game has an avatar/NPC)
d. Rough action: how avatars/NPCs move, interact with the environment
e. How the controls affect the animation

Notes:

1. If your game is not level-oriented, but e.g., a scroller, then show the initial screen-sized exploration or action area that seems roughly equivalent to the detail of a mini-level.
2. For a board-game, the pieces may have little movement on their own.
3. For a puzzle-game, there may be no avatar to speak of.

Weighting of Project Issues

- 60% for a playable game
- 5% for a clear **Rough Concept Spec**
- 15% for a clear understandable **Game Sample**
- 5% for clear **Summary and Design Report**
- 5% for clear well-documented source code.
- 5% for a reasonable README.txt file
- 5% for following the Submission rules

Project Summary and Design Report

A project summary and design report is a short document, as a pdf file, that is an extension of your original game design documentation. It should be included in the zip archive of your submission.

The final project is judged not only on the completeness of the solution but on the quality of the students understanding of the project as demonstrated by a written project summary.

A good summary and design report would cover the following topics:

● **Introduction**

What is your project? Game genre, and sub-genre if applicable. Describe in detail what your project is so a layperson can understand the objective, as well as the utility.

● **Design**

Detail the rules of the game, how it is played, example scenarios. Provide enough detail such that a savvy person can create a software architecture from this information. Factors to consider while writing your game design are:

- Rules
- Sources of uncertainty
- Win state, lose state
- Expected skills the player must have prior to starting the game
- Controls
- Expected duration of a game
- Scoring
- Visual representation of game state

● **Software Architecture Detail**

The software design you used for your program. Provide details such as algorithms used, organization of the software, the classes/objects (each with title, responsibility, and their collaborating classes – think CRC cards), and their major functions/methods, etc.

● **Game Demonstration**

Detail what was implemented and how. Use screen shots to illustrate your results.

● **Bibliography**

Write a bibliography which details the resources you used to understand and accomplish this project. **Give credit where credit is due.**

Project Presentations and Demonstrations

On the last week of class, you will present your Final project to the class. You should come prepared to talk about your final project and demonstrate it. Your presentation should be **no more than 5 minutes long**.

Be prepared with a **thumb (flash) drive and maybe a laptop**. We will hook it up to the projector. Hopefully, the connector will work in the laptop, if used.

Be prepared to answer questions during and after your presentation. During your presentation, it is suggested that the following topics be covered (the timing is a suggestion, only).

- Game Opening Screen: **Show** the game **Title** and **Maker Team Name**
- Brief overview of the game (~ 1/2 minute) – including the game genre (eg, Action Side-Scroller)

- Interesting algorithms, implementation issues (~ 1/2 minute)
- Project demonstration (~ 2 - 3 minutes)
- Thank the audience

Your project must be demonstrated in class during your presentation. Should you require specific technology or materials to accomplish your demonstration please see the instructor to make arrangements.

Academic Rules

Correctly and properly attribute all third party material and references, lest points be taken off.

Submission Rules (for an easy 5%)

Your submission must, at a minimum, include a plain ASCII text file called **`README.txt`** (e.g., title, contact info, files list, installation/run info, bugs remaining, features added) all necessary source files to allow the submission to be built and run independently by the instructor. [For this project, no unusual files are expected.] Note, the instructor doesn't necessarily use your IDE.

All source code files must include a comment header identifying the author, author's contact info (please, no phone numbers), and a brief description of the file.

Do not include any IDE-specific files, object files, binary executables, or other superfluous files.

If you need to include a text file, use either a .txt file, or a .pdf file if formatting is required. Please don't include a .doc or .docx file.

Place your submission files in a **folder named with this format:**

`X-pY_teamname` or `X-pY_lastname-firstinitial`

where **`X`** is the course **number** (e.g., **123** for CS-123 – **BUT NOTE THAT "123" IS NOT YOUR COURSE NUMBER**) and **`Y`** is or the project **number** (eg, 9 for Project #9). For example in CS-**123** for Project #**9**, if you were a 1-person team and named were **Tim Crazy**, then you would use `123-p9_Crazy-T` for the folder name, or if your team members were **L**oest, **A**monge, and **D**eStars, then you would use

`123-p9_LAD`

for the folder name. If you only have a 2-person team, use "X" for the third team letter.

Then zip up this folder. **Name the .zip file the same as the folder name**. Please don't use a .rar extension.

`123-p9_Cruise-T.zip`

Note that if you have "file extensions" hidden on your files browser, you will not see the **.zip** file extension.

The project is due by the time and on the due date specified in the class's bulletin-board project assignment post. Turn in by **sending me email** (see the Syllabus for the correct email address) with the zip file attached.

The email subject title should also include the folder name.

NB, If your emailer will not email a .zip file (eg, because of file size), then change the file extension from .zip to .zap, attach that, and tell me so in the email.

Please include your name and campus ID at the end of the email (because some email addresses don't make this clear) – and if it's a team effort then include everybody on the team.. If there is a problem with your project, don't put it in the email body – instead tell me by putting it in the README.txt file.