### **GAME DESIGN TUTORIALS**

The tutorials that you follow in this course are designed to teach you about the software and procedures for completing game builds. The steps for completing game builds are organized into numbered stages. Sometimes, you will see words from a formal **vocabulary** that you must define. Sometimes, you will have to make written predictions by forming a **hypothesis** statement based on completed steps or stages. Additionally, you will have to write **evaluations** after testing certain parts of your game. These will be done digitally on a separate document called a **tutorial guide**. This will be downloaded from the teacher website, completed, and saved. It will be checked by your teacher visually or uploaded to a teacher website. Text features typical to the tutorials are shown and explained in the list below.

- vocabulary New words will be listed in your tutorial guide. You will be required to show their meaning by adding written definitions for each. When a vocabulary word is first introduced in a tutorial it will be boldfaced in dark red as shown. Definitions and deeper explanations can be found in the Glossary and Concepts Explained.
- **STAGES** Tutorials are written in numbered steps and these steps are grouped in to **stages**. This is done to focus on specific parts of the game programming and the way they work. The stages are identified with numbers or numbers and letters with boldfaced red text (eg. **STAGE 1, STAGE 3a, STAGE 5b,** etc.).
- **HYPOTHESIS STATEMENTS:** Throughout the tutorials, you will find boxes labeled for hypothesis statements. You will be prompted to write about the specific parts of the program completed in the preceding stages. The idea is predict or anticipate the behaviors of the programming, properties, or other features that you create in the tutorial.
- **TEST & EVALUATION:** As you complete stages or full tutorials, you will need to test what you have done to prove if your hypothesis statements are **valid** or **invalid**. If what you see is not what you anticipated, you will need to review the tutorial and fix the problem. You will write about your test of the scripts and/or properties and explain why your hypothesis is valid or invalid. Of course, you should also recall the importance of the **evaluate the solution step** from design process.
- Testing prompts are used to mark the point in a tutorial when it is appropriate to test your game ( ▶ ) and also will also caution you not to test
  This is important to make the process of learning through hypothesis and evaluation more meaningful.
- File naming conventions are used to make the name of your game and other asset files consistent, organized, and easy to recognize. Typically, you will use your name (or initials) and the game name (or an abbreviation) to name and save your game files (eg. *fk\_witchhunt*). Use underscores (\_) instead of spaces when creating file names. This style will also be used to identify game assets that are being used to build your game (*eg. sprite\_happyface, object\_cannon, music\_background, sound\_bell,* etc.). Words like *object* and *sprite* can be abbreviated as *obj* or *spr* as long as this is done consistently. You will be reminded to save your work.
- Specific actions for you to perform, buttons or keys to be pressed are marked with boldface text (eg. Open the *obj\_boss* again and click **Add Event** then **Other Event**). There are also pictures to help you match your actions with the tutorial instructions.
- The **tutorial guide** is the document you must download and complete as you are moving through the tutorial. You will document vocabulary, hypothesis, and evaluation and other writing that goes along with your tutorial. This may be done on paper or digitally. Sometimes, you may be required to submit it for grading by uploading it to a teacher website account or by saving it to a folder on your desktop. Study the sample tutorial guide shown on the next page to see what yours may look like when completed.

# Sample Tutorial Guide:

3.

Note that tutorial guides can have up to ten stages for a game build. This shows a sample of typical entries.

## TUTORIAL GUIDE: GAME DESIGN TUTORIALS (SAMPLE) VOCABULARY LIST

1. vocabulary - New words with written definitions to be listed in your tateniel public

2. stages - The groupe revolved stops and these stops are grouped in to stopes to focus or escelife parts of the gave programming.

4. hypothesis statements - A predictive to articipate the behaviors of the programming properties, or other features.

5. test and evaluation - Written statements explaining why a solution is wall on invalid and what must be done to fix the behavior.

5. testing prompts - Used to mark the point in a tectorial when it is appropriate to test your general also will and certain you not to test

- File naming conventions Crusting the new of your pare and other cost files of they are consistent, organized, and easy to receiptive (op fb mitchburt, carite happyfree effect county music heckground count hell etc.).
- tutorial guide The document gas must doubled and complete as you are moving during the tatonial. It contains oncolulary, hypothesis, analysis, and other uniting that goes along with your tatonial.

### STAGE 1 HYPOTHESIS STATEMENTS:

When I preas the space bar, the front end should disspecar. The beach ball spicet will specar and begin bouncing. It will keep bouncing wildly with I control it by asing the armon keys to nooe it left or night.

### STAGE 1 TEST & EVALUATION:

Although the beach ball did appear, I could only control it and now it to the right. The problem is in the beach ball properties for the left arrow Reyboard event. If I respect those properties, I should be able to edit the section os that the object can be noved both left and right.