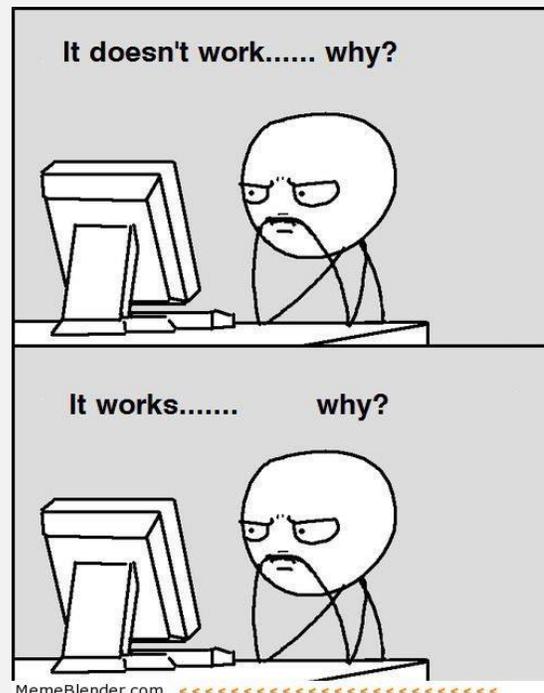


Information Processing

LEARNING GOAL & SCALE: Standard	
4	Student will be successful in level 3 and: Use a high-level language and number/non-numeric data to create a basic game in Unity or Unreal Engine.
3	Student will: Implement enhanced program structures. (standard 47.0) <ul style="list-style-type: none"> ✓ Use iteration- if you fail – try again/find a solution and improve. Change design as you go to improve customer satisfaction ✓ Incorporate “help” text (on title screen or tutorial level) ✓ Create an interactive game ✓ Design Screen layouts for use in interactive programs
2	Student will: Define: variable, iteration, interactive, layout and immersion
1	With help from the teacher, the student has partial success with the current content.

Work Scenario: Create a game (or improve on a past game) that has solid goals, rules and feedback. Use playtesting with other students to ensure your game is fun and has high entertainment value.

Learning Target: Understand how to implement enhanced program structures using code to immerse your players or customers in a gaming experience through solid goals, rules, feedback using iteration.



Game Requirements (Unity or Android Studio- if you prefer a different platform discuss it with me first. Unreal is not an option on school computers at this time):

1. 1 or 2-person team (roles: Programmer and Asset Manager)
2. Recommendations (not required but recommended): Title screen & credit screen, Tutorial level or instructions on menu/title screen, at least 2 playable levels
5. Game must have a solid goal(s), rules that require creativity and/or strategic thinking, and a feedback system (i.e. lives, score, health, points, high score, death, etc.).
6. Cohesive design, consistency, no bugs.

Project Steps:

1. Discuss design with your team member or peers to get ideas.
2. Outline goal, rules, and feedback. a. Once you know your overall genre and gameplay – pick a design theme/style.
3. Asset checklist- detailed list of assets needed to create the game and the sizes you want them to be. This includes characters, backgrounds, buttons, health bars, weapons, etc.
4. Code checklist- detailed list of what code you need to program and find tutorials for if needed. Examples include player movement (get specific- what kind of movement- 4 way, up/down, flight simulator, left/right only), enemy AI, weapon use, health/lives/score, transition between rooms, etc.
5. Build the game!