

# Project 1 - *EVASION*

## *Technical Specifications*

**Micah Hamann (1017044)**

---

[ProjectOneScript → Start\(\)](#)

[ProjectOneScript → Update\(\)](#)

[ProjectOneScript → SpawnEnemy\(\)](#)

[ProjectOneScript → GenerateEnemyStats\(\)](#)

[ProjectOneScript → PlayerAttack\(\)](#)

[ProjectOneScript → PlayerWhack\(\)](#)

[ProjectOneScript → PlayerDodge\(\)](#)

[ProjectOneScript → EnemyAttack\(\)](#)

[ProjectOneScript → EnemyDefeated\(\)](#)

[ProjectOneScript → LevelUp\(\)](#)

[ProjectOneScript → Victory\(\)](#)

[ProjectOneScript → DebugOutAttackStats\(\)](#)

[ProjectOneScript → DebugEnemyAttack\(\)](#)

## ProjectOneScript → Start()

- What are the inputs (variables/data required for this to work)? (if any)
  - Game Text Field
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Game Text Field
  - ResetGameStats()
    - Player stats: LVL, HP, Strength, Evasion, EXP
    - Player character text field
    - Player stat text field
    - Enemy stats: LVL, HP, Strength, Evasion, Potential EXP, Name
    - Enemy Character text field
    - Enemy stats text field
    - Game state booleans:
      - Is player dead
      - Is player turn
      - Is game end
      - Is player dodging/whacking
      - Is player ready for a new enemy
      - Is player ready for LVL up
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Calling ResetGameStats() sets all the necessary variables back to their initial start values so when you restart after winning or dying, the game is as new.
  - Then set the main game text field to the welcome message.

## ProjectOneScript → Update()

- What are the inputs (variables/data required for this to work)? (if any)
  - Game text field
  - Player text fields
    - All player stat variables
  - Enemy text fields
    - All enemy stat variables
  - Enemy HP
  - Game state bools
    - Is player dead?
    - Is player turn?
    - Is game end?
    - Has player won?
    - Is player dodging/whacking?
    - Is player ready for a new enemy?
    - Is player ready for LVL up?
    - Is player's stats increasing?

- Is player gaining EXP?
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Functions called:
    - SpawnEnemy()
    - Default Animation()
    - EnemyAttack()
    - Victory()
    - LevelUp()
    - Start()
    - PlayerAttack()
    - PlayerWhack()
    - PlayerDodge()
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Game will display player/enemy stats and will change depending on the state of the game (i.e if the player is dodging, the evasion stat displayed will temporarily increase). This is done using bools and if statements.
  - Game will check if the player is dead (health < 1) and will change IsPlayerDead bool accordingly.
  - Upon pressing the SPACEBAR key, the game will decide to call the following functions depending on which game state bools are true or false:
    - If an enemy has just been killed or the game has just started, the isReadyForNewFight condition will be true and SPACEBAR will call the SpawnEnemy() function.
      - The game state to changed to show it is the player's turn
    - If the player has just attacked and the enemy is not dead, the SPACEBAR will continue to the enemy attack phase and call the EnemyAttack function.
      - If the enemy is dead, SPACEBAR will call the EnemyDefeated() function.
      - If enemy is still alive the game state changes to Player's turn
    - If a player has reached level 5, this marks the win condition as being true and SPACEBAR will call the Victory() function.
    - If the player's EXP reaches its limit, SPACEBAR will trigger the LevelUp() function
    - If the player's health is less than 1, SPACEBAR will trigger a death message in the player and game text fields.
  - If it is the player's turn (isPlayerTurn = true), pressing the spacebar is disabled and the player can only press A, W or D keys each corresponding to a specific battle option described in the game text field.
    - If player presses A, the PlayerAttack() function is called and change state to Enemy's turn
    - If player presses W, the PlayerWhack() function is called change state to Enemy's turn

- If player presses D, the PlayerDodge() function is called change state to Enemy's turn
- If the player is dead or victorious, all options are disabled besides the ESCAPE key which calls the Start() function.
- The game text field is constantly changing to describe what is happening and what keys the player can press.

## ProjectOneScript → SpawnEnemy()

- What are the inputs (variables/data required for this to work)? (if any)
  - EnemyType = A random integer to determine the type of enemy that is spawned
  - EnemySpawnLevel = A random integer to determine the strength of the enemy that is spawned
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Game text field
  - Enemy Character text field
  - Enemy Name
  - GenerateEnemyStats()
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Two random numbers are inserted into the function using a random number generator function. The 2nd number decides the enemy level, the 1st decides the following:
    - If the random number is 1 or 2 an CROCODILE is spawned of which has weaker stats
      - Change enemy sprite and text to CROCODILE
    - If the random number is 3 or 4 a DINOSAUR enemy is spawned which is a bit stronger
      - Change enemy sprite and text to DINOSAUR
    - If the random number is 5 or 6 a MONSTER enemy is spawned which is stronger
      - Change enemy sprite and text to MONSTER
    - If the random number is 7 or 8 a DEATH enemy is spawned which is the strongest enemy. The random number generator is set up so that 7 and 8 are only possible when the player is LVL 3 or 4.
      - Change enemy sprite and text to DEATH
    - The Enemy sprite and stat display updates accordingly to the type and level of the enemy that is spawned

## ProjectOneScript → GenerateEnemyStats()

- What are the inputs (variables/data required for this to work)? (if any)
  - Enemy Type

- Enemy Spawn Level
  - Boss stat
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Enemy Level
  - Enemy Max HP
  - Enemy Current HP
  - Enemy Strength
  - Enemy Evasion
  - Potential EXP
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Enemy Level updates to match Enemy Spawn Level
  - Enemy Max HP becomes:
    - Player level multiplied by Enemy Level + boss stat
    - This is so the Enemy Max HP scales with the player's level and is beefed up when facing DEATH boss enemy
  - Enemy strength becomes enemy Level plus enemy type
  - Enemy evasion becomes enemy type multiplied by 3 plus a fixed number of 10
  - Potential EXP (this is the EXP given to the player when the enemy is killed) is equal to the player level plus the enemy level both multiplied by 2 and with a random number between 1-4 added. This is to create some variance in EXP.

## ProjectOneScript → PlayerAttack()

- What are the inputs (variables/data required for this to work)? (if any)
  - Player presses A key
  - Random number
  - Current enemy HP
  - Player strength
  - Enemy evasion
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Enemy current HP
  - Game text
  - Player sprite text
  - Enemy sprite text
  - DebugOutAttackStats()
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Game generates a random number between 0 and 100 and if that number is more than enemy evasion stat, the enemy loses health equal to player's strength.
  - Debug log shows what attack option player chose and the resulting damage done to the enemy.
  - Player sprite text changed to attack sprite.
  - Game checks to see if the enemy is still alive. If not, game text and enemy sprite is changed to show that the enemy has been slain.

- If the random number is less than the evasion, it is considered a miss and no damage is dealt.

## ProjectOneScript → PlayerWhack()

- What are the inputs (variables/data required for this to work)? (if any)
  - Player presses W key
  - Random number
  - Current enemy HP
  - Player strength
  - Enemy evasion
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Enemy current HP
  - Game text
  - Player sprite text
  - Enemy sprite text
  - DebugOutAttackStats()
  - Is whacking bool
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Game state changes to show that player chose to Whack which temporarily changes player's strength (x2) and enemy evasion (+20).
  - If the random number is less than the evasion, it is considered a miss and no damage is dealt.
  - Game generates a random number between 0 and 100 and if that number is more than enemy evasion stat (+ 20), the enemy loses health equal to **x2** player's strength.
  - Debug log shows what attack option the player chose and the resulting damage done to the enemy.
  - Player sprite text changed to WHACK sprite.
  - Game checks to see if the enemy is still alive. If not, game text and enemy sprite is changed to show that the enemy has been slain.

## ProjectOneScript → PlayerDodge()

- What are the inputs (variables/data required for this to work)? (if any)
  - Player presses D key
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Game Text
  - Player sprite text
  - Is Dodging bool

- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Game state changes to show that the player is dodging. This temporarily changes the stat text to show a 40 increase in player evasion and a x1.5 increase in enemy strength.
  - Debug log shows what attack option the player chose and the resulting damage done to the enemy (which when dodging is always 0 damage)
  - Game text shows that the player chose to dodge
  - Player sprite text changes to show dodge sprite

## ProjectOneScript → EnemyAttack()

- What are the inputs (variables/data required for this to work)? (if any)
  - SPACEBAR input from player
  - Is enemy turn bool
  - Is player dodging
  - Is player whacking
  - Random number
  - Player evasion
  - Current player HP
  - Enemy strength
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Current player HP
  - Game text
  - Player sprite text
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Game checks to see if player is dodging
  - If not, a random number is generated and if it is above the Player's evasion stat:
    - The value of enemy strength is subtracted from the player's current HP. The player sprite text is changed to the hit sprite. Swap to player's turn
    - Else, the enemy misses and no damage is dealt and swap game state to Player's turn
  - If a player is dodging, the same applies except a value of 40 is added to the player's evasion when checking if the enemy hits. If it is a hit:
    - The value of enemy strength multiplied by 1.5 (rounded and converted back to integer) is subtracted from the player's current HP. Swap to player's turn.
  - The game then generates a new evasion stat for the enemy between a value of 5 and the enemy's base evasion plus 30.
  - Debug log shows whether or not the enemy hit and how much damage was dealt.

- Game checks if the player's health is  $< 1$ . If so, game text shows that the player "has fallen to the ground" and makes sure the game state does not swap to the player's turn.

## ProjectOneScript → EnemyDefeated()

- What are the inputs (variables/data required for this to work)? (if any)
  - Current player EXP
  - Potential EXP
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Current player EXP
  - Is level up bool
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - The enemy's potential EXP is added to the player's current EXP
  - Game checks to see if current EXP is equal to or more than their max EXP.
    - If so, the level up bool is changed to true.
    - Else, level up remains false

## ProjectOneScript → LevelUp()

- What are the inputs (variables/data required for this to work)? (if any)
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Game text
  - Player sprite text field
  - Player Level
  - Player Strength
  - Player evasion
  - Current EXP
  - Max EXP
  - Player current Health
  - Player Max health
  - Enemy sprite text
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Change stat increase to true (this tells the game in the Update function to display the follow stat increases in green text)
  - Update player sprite to default
  - Player Level increases by 1
  - Player strength is multiplied by 1.75 which is then converted to an integer. Also add 1 more to stat.
  - Player evasion increases by a fixed value of 5



- Max HP increased by 10 and current HP value is reset to equal Max HP. This brings the player back to full health.
- Set enemy sprite to empty
- Game text displays a level up message to the player.
- Game checks to see if player is at level 5, if so, the win condition is set to true
- Debug out all the stat changes as a result of the level up.

## ProjectOneScript → Victory()

- What are the inputs (variables/data required for this to work)? (if any)
  - SPACEBAR input
  - Win condition is true
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Game text
  - Player sprite text
  - Enemy sprite text
  - Is game end bool
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - The game text is changed to a congratulations message
  - Player sprite is changed to the victory sprite
  - Enemy sprite set to empty
  - Set the “is Game End” condition to true.
    - This allows the player to press the ESCAPE key to restart the game as stated in the Update function

## ProjectOneScript → DebugOutAttackStats()

- What are the inputs (variables/data required for this to work)? (if any)
  - Attack Type (string)
  - Buff multiplier
  - Hit or miss (string)
  - Damage multiplier
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Debug Log
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - When this function is called after each attack type and condition, I manually fed in the attack type, whether or not the attack hit or miss, and the temporary stat changes.
  - Debug log then shows the attack type and the damage done by the player

## ProjectOneScript → DebugEnemyAttack()

- What are the inputs (variables/data required for this to work)? (if any)
  - Hit or miss (string)
  - Temporary stat multiplier
- What are the outputs(variables/data changed or functions called by this running)? (if any)
  - Debug Log
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
  - Debug log shows whether or not the enemy hits (or misses) the player and the damage done to the player.