

# A3 Developer's manual

Corey Horsburgh AU 3487170

## Code Explanations/Descriptions

### Main.js

- **Loader:** Loads all of the images that will be used in the game. One of the default Pixijs functions.
- **setup():** This function sets up all the objects for the game, adds them to applicable arrays and also sets up the keyboard to take in input. Most variables are also given starting values here and the game is set to start on the title screen once this function is over. This function is only run once when the game is first loaded.
- **gameLoop():** This function tracks the current state that the game is in (title, menu, play or end).
- **play():** This function is for when the game is in the play state. It handles many things such as player actions and interactions with objects, as well as calling many other functions for different objects when appropriate. It also handles tracking the time of the game and score values. It will set the state of the game to "end" once the timer hits 0. The other functions it invokes are `hitTestRectangle()` for testing collision between many objects, `enemiesMove()` to move the enemies, `enemyBoundsCheck()` to keep the enemies in the play area, `updateEnemies()` to update enemy values like health and status and `spawnItem()` and `spawnEnemies()` for spawning items and enemies respectively.
- **end():** This function is for when a game ends but is not on the main menu. It stops the player and enemies from doing anything and gives the player the options to replay with the same settings or return to the main menu.
- **titleScreen():** This function is for when the game is in the title state. It hides all objects except the background and makes the background a button. Clicking it brings you to the main menu.
- **menu():** This function is for the main menu state of the game. It hides game objects like the character and enemies and makes the menu objects visible. It also makes certain menu objects buttons to select things like characters, maps, time limits and the play button which calls the function to start a match.
- **onMenuClick():** Sets the state to "menu."
- **characterSelectX():** There are three of these, one for each character. They set the character stats and sprite to the appropriate values. They also change the location of an arrow that indicates which character is selected.
- **mapXSelect():** There are four of these and set which map will be played on. They also change the button's sprites to indicate which is selected.
- **XMinuteSelect():** There are also four of these and they set the time limit for matches. They also change the button's sprites to indicate which is selected.
- **musicClicked()** and **soundClicked():** These mute and unmute the music and sound respectively. They also change the appropriate button sprites to reflect if they are muted or not.
- **pauseClicked():** This will pause and un-pause the game, and make an icon appear to indicate when it is paused.

- `hitTestRectangle(r1, r2)`: This checks if two sprites are touching or not. It uses this by taking their total widths and heights and seeing if they overlap. It takes in two Pixijs "Sprite" objects as its arguments and returns a Boolean of whether or not they are touching.
- `keyboard(keycode)`: Handles keyboard input for key down and key up events. It takes in the keycode for a key on the keyboard and makes a key object that contains info on if the key is up or down, if it has been pressed or released. It returns this key object.
- `enemiesMove(slime)`: Makes the enemies move. They will choose a random number between 0 and 4 and will move in a certain direction based on this number. It takes in a "Slime" object to make move.
- `enemyBoundsCheck(slime)`: Checks for enemies to make sure they do not go off screen or inside of stage objects. Uses their position and height and widths to determine if they are at the edge of the screen or touching a stage objects by calling `hitTestRectangle()`. If they are it calls `enemiesMove(slime)` to make them choose a new direction to go in. It takes in a "Slime" object to check its location.
- `updateEnemy(slime, player)`: Updates various values of the enemies including if they are being hurt by the player, their health, if they are dead or not and respawning them if they are dead. It takes in a "Slime" object to update, as well as the "Player" object for checking how much damage they did to the slime (if they attacked it). It uses `hitTestRectangle()` to determine if the player hit the slime with an attack. When a slime respawns it calls `enemiesMove()` to get it moving again.
- `startLevel()`: This is called whenever a new match is started. It hides all the menu objects and makes the player and any stage specific objects visible. It also makes all enemies on the chosen stage visible and will spawn 5 random enemies from the list for that map in random locations to start. It uses `spawnEnemies()` to do this and `hitTestRectangle()` to make sure it spawns the initial 5 enemies in valid locations (not in stage objects). It then calls `updateEnemy()` to make sure the enemies know their appropriate status and information. At the end it sets the game's state to "play."
- `spawnItem()`: Spawns a random item from the list to a random location on the map. Makes sure that the item doesn't spawn in a location the player cannot get it. Also resets the timer for collecting items. It uses `hitTestRectangle()` to make sure that items are spawned in valid locations.
- `spawnEnemies()`: Chooses a single random enemy from the list of available enemies for the current map to spawn. It will go through the list until it selects an enemy that has not been spawned yet. If the maximum number of enemies for the map are already spawned it will not do anything. Also makes sure to not spawn enemies in stage objects like blocks or lava using `hitTestRectangle()`. It also calls `updateEnemy()` for the enemy it spawned.