**Due Date**

This assignment must be completed *prior to the start of the lab during Week 4*. This assignment will be graded during the lab so be prepared to demonstrate your work to the instructor and/or TA.

### Description

This assignment involves building a simple side-scrolling physics game that demonstrates your understanding of the concepts covered in Snippets 1 through 18, including physics effects, text, levels and tile sets.

Your grade will be determined by how well your project meets the requirements specified below.

***Game States:***

* Implement the following game states using the “Levels” concepts covered in Lesson 16:
  + Start Screen
  + Win Screen
  + Lose Screen
  + Gameplay
* NOTE: The code for getting the name of the level, as shown in Lesson 16 is incorrect. Instead, you may use the following:
  + currentLevelName = self.Space.CurrentLevel.Name

***Start Screen:***

* Using SpriteText objects, display the following text:
  + A game title for your project
  + A start message appropriate for your project. For example:
    - “Press SPACE to begin”
* When the specified key is pressed, change to the Gameplay level.

***Win Screen:***

* Using SpriteText objects, display the following text:
  + A “win” message appropriate for your project.
  + A resume message appropriate for your project. For example:
    - “Press BACKSPACE to continue”
* When the specified key is pressed, change to the Start Screen level.

***Lose Screen:***

* Using SpriteText objects, display the following text:
  + A “lose” message appropriate for your project.
  + A resume message appropriate for your project. For example:
    - “Press BACKSPACE to continue”
* When the specified key is pressed, change to the Start Screen level.

***Gameplay Environment:***

* Using the Tile Editor, as discussed in Lesson 18, create an interesting side-scroller environment using tiles that represent ceilings, floors and walls.
* Create ten or more regions with physics effects within the environment. Demonstrate three or more distinct types of physics effects using these regions. For example:
  + WindEffect
  + VortexEffect
  + FlowEffect
  + PointGravityEffect
* Create five or more regions representing hazards within the environment.
  + Load the Lose Screen level when the player enters a hazard region.
* Create a single region representing an “exit” from the level.
  + Load the Win Screen level when the player enters the “exit” region.
* Position the required regions in such a manner that the player is required to exercise at least a minimal level of skill in order to successfully navigate to the “exit”.
* For grading purposes, it should be possible to complete the entire level in less than 60 seconds.

***Player Object:***

* Create an image file to represent the player.
  + This image file must contain two or more frames of animation.
  + The type of object represented by this image is at the discretion of the student.
  + For the purposes of this assignment, the artistic quality of the image is not important. However, each frame of animation should be distinct.
* Implement a sprite using the image file as a SpriteSource.
* Implement a force-based movement controller using the concepts covered in Lesson 13.
* Set the starting position so as to require the player to navigate through the environment in order to reach the “exit”.
* The player should be unable to leave the gameplay environment, as defined by the floors, walls, ceilings and hazards.

***Camera:***

* Implement a side-scrolling camera controller that follows the player.
  + The camera’s Y position should be either fixed or constrained so that the player cannot see beyond the top and bottom edges of the gameplay environment.
* Optional: Additional camera controller functionality is permitted, as long as the player remains visible at all times.

***Heads-Up Display (HUD):***

* Implement a HUD layer with at least one of the following elements:
  + A text object displaying information about the game, such as a current score, current speed, or the elapsed time since the beginning of the game.
  + A sprite object displaying information about the game, such as the player’s current status, current heading, etc.

**Build Requirements**

The project must build cleanly, with no errors or warnings.

**Assignment Grading Guidelines**

* Projects will be graded in class on the day that they are due.
* Use the “Export” feature of the Zero Engine to create a stand-alone executable.
* Projects that are modified ***after the start of class*** will be considered late and automatically receive a grade penalty of -25%.
* Projects that have not been graded within one week after they are due will automatically receive a grade of 0% (F).