Flyer Game Project Rubric

Key Concept	Extensive Evidence	Convincing Evidence	Limited Evidence	No Evidence
Program Development - Project Guide	The project guide is complete and reflects the project as submitted.	The project guide is mostly complete and generally reflects the submitted project.	The project guide is filled out but is not complete or does not reflect the submitted project.	The project guide is incomplete or missing.
Program Development - Program Sequence	You *sequenced the program well*[1] and properly separated *code in and out of the draw loop*[2].	You properly separated code in and out of the draw loop; however, you have a few incorrectly sequenced lines of code, which resulted in a few elements being hidden behind others unintentionally.	You have several sequencing errors, resulting in many unintentionally hidden or overlapping elements. Some code is improperly placed in or out of the draw loop	Errors in program sequencing are significant enough to keep the output from resembling the intended scene, or the Draw loop is not used to create animation
Modularity - Multiple Sprites	At least 4 *sprites are created and their animations are set properly*[3] . The velocities of at least 2 obstacle sprites are properly set outside the draw loop.	At least 3 sprites are created and their animations are set properly. The velocity of at least 1 obstacle sprite is properly set outside the draw loop.	At least 2 sprites were created and their animations were set. There are no velocities for obstacles set properly outside the draw loop.	Either the program only contains the "player" sprite provided by the starter code, or the sprites are not properly created.
Algorithms and Control - Player Control Conditionals	Your program has *conditionals to respond to multiple types of user input that control the player sprite's movement*[4] inside the draw loop, which causes the sprite to *respond as expected*[5].	Your program has conditionals to respond to multiple types of user input that are meant to control the player sprite's movement[4], however the sprite does not *respond as expected*[5].	Your program responds to at least one user input key but does not have the additional conditionals to control the sprite movement.	Your program does not have conditionals to respond to user input.
Algorithms and Control - Looping Conditionals	Your program has *conditionals to control the obstacle sprites' looping behavior*[6] inside the draw loop.	Your program has conditionals in the draw loop meant to control the obstacle sprites' looping behavior; however, one of the obstacles does not behave as expected.	Your program either does not have all conditionals in the draw loop needed to control the obstacle sprites' looping behavior or has required conditionals, but none of them work as expected.	Your program does not use any conditionals to control looping behavior.
Algorithms and Control - Interaction Conditionals & Collision Detection	Your program has a *conditional and at least 2 collision detection blocks to control multiple sprite interactions*[7] with the player inside the draw loop.	Your program has a conditional and at least 1 collision detection block in the draw loop meant to control multiple sprite interactions with the player; however, one of them does not work as expected.	Your program either does not have a conditional in the draw loop required to control player interactions with the target or has the required conditional, but it does not work as expected.	Your program does not use any conditionals to control sprite interactions.
Optional "Stretch" Feature				
Variables	Your program properly creates at least 2 non-sprite variables (such as *score* or *health*), updates them during the game, and properly displays the variable totals on the screen.	Your program properly creates at least 1 non-sprite variable, updates it during the game, and properly displays the variable total on the screen.	Your program properly creates at least 1 non-sprite variable and updates it during the game but does not properly display the variable total on the screen.	Your program does not create or update any non-sprite variables.

The below explanations and examples give criteria guidance for students choosing to recreate the sample Flyer Game.

1. Sequenced the program well

- If the program code is not sequenced correctly, some elements, such as shapes, sprites, and text, may be unintentionally hidden behind others.
- In the specific case of sprites, if the `drawSprites()` code is not sequenced correctly, some or all sprites may not appear on the screen.

2. Code in and out of the draw loop

- Code outside the draw loop is used to set up the program and its starting elements (creating sprites, setting starting properties, sprite velocities that won't change, etc).
- Code inside the draw loop is for things that are changing as the program is running, such as user interaction. This also includes updating properties or non-sprite variables as well as any shapes for the background, text, and the `drawSprites()` block.

3. Sprites are created and their animations are set properly:

- Sprites are created and their animations are set with these two lines of code **outside** of the draw loop:

```
var sprite = createSprite(200, 200);
sprite.setAnimation("animation_1");
```

4. Conditionals to respond to multiple types of user input that control the player sprite's movement

- This includes the use of 3 conditionals that use different keyboard input (left arrow and right arrow) and/or mouse input.

5. Respond as expected

- The sprite's movement should simulate the player going up when the "up" key is pressed, followed by the sprite accelerating falling back down the screen on its own. The sprite should also move left and right when those keys are pressed..

6. Conditionals to control the obstacle sprites' looping behavior

- This includes the use of at least 1 conditional per obstacle sprite to control the looping on both the x and y axis'.

7. Conditional and at least 2 collision detection blocks to control multiple sprite interactions

- This includes the use of at least 1 conditional that checks if the player sprite is touching the target sprite and then moves the target to a random location
- This also includes the use of at least two collision detection blocks, such as `collide()` or `bounceOff()`, to have the obstacles interact with the player sprite