- **Ideas and the Requirements They Might Fill:**

  Candy crush-ish game:
  -> user input (you have to decide which rows to shift)
  -> graphics/images (whatever the columns of objects you are shifting are)
  ->start screen
  ->small enough window
  -> collectibles; (we could include the extra object things that delete entire rows or things that are the same colors)
  -> timer
  -> intersession progress (save high scores)
    - If we wanted to add more, we could have players advance through different stages/themes and play a mini-game if they passed a stage
      * a pacman game that would have enemies that could kill pacman or that pacman would need to eat

  * We could also do something where we included animations (the sprite sheet stuff doesn't look too hard and candy crush does little celebration things when you clear rows so it would be easy to do that)

  Supermario-ish game:
  ->user input
  ->graphics
  ->start screen
  ->small window
  ->collectibles
  -> intersession progress
  ->enemies
  ->save score
  -> save points
  ->health meter
  → etc.

  Save the Planet from Dying Game:
  Have a map of the world that is getting infected by a disease and have the player have to combat it. Have an avatar that needs to play a mini game to get the cure (have the cure be a collectible)

  -> user input
  ->graphics

->start screen
-> small window
-> collectibles
-> Idk...would need to be more thought out….

AVAILABLE SPRITE SHEETS:

Requirements:

User Input

Either through the keyboard or mouse, you should have appropriate and working user controls.

Graphics/Images

You should use some appropriate images in your game.

Start Screen

Game has a start screen with game name, student names (and IDs), and basic game instructions.

Small Enough Window

Some TAs don't have large screens; your game window may not be larger than `gamebox.Camera(800, 600)` (it may be smaller…)

Optional:

You must include at least four of the following:

Animation

Use a sprite sheet to have an animated character.

Enemies

Have characters that can hinder the player character from accomplishing the goal.

For full credit, enemies must move.

Collectibles

Add collectables (i.e. coins) to the level that can be picked up by the character with a counter that appears on the screen.

We typically get many “does *X* count as a collectible?” questions. Standard collectibles exist in the environment, vanish when you touch them, and give you some benefit from collecting them.

Scrolling level

Make the level much larger than the screen (You may need to add a background image to make this more obvious.)

Timer

Have a countdown (or count up) timer for your game. For full points, this needs to make some kind of sense within the context of the game itself.

Health meter

Have a health meter that changes as you hit enemies/obstacles. It’s only a *health* meter if it getting to 0 results in loss of game (or re-spawning at a save point).

Two players simultaneously

Two players who are able to interact with one another within the game.

Multiple levels

Make several distinct levels that the player can advance through. If you do this, make it clear that there are several levels so that your code coach knows to try to reach more than one.

Save points

When the player reaches some milestone, future failures cause the player to respawn at that save point instead of having to start over.

Inter-session progress

Have your game write some information to a file when it closes, and use that to update the next play. For example, you could have a cumulative high-score chart; a

“save game” option; or even log what the player does and make “instant replays” or the like.

Something More

Want to add another feature, not listed above, and have it count? Describe it in your game checkpoints and see if your code coach thinks it is worth points! Note, it will typically need to be at least as programming-complicated as the examples above…