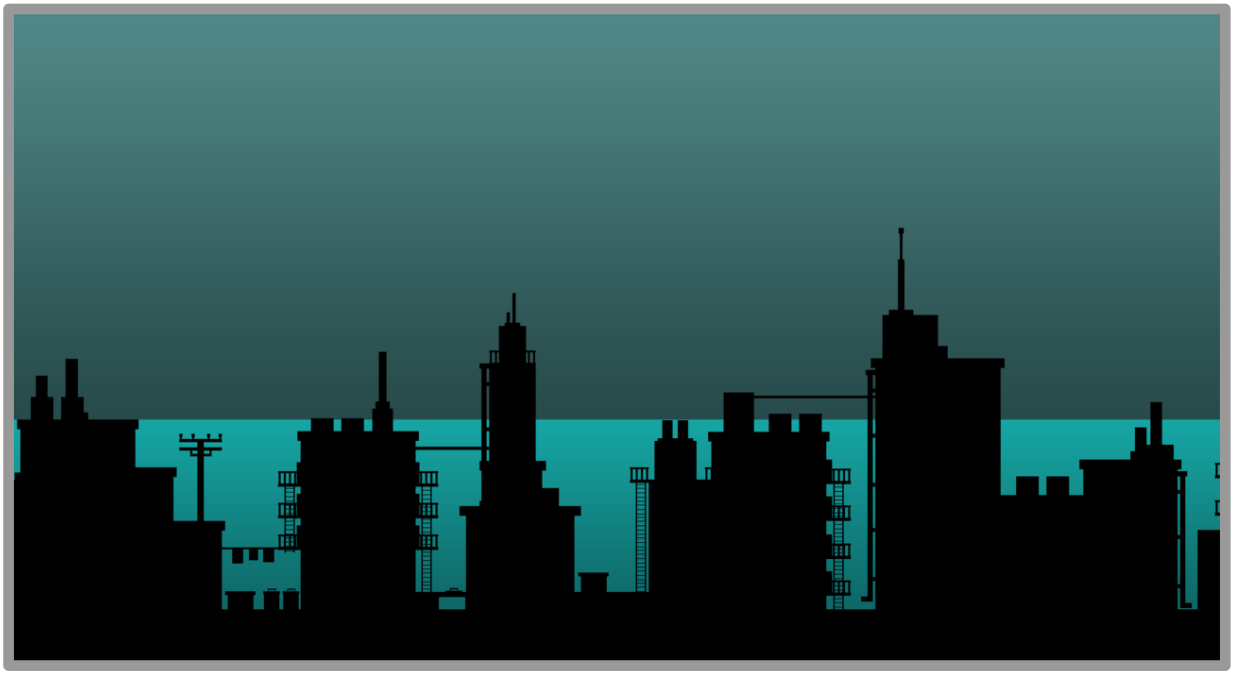
JETPACK ADVENTURE

INITIAL DESIGN

A python game designed and created by Alden Shin-Culhane and Nathan Souphanthong



Program Overview

Jetpack Adventure is a side-scroller, collision style game inspired on the popular mobile app "Jetpack Joyride". The objective of our game is for the user to navigate through an infinite length level controlling a character named "Jooster", while dodging obstacles like lasers (stationary/mobile), and missiles along the way. The character is always in motion, moving forward, but the user can use the spacebar to control a jetpack that moves the character upwards. When the jetpack is not in use, gravity naturally brings the character down. The final goal of Jetpack Adventure is to reach the highest possible score, which is measured by how many obstacles the user clears.

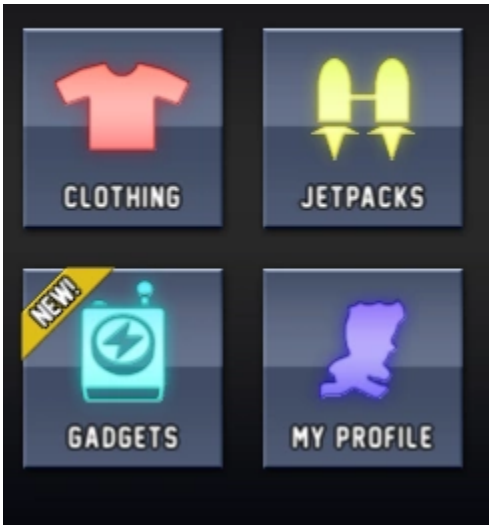
While the user controls the character and avoids the obstacles, coins and vehicle power ups spawn randomly throughout the level at different distances. Coins are saved until they're spent and allow the user to purchase clothing and gadgets (e.g. t-shirts, head starts) that enhance the user's experience. These items can be purchased in a shop on the home screen. Vehicle power ups spawn randomly after a certain distance has been reached, they last for a temporary amount of time, they give the character movement advantages, and they make the character invincible for the duration of the vehicle (when an obstacle is hit while in a vehicle is in use, the vehicle breaks but the character doesn't die; the game doesn't end). As the game progresses, the speed at which the character moves increases, making it more challenging to dodge obstacles.

Screen Layouts and User Input

- The character is continuously moving forward. The score is represented at the top left corner, as well as coins collected
- The missle located in the middle of the screen is one of the many obstacles the user has to avoid when playing the game

- When spacebar is pressed, the character moves upward.
- If the user were to release space bar the character would fall down to the ground (diagram 1)
- Stationary laser shown at the top right of the screen (differs in length throughout the game)

- Character is seen collecting coins that spawn in different patterns during the game
- When the headstart gadget is used, it gives the character a 5000 unit head start, the character is invincible and quickly travels to 5000 units without being affected by obstacles



- In the home screen there is a button to buy gadgets, clothing, and change the overall aesthetics of the character

- Character must touch the vehicle power up to receive a randomized powerup that can be used until the character comes into contact with obstacles
- When a block is hit, the vehicle is deployed and replaces the player model

Description of Key Variables

Variable and Type	Functions
Var playerX: int	Location of player on the x axis
Var playerY: int	Location of player on the y axis If user pressed spacebar: playerY -= 1
Var shirt: string	Changes the colour of the character's shirt
Var pants: string	Changes the colour of the character's pants
Var shoes: string	Changes the colour of the character's shoes
Var jetpack: string	Changes the colour and shape of the character's jetpack
Var gravity: int	Constantly brings the character down
Var speed: float	The speed at which the character is travelling, increases over time
Var bird: string	Draws a temporary bird vehicle that gives a different look, different movement, and one extra life while it lasts
Var motorcycle: string	Draws a temporary motorcycle vehicle that gives a different look, different movement, and one extra life while it lasts
Var gravitySuit: string	Draws a temporary gravitySuit vehicle that gives a different look, different movement, and one extra life while it lasts

List obsctacleX[ ]	Sets a list for where the obstacles are being generated
List obsctacleY[ ]	Sets a list for where the obstacles are being generated
List obsctacleVisible[ ]	Sets an obstacles to being visible or invisible depending on if it's been passed or not
Var keys: string	Collects user input
Var fps: int	Sets the fps for the game
Var elapsed: float	Sets the time elapsed in the game
List coinCount[ ]	The number of coins the user accumulates
Var offscreen: int	The score the user gets from the number of obstacles that they pass

Program Plan/Pseudocode

```
'''
--------------------------------------------------------------------------------------------
Author:      Alden Shin-Culhane and Nathan Souphanthong
Created:     December 13, 2017
Title:       Final Project.py
--------------------------------------------------------------------------------------------
------------------------------------------------------------
      WELCOME TO JETPACK ADVENTURE
------------------------------------------------------------
'''
Import pygame and all other libraries needed (random, math, time)

Set screen

Define Colours & fonts

Set music
Start music

Set sounds

# Start functions
```

```
Define menu:
    while menu is True:
        Draw background
        Get user input for when they click on a button
        Set and draw buttons they can press
            Break out of menu when they press a button
        Draw text
        Update display
        If “shop” is pressed:
            Call shop function
        Elif “gadgets” is pressed:
            Call gadget function
        Elif “start” is pressed
            Call redrawGameWindow function


Define shop:
    while shop is True:
        Draw background
        Get user input for when they click on a button
        Set and draw buttons they can press
            Break out of shop when they press a button
        Draw text
        Update display
        If “clothes” is pressed:
            Call clothes function
        Elif “jetpacks” is pressed:
            Call jetpack function
        If “back” is pressed:
            Call menu function

Define clothes:
    while clothes is True:
        Draw background
        Get user input for when they click on a button
        Set and draw buttons they can press
            Break out of clothes when they press the back button
        If they buy a shirt:
            Change the character’s shirt colour
            Subtract the number of coins it costs from their balance
        Elif they buy pants:
            Change the character’s pants colour
```

```
                Subtract the number of coins it costs from their balance
            Elif they buy shoes:
                Change the character's shoe colour
                Subtract the number of coins it costs from their balance
            Draw text
            Update display
            If "back" is pressed:
                Call shop function

Define jetpack:
    while jetpack is True:
        Draw background
        Get user input for when they click on a button
        Set and draw buttons they can press
            Break out of jetpack when they press the back button
        If they buy a jetpack:
            Change the character's Jetpack shape and colour
            Subtract the number of coins it costs from their balance
        Draw text
        Update display
        If "back" is pressed:
            Call menu function

Define gadgets:
    while gadgets is True:
        Draw background
        Get user input for when they click on a button
        Set and draw buttons they can press
            Break out of gadgets when they press the back button
        If they buy a headstart:
            Make the character's X coordinate start at 5000 they next time they play
            Subtract the number of coins it costs from their balance
        If they buy an extra life:
            Don't break of game loop the first time an obstacle is hit
            Subtract the number of coins it costs from their balance
        Draw text
        Update display
        If "back" is pressed:
            Call shop function

Define bird vehicle:
    Replace the character model with a bigger bird looking vehicle
```

```
        Change movement
        Change user input
        Last for a temporary amount of time
        If an obstacle is hit while the bird's duration isn't up:
            Break out of bird vehicle

Define motorcycle vehicle:
        Replace the character model with a bigger motorcycle vehicle
        Change movement
        Change user input
        Last for a temporary amount of time
        If an obstacle is hit while the motorcycle's duration isn't up:
            Break out of motorcycle vehicle

Define gravitySuit vehicle:
        Replace the character model with a gravitySuit
        Change movement
        Change user input
        Last for a temporary amount of time
        If an obstacle is hit while the gravitySuit's duration isn't up:
            Break out of gravitySuit vehicle


Define end screen:
        while end screen is True:
            Draw background
            Draw Game Over message (text)
            Show score
            Show time played
            Update display

Define messages:
        Start timer
        Star score counter
        Draw them on screen in the top left corner

Define redrawGameWindow:
        Set gravity
        Draw scrolling background
        Draw the player
        If obstacles are visible:
            Draw obstacles
```

```
        If an obstacle is off screen, make it invisible
        Call draw messages
        Display update
        Time delay

# Main program

Declare Variables

Set lists

Set time

Generate obstacles

Randomly create obstacles in a large range
        Create obstacles with an X from 0 - a large number
        Create obstacles with a Y from the bottom to the top of the screen
        Create obstacles with a random size
        Obstacle visibility is set to True
If the user's score equals 50, speed up the random generation of obstacles

Randomly generate coins in a large range
        Create coins with an X from 0 - a large number
        Create coins with a Y from the bottom to the top of the screen
        Coin visibility is set to true
        If a coin is hit:
                Add 1 to the player's coin balance

Randomly generate vehicle blocks less frequently than everything else
        Create vehicle blocks with an X from 0 - a large number
        Create vehicle blocks with a Y from the bottom to the top of the screen
        Vehicle visibility is set to True
        If a vehicle block is hit:
                Call appropriate vehicle function

inPlay is set to True
while inPlay:

        Call menu function

        Set fps
```

```
        Start timer

        Get user input

        If ESC is pressed:
                inPlay is set to False

        If spacebar is pressed:
                The character's Y -= 1

        If the character hits an obstacle:
                inPlay is set to False

        If the player's score is 50:
                Player's speed increases

        Elif the player's score is 100:
                Player's speed increases

        Elif the player's score is 150:
                Player's speed increases

        Elif the player's score is 200:
                Player's speed increases

        Elif the player's score is 250:
                Player's speed increases

        Elif the player's score is 300:
                Player's speed increases

        If an obstacle goes off screen:
                Score += 1

        (etc. for every multiple of 50 score the player reaches)

while True:
        Stop music
        Play sounds
        Call end screen function
```

Quit pygame

Summarize game
Print score
Print time played

Other ideas

- Make the menu screen and end screen have an animated character moving around for aesthetics
- Make a sidekick that follows the character during the game (only purpose is to collect coins, is not affected by obstacles)
- Change the background once the character reaches a certain score/time played/distance
- Add a slot machine that has a chance of giving you gadgets or nothing, costs coins to play
- Add more gadgets