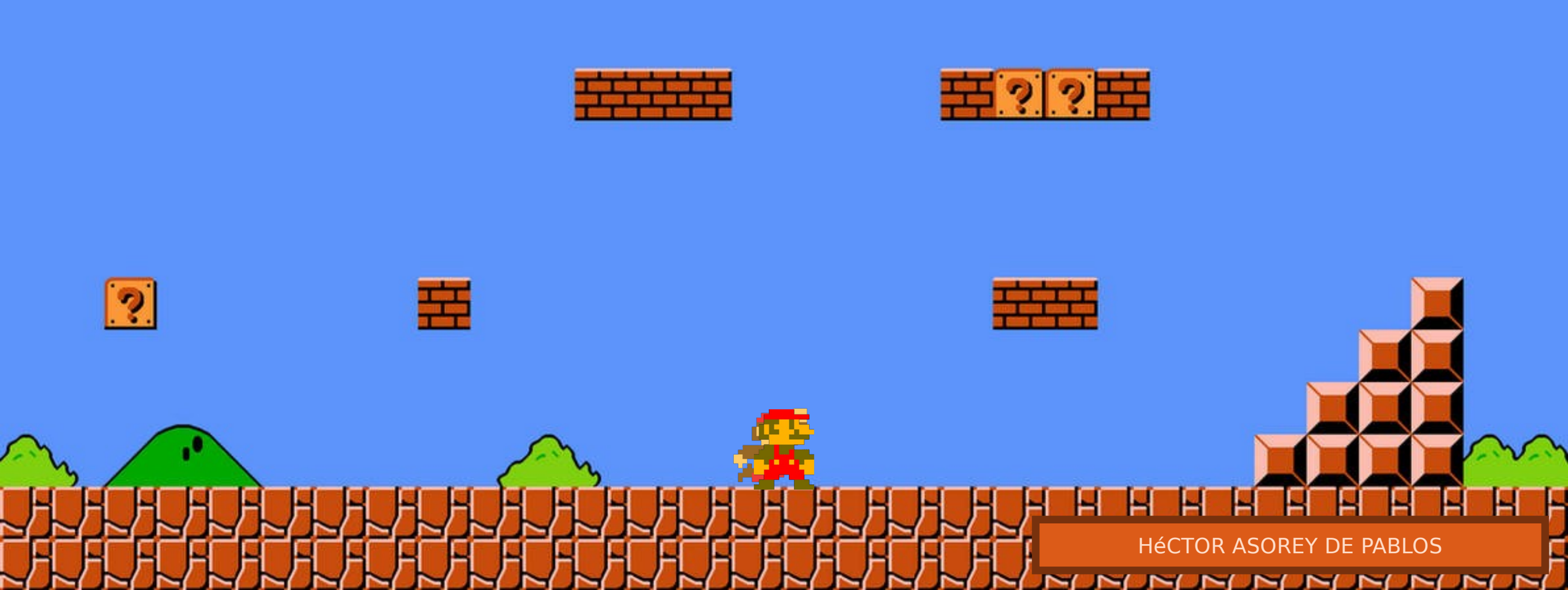
APRENDIZAJE POR REFUERZO PARA RESOLVER EL JUEGO “SUPER MARIO BROS”



HéCTOR ASOREY DE PABLOS

DESCRIPCIÓN DEL JUEGO

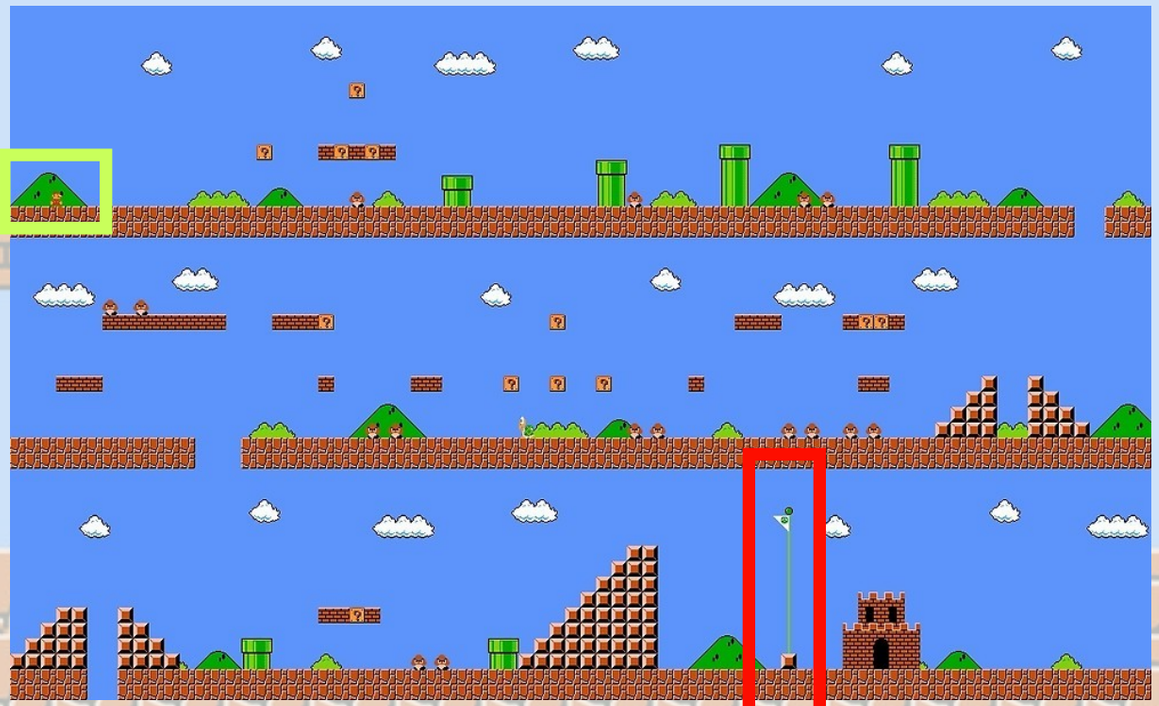
- El objetivo del juego es llevar a Mario a la bandera final para completar el nivel

- Para ello, Mario debe esquivar, saltar o derrotar distintos obstáculos

- Mario obtiene puntuación al derrotar enemigos, recoger monedas y al terminar el nivel

Inicio del nivel

Final del nivel



ACCIONES DE MARIO

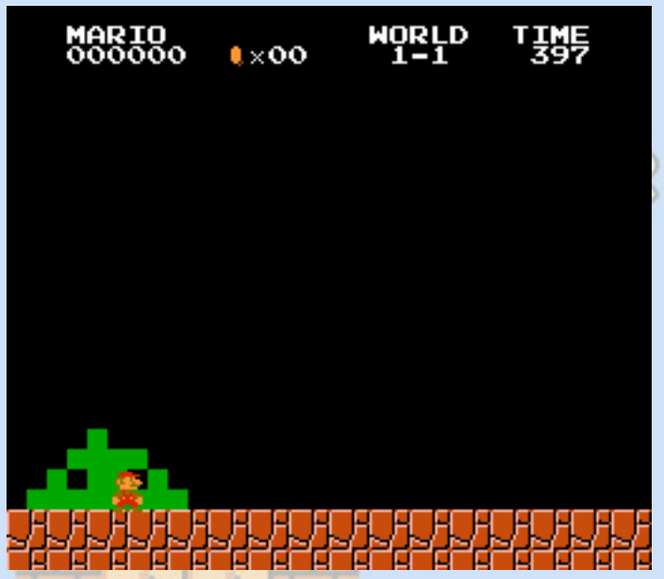
- Mario puede andar o correr a izquierda o derecha, y saltar, como acciones básicas. En caso de tener algún Power-Up, puede por ejemplo, lanzar bolas de fuego.

- En el entorno “gym_super_mario_bros” podemos limitar las acciones que puede tomar el agente. Así, gracias al uso de JoypadSpace, haremos que Mario sólo pueda desplazarse hacia la derecha, puesto que Mario debe avanzar hacia la derecha hasta la bandera para finalizar el nivel.

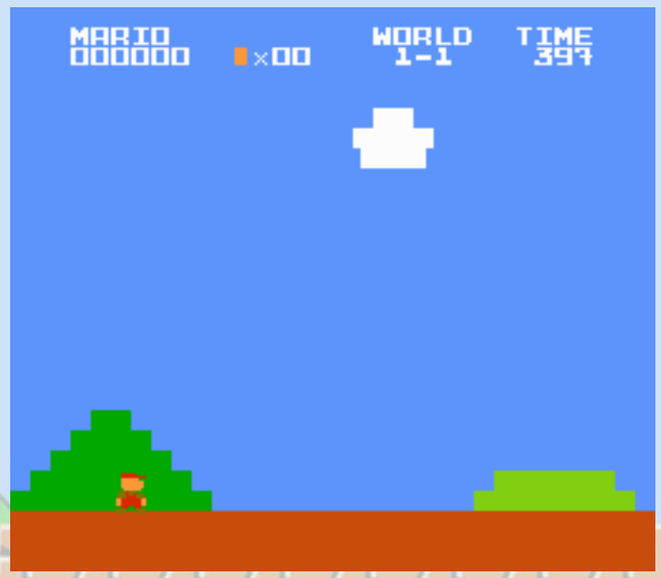
OPCIONES DEL ENTORNO

- “gym_super_mario_bros” ofrece el entorno del juego con diferentes preprocesos hechos.

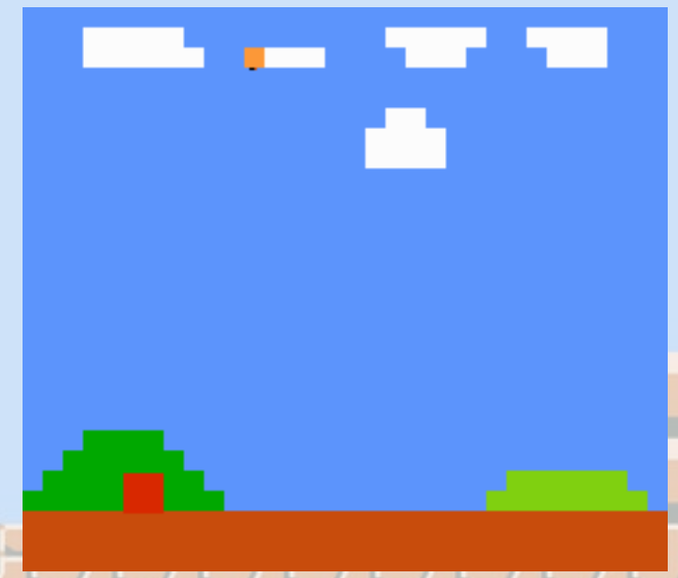
v0: ningún preproceso



v1: downsampling



v2: pixelación



v2: todo rectángulos

MODELOS UTILIZADOS

- Para este proyecto, se han entrenado varios modelos:
  - Modelo 1: DQNAgent con 20000 pasos
  - Modelo 2: Doble DQNAgent con 33000 pasos
  - Modelo 3: PPO con 800000 pasos

```python
def build_cnn_model():
    model = Sequential()
    model.add(Conv2D(32, (8, 8), strides=(4, 4), activation='relu',
    input_shape=(5, height, width, n_channels)))
    model.add(Conv2D(64, (4, 4), strides=(2, 2), activation='relu'))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(Dense(n_actions, activation='linear'))
    return model

model = build_cnn_model()
```

CNN

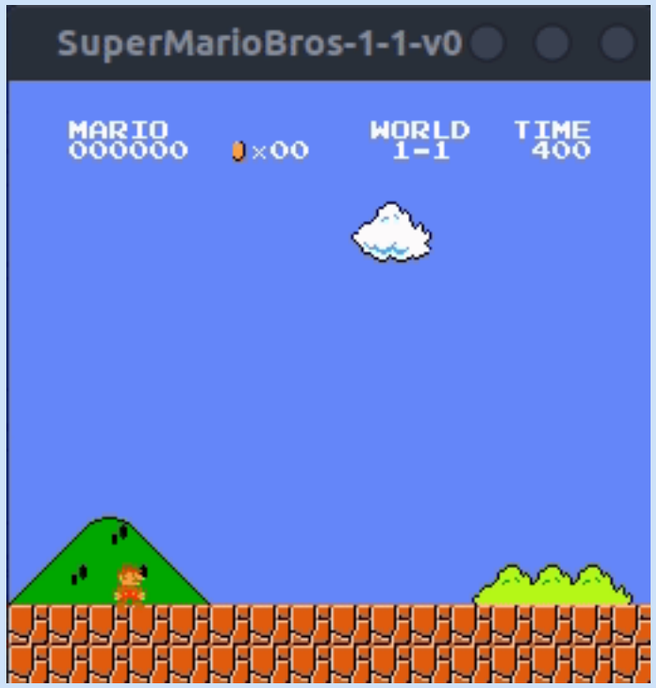
```python
dqn = DQNAgent(model=model,
               nb_actions=n_actions,
               memory=memory,
               nb_steps_warmup=4000,#50000,
               target_model_update=2000,#10000,
               policy=policy,
               #Quitar la línea de abajo en caso de no querer Double DQN
               enable_double_dqn = True)
```

Modelos DQNAgent

```python
model2 = PPO('CnnPolicy', env, verbose=1,
             tensorboard_log='./logs/',
             learning_rate=0.000001, n_steps=512)
```

Modelo PPO

RESULTADOS



Modelo 1: DQNAgent

Modelo 2: DDQNAgent

Modelo 3: PPO

GRACIAS