



# KEEPCODING

## Proyecto fin Bootcamp Aprender a programar desde cero

Pygame



# KEEPCODING

## Índice

<b>A zambullirse donde cubre .....</b>	<b>1</b>
<b>El juego: The quest .....</b>	<b>2</b>
Mecánica del juego: .....	2
Pantalla Inicial o Portada.....	5
Records del juego .....	5
<b>Forma de trabajo .....</b>	<b>6</b>



## A zambullirse donde cubre

Hasta ahora habéis estado ejercitando vuestras habilidades de programadores (tanto las básicas de pensamiento computacional como las avanzadas de programación en python y desarrollo de distintos tipos de pequeñas aplicaciones - juegos, escritorio y web -). Os toca soltaros de la orilla y nadar en lo hondo.

La idea es que practiquéis lo aprendido y que seáis capaces de desarrollar una pequeña aplicación (un prototipo) lo más completa funcionalmente y sin errores. Tenéis tres semanas para hacerlo durante las cuales deberéis investigar, decidir, estudiar y aprender cosas nuevas (eso ocurre en cualquier proyecto de la vida real).

Los dos puntos fundamentales son **aprender** y **disfrutar**. Utilizad lo aprendido y el resto de cursos de la plataforma para completar vuestro prototipo.

El objetivo es entregar el proyecto lo más avanzado posible, valorándose tanto el modo de resolución como el alcance. Sois libres de modificar, añadir o reemplazar funcionalidades que mejoren la aplicación. Ahora bien, si la mejora sirve para escaquearse del trabajo sólo irá en contra de la valoración final del proyecto.



## El juego: The quest

**Historia del juego:** La búsqueda comienza en un planeta tierra moribundo por el cambio climático. Partiremos a la búsqueda de un planeta compatible con la vida humana para colonizarlo.

### Mecánica del juego:

#### INICIO DE NIVEL

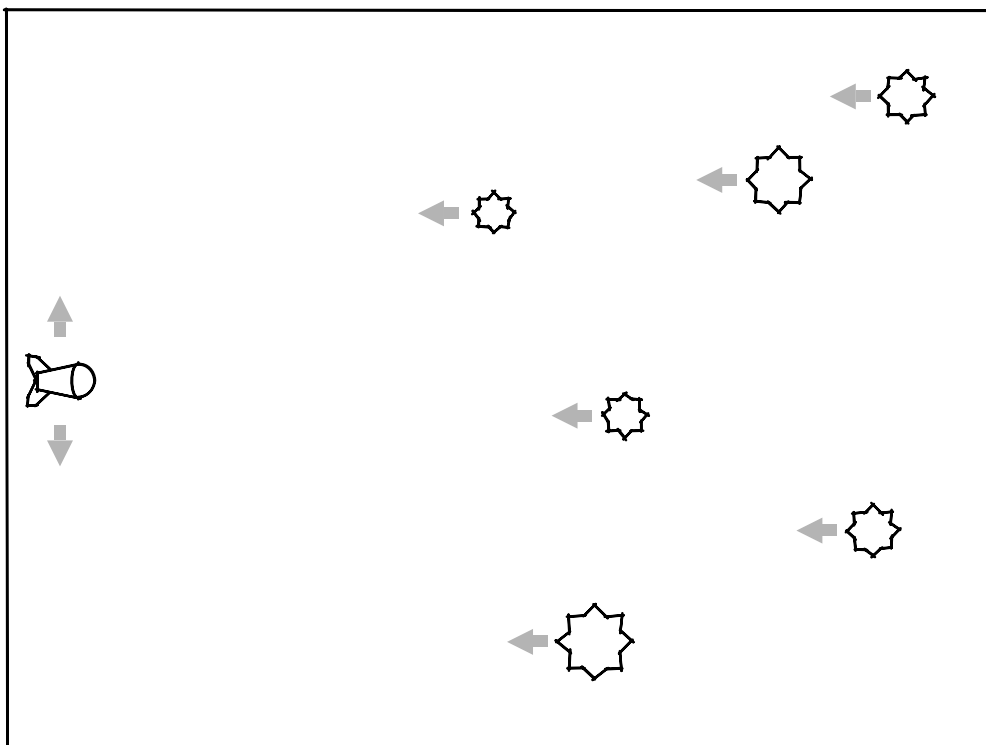
- Nuestra **nave** aparecerá en el margen izquierdo de la pantalla en el centro vertical.
- La nave se moverá hacia arriba y abajo con las teclas del cursor.
- La nave se parará en su ascenso y descenso si deja de pulsarse la tecla del cursor que corresponda

#### Restricciones

- La nave no podrá avanzar ni retroceder de derecha a izquierda ni viceversa.
- Una vez alcanzado el nivel superior o inferior de la pantalla la nave se quedará parada. No desaparecerá de la pantalla ni aparecerá por el otro lado.

#### Opciones

- El movimiento hacia arriba y hacia abajo de la nave podrá ser acelerado si se mantiene pulsada la tecla de flecha.





## DESARROLLO DE NIVEL

- Aparecerán **obstáculos** de diferente tipo y tamaño en alturas aleatorias en el margen izquierdo de la pantalla.
- Los obstáculos avanzarán en línea recta con diferentes velocidades de derecha a izquierda y desaparecerán al llegar al margen derecho de la pantalla
- Si cualquiera de los obstáculos choca con la nave, esta se destruirá con una explosión (sonido y animación) y acabará la partida
- La dificultad del nivel viene determinada por la frecuencia de creación de obstáculos y su velocidad.
- Un nivel siempre debe ser más difícil que los anteriores

## Opciones

- El nivel se terminará transcurrido un determinado tiempo o número de obstáculos esquivados (a decidir por el desarrollador)
- Puede acelerarse la frecuencia de aparición de obstáculos dentro de un nivel o bien que sea constante.
- El mínimo de niveles serán 2, pudiendo crearse tantos como se quiera
- La imagen de fondo de cada nivel puede ser la misma o distinta para cada nivel
- Se puede dejar la imagen de fondo y que el movimiento lo aporten los obstáculos, o animar la imagen de fondo para simular avance incluso sin obstáculos.

## PUNTUACIONES

- La puntuación se incrementará a medida que la nave del jugador va esquivando obstáculos y progresa. Cuanto más tiempo aguante (más obstáculos esquivados) mayor será la puntuación.
- La puntuación empezará en cero al iniciar la partida
- Aterrizar en un planeta puede dar puntos extra (fin de nivel)
- La puntuación debe aparecer continuamente en pantalla. En la esquina superior izquierda.

## FIN DE NIVEL

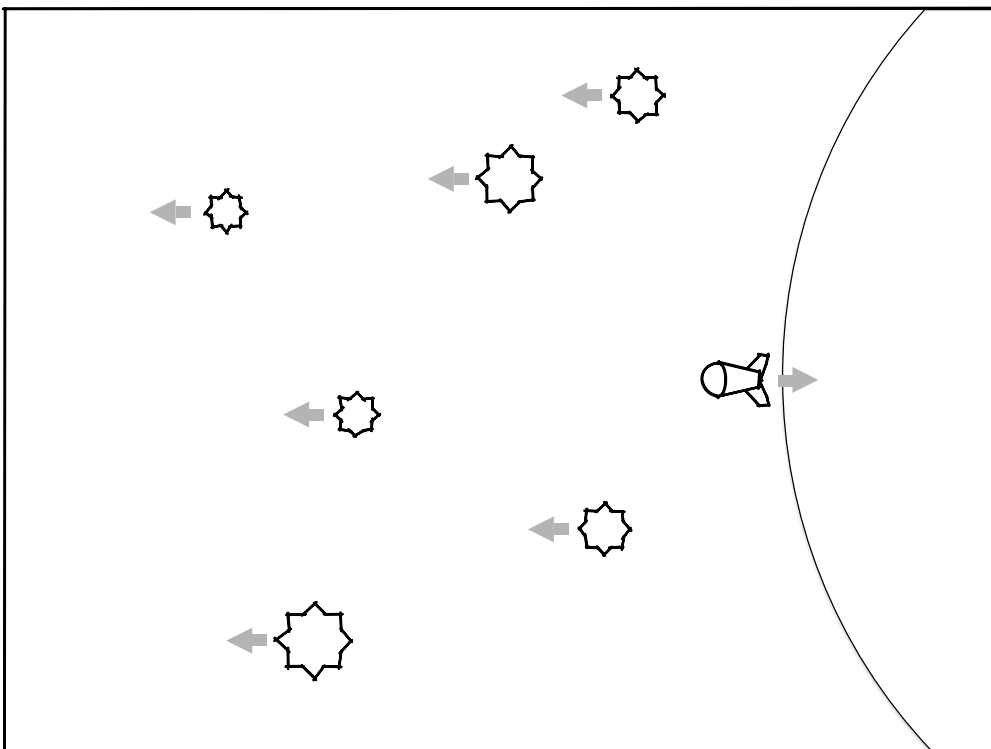
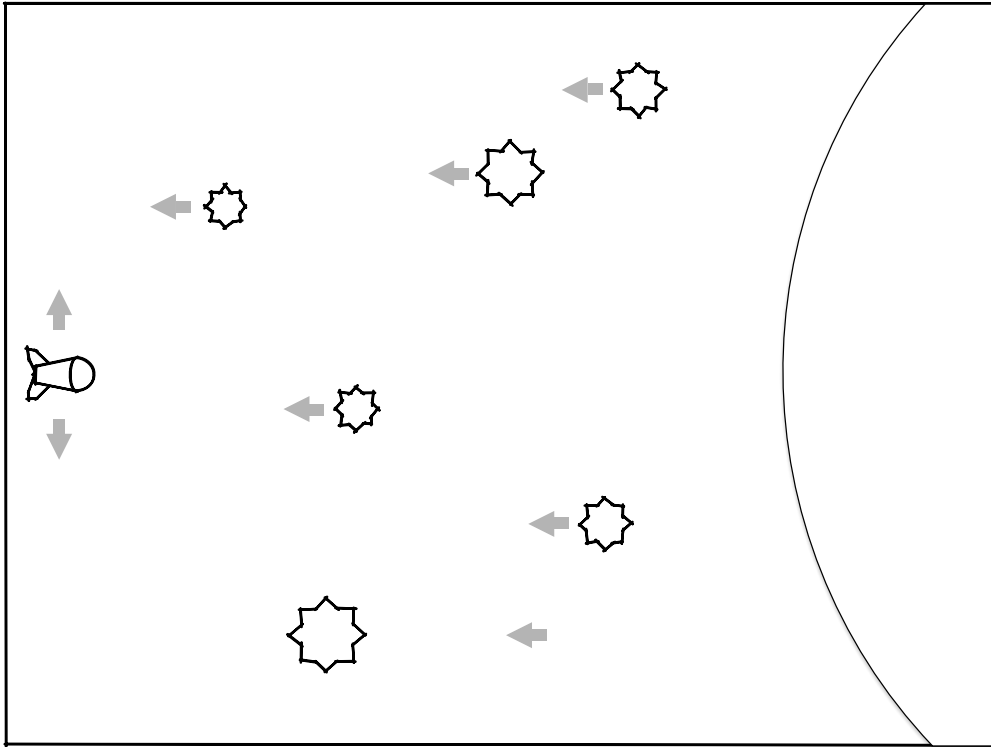
- Una vez se produzca la condición de final de nivel (tiempo o número de obstáculos generados y esquivados) se alcanzará el final de nivel.
- Aparecerá un planeta por la parte izquierda de la pantalla.
- Dejarán de aparecer obstáculos pero los que estén en pantalla deberán continuar su movimiento hasta el borde derecho o chocar con la nave.
- La nave girará 180 grados y aterrizará sobre el planeta de forma automática
- Aparecerá un cartel que indique "**Pulse <tecla elegida por el programador> para continuar**"
- Si fuera el último nivel el cartel sería de felicitación e indicaría acción para reiniciar el juego.
- Si el jugador no iniciara la partida transcurrido un tiempo, se volverá a la portada.

## Restricciones

- En ningún caso el aterrizaje automático puede implicar la colisión con un obstáculo (si aún estuvieran en pantalla). Una vez el juego toma el control de la nave no se pueden producir colisiones.



# KEEPCODING





## Pantalla Inicial o Portada

---

Antes de iniciar la partida aparecerá una pantalla inicial que

### PANTALLA INICIAL

- Mostrará el nombre y la historia del juego (la búsqueda de otro planeta).
- Indicará como iniciar la partida (con un botón o pulsando una tecla)
- Si el usuario pulsa esa tecla o botón se iniciará la partida
- Mostrará las instrucciones del juego.

### Opciones

- Las instrucciones podrán aparecer directamente en la pantalla inicial, o indicar que pulsando una tecla se mostrarán las instrucciones en otra pantalla o encima de la actual

## Records del juego

---

### RECORDS DEL JUEGO

- Podrán almacenarse las 5 o 10 mejores puntuaciones con las iniciales del jugador.
- Se almacenarán en una base de datos de SQLite
- Al finalizar la partida, si la puntuación fuera mayor que cualquiera de las 5 almacenadas, se habilitará una pantalla de entrada de datos en la que se podrán incluir 3 iniciales (como en los videojuegos antiguos)
- El juego actualizará la tabla de records y los presentará en la pantalla de inicio o en la de final de partida (a elegir por el desarrollador)



## Forma de trabajo

El proyecto será individual, oficialmente tenéis tres semanas para hacerlo comenzando este periodo el **3 de febrero de 2020** y debe ser entregado el **23 de febrero antes de las 23:59**.

Se os indicará antes del inicio qué formulario utilizar para realizar la entrega.

La entrega consistirá en indicarnos el **enlace al repositorio remoto** en que habéis dejado vuestro programa. Este repositorio deberá incluir en el fichero **readme.md** las instrucciones especiales de instalación si las hubiera. No obstante sabéis utilizar entornos virtuales y el fichero **requirements.txt** con ello debería bastar. De la misma forma deberéis incluir en el directorio **migrations** la creación de las tablas necesarias y las instrucciones para realizarlo en **reame.md**. Proteged también el fichero **config.py** con claves si lo hubiera (la key de acceso a la API de coinmarketcap).

A lo largo de esas tres semanas dispondréis de mi asesoramiento en slack como hasta ahora y se habilitará la noche del jueves de cada una de las tres semanas la posibilidad de realizar una tutoría/seguimiento conjunto de los proyectos si hay petición por vuestra parte.

Los proyectos serán corregidos en los días siguientes a la entrega y os informaré por correo de vuestra calificación.