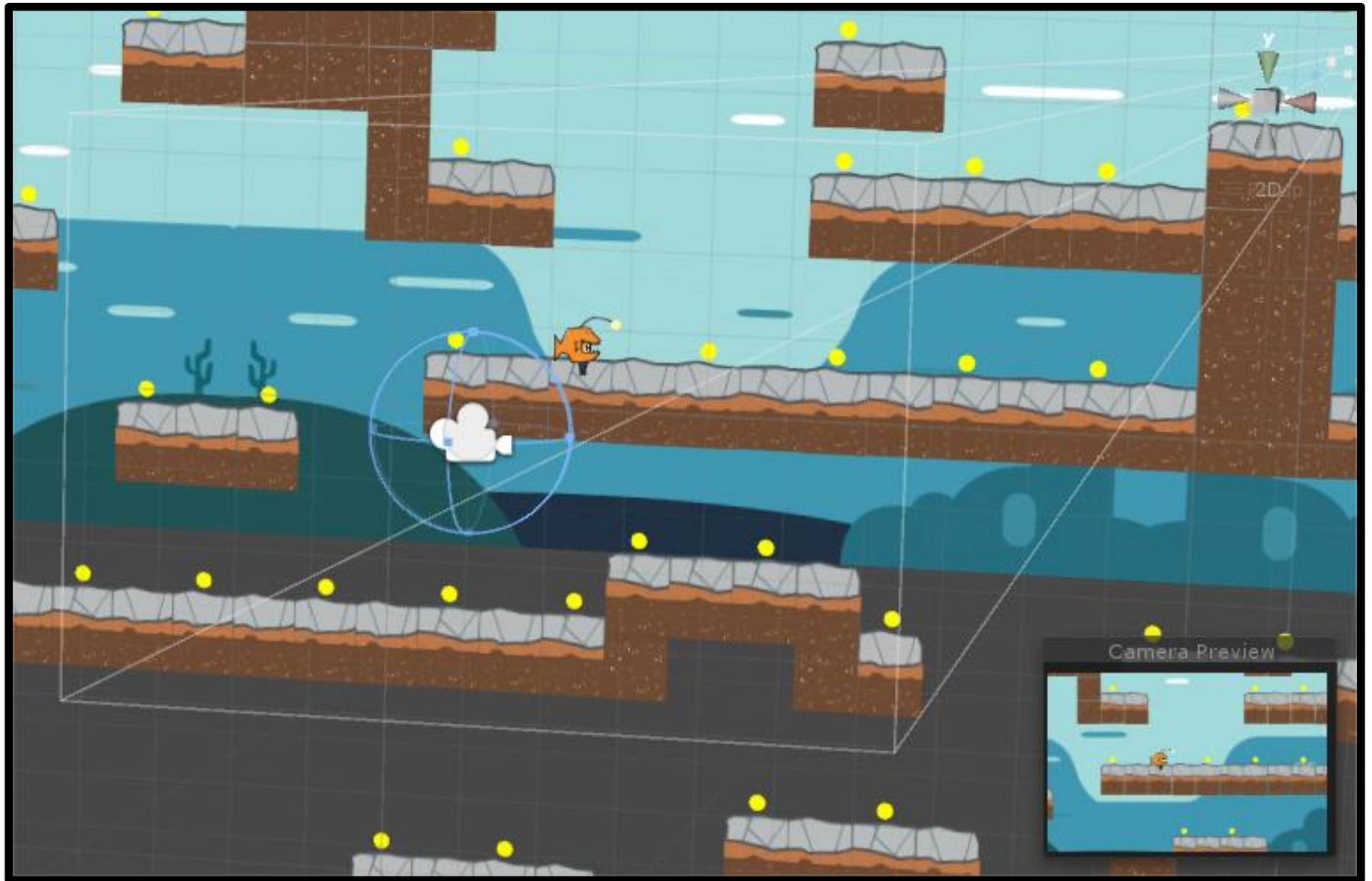


# **Low-Level Game Development 2**

# **V9 - Camera**

## **Camera Object**



# **V10 - Camera**

## **Relative Drawing**

**Absolute Coordinates**

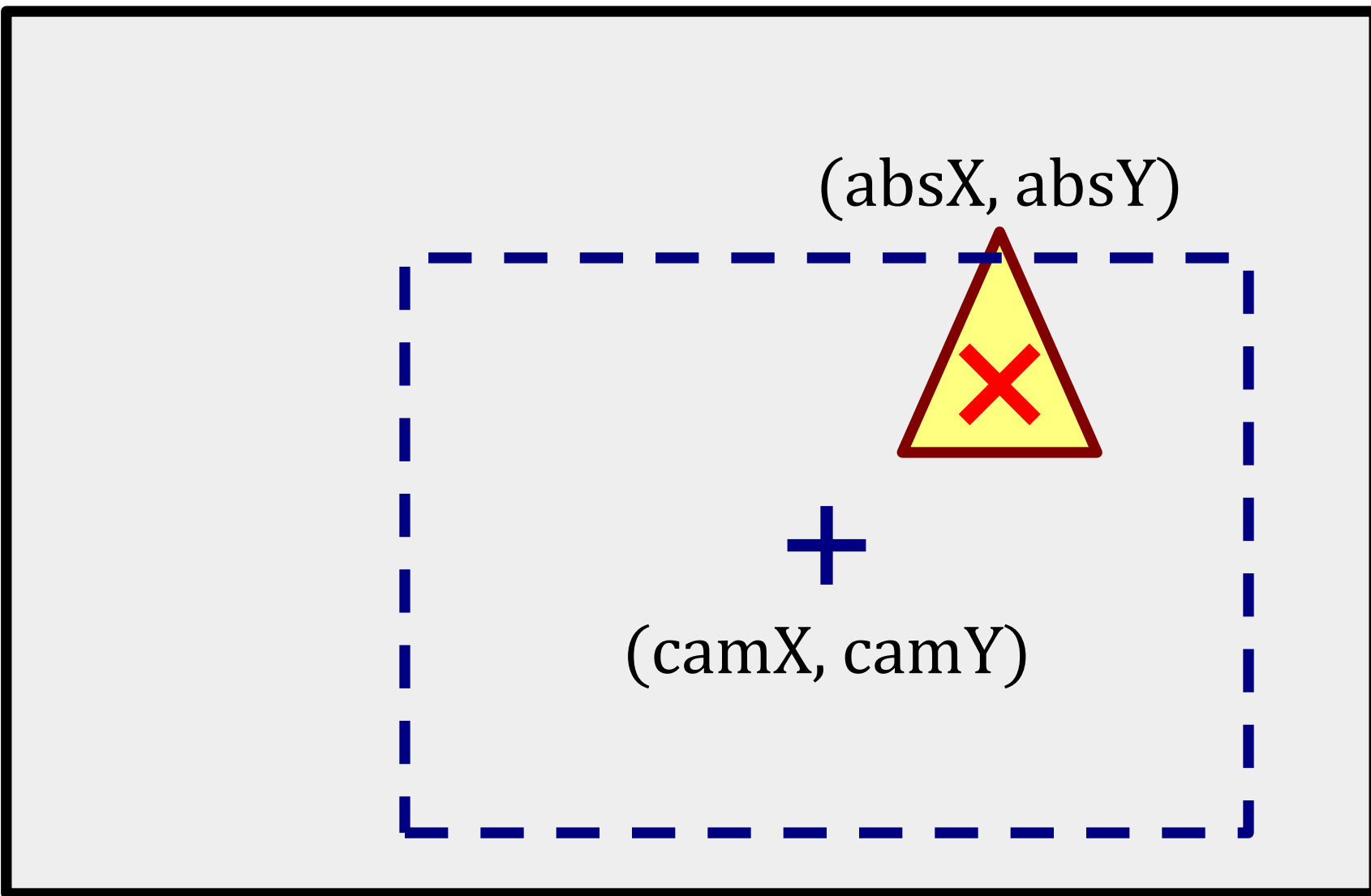
**Relative Coordinates**

$(x,y)$

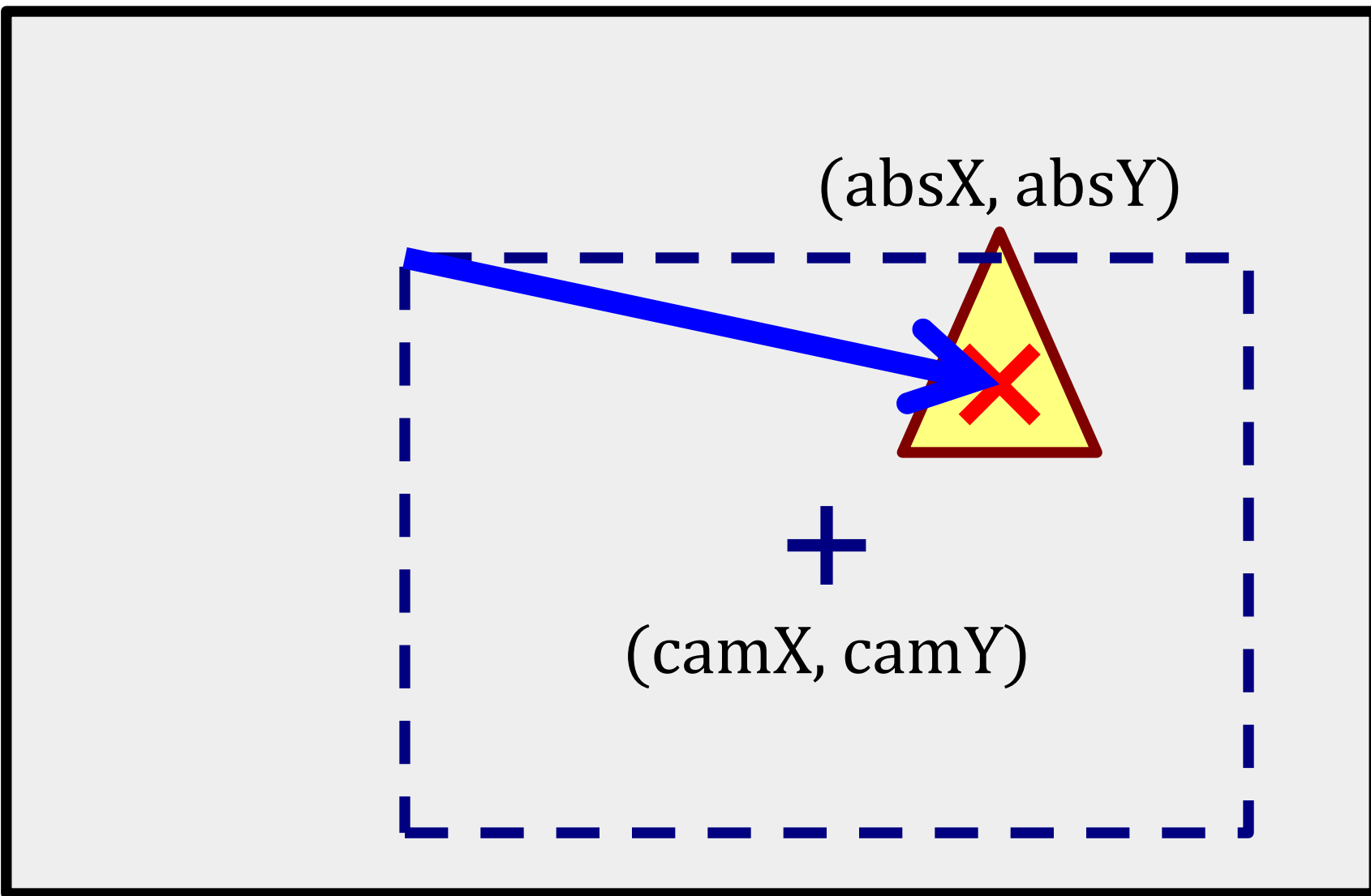


$(\text{abs}X, \text{abs}Y)$



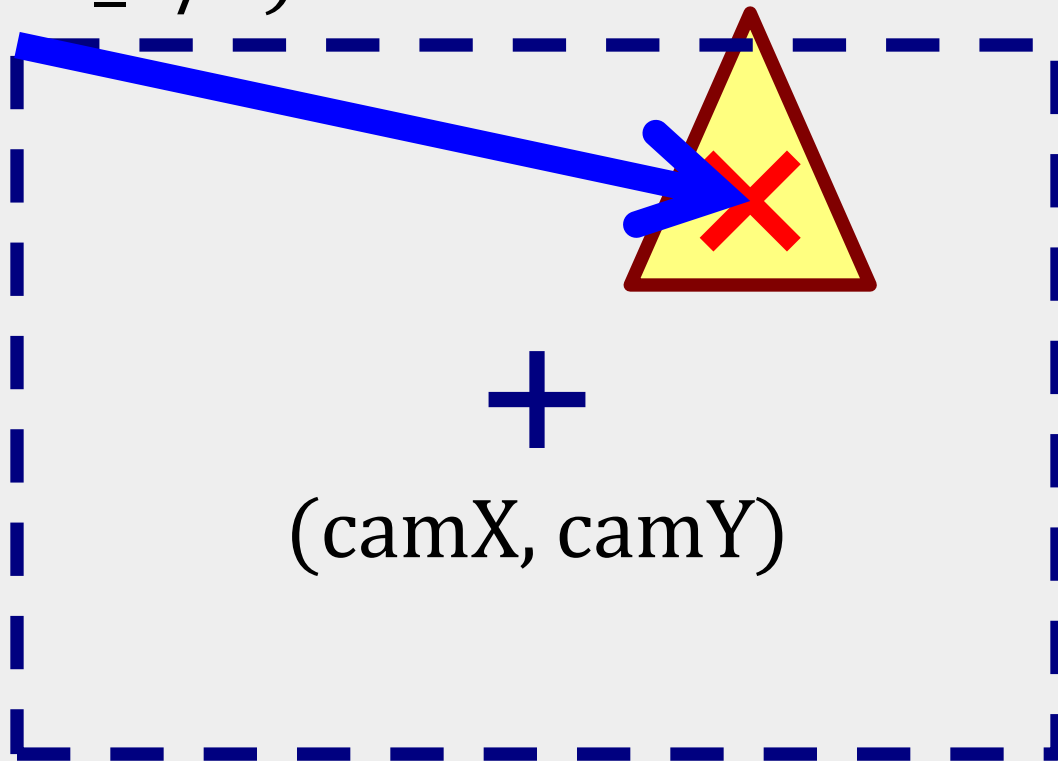






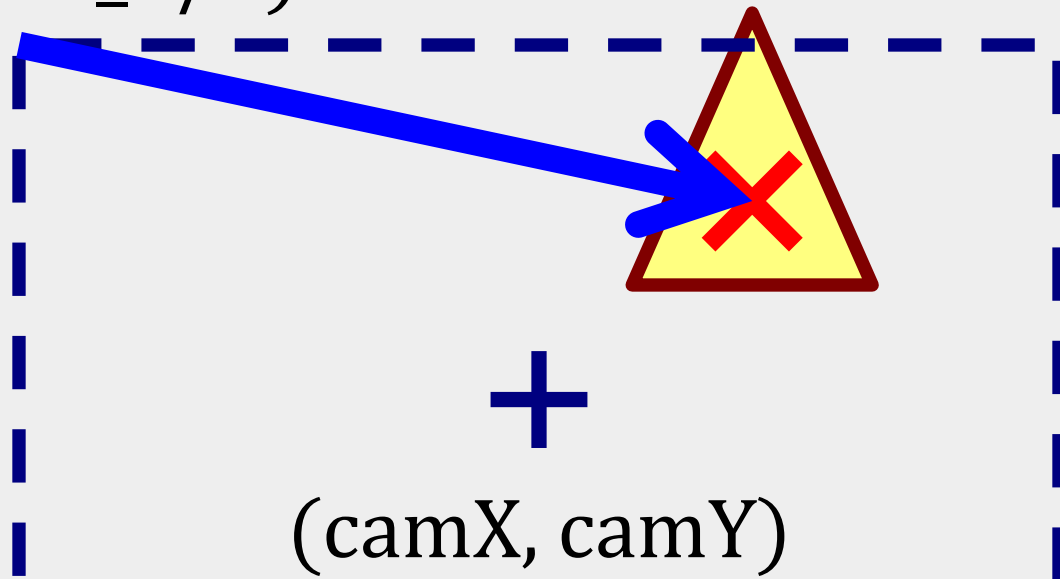
$(\text{camX} - \text{RES\_X}/2,$   
 $\text{camY} - \text{RES\_Y}/2)$

$(\text{absX}, \text{absY})$



$(\text{camX} - \text{RES\_X}/2,$   
 $\text{camY} - \text{RES\_Y}/2)$

$(\text{absX}, \text{absY})$



$$\text{relX} = \text{absX} - (\text{camX} - \text{RES\_X}/2)$$

$$\text{relY} = \text{absY} - (\text{camY} - \text{RES\_Y}/2)$$

# Absolute to Relative

$$\text{relX} = \text{absX} - (\text{camX} - \text{RES\_X}/2)$$

$$\text{relY} = \text{absY} - (\text{camY} - \text{RES\_Y}/2)$$

# Relative to Absolute

$$\text{absX} = \text{relX} + (\text{camX} - \text{RES\_X}/2)$$

$$\text{absY} = \text{relY} + (\text{camY} - \text{RES\_Y}/2)$$

# Absolute to Relative

$$x -= (\text{camX} - \text{RES\_X}/2)$$

$$x -= (\text{camY} - \text{RES\_Y}/2)$$

# Relative to Absolute

$$x += (\text{camX} - \text{RES\_X}/2)$$

$$x += (\text{camY} - \text{RES\_Y}/2)$$

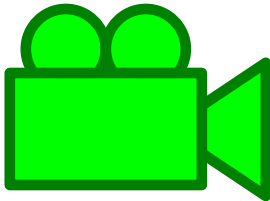
# **V11 - Camera**

**Some game adjustments**

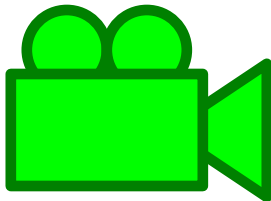
# **V12 - Camera**

**Smooth Camera**

**Player**



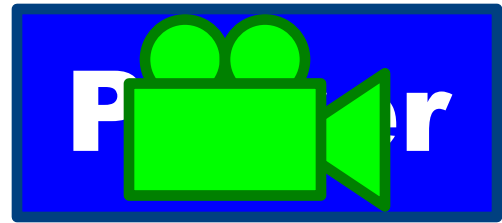




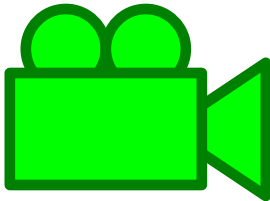
**Chase**

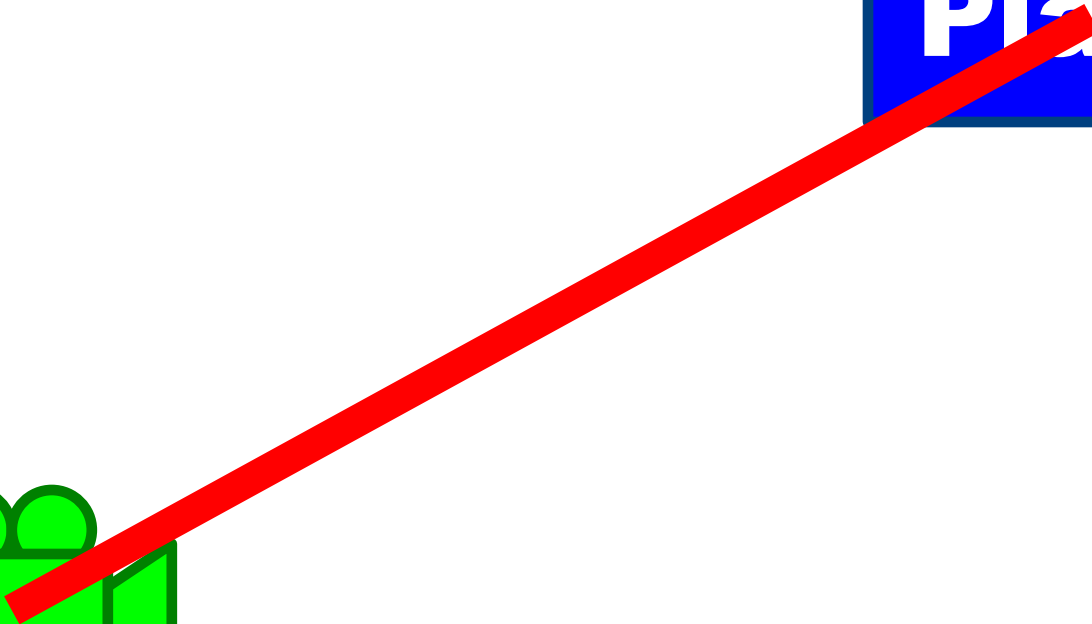
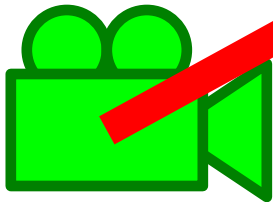


**Player**

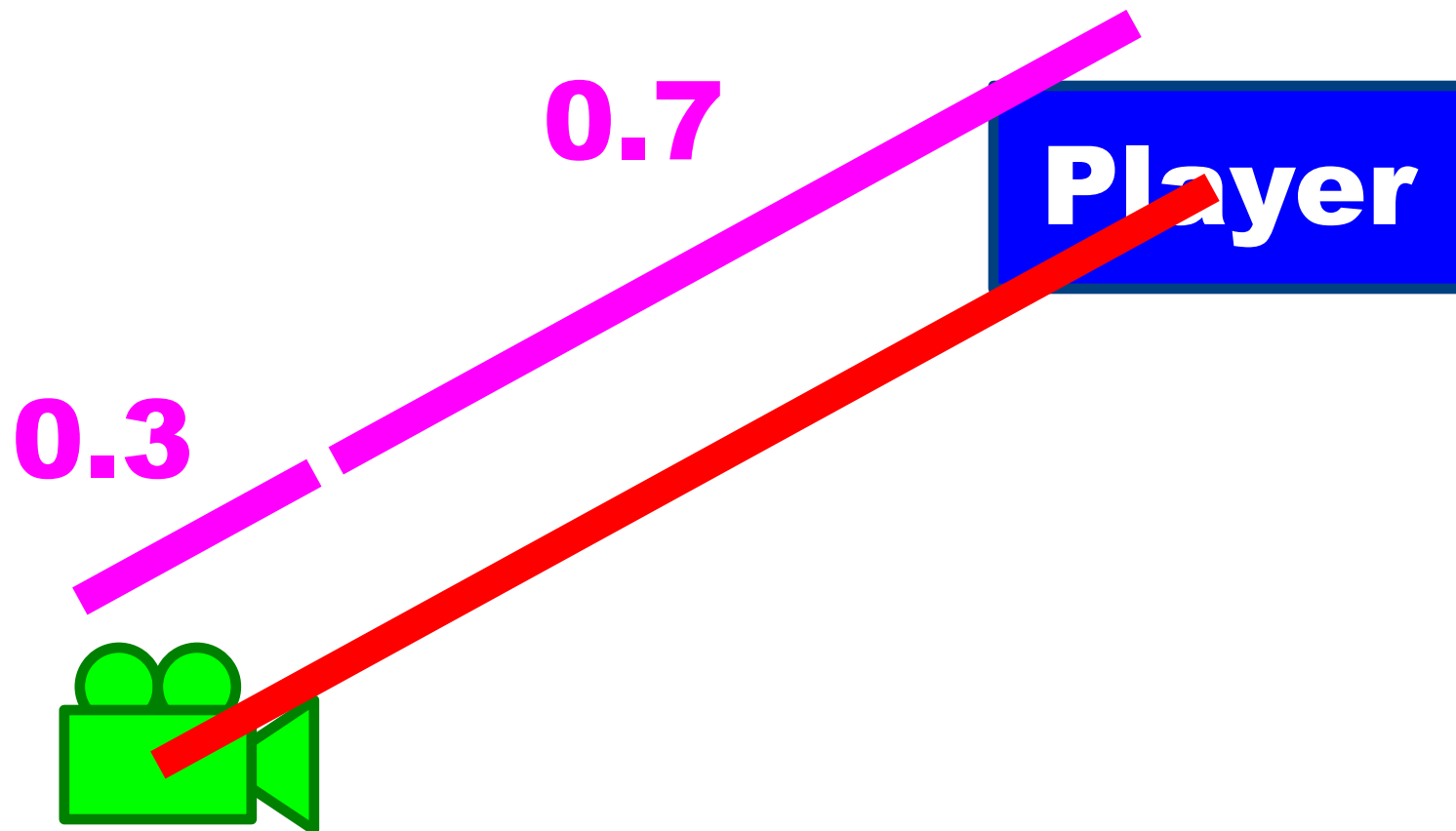


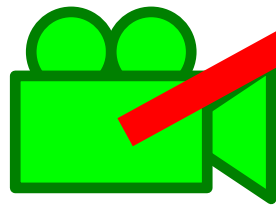
**Player**



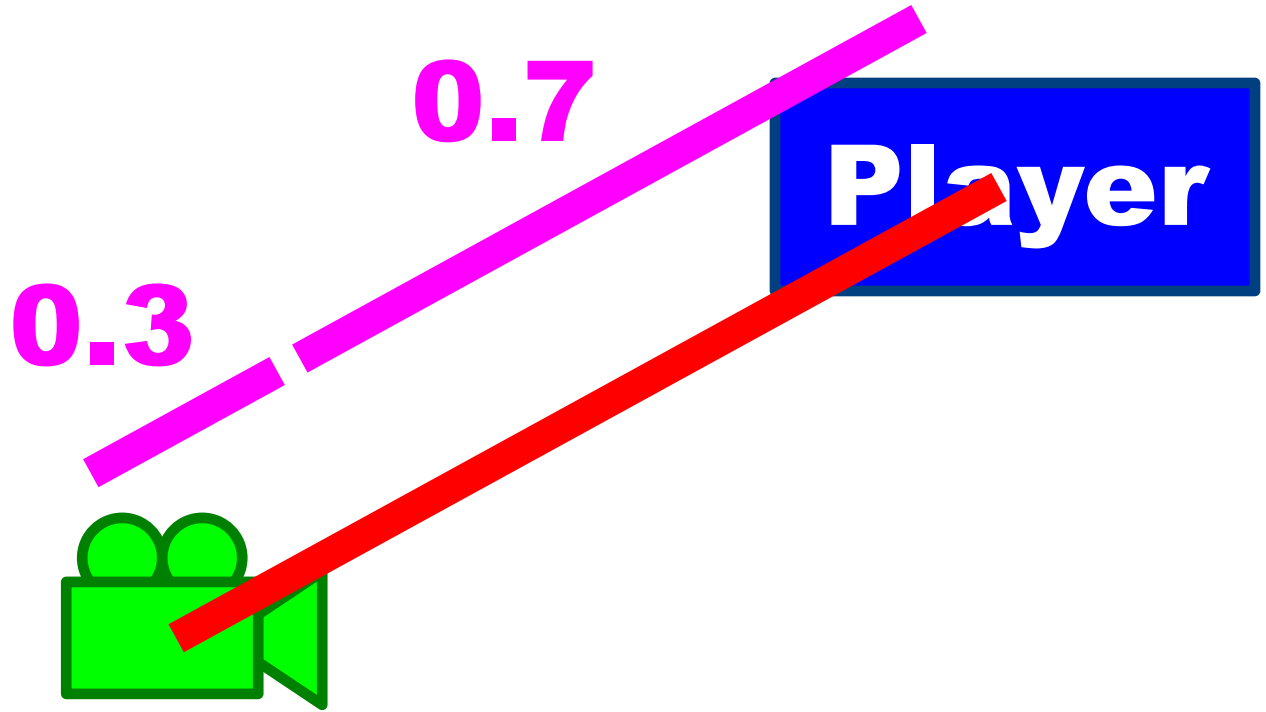


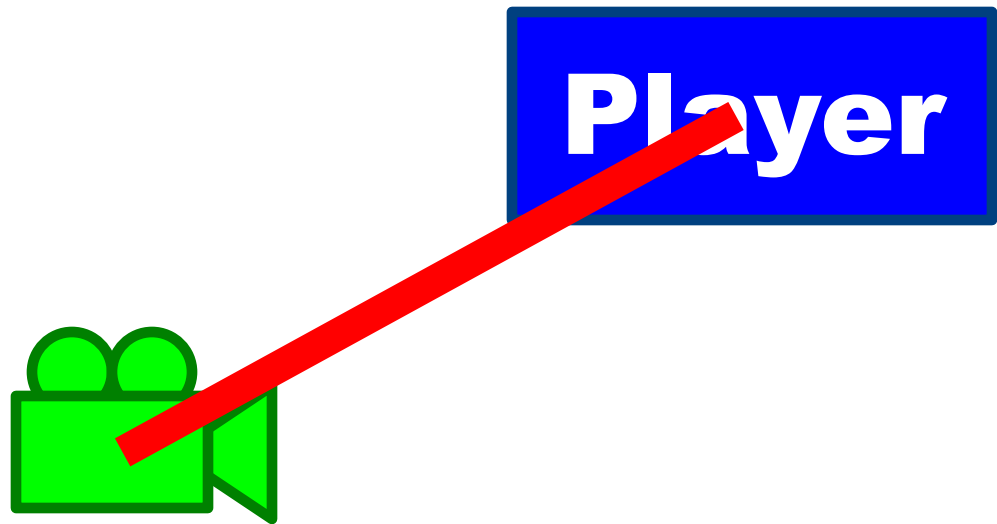
**Player**





**Player**



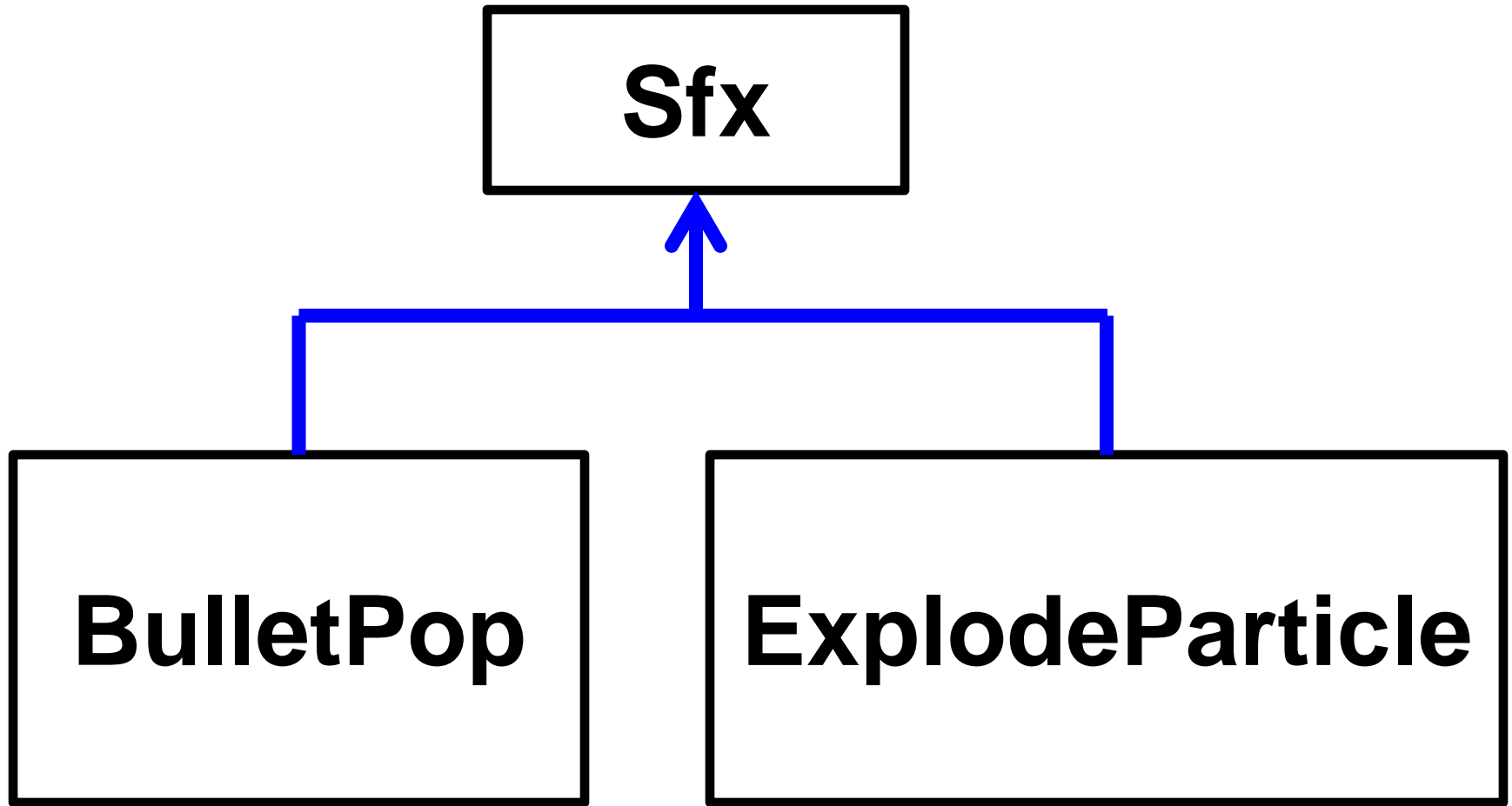




**V13 - Sfx**

**Sfx Abstract Class**

**Sfx**



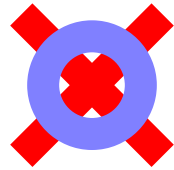
**V14 - Sfx**

**BulletPop Sfx**

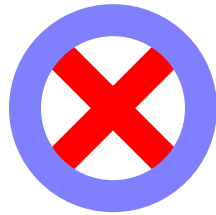




animframe = 0

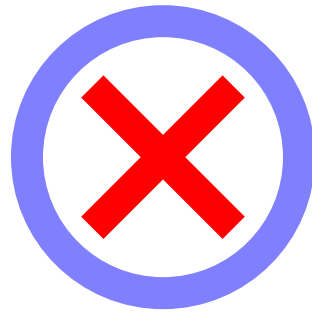


animframe = 0

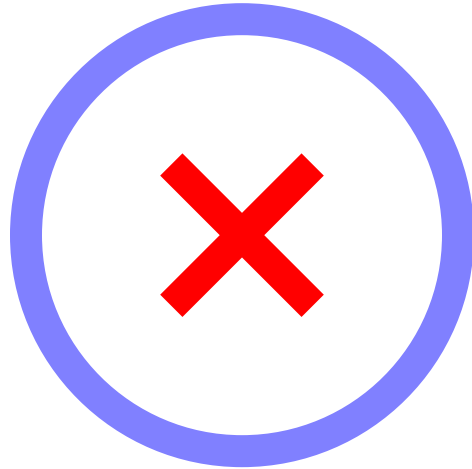


animframe = 1





animframe = 2



animframe = 3



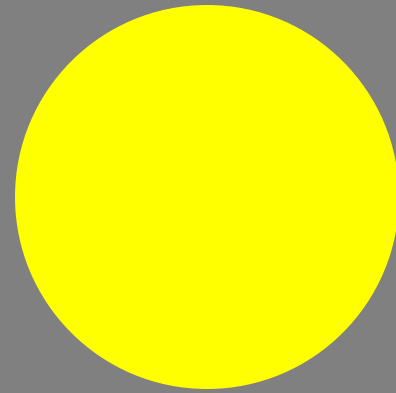
animframe = 40



```
radius = 1 + 60*animframe/duration;
```

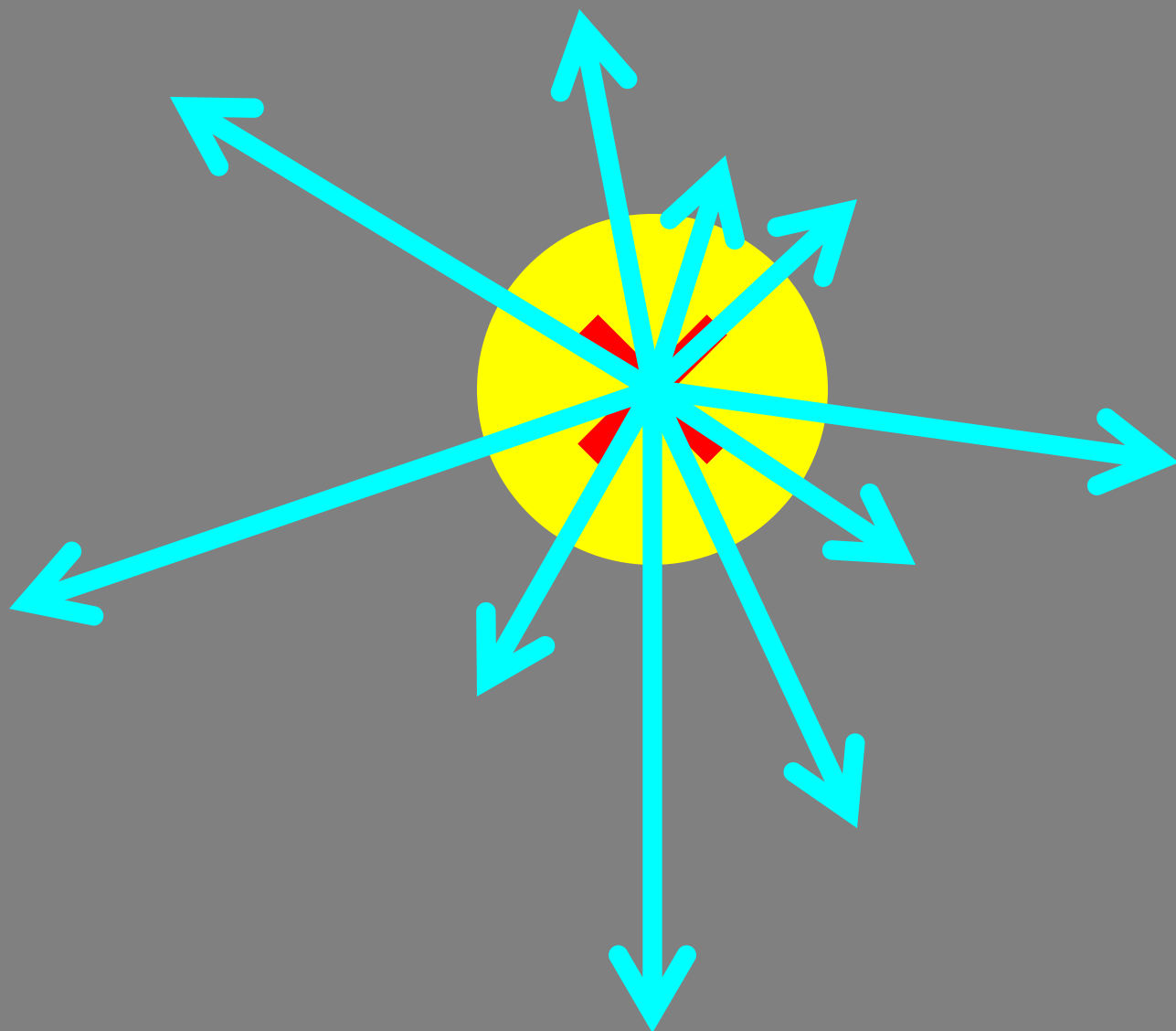
**V15 - Sfx**

**Explosion Particles**



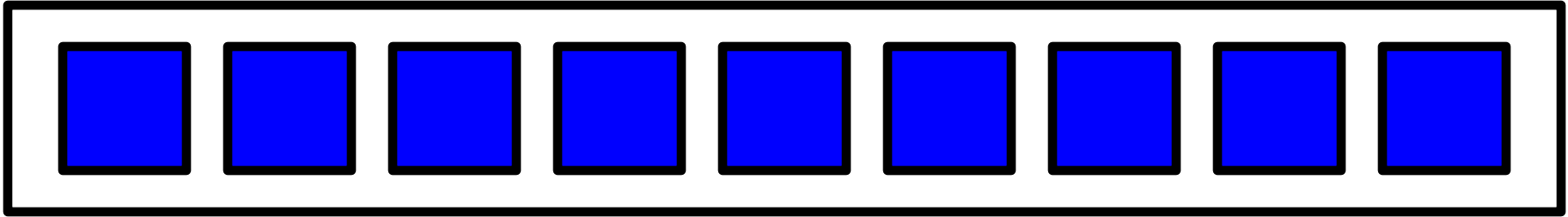


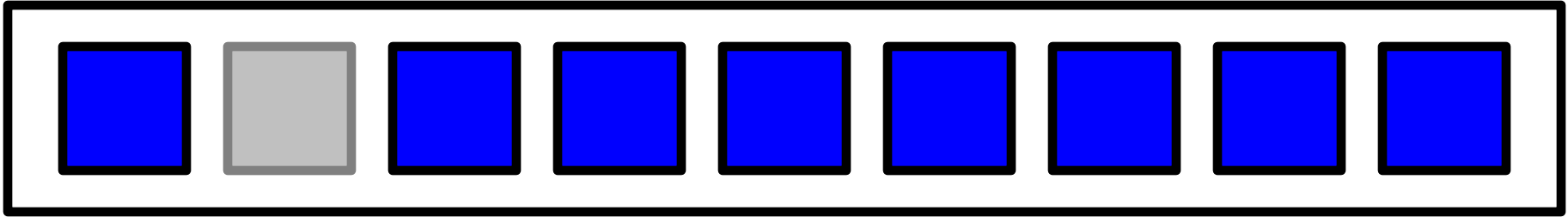


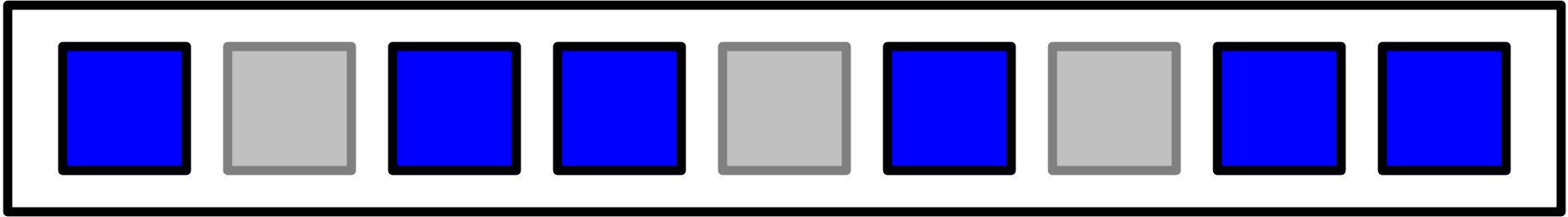


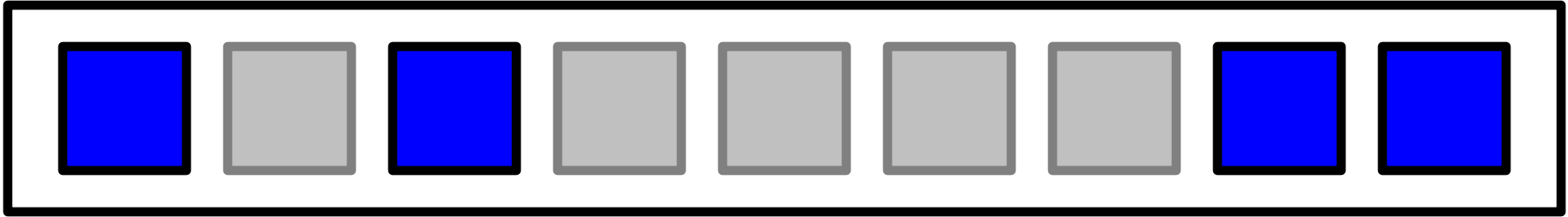
# **V16 - Array Deletion**

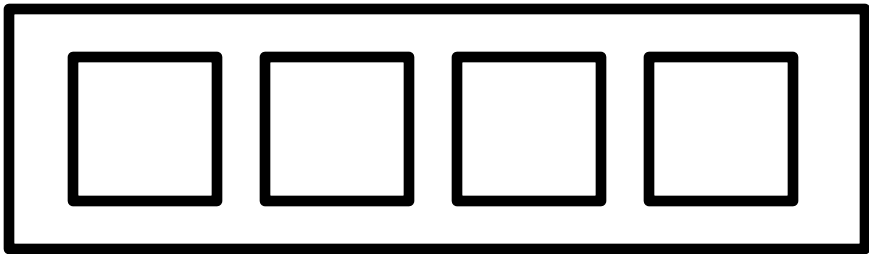
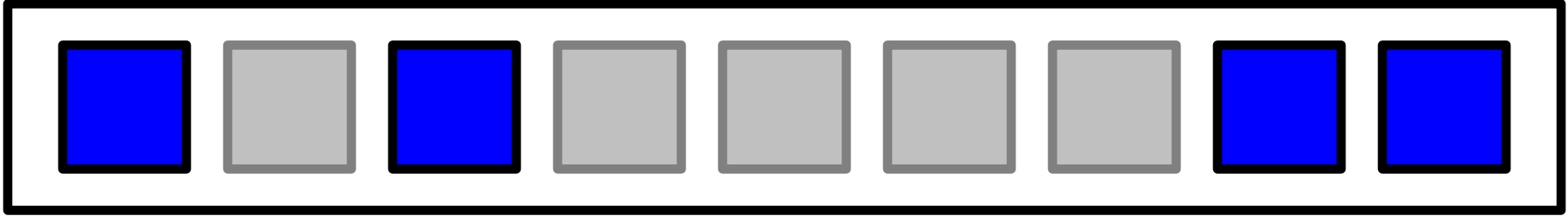
**Clean up arrays regularly**

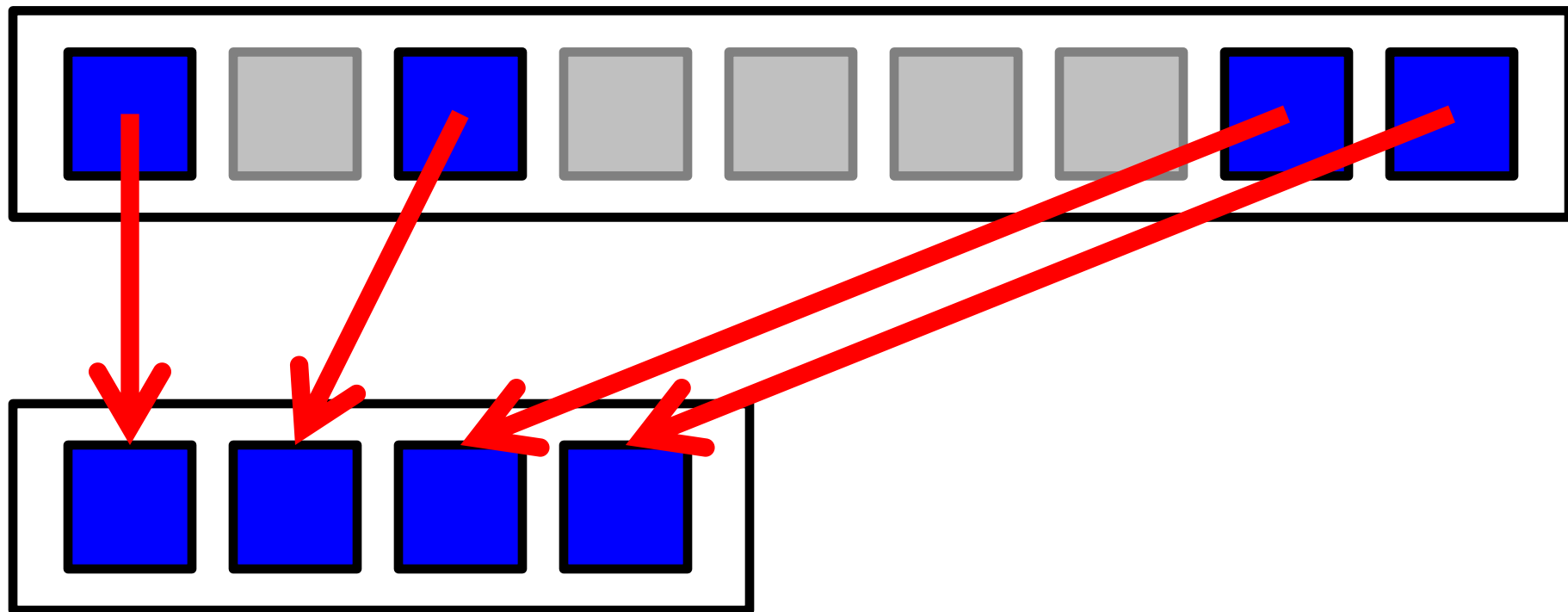




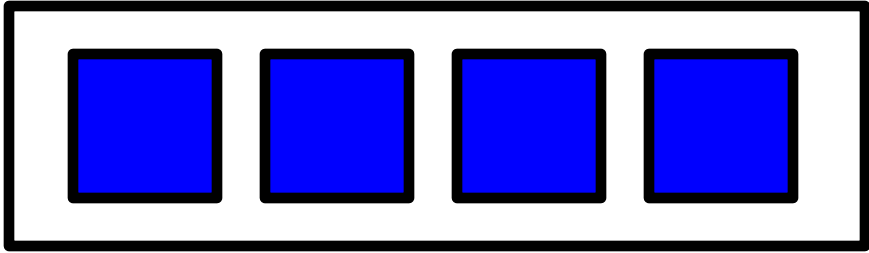












# **V17 - Menu**

## **Menu Screen**

**Infinite while loop**

```
graph TD; A[Infinite while loop] --> B[Read events]; B --> C[Update frame]; C --> D[Draw frame]; D --> A;
```

The diagram illustrates an infinite while loop structure. It begins with a yellow box labeled "Infinite while loop". An arrow points down to a light blue box labeled "Read events". Another arrow points down to a light blue box labeled "Update frame". A third arrow points down to a light blue box labeled "Draw frame". A curved arrow then loops back from the bottom of the "Draw frame" box to the top of the "Infinite while loop" box, completing the cycle.

**Read events**

**Update frame**

**Draw frame**

```
bool inGame;
```

**Infinite while loop**

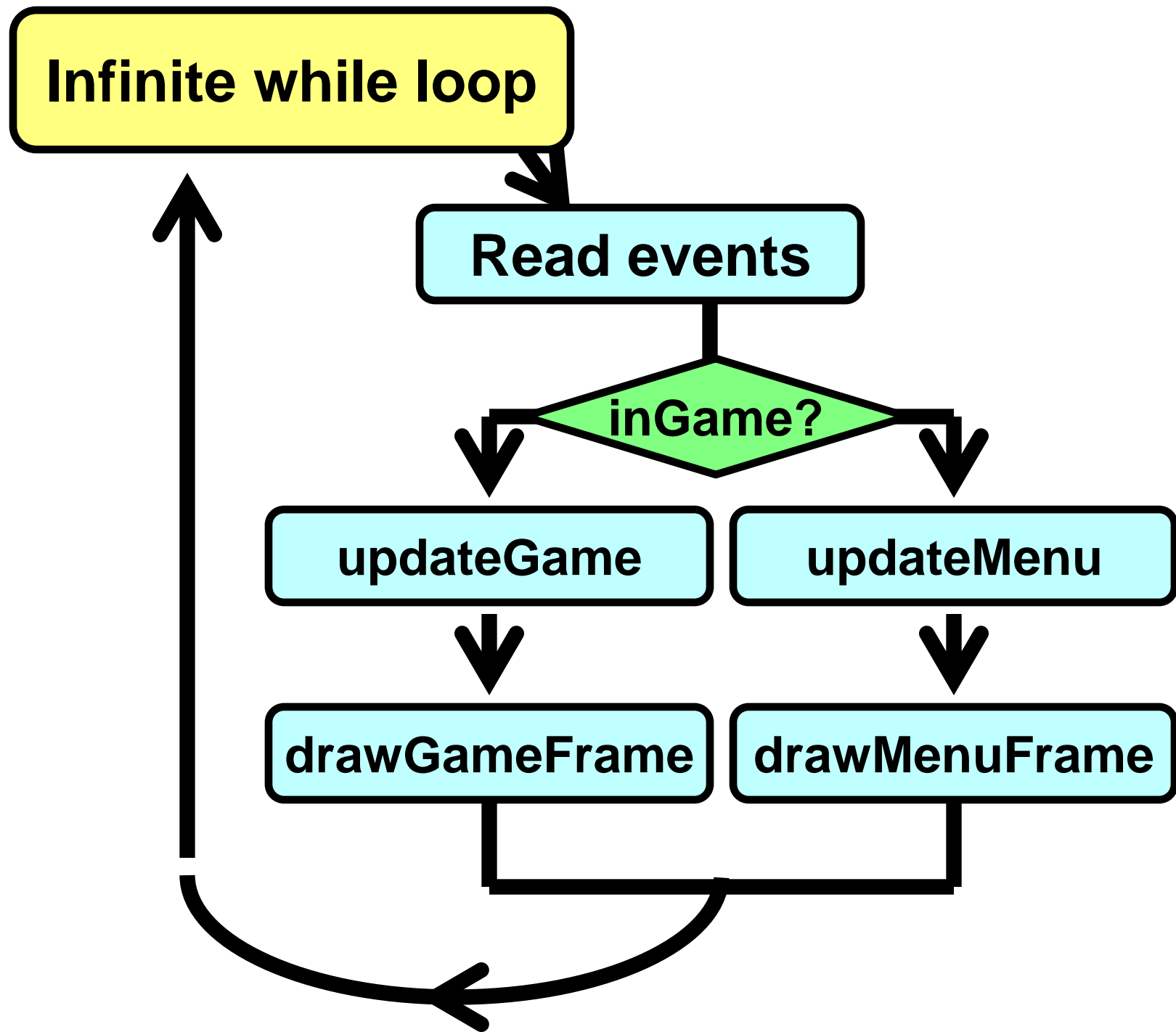
```
graph TD; A[Infinite while loop] --> B[Read events]; B --> C[updateGame]; C --> D[drawGameFrame]; D --> A;
```

The diagram illustrates an infinite while loop structure. It begins with a yellow box labeled "Infinite while loop". An arrow points down to a light blue box labeled "Read events". From "Read events", an arrow points down to another light blue box labeled "updateGame". From "updateGame", an arrow points down to a third light blue box labeled "drawGameFrame". Finally, a curved arrow points from the bottom of "drawGameFrame" back up to the "Infinite while loop" box, completing the loop.

**Read events**

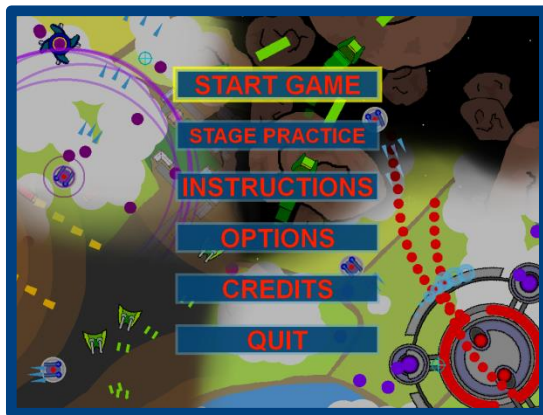
**updateGame**

**drawGameFrame**



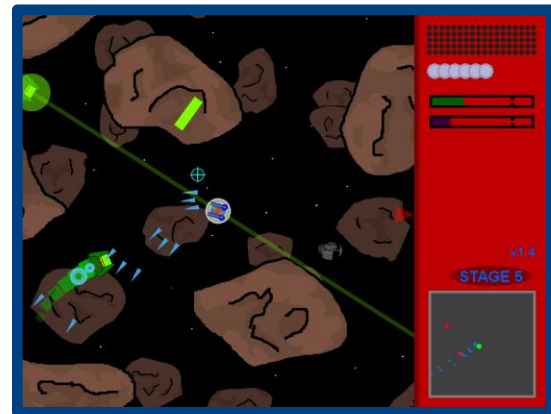
**DISPLAY**

# DISPLAY

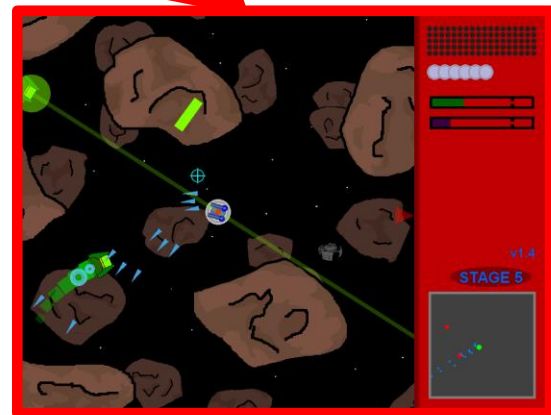
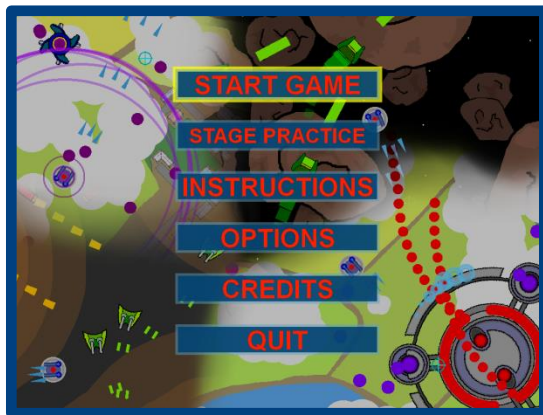
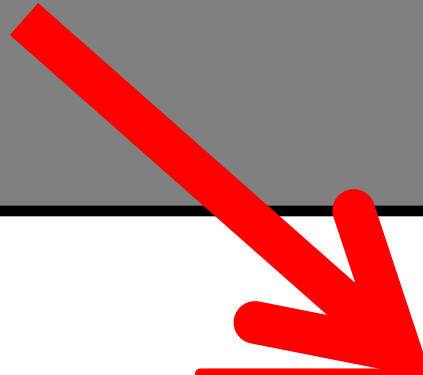




# DISPLAY



# DISPLAY



# **V18 - Menu**

**Start Game / Options**

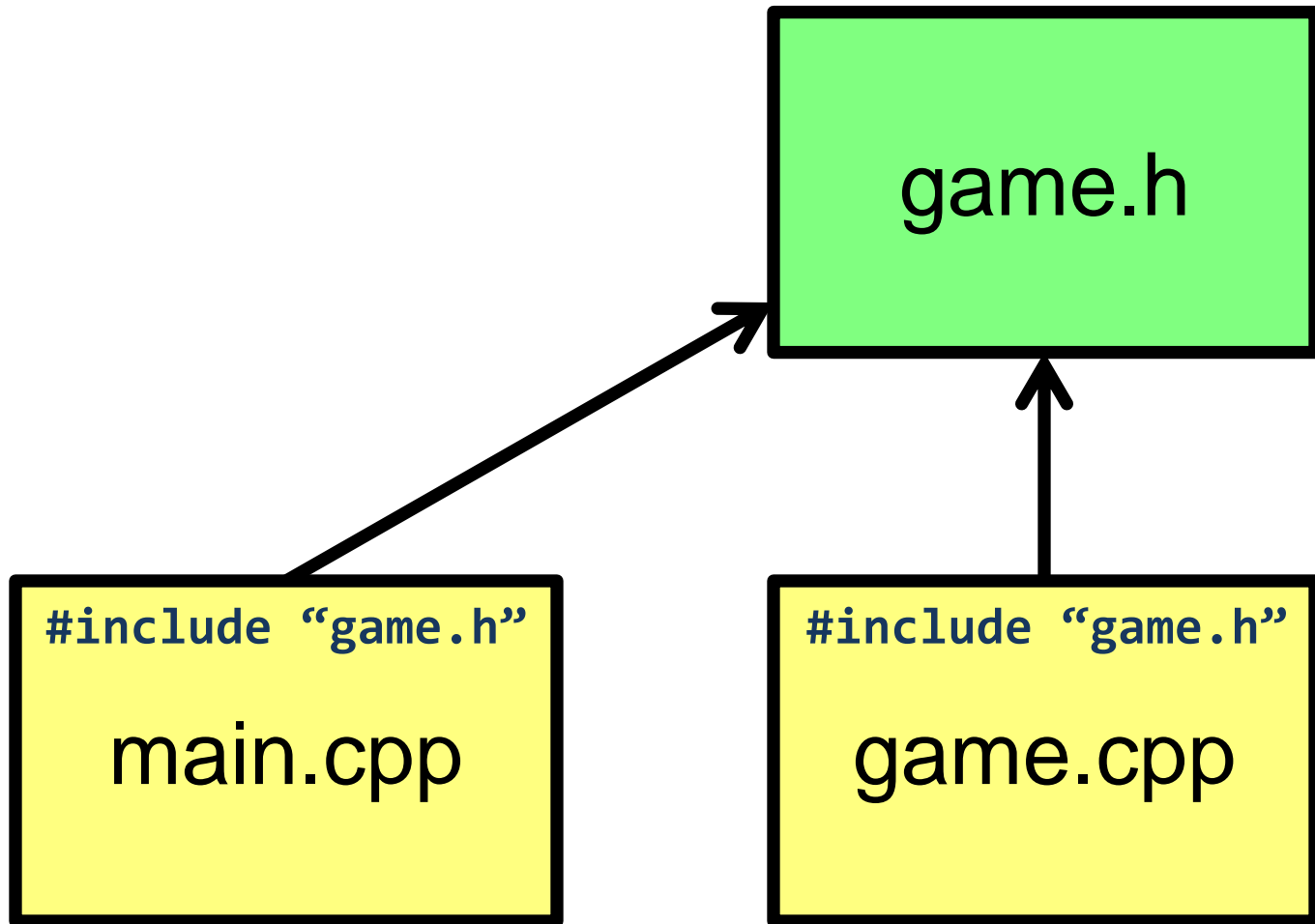


main.cpp



main.cpp

game.cpp

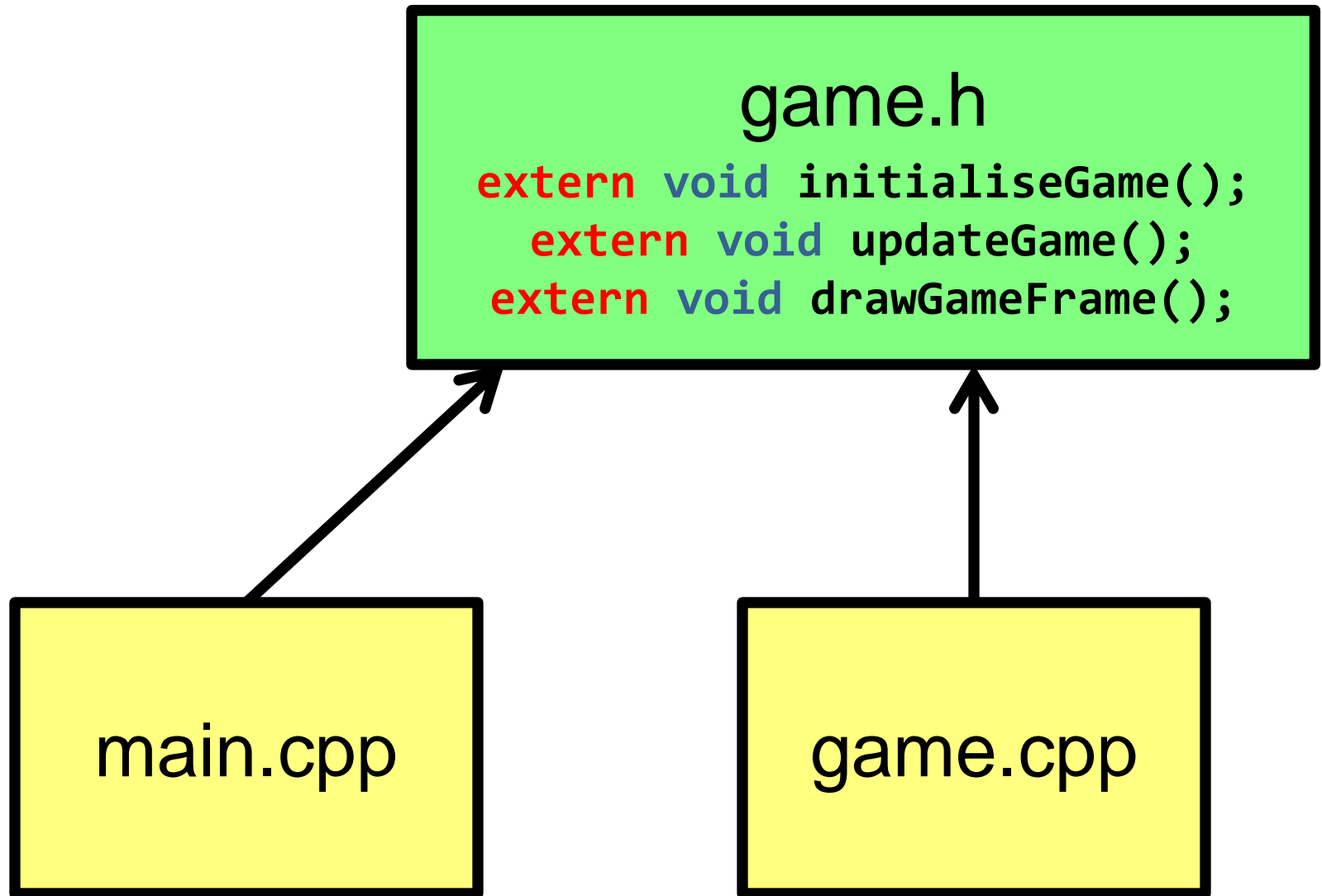


game.h

```
extern void initialiseGame();  
extern void updateGame();  
extern void drawGameFrame();
```

main.cpp

game.cpp



game.h

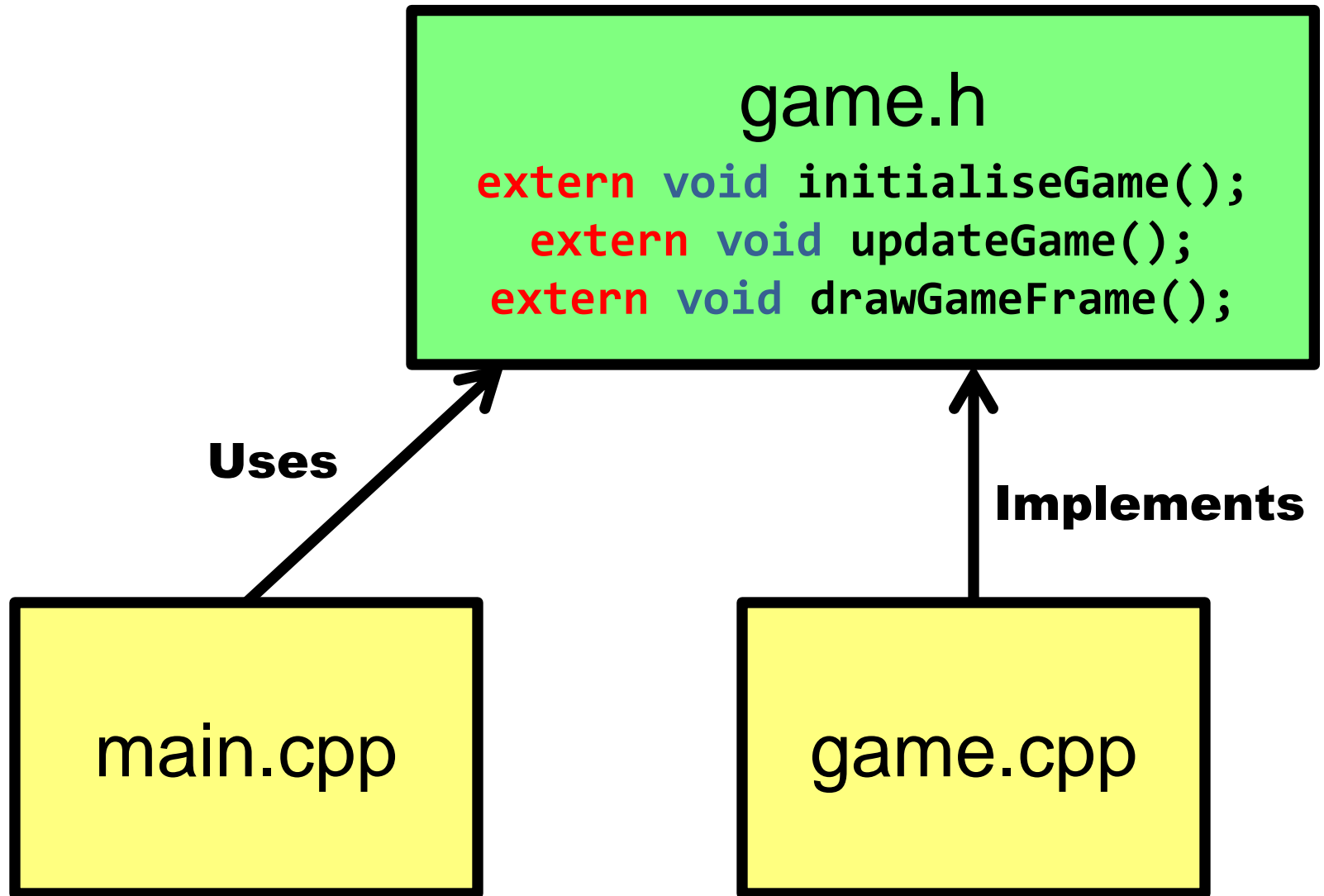
```
extern void initialiseGame();  
extern void updateGame();  
extern void drawGameFrame();
```

**Uses**

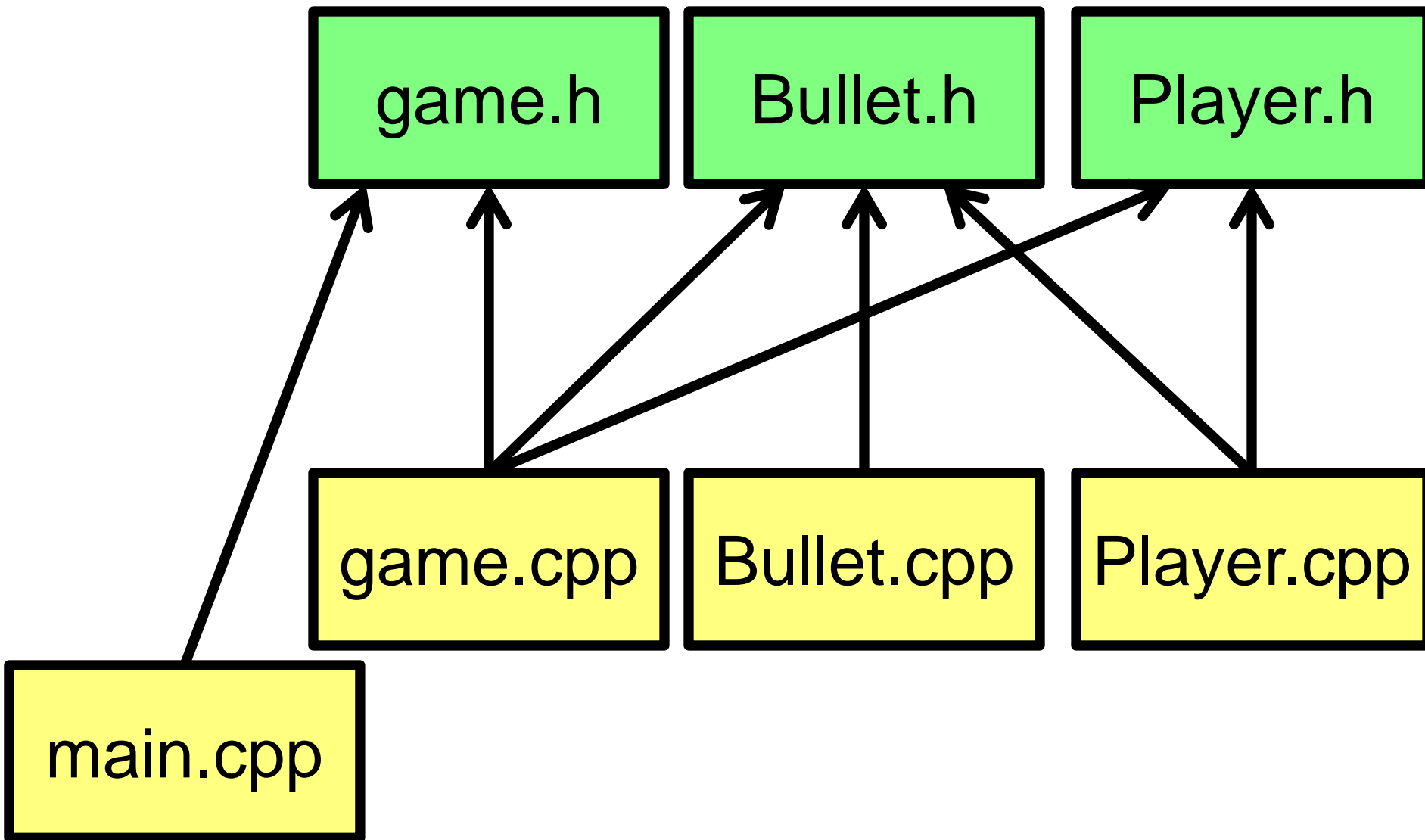
main.cpp

**Implements**

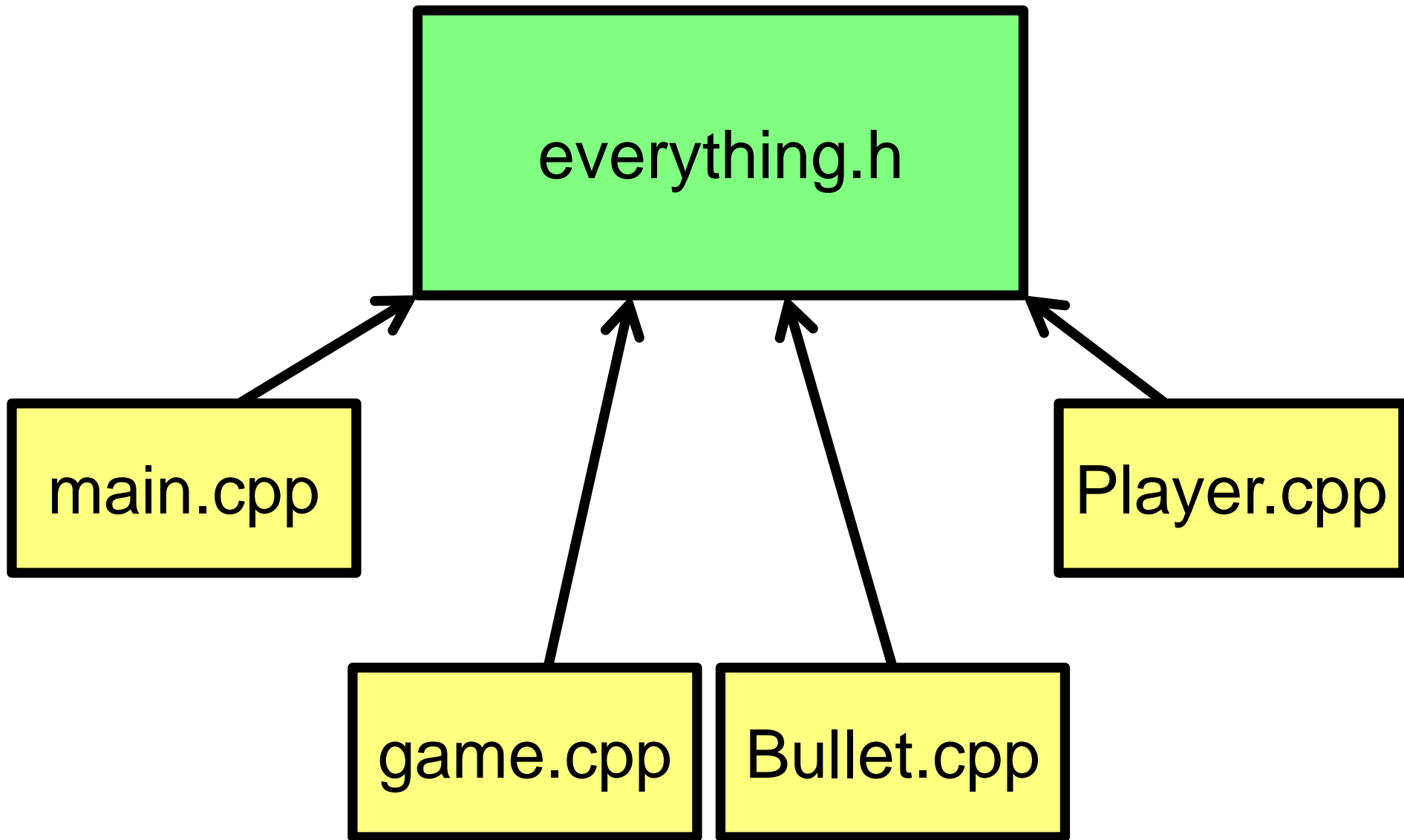
game.cpp







# For the Lazy:



# **VFinal - Splitting**

**Splitting into multiple files**

# ACKNOWLEDGEMENT

This presentation benefitted from

## PowerPointLabs

a PowerPoint plugin for creating  
better presentations with less effort.

PowerPointLabs  
is available for free at  
<http://PowerPointLabs.info>