

# **Comp 388/488 - Introduction to Game Design and Development**

---

Spring Semester 2017 - Week 13

Dr Nick Hayward

# Contents

---

- Game Designers
- Games and dramatic elements
- Pygame - Game Example
- References

# Games and dramatic elements

---

## considerations of game characters

- Characters in our games may also exhibit certain traits
  - *often unique to an interactive gaming environment*
- e.g. ability of a protagonist to become an agent in the game
  - *and channel empathy from a player to the game*
- traits of a character, in particular a game's protagonist
  - *need to be considered at each stage of a game's design and development*
- help us question motivation for a particular aspect of a game
  - *perhaps a backstory that leads to a mini-challenge for our character*
- need to consider how the character as agent enables our player to complete this mini-challenge
  - *what is the justification for including this mini-challenge in our game?*
- if we start to simply add challenges, conflict, or perhaps obstacles
  - *without a consideration of agency or motivation*
  - *a game may become disjointed and lack flow for the experience*
  - *the story, its characters, and gameplay may not make sense to the player*
- such characters need not be preconceived or developed by the game's designer
- avatars may also play a role as agent within a game
  - *e.g. in Blizzard's World of Warcraft*
- avatars will often be created, designed, and managed by a player
- players may invest a great deal of time, energy, and resources into such avatars
- agency and empathy provided by these characters
  - *will often fuel a player's gameplay and social role in a gaming environment*
  - *such empathy may be increased with greater player engagement with avatars...*

# Video - World of Warcraft

---

## avatars



- Original article - BBC News - World of Warcraft: Finding love with an online avatar
- Source - YouTube

# Games and dramatic elements

---

## characters and classes in Diablo

- interesting and fun aspect of original Diablo game was use of *classes* for characters
- instead of simply providing a single option for the protagonist
  - *Diablo provided three classes*
  - *classes = Rogue, Sorcerer, and the Warrior*
- expanded to six classes for Diablo III with various expansion packs
  - *expected to increase to seven in 2017*
- each character class provides different attributes, skills, and agency for the game
- not simply a matter of providing different types of characters and skills
  - *allows different players to empathise in varying ways with the game*
- no sense of one size fits all
  - *a player is provided with different ways to enjoy and complete the game*
- choice of game agent may also introduce variant paths through the game
- a player is provided with different perspectives on the story, challenges, and general gameplay
- many other games that employ a similar option for characters
  - *e.g. Nintendo's Mario Kart selector...*

# Image - Mario Kart

select a character and kart



Nintendo's Mario Kart

# Games and dramatic elements

---

## characters and emergent systems

- we may introduce *emergent systems* to our gaming environment
  - *creating a sense of autonomous, generated gameplay and challenge*
- add a semblance of *free will* to our characters
  - *creates a noticeable variant to standard player control*
- a traditional character's agency
  - *may be directly influenced, monitored, and controlled by the player*
- introduction of *free will* for certain characters
  - *control limitations may no longer apply*
- AI-controlled characters or emergent systems
  - *may now start to exhibit examples of autonomous behaviour*
- potential for interesting conflict may arise as a simple result of expectations
  - *e.g. player control vs a sense of limited free will for certain characters*
- The Sims - Free Will

# Video - The Sims 4

---

## Free Will



- Source - Sims 4 100% free will - YouTube



# Games and development

---

## quick exercise

Consider the following game characters and objects,

- a mediaeval knight
  - *carries a sword, may ride a horse, fighting skills, finite health...*
- a squire
  - *attends to the knight*
- a semi-intelligent/aware mob object - e.g. an ogre
  - *carries a club, may ride horse-like animal, fighting skills, renewable health...*
- a series of huts, caves &c. in the gaming world

Each of these characters or objects may be pre-defined or created with a sense of free will.

Define the following,

- rules for each character and object
- a brief outline for a game with these characters and objects

Then consider the following,

- how might free will affect the rules and outline for your initial game?
- what type of unexpected glitches, interactions, and features may result due to free will in this game?

# Python and Pygame - Game Example I

---

## fun game extras - add explosions to player's ship

- add explosions to a collision with a player's ship
  - *again, update game loop for these collisions*

```
# add check for collision - enemy and player sprites (True = hit object is now deleted from game window)
collisions = pygame.sprite.spritecollide(player, mob_sprites, True, pygame.sprite.collide_circle)

# check collisions with player's ship - decrease shield for each hit
for collision in collisions:
    # decrease player's shield for each collision
    player.stShield -= 20

    # add animation for explosion images if collision
    explosion = Explosion(collision.rect.center)

    # add explosion sprite to game sprites group
    game_sprites.add(explosion)

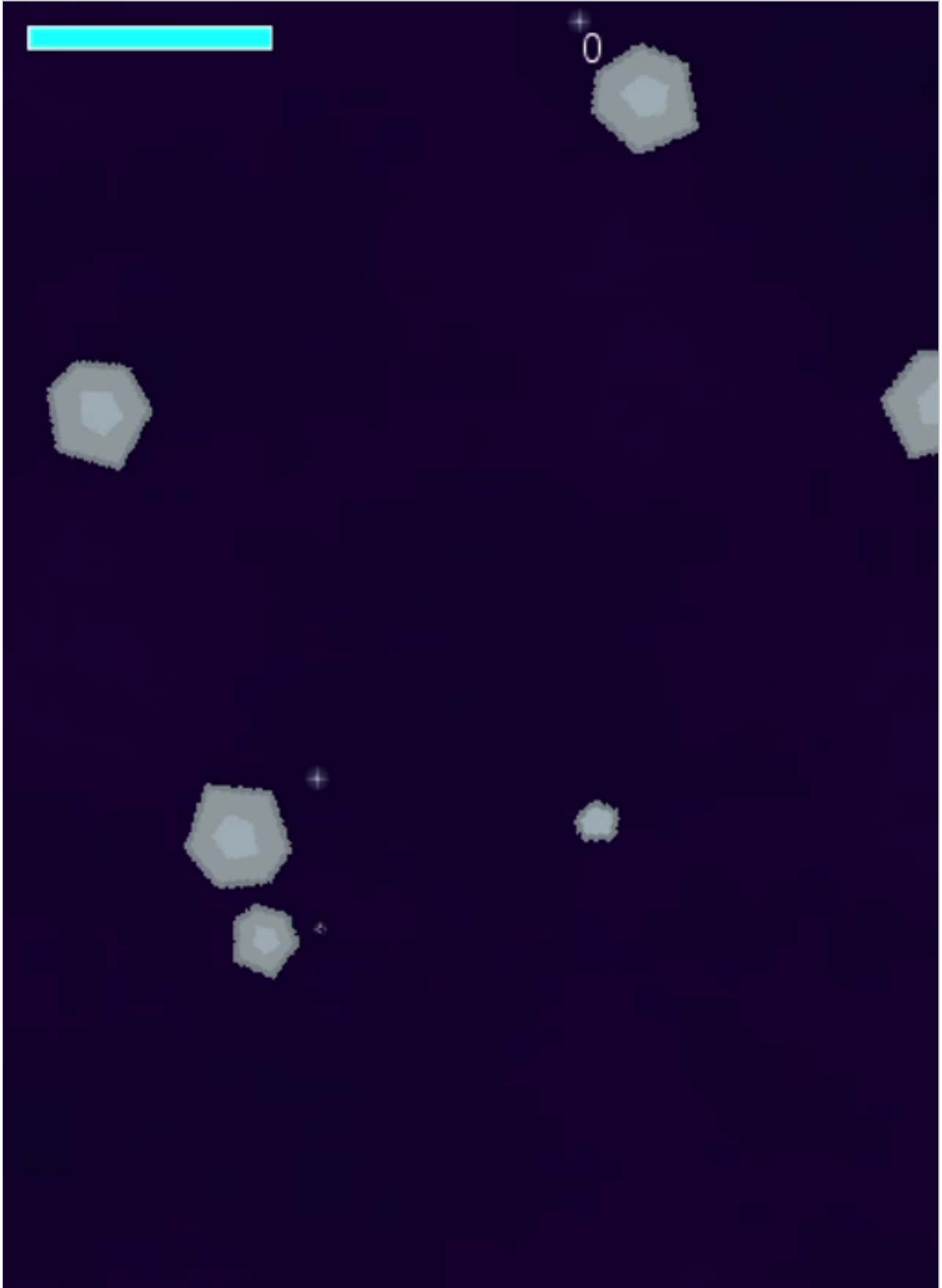
    # create a new mob object
    createMob()

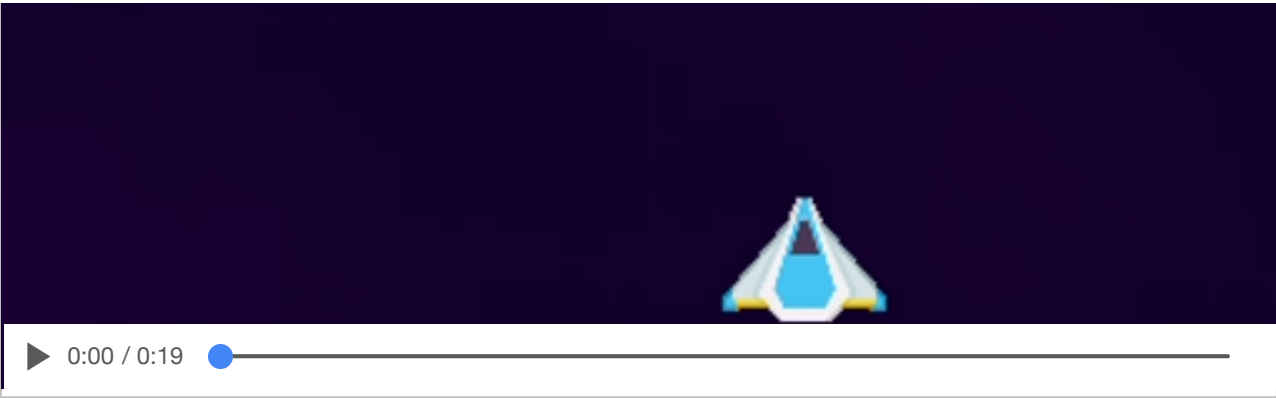
    # check overall shield value - quit game if no shield
    if player.stShield <= 0:
        running = False
```

## Video - Shooter 1.2 - Part 2

---

**add some fun explosions - player's ship**





# Python and Pygame - Game Example I

---

## fun game extras - scale explosion images - basic scale

- still a lingering issue with these collisions and explosions...
- explosions are not reinforcing the gameplay for our shooter style game
  - *no differentiation in the relative size of an explosion*
  - *no semblance of feedback to our player*
- one option to this issue
  - *perhaps add standard scale transform to image for each explosion sprite object*

```
# explosions
explosion_imgs = []

# iterate over explosion images in directory
for i in range(9):
    file = 'explosion{}.png'.format(i)
    # load image from os
    expl_img = pygame.image.load(os.path.join(img_dir, file)).convert()
    # set colour key for image
    expl_img.set_colorkey(BLACK)
    # append to specified list for explosion images
    explosion_imgs.append(expl_img)
```

- render a smaller, less overwhelming explosion for each collision

# Python and Pygame - Game Example I

---

## fun game extras - scale explosion images - relative scale - part I

- useful to be able to scale these explosions relative to the actual size of a given sprite object
  - e.g. a smaller relative explosion image for a smaller mob object
  - or, a relatively sized explosion against the player's ship
- update our class for the Explosion object
  - dynamically modify each explosion image in the animation relative to a specified size
- scale each frame of explosion animation to match the size of the collision object, e.g.

```
# create a generic explosion sprite - use for asteroids, player explosions &c.
class Explosion(pygame.sprite.Sprite):
    # initialise sprite
    def __init__(self, center, size):
        pygame.sprite.Sprite.__init__(self)
        # specify size for explosion sprite
        self.size = size
        # get initial image for explosion
        self.image = pygame.transform.scale(explosion_imgs[0], self.size)
    ...
```

- start by adding a parameter for size
  - pass a variable size for each collision object
- use this size to scale the initial image for the explosion animation

# Python and Pygame - Game Example I

---

## fun game extras - scale explosion images - relative scale - part 2

- each frame of the animation will also require scaling of the explosion image, e.g.

```
# change image as time progresses for explosion sprite
def update(self):
    # get current time
    now = pygame.time.get_ticks()

    # check if enough time has passed between animations
    if now - self.last_update > self.frame_rate:
        self.last_update = now

        # if enough time passed - add 1 to frame
        self.frame += 1

        # check if end of explosion images reached
        if self.frame == len(explosion_imgs):
            # kill if end of image reached
            self.kill()
        else:
            center = self.rect.center

            self.image = pygame.transform.scale(explosion_imgs[self.frame], self.size)

            # update rect for image
            self.rect = self.image.get_rect()

            self.rect.center = center
```

- as we output each frame of the explosion animation
  - scale this image to match the passed *size* for the explosion object



# Python and Pygame - Game Example I

---

## fun game extras - scale explosion images - dynamic collision size

- different size mob objects will have a matching explosion animation
  - *update in the game loop, e.g.*

```
# add check for sprite group collide with another sprite group - projectiles hitting enemy objects - use
collisions = pygame.sprite.groupcollide(mob_sprites, projectiles, True, True)

# add more mobs for those hit and deleted by projectiles
for collision in collisions:
    # calculate points relative to size of mob object
    game_score += 40 - collision.radius

    # play explosion sound effect for collision
    explosion_effect.play()

    # get size of collision object
    col_size = collision.rect.size

    #print("collision size = " + str(col_size))

    # add animation for explosion images if collision
    explosion = Explosion(collision.rect.center, col_size)

    # add explosion sprite to game sprites group
    game_sprites.add(explosion)

    # create a new mob object
    createMob()
```

- same for the player's object...

### resources

- notes = extras-part1-explosions.pdf
- code = objectexplosions2.py

### game example

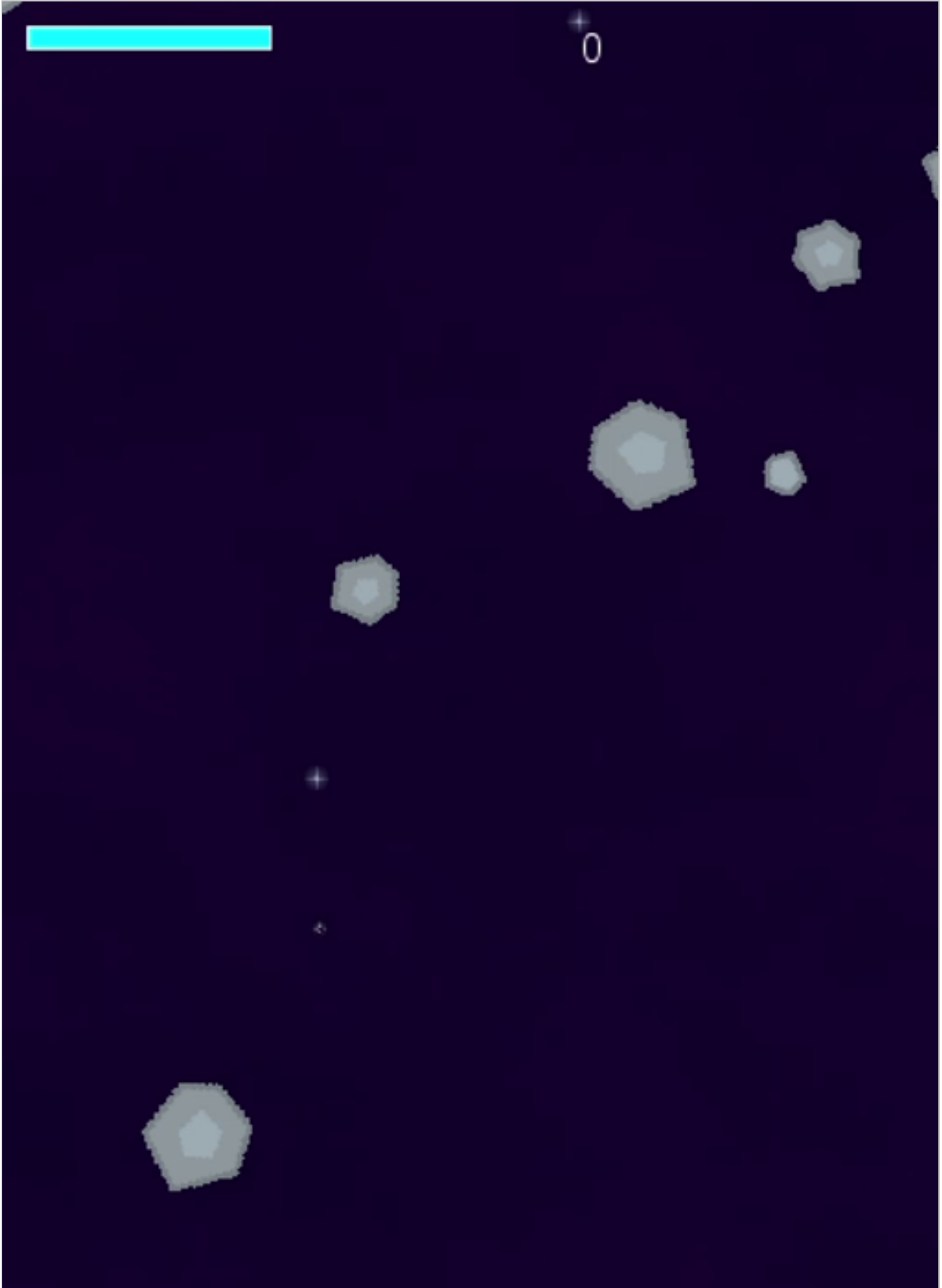
- shooter1.2.py
- add some fun explosions
  - *create sprite object for explosion*
  - *cycle through images to create explosion animation*

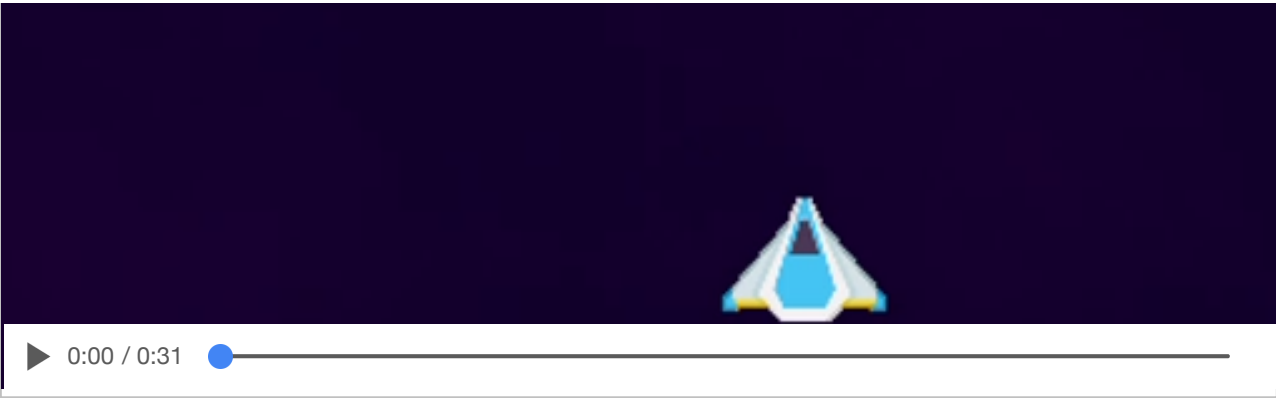
- *add explosion for each collision*
- **extra explosions**
  - *explode a player's ship for a collision*
- **scale explosions**
  - *rescale and size explosions in game window*

# Video - Shooter 1.2 - Part 3

---

scale explosions





# Game designers

---

## Designer example - Jenova Chen

- Jenova Chen is a Chinese game designer and director, now based in Los Angeles, USA
- after creating an experimental game called *Cloud* with Kellee Santiago
  - *whilst a student at USC's Interactive Media Division*
  - *Chen briefly worked on Spore*
- Chen is best known for games such as
  - *Cloud, fLOW, Flower*
  - *and most recently Journey*
- co-founded *ThatGameCompany* with Kellee Santiago
- landed a three game deal with Sony, which included *fLOW, Flower*, and *Journey*
  - *games exclusive to PlayStation consoles*
- his games are known for experimental use of narrative and structure
  - *and attempts to simply push what we perceive as a game...*
  - *e.g. his development of Cloud as a student*
- his collaboration with Austin Wintory on the music for *fLOW* and *Journey*
  - *represents a desire and commitment to integrate various dramatic elements*
  - *music, sound effects, shapes, colour &c. into the overall gaming experience*
- underlying trend and theme to the design of his games
  - *tries to make games that don't fit cultural preconceptions*
  - *interested in sparking universal emotions and feelings beyond culture...*

### Resources

- *Cloud*
- *fLOW*
- *Flower*
- *Journey*

- [Journey - Wikipedia](#)
- [ThatGameCompany](#)

# Image - Journey

---



- Source - ThatGameCompany

# Games and dramatic elements

---

## games and narrative structure

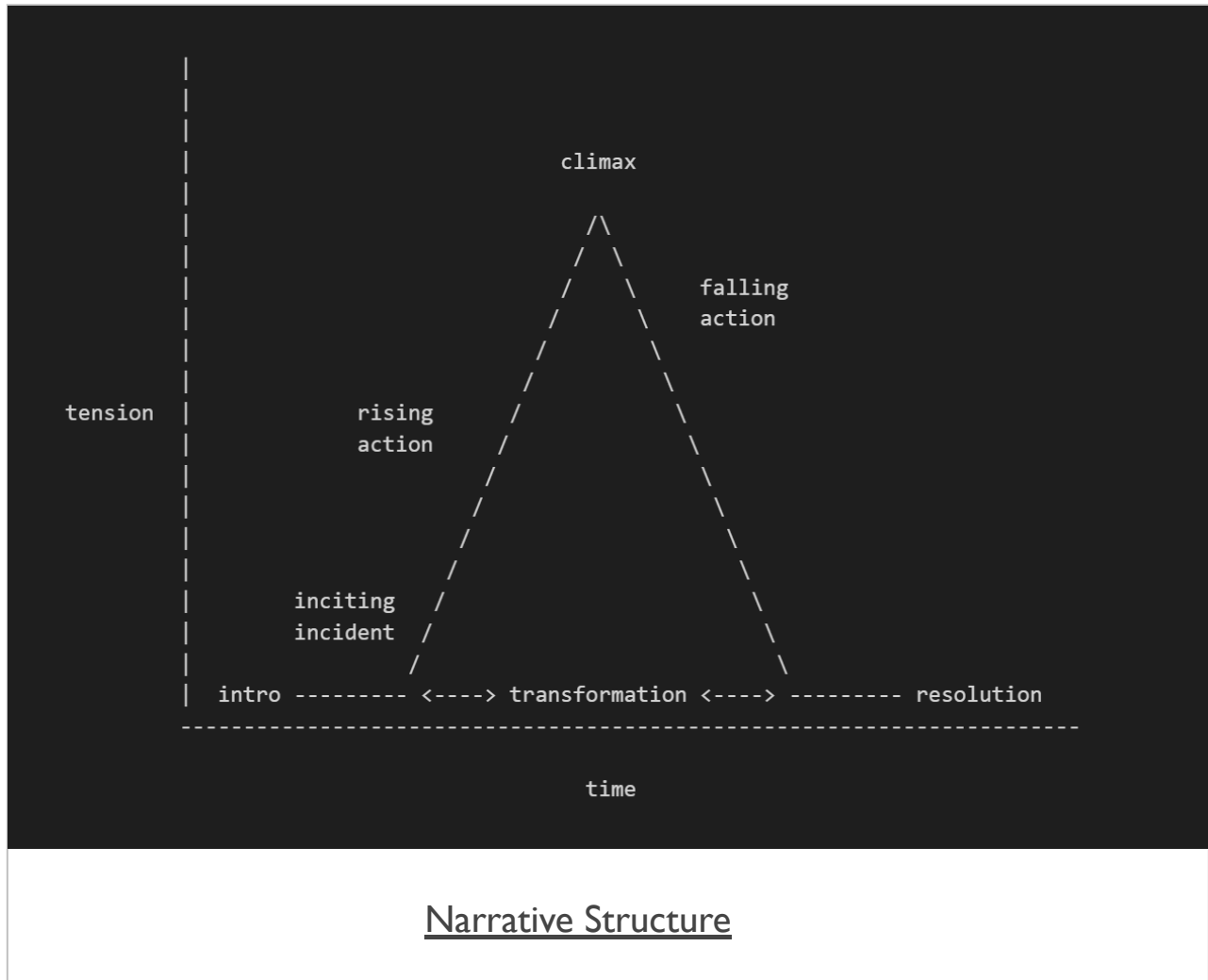
- **traditional drama** perceives the following categories as useful options for conflict
  - *a single character vs another single character*
  - *a single character vs their environment*
    - a character battling the forces of nature &c.
  - *a single character vs a machine*
    - many examples in movies...
  - *a single character vs their own inner demons*
    - a consideration of experience, morals, insanity &c.
  - *a single character vs perceptions of fate*
    - something is inevitable, bound to happen, can't be changed &c.
- a **game** will employ similar categories for its players, in particular the protagonist
  - *a single player vs another single player*
  - *a single player vs the game*
  - *and so on...*
- as these categories are played out in our games
  - *the sense of conflict they create will usually follow a discernible pattern*
  - *this pattern will escalate to a final resolution*
- escalating conflict will create a sense of tension in the gameplay
  - *usually matched and reflected in the story*
- gameplay may respond to the story, including corresponding elements
  - *such as music, visuals, speed, and a sense of risk*
- this tension will also tend to get worse, or more dramatic
  - *before it is resolved and gets better*
- this forms a classic **narrative structure** or **narrative arc**
  - *it becomes a useful tool for storytelling in games*
- forms the framework and support for all dramatic media



- **games** are not excluded...

# Image - Narrative Structure

## conflict in a game's story



- Source - Building An Arc: Bringing Narrative Structure To Your DJ Sets

# Games and dramatic elements

---

## ***Journey to a narrative structure***

- a recent example of narrative structure in gaming was the 2012 release *Journey*.
  - *designed by ThatGameCompany, and directed by Jenova Chen*
- its underlying design and story was inspired by *The Hero's Journey*
  - *a structure and outline for myth and story telling prescribed by Joseph Campbell*
- Campbell defined twelve stages on the *Hero's Journey*
  - *set a structure that follows the narrative arc along the path of the story*
- initial incident is an effective acknowledgement of the limits of the current environment
  - *the encompassing world for the hero*
- the hero must now leave this environment, this comfort zone of sorts
  - *embark into unknown, commonly dangerous territory*
- this journey will normally include many trials and tests
  - *the challenges we expect to introduce to many games*
- trials are not simply physical, but may also include
  - *aspects of temptation, mental reasoning, emotional dilemmas...*
- player will normally be expected to reach a defined low point on the journey
  - *the abyss that defines and shapes the counterpoint to the story and game*
- introduction of an extreme low point, the *abyss*
  - *allows the character to metaphorically die*
  - *then be reborn ready for the final challenges of the journey*
- the hero will then return to a point of calm and resolution
  - *transformed and free of the issues, fear, and doubts that initially defined them...*

# Image - Journey

---



- Source - ThatGameCompany

# Video - Journey

---

**available on PS4**



- Source - Journey PlayStation 4 Official Launch Trailer - YouTube

# Games and development

---

## quick exercise

For the earlier game, characters, and objects you defined, consider the following

- reduce this game's outline to its bare essentials
  - *i.e. which shapes, patterns, colours, objects &c. are still necessary to define your game's story?*
- outline the narrative structure for this game using these bare essentials
  - *where is the conflict in this story? the rising action, climax, falling action &c...*
  - *how is the resolution achieved for this game's narrative structure?*

Then, re-consider the role and influence of free will or emergent systems on this narrative structure

- what type of unintentional features, dead ends &c. may be introduced?
- how do you allow for such potential issues in your narrative structure?

# Demos - Pygame

---

## fun game extras

- `objectexplosions2.py`

# Demos - Pygame - Game I Example

---

- shooter1.2.py



# Games

---

- [Diablo - Wikipedia](#)
- [Diablo III - console](#)
- [Journey - ThatGameCompany](#)
- [Journey - PS3](#)
- [Journey - Wikipedia](#)
- [World of Warcraft](#)

# References

---

- Bogost, I. *Persuasive Games: The Expressive Power of Videogames*. MIT Press. Cambridge, MA. 2007.
- Bogost, I. *The Rhetoric of Video Games*. in *The Ecology of Games...* Salen, E. MIT Press. Cambridge, MA. 2008.
- Bogost, I. *Unit Operations: An Approach to Videogame Criticism*. MIT Press. Cambridge, MA. 2006.
- Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. Harper & Row. New York. 1990.
- Huizinga, J. *Homo Ludens: A Study of the Play-Element in Culture*. Angelico Press. 2016.
- Murray, J. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Free Press. New York. 1997.
- Poundstone, W. *Prisoner's Dilemma*. Touchstone. New York. 2002.
- Salen, K. & Zimmerman, E. *Rules of Play: Game Design Fundamentals*. MIT Press. 2003.

# References - Pygame - Game Notes

---

- [extras-part I -explosions.pdf](#)

## References - Various

---

- BBC News - World of Warcraft: Finding love with an online avatar
- Dubspot - Electronic Music Production and DJ School
- The Sims - Free Will
- ThatGameCompany - Hiring

# Videos

---

- World of Warcraft - Finding love with an online avatar - YouTube